Brunel University
Uxbridge

Investigating the Impact of Individual User Differences and Environmental Factors on Spatial Knowledge Acquisition from Virtual Environments

A dissertation submitted in satisfaction

of the requirements for the degree of

Doctor of Philosophy in Computer Science

by

Markos – Akrivos Kyritsis

2009

Abstract

Trying to ‘learn’ the spatial layout of an environment is a common problem in certain application domains, such as military and emergency personnel training. Until recently this training was accomplished solely by providing maps and briefings of an environment. These methods, however, only provide topological (survey) knowledge of the environment, which pays little attention to the details of routes and landmarks that can only be acquired through the acquisition of procedural knowledge via navigation. Unlike previous experiments concerning spatial knowledge acquisition this work does not attempt to determine whether spatial knowledge acquisition is feasible. Such investigations have yielded a variety of results, yet all agree that spatial knowledge acquisition from a virtual environment is feasible if given enough exposure time. Accordingly, the aim of this thesis is to contribute towards a better understanding of how various individual differences and environmental factors impact the exposure time requirements needed for a person to acquire spatial knowledge from a virtual environment.

Although the results of our investigation should be used with caution, we show that a one-size-fits-all situation is not possible when estimating the required exposure time that a user needs to acquire spatial knowledge. Moreover we provide a guide that allows a trainer to predict the required exposure time a person will require, by using the person's personal profile, and the environment's particular factors. In addition, we found that one of the tests we used during our investigation caused unnecessary frustration and confusion to our participants. This test is a standard way of finding a participant's orientation skill, and is commonly used in the area of spatial knowledge acquisition. Therefore, by recreating a new electronic version of the test and comparing the scores from both the new test and the old one our investigation showed that the scores on the new test were significantly higher for all participants. The training time was also lowered significantly. Our updated electronic version will be useful in future research. This test is available online at: www.newgztest.com.

Declaration

I confirm that all the work in this thesis is my own. All sources have been referenced and acknowledged accordingly.

Markos-Akrivos Kyritsis

Acknowledgments

First and foremost I would like to thank my supervisor Dr Stephen Gulliver for helping me through the PhD process. He showed a significant amount of patience and understanding, and I will forever be in his debt. I would also like to thank Dr Sonali Morar who initially supervised me before moving to the U.S. and also Professor Robert Macredie for guiding me in the right direction.

I am extremely grateful to Dr Darken for sending me the Guilford-Zimmerman orientation survey, and for being an inspiration to my work on spatial knowledge acquisition.

Finally, I would like to thank my lovely wife Ecaterina for her support and love, my mother Yvonne for her love and understanding, and my father George, who is the man responsible for who I am today. I couldn't have done it without you guys.

Table of Contents

Publications

Kyritsis M. Gulliver S. R., Morar S., Macredie R. (submitted) Empirical Study Investigating how User Individual and Environmental Differences Impact Spatial knowledge Acquisition Exposure Requirements, International Journal of Human Computer Studies.

Kyritsis M., Gulliver S., Morar S. (2010) “Spatial Knowledge Acquisition: Understanding the Factors Influencing Training Time.” Virtual Worlds and E-Commerce: Technologies and Applications for Building Customer Relationships, to be published by IGI Global.

Kyritsis M., Gulliver S., (Accepted) “Guilford Zimmerman Orientation Survey: A Validation”, ICICS 2009.

Kyritsis M. Gulliver S. R., Morar S., Macredie R. (2009) "Impact of cognitive styles on spatial knowledge acquisition". 2009 IEEE International Conference on Multimedia & Expo, June 28-July 3, 966-969.

Morar S., Kyritsis M. (2006) Mobile Gaming: Investigating the Effects of Screen and Keyboard Size on Game Playability and Interaction. Journal of Intelligent Systems, 15, 129-152

List of Figures

List of Tables

Chapter 1 - Introduction to the Problems Associated with Spatial Knowledge Acquisition from Virtual Environments

1.1 Introduction to Spatial Knowledge acquisition

Spatial knowledge acquisition (SKA) research attempts to clarify whether people can transfer geographical knowledge from a virtual environment (VE) into the real world. Research results in this area have proven to be quite contradictory, as many researchers have concluded that spatial knowledge acquisition is not possible (Darken and Banker, 1998; Goerger et al., 1998), while others state that it is possible (Witmer et al., 1996; Wilson et al., 1996; Waller et al., 1998; Foreman et al., 2003). Being able to transfer knowledge through navigation rather than map reading, can prove to be useful in a variety of areas ranging from military and emergency training (Bliss et al., 1997; Egsegian et al., 1993), to helping people with disabilities (Foreman, et al., 2003). Unlike traditional methods for learning environmental space, which rely on maps and compasses to provide a topological understanding of the environment (also known as survey knowledge), SKA focuses on the learning benefits of direct navigation in a representation of the actual environment (also known as procedural knowledge). This type of learning has the distinct advantage of providing detailed spatial information, which is difficult to acquire from a map (such as unique object and geographical landmarks). There is also strong evidence to support the theory that learning in a procedural manner can provide better distance estimation during navigation (Thorndyke and Hayes-Roth, 1982). Both the thorough learning of landmarks, as well as better distance estimation are very important during the process of navigational updating, which relies on both landmark-based processing, and dead-reckoning (a set of internal calculations performed during navigation that helps estimate distance and bearings), and can therefore decrease disorientation (Montello, 2005). Therefore, SKA research aims to understand the process involved when a person acquires spatial knowledge from a VE, and applies that knowledge when navigating in the real world.

1.1.1 – Problems with Spatial Knowledge Acquisition

Research has indicated that learning in any environment, regardless of whether it is virtual or real, relies on the ability of users to develop an understanding of space by creating a cognitive map of the environment (Asthmeir et al., 1993; Silverman and Spiker, 1997; Goillau et al., 1998; Clark and Wong, 2000; Riva and Gamberini, 2000). Cognitive maps are developed from both procedural and survey knowledge (Thorndyke, 1980; Golledge, 1995; Witmer, 1995; Goerger et al., 1998). The process of converting the knowledge acquired from exploration in a VE into a cognitive map, and then applying it in the real world, is known as Spatial Knowledge Acquisition (SKA). SKA has been considered by many researchers (Witmer et al., 1996; Darken and Banker, 1998; Waller et al., 1998, Goerger et al., 1998; Darken and Peterson, 2001), generating conflicting results on whether spatial knowledge of a real environment can be acquired from a virtual representation. Some researchers have reported success (Witmer et al., 1996; Wilson et al., 1996; Waller et al., 1998; and Foreman et al., 2003), yet others have said learning from a virtual space is not feasible (Darken and Banker, 1998; Goerger et al., 1998). However, most researchers agree that spatial knowledge acquisition research is feasible if enough exposure time is given (Waller et al. 1998; Darken et al., 1999; Koh et al., 1999).

Darken et al. (1999) determined that a one-size-fits all exposure time is not possible, instead arguing that there are various individual differences that affect the knowledge, aptitude, abilities, strategies, and impact on perceptual motoric and memorial knowledge of an individual, and therefore ultimately influence the navigational ability of a user. This leads to a significant difference in the exposure time that users need to acquire spatial knowledge from a VE.

Although research suggests that individual user differences play a critical role in navigation and learning, research also suggests that the environment itself can be just as influential. This thesis will demonstrate how various environmental factors can have either a negative or positive effect on learning in a VE. We have found that the size of the environment can alter the navigational complexity of the environment, therefore leading to higher exposure time requirements for SKA to take place (Bliss et al., 1997; Darken and Banker, 1998; Darken and Peterson, 2001). We have also found, from previous studies in the field, that spatial complexity as well as the amount of visual references available for navigational updating (commonly referred to as landmarks) are very important (Darken and Sibert, 1996; Hermer and Spelke, 1996; Witmer and Sadowski, 1998; Vinson 1999; Gouteux and Spelke, 2001).

The question that arises, therefore, is: How much exposure time is actually required by a user to acquire spatial knowledge from a particular environment? Answering this question is the aim of this thesis, and to satisfy this aim we will discover which individual user differences and environmental factors impact on the exposure time required by a person to acquire spatial knowledge from an environment.

1.1.2 – Factors Contributing to the Required Exposure Time

We argue that the factors that impact on the required exposure time for a person to gain spatial knowledge from the environment in which they are navigating, depends on both the person and the environment. Our research indicates that the three dominant factors that make up an environment are: size, spatial complexity and the amount of unique object landmarks (referred to as landmark potential). Size simply relates to how large the represented environment is. Spatial complexity is more generic, and can mean how many rooms and corridors make up an environment, or how rocky is the terrain. Ultimately, however, complexity is related to the effect of visual obstruction (causing the user disorientation due to lack of visual cues). Finally, the landmark potential of an environment implies how many unique object landmarks are available as visual cues to the navigator. Landmark potential can be measured by the number of landmarks in a room, and the frequency of their occurrence. For example, having six unique landmarks in a room and none in the other rooms is not as useful during navigation, as having six scattered unique landmarks in different rooms. These environmental factors can have a variable impact on exposure time required for the acquisition of spatial knowledge, depending on the cognitive and biological characteristics of the navigator, which are commonly referred to as individual user differences. The user differences that we looked at during our study were: gender, cognitive styles, orientation skill, previous knowledge of similar environments and knowledge of the training system. We show that both the individual user differences and the environmental factors impact the required exposure time. We aim to show, through a series of experiments, that some environment types are harder for certain groups of people to learn, whilst others are difficult in general for all people. We also aim to identify whether any of the individual user differences can be trained in order to lower the required exposure time to acquire spatial knowledge. For example can prior training with the VE interface decrease the required exposure time?

1.2 Aims and objectives

1.2.1 – General Aim of the Thesis

The aim of this study is to identify and justify how environmental factors and individual differences impact on the exposure time required by people to acquire spatial knowledge from a virtual environment. To do this, we use pre-tests that separate experimental participants into appropriate groups, depending on their individual differences. Four conditions must be met in our experimentation:

- To discover which environmental factors and individual user differences impact the required exposure time.
- To measure the importance of each individual user difference on the required exposure time.
- To measure the importance of each environmental factor on the exposure time.
- To provide a set of guidelines that can help VE trainers predict the amount of training time required by a person in order to acquire spatial knowledge; depending on the environment and their particular cognitive and biological attributes.

1.2.2 – Objectives

In order to achieve the research aim, the following three objectives have been defined.

- Identify, through previous research, the factors (cognitive and environmental), which are responsible for the change in exposure time requirements.
- Design a set of experiments that examines how these factors contribute to the required exposure time. This can be accomplished by creating a test-bed that enables the development of virtual environments that examine the individual differences and environmental factors on exposure time. Various pre-test will

be required in order to group recruited participants according to their individual differences.

- Run the experiments, analyse results and present a set of guidelines that can help VE trainers predict the amount of required exposure time that a person will need in a particular environment.

1.3 Expected Contributions from the Results of the Experiments

We anticipate that some individual differences will impact on exposure time in general, while others impact on exposure time only in certain environments. By collecting the results from our experiments, we hope to identify patterns that can help us understand how the various environmental factors, and individual user differences, impact on the exposure time required for a user to acquire spatial knowledge from an environment. Using these patterns, we aim to create a set of guidelines for VE trainers that will function as an indication of how much exposure time a person needs during training. These guidelines will not only help ensure that a person has acquired the spatial information from an environment, but is also an important step in SKA research, since it will clarify why research in the area is of a contradictory nature.

Another contribution we hope to make involves the development of a suitable methodology for grouping participants involved in spatial knowledge acquisition research according to their cognitive/biological abilities, and will also present a formal way of creating environments depending on the environmental factors the research aims to investigate.

1.4 Structure of the thesis

This section outlines the structure of the thesis, presenting the reader with the material, design, experiments and results in a logical manner. The chapters represent different stages of the thesis, and discuss relevant information starting from the literature review, through to the experimental process, conclusion, and findings/contributions. The following contains a brief summary of the content of each chapter.

- Chapter 2 – Introduces the reader to the field of spatial knowledge acquisition and the problems associated with the area. We also discuss the individual user differences and environmental factors we identified from previous research that appear to impact on the exposure time required to acquire spatial knowledge. Finally we discuss the aim of the thesis.
- Chapter 3 – Discusses the methodology that will be adopted by the thesis in order to tackle the issues presented in chapter two. The chapter presents the tests we used in order to put participants in specific individual difference groups, including the Guilford - Zimmerman orientation survey, which was used to separate participants with a 'high' orientation skill from participants with a 'low' orientation skill. The chapter also discusses how we developed the environments required to investigate the importance of each environmental factor we identified in chapter two. Finally the chapter presents the actual experimental process.
- Chapter 4 – Discusses the results of the experiments, analyses the data, and raises further questions that are tackled in chapters five and six. The initial contributions from the results of the thesis are shown here. A guideline is provided, which indicates how the required exposure time that a person needs to acquire spatial knowledge from an environment changes according to their individual user differences. The chapter also raises a problem with the traditional Guilford - Zimmerman orientation survey which is then tackled in chapter five.
- Chapter 5 – Design, development and testing of the web-based Guilford – Zimmerman aptitude survey created to tackle the issues found in chapter four. Presents the new survey, discusses the mechanics of the survey, discusses the experimental design and process of testing the new survey.
- Chapter 6 – Discusses how increasing the environmental factors contributes to a rise in the required exposure time, and the rate of which

these factors increase. The chapter also presents a guideline that can be used to predict the required exposure time for a particular user, in a particular type of environment.

- Chapter 7 – Conclusion, further work, and a summary of contributions found in the thesis to the domain of spatial knowledge acquisition in virtual environments.

Chapter 2 - Spatial Knowledge Acquisition Research

2.1 Introduction to Spatial Knowledge Acquisition Research

This chapter presents detailed information on the area of Spatial Knowledge Acquisition (SKA). We introduce the problem of contradicting literature in the area of SKA, and discuss how the amount of exposure time given to a person during Virtual Environment (VE) training is responsible for the feasibility of SKA. We then show how various individual user differences (such as gender), as well as environmental factors (such as size), impact on the required exposure time that a particular person will need in a specific environment during the process of SKA. Ultimately, this chapter presents the research problem of this thesis, which is to understand how much each individual user difference and each environmental factor impacts on the exposure time required to acquire spatial knowledge from an environment.

The ability to 'learn' the environment, before engaging in navigation, is an area of interest for a variety of application domains, such as emergency training (Bliss et al., 1997; Egsegian et al., 1993) and when helping people with disabilities (Foreman et al., 2003). The more traditional approach to training is accomplished by providing maps and briefings of an environment before navigation. These methods, however, only provide topological (survey) knowledge of the environment, which whilst being more flexible, pays little attention to the details of routes and landmarks (Thorndyke, 1980; Golledge, 1991). Procedural learning has a distinct advantage over survey knowledge as can be seen in an experiment conducted by Thorndyke and Hayes-Roth (1982). In this experiment, participants who had procedural knowledge of an environment, estimated route distances significantly better than participants who had acquired survey knowledge of the environment. There is also a general understanding that navigation relies heavily on previously acquired visual information. An example of this is landmark-based navigational updating, which is the process of re-orientation during navigation in a previously visited environment (Montello, 2005). This process relies on previously seen "visual references" in order to adjust bearings during navigation. Maps and other traditional navigational equipment cannot provide visual information in the same way that a real environment can, or a virtual representation of that environment. Therefore, virtual environment training promises the ability to provide procedural knowledge through exploration, and because of this has caught the attention of a variety of researchers all attempting to discuss whether virtual training is more efficient than training through more traditional methods (Witmer et al., 1995; Goerger et al., 1998; Waller et al., 1998; Foreman et al., 2003).

Learning in virtual environments partially relies on the ability of users to develop an understanding of space by creating a cognitive map of the environment (Asthmeir et al., 1993; Cobb and d'Cruz, 1994; Bliss et al., 1997; Silverman and Spiker, 1997; Goillau et al., 1998; Clark and Wong, 2000; Riva and Gamberini, 2000). Cognitive maps are mental representations of space that people develop in order to acquire an understanding of space within an environment, both virtual and real, through either procedural knowledge or survey knowledge (Thorndyke, 1980; Golledge, 1991; Witmer et al., 1995; Goerger et al., 1998). When learning in a procedural manner, cognitive maps are created through the act of navigation (Montello, 2005). Navigation in itself is made up of two separate and very distinct processes. The first of these processes is locomotion, which is the movement of a person within an environment. The second process is way-finding, which is the planning of routes that a person undergoes when trying to get to a specific destination (Montello, 2005). It is understood that during self-directed locomotion (where the person is actively moving about in the environment solving behavioural problems, such as avoiding obstacles, rather than being moved in a vehicle), there is a tendency to acquire more spatial knowledge (Feldman and Acredolo, 1979). Virtual environment training provides this benefit of self-directed locomotion, without the possible hazards of a dangerous life-threatening situation, and is therefore very suitable for emergency training.

Both procedural and survey knowledge can be learned to such an extent, that it can be transferred into the real world (Witmer et al., 1995; Howes et al., 1998; Rose et al., 2000). The process of transferring the knowledge acquired from exploration of a virtual environment into a cognitive map and then applying it into the real world is called spatial knowledge acquisition.

So far research on spatial knowledge acquisition through virtual environments, has provided a variety of results, sometimes of a contradictory nature. The findings, although conflicting, appear to be subject to a key influencing factor, 'required exposure time' (Witmer et al., 1996; Darken and Banker, 1998; Waller et al., 1998; Goerger et al., 1998; Darken and Peterson, 2001). This factor is the exposure time that a user will spend learning the environment in order to achieve spatial knowledge acquisition, and according to previous research in the field of navigation and SKA, seems to be affected by the environmental properties and also the particular cognitive abilities of the users navigating through it (Darken et al., 1999; Darken and Peterson, 2001; Stanley et al., 1998).

In order to fully understand the effectiveness of spatial knowledge acquisition through virtual environments, this thesis aims to identify the factors that influence navigational complexity of an environment, and also the individual user differences that may have an effect on the learning abilities and strategies of users. The results of this investigation will look at how exposure time is affected by these factors and individual user differences, by presenting their relative importance. These findings will contribute towards a better understanding of how much exposure time is required in order to acquire spatial knowledge, from a certain type of virtual environment, depending on the individual user differences and environmental factors.

2.2 The Relevance of Exposure Time on Spatial Knowledge Acquisition

Witmer et al. (1996), Wilson et al. (1996), Waller et al. (1998), and Foreman et al. (2003) conducted various experiments in order to conclude whether spatial knowledge acquisition can be acquired from a VE representation of the real world. These experiments involved a group of participants navigating through virtual space and acquiring spatial knowledge in a procedural manner, and then comparing the results to a group that learned the environment through maps (which was defined as “conventional” or “traditional” in these studies) in a non-procedural manner (survey knowledge). These experiments concluded that the participants who acquired the knowledge from a VE representation of a real world space, performed better when asked to navigate in that actual space, therefore showing that they had acquired more spatial knowledge. However, this is only the case if a long exposure time is given to the participants. If a short exposure time is given, then the participants who used the “traditional” methods of spatial learning performed better during navigation in the real world. We hypothesise that this has more to do with the learning curve involved in acquisition of procedural knowledge. Maps are draft representations of an environment with key landmarks and spatial layout. The environment itself, however, is often much more packed, and requires more time to learn, not only because of the totality of visual information, but also because actual navigation needs to take place.

Darken and Banker (1998) and Goerger et al. (1998) disagree with Witmer et al. (1996), Waller et al. (1998), Wilson et al. (1996) and Foreman et al. (2003) and argue that spatial knowledge acquisition is not always feasible. Darken and Banker (1998) reported that experts perform better using conventional methods such as maps, while Goerger et al. (1998) reported that all participants had greater success learning from traditional methods. Although not shown in their study, Goerger et al. (1998) acknowledge, that with longer exposure times, virtual reality training may in fact be more beneficial, however this is hard to determine since the exposure times that a user spent in each experiment differed. Waller et al (1998) allowed for two minutes, Darken and Banker allowed for a set 60 minute exposure, and Georger et al. (1998) allowed for a set 30 minute exposure, yet they referred to this as a short exposure time. In these studies, the allowed exposure time was inconsistent, and no explanation was given as to why these exposure times were chosen. We hypothesise that this exposure time would be affected by the various environmental factors and individual user differences.

In an attempt to clarify this situation, Darken et al. (1999) discussed why spatial knowledge acquisition research often delivers contradictory results. They explain why Witmer et al. (1995), Bliss et al. (1997), and Koh et al. (1999) all conclude that spatial knowledge acquisition is possible, whilst Darken and Banker (1998) and Goerger et al. (1998) concluded that spatial knowledge acquisition is not feasible. Darken et al. (1999) agree with the argument made by Koh et al. (1999), that individual user differences are an extremely important factor in the development of cognitive maps, and expand by saying that a one-size-fits all situation may not be possible when deciding on the required exposure time. Darken et al. (1999) also discuss that cognitive and biological differences affect a series of cognitive processes, which are critical to navigation. They stated that previous knowledge, aptitude, orientation ability, strategy, perceptual motoric, as well as memorial knowledge, all influenced the navigational skill of the user. This is backed up by Koh et al. (1999) and Waller et al. (2001) who both discuss the importance of individual differences when acquiring spatial knowledge from an environment. According to Koh et al. (1999) and Waller et al. (2001) there is a need to identify these individual differences and to understand how they affect performance when acquiring spatial knowledge. Therefore, this thesis aims to identify and discuss the individual differences of users that can affect navigation skills, and therefore the exposure time required to acquire spatial knowledge from a VE. Understanding how these individual differences affect navigational skill should help researchers understand the required exposure times necessary for a specific user to acquire spatial knowledge from a particular environment.

The individual differences of users that navigate through an environment is not the only factor that seemingly influences the required exposure time. Darken and Peterson (2001) analysed how spatial knowledge acquisition is affected by a variety of factors, and reported that required exposure time may in fact be environment dependent as well as dependent on individual user differences. They explained that some environments provide more cues than others and, therefore, that the exposure time needed may alter according to those cues (Darken and Peterson, 2001). What Darken and Peterson (2001) identified is that, regardless of whether the training interface is supported with a map or whether other visual cues are used in combination with the environment, the structure of the environment itself may contain factors that support user navigation, leading to a smaller exposure time requirement when acquiring spatial knowledge. It may seem obvious that as the size of an environment increases, so does the time it takes to navigate through it, and consequently the ability to create a cognitive map of the environment; but size, is only one influencing factor that will be discussed in this chapter.

It seems that both environmental factors and individual user differences are responsible for how long it will take a person to acquire spatial knowledge from a virtual environment. Figure 2.1 presents a diagram, created by the author, which summarises what we have discussed so far. The diagram shows how the process of spatial knowledge acquisition is affected by the various environmental factors. In the diagram, and arrow represents a link between a parent node, and its children. The rectangles are properties that are important parts of the SKA process. For example, we have already discussed that SKA may be feasible if enough exposure time is given, and we have discussed how exposure time is affected from the overall navigational complexity which is affected by both individual user differences, and environmental factors. By following the diagram, we can see how the arrows lead us to both the importance of the individual user differences, and environmental factors.

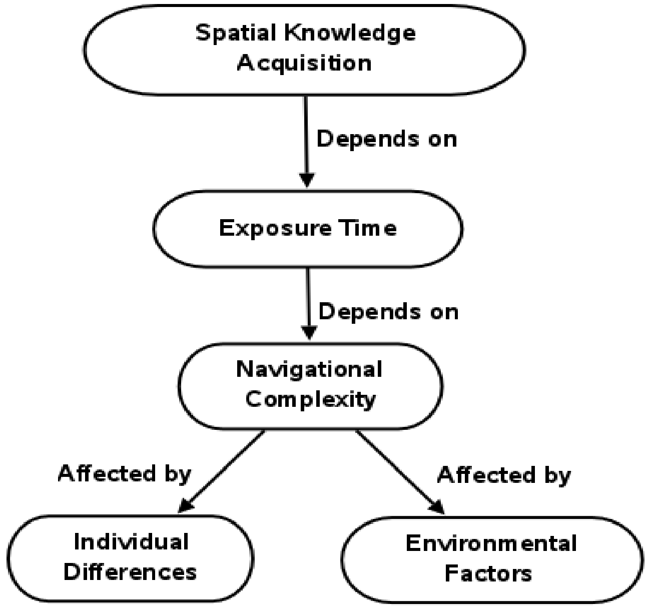


Figure 2.1 – Illustrative summary of research so far

Figure 2.1, however, does not show any interactions between individual user differences, and environmental factors. We cannot know if certain types of environments impact required exposure time for different types of users. For example, we are unable to answer whether males or females require a different exposure time in certain types of environments – a conclusion that would be of considerable use. We aim to know more about these types of issues by looking at research that indicates associations between environmental factors and individual user differences. Initially, we need to consider which environmental factors, and individual user differences, will most likely affect spatial knowledge acquisition. We begin by discussing the environmental factors in section 2.3.

2.3 The Environmental Factors that Affect Exposure Time

Many of the factors that affect navigational complexity, which apply to the physical world are applicable in the virtual world as well (e.g. size) (Bliss et al., 1997; Darken and Banker, 1998). Darken and Peterson (2001), broke down an environment to a space made up of building blocks or 'landmarks' that are connected by routes, which are interconnected to form nodes. These interconnected routes and nodes make up the spatial layout of the environment.

As was discussed earlier, environments can be complex in nature and can therefore affect the exposure time required to acquire spatial knowledge (Darken and Peterson, 2001). For example, Darken and Sibert (1996), report size as the major influencing factor of spatial knowledge acquisition.

Although we have used size as the main example so far, it is only one of a number of factors that can influence the process of navigation. Research indicates that there are other factors such as complexity of the spatial layout and landmark potential (Darken and Sibert, 1996; Hermer and Spelke, 1996; Witmer and Sadowski, 1998; Vinson, 1999; and Gouteux and Spelke, 2001). These factors appear to influence required exposure time in an environment, and research in the field is again contradictory on which factors more greatly influence the exposure time. Hermer and Spelke (1996), Goerger et al. (1998) and Gouteux and Spelke (2001), all show that the spatial layout complexity is critical to navigation, whilst, according to Witmer et al. (1995), Witmer and Sadowski (1998) and Vinson (1999), the number of unique object landmarks, as well as the graphical detail of these landmarks is important.

Research indicates that navigational complexity influences various processes that directly affect navigation. These processes are identified by Stankiewicz and Kalia (2004) as: perception (input of cues and other environmental information at a given time during navigation); accessing the cognitive map (ability of each person to develop a cognitive map of the environment and then apply it to navigation); spatial updating (ability to navigate from different positions in the environment); and decision making (logical process to reach a certain goal depending on current position and perception). Moreover, Stankiewicz and Kalia (2004) discuss how various environmental factors such as size can influence these processes and make navigation more complex. For example, for an environment of a large size, perception is burdened since there is a larger area that must be processed by the person navigating, and creating a cognitive map is harder since more spatial memory is required. Spatial updating takes longer since there are more places and objects to consider, and finally decision strategy is influenced, since more decisions must be made to reach a certain goal (Stankiewicz and Kalia, 2004).

So far we have seen that the three major environmental factors that appear to influence exposure time are: size; spatial layout complexity; and landmark potential. The following sub-sections will discuss these factors in more detail.

2.3.1 - Size

In our work, environmental size refers to the overall raw navigational space available to the user. The differentiation between large-scale and small-scale environments is not clearly defined in literature, however in the experiments of Darken and Banker (1998) a large-scaled environment was described as being 1200*700 metres. We could consider this to be large when comparing it to other virtual spaces that represent a house or building, such as the one in Goerger et al. (1998). Obviously, without visual obstructions, size would not be a confounding factor, however, in any non-flat featureless environment, as size increases, so does the time taken too locomote from one place to another in order to acquire spatial knowledge. Darken and Sibert (1996) made it clear that the alteration of size plays a critical role in exposure time required to acquire spatial knowledge. They attempted to lower the navigational complexity of large-scaled environment by further introducing various visual aids such as maps. Evidence shows that if the environment is large, then a navigator can greatly benefit from landmarks or other navigational aids (Darken and Peterson, 2001). It may be obvious that the navigational ability of a user would be decreased in a large environment, since it would take more time to explore a larger environment during locomotion, and to absorb the spatial information which must then be transformed from visual working memory into long-term memory (O'Keefe and Nadel, 1978). What may not be so obvious is that navigational strategy is also affected. Butler et al. (1993) demonstrated that distance plays a significant role when navigating. They found that users will most frequently choose to navigate through shorter paths, even if those paths are more complex. Therefore, as size increases so does the amount of exposure time required by a person to acquire spatial knowledge from the environment. The question that seems to arise, however, is how much time is suitable for a set size.

It seems that for a large mountainous region, as used by Darken and Banker (1998), a 60 minute exposure time was considered ‘short', however, to our knowledge no justification for this exposure time was given by the author. For a seven story building, as designed by Goerger et al. (1998), an exposure time of 30 minutes was considered ‘short’, however again no justification for this exposure time was given. We have no indication at all, whether navigational complexity increases linearly as size increases when navigating through a virtual environment. If the relationship is not linear, then size becomes more and more critical to consider with respect to exposure time required for SKA.

Although an environment can have a variable size, having an absolutely flat environment, with no obstructions to reference points is quite rare in the real world, and since this thesis looks at virtual environments that represent the real world, it is appropriate to take into account the issues that obstructions have on navigational complexity and subsequently on required exposure time. This is referred to as spatial layout complexity.

2.3.2 - Spatial layout complexity

Spatial layout is the geometrical structure of an environment (Gouteux and Spelke, 2001). When trying to determine what makes a layout more complex, without involving size, spatial layout complexity is the number of objects, such as walls, that obstruct a user's line of sight from various reference points, such as visible landmarks. This is demonstrated in the work of Kalia and Stankiewicz (2007), who measured the spatial layout complexity in terms of corridors. They found that as the number of corridors in an environment increased, so did its complexity.

Another type of spatial complexity is achieved by adding fog, or decreasing the distance of rendered objects (known as view frustrum), which also limits visibility (Stankiewicz and Kalia, 2004). Although this is quite critical when considering research areas such as gaming, this type of complexity is not likely to be a factor in spatial knowledge acquisition research, since this research focuses on learning spatial layout, and such natural effects would be removed from the VE during the process of acquiring spatial knowledge.

In virtual environments the architecture of an environment is important to navigation, as demonstrated by Passini (1984), who discussed how Manhattan's rectangular grid, with visual aids such as numbering of streets and avenues, makes navigation very simple. However, in some cases it is simply not possible to provide architectural simplification (e.g. in a natural mountainous region). Darken and Sibert (1996) explained that environments, which do not provide any navigational aids such as road signs, will prove harder to navigate through, and will ultimately lead to a loss of awareness and disorientation. In an office building, one expects to find signs that point towards different levels, or corridors that do not simply lead to dead ends. A 'natural environment' on the other hand has few restrictions and does not follow any architectural laws, therefore increasing the time required by a person to acquire spatial knowledge, by making the environment more complex.

In general, complex environments, both natural and man-made, tend to have a lot of visual obstructions to important visual references. In mountainous regions, this is accomplished through the many slopes, while in caves or buildings, this is accomplished through the walls that obstruct various structural or object references. These visual references are objects that are usually referred to as landmarks, and the number of distinct landmarks available to a user plays an important role in the navigational complexity of an environment.

2.3.3 - Landmarks

Darken and Sibert (1996) report that adding landmarks to an environment can enhance navigation in more important ways than individual user differences could. Lynch (1960), Vinson (1999) and Stankiewicz and Kalia (2004) identify that an environment is made up of a variety of landmarks that individuals use to navigate. Stankiewicz and Kalia (2004) broke down the term landmarks into two distinct types: structural landmarks and object landmarks. Structural landmarks are distinct geographical features of an environment that can be used for navigating (such as a T-junction, or a different coloured room), whilst object landmarks are objects in the environment that are independent of the structure (such as a statue). In general, landmarks create differentiation between different parts of an environment (Weisman, 1981). This would mean that an environment with similar structural geometry throughout will be considered to have less structural landmarks, while an environment with varied structure could have more potential for structural landmarks. Although it is rather difficult to understand and predict which landmarks a user will choose for navigation purposes, there are some theories. Stankiewicz and Kalia (2004) explained that different landmarks can be more or less beneficial to the user. According to their research, landmark potential and its effect on navigation is defined through three properties that a landmark may possess.

The first of these properties is persistence. This is whether the landmark is mobile, so a parked car for example may not be the best landmark as it has a high chance of moving from that space by the time the user revisits the site. This may cause confusion as users often navigate on object landmarks rather than geographic structure (Newman et al., 2007). The second property is whether the landmark is perceptually salient; this simply means how visible the landmark actually is, and can be determined from such factors as landmark size, obstruction from other objects etc. The third and final property of a landmark is whether or not the landmark is informative. This is important as it informs an individual of their location. Stankiewicz and Kalia (2004) explain that for a landmark to be informative it must be distinctly different from other landmarks, in fact the reason why users have difficulty using landmarks such as statues (Ruddle et al., 1997) is because they cannot easily distinguish between the statues, unless they approach them for a closer inspection. If all three of these factors are satisfied then the landmark can be useful during landmark-based navigational updating, which, as we discussed before, is a process of reorientation that relies on landmarks.

Stankiewicz and Kalia (2004) discovered that participants tend to learn structural landmarks better than object landmarks and that when spatial knowledge acquisition does occur, the environment structure can be remembered by a user, even as far as a year after the initial encounter. For example, most people will remember the layout of their first school, but few will remember the location of specific objects.

Vinson (1999) identifies the importance of correct landmark placement and explains that if the landmarks are correctly placed, they can play a critical role in lowering the navigational complexity. Vinson (1999) presents various types of landmarks previously identified by Lynch (1960), which can be used to ensure that the environment is informative to the user and helps them obtain spatial awareness. Table 2.1 demonstrates the types of landmarks that could be used in an outdoor environment.

Table 2.1 – Landmark types and functions adapted from Lynch (1960) and Vinson (1999)

Types	Examples	Functions
Paths	Street, canal, transit line	Channel for navigator movement
Edges	Fence, river	Indicate district limit
Districts	Neighbourhood	Reference point
Nodes	Town square	Focal point for travel
Landmarks	Statue	Reference point into which one does not enter

Vinson (1999) deducted that the landmarks, which are frequently available, and visible from various positions in the environment (i.e. paths), are useful in navigation. Frequent landmarks appear to increase navigational performance. For this to apply, however, landmarks must be unique. In natural environments, such as a forest, there is a large amount of non-unique landmarks, such as trees and rocks. These landmarks overpopulate the area, and cannot be used as reference points (which is a very important part of navigational updating), since one tree may not be distinctly different from another. However, man-made structures in a forest environment would stand out as distinct landmarks (Whittaker, 1996). Therefore, it is not just the frequency of landmarks readily available throughout the environment, but also the number of distinct landmarks available that can help a user orientate, decreasing navigational complexity. This information seems to relate to the need for a landmark to be informative, as was suggested by Stankiewicz and Kalia (2004). The assumption is that for a virtual environment, landmark potential can be measured by the number of visible, non-dynamic (not moving) and distinct landmarks available to the environment as a whole, and the frequency of these landmarks per sector (for which a sector may be the maximum view available to the user: a room or a corridor).

2.3.4 – Indications of the importance of Environmental Factors in VE Training

We have discussed that various environmental factors affect navigation and therefore learning in a Virtual Environment (VE). What we have not discussed, however, is how much impact these environmental factors actually have on spatial knowledge acquisition. Although we understand that a larger size will prove burdensome to the user, we do not have a clear picture of how much an increase in size will correspond to an increase in exposure time. Moreover, we do not know whether a large increase in complexity will be more burdensome during navigation and learning than, for example, a slight increase in size. Understanding the impact these factors have on navigation and learning should help determine how much exposure time will be needed to train an individual in a certain environment.

Newman et al. (2007) presents how navigational complexity is affected by various changes in the environment, which includes changes to the layout and by changing, removing or adding landmarks. Newman et al. (2007), developed an environment made up of roads and buildings, as well as different types of city landmarks such as shops, and looked at the relationship between the spatial layout and the landmark potential. They hired various students to assume the role of a taxi driver within a virtual city. The participants had to drive around the city picking up passengers and bringing them to various positions. The tasks themselves were reportedly quite simple and some training was given. They then looked at how the spatial layout and the landmark potential affects spatial knowledge acquisition by investigating the time it takes for the participants to complete the task. Their investigation reports that users can acquire spatial knowledge through spatial layout alone, and also that if layout and landmarks conflict (e.g. a sign post is now moved down the street), users will prefer to navigate on landmarks over the spatial layout, and will therefore find themselves completely disoriented. According to this research as long as the landmark is persistent (e.g. does not change location in space as a car might do), then the spatial layout holds more weight. If however a landmark unexpectedly changes location, then confusion occurs and the user will attempt to navigate using the landmark. The restriction on this paper was that Newman et al. (2007) did not take into account the size of the environment, or the frequency of unique landmarks readily available in the environment. Instead, the developed environment was of a set size, that had a certain number of landmarks that were used a certain number of times. Each time the spatial layout of the environment was changed, the landmarks were placed in different locations. This research focused on understanding navigational complexity in terms of landmarks and layout. The other restriction on this paper was that it did not take into account the individual differences of the participants, which according to Darken et al. (1999), Koh et al. (1999), and Waller et al. (1998) are an extremely important part of understanding required exposure times.

In summary, we have seen through research, that size, spatial layout complexity, and landmark potential, are very important to the process of SKA. We can conclude that research in the area is inconclusive as to how much weight each factor applies on the exposure time required for a person to acquire SKA. Research does indicate however, that most of the factors tend to compliment each other, and as one increases, the others are affected as well. The isolation of each factor may prove difficult, but necessary if we are to truly understand their importance in SKA.

2.4 Individual Cognitive and Biological Differences with Respect to Navigational Competence

As discussed earlier, Darken et al. (1999), Koh et al. (1999) and Waller et al. (1998) all discuss the importance of individual differences, and their relevance to the exposure time needed to acquire spatial knowledge acquisition from a VE. Darken et al. (1999) identified the skills that affect navigational competence as knowledge, aptitudes, abilities and strategies. Individual differences have been considered for many years in Visuospatial research, which considers a very broad spectrum of research on the understanding of images and space, as well as spatial knowledge acquisition (Hegarty and Waller, 2004). This thesis will now consider the research that presents cognitive and biological differences that affect these skills. We group the individual user differences into: gender, experience/knowledge, age, and orientation skill. As suggested by Darken et al. (1999), each of these human attributes can influence the navigational skills of the user when they navigate in a novel environment. These attributes affect navigation as a whole, e.g. orientation skill due to hippocampus development (O'Keefe and Nadel, 1978; Smith and Millner, 1981; Maguire et al., 1996; Maguire et al., 1999; Maguire et al., 2000), or they may simply affect user navigation when an environment lacks various cues.

2.4.1 - Gender issues in navigation

There is evidence that gender plays a significant role in acquiring spatial knowledge from a virtual environment. Waller et al. (1998) reports that females are particularly disorientated in a virtual maze, reporting large bearing errors when they were asked to draw the maze they had just navigated through. They report that the performance of females, when acquiring spatial knowledge, lagged behind that of men. They also seem to face more difficulty when pointing to objects in the virtual environment using an analogue input device such as a joystick. However they did not have trouble navigating through the environment after training in the real maze. According to Waller et al. (1998), this suggests that women have more difficulty learning the spatial characteristics of a virtual environment than men, which agrees with the findings of Astur et al. (1998).

Although it would seem that women's ability is more constrained when learning spatial characteristics of a virtual environment, their difficulty when navigating in the maze may be constrained by strategy rather than ability. Both Sandstrome et al. (1998) and Moffat et al. (1998) have provided explanations as to why male users navigate better in a maze. One of the deficiencies of a maze is that it relies heavily on geometrical navigation, rather than the use of landmark cues. After a series of experiments concerning navigation in landmark rich and landmark poor environments, Sandstrome et al. (1998) concluded that women rely heavily on the use of object landmarks for navigation. Men on the other hand, seem to use both structural landmarks, and object landmarks for navigation and development of cognitive maps. We hypothesise that the main reason that females performed worst in the Waller et al. (1998) experiments is because they rely more on landmark-based navigational updating. As we discussed earlier, navigation is made up of two processes, locomotion, and way-finding (Montello, 2005). Locomotion, which is the act of moving around in the environment whilst solving various behavioural problems, such as avoiding obstacles, is constantly updated through two more processes. These processes are landmark-based updating, and dead-reckoning. Landmark-based updating is a fixed reference system, which acts as a “beacon” while a person is navigating (Montello, 2005). It seems that females pay close attention to landmarks, and can re-orient themselves very well as long as they can update their navigation using landmark-based updating. We believe that the problem lies with the process of dead-reckoning. This process is an internal process, which is more mathematical in nature. A person navigating will keep track of various components of locomotion, such as velocity, acceleration, bearing, etc. (Gallistel, 1990; Montello, 2005). Our assumption is that females have trouble performing these internal calculations.

The difficulty that women face when navigating through an environment with limited landmarks, suggests that the required exposure time required by women to acquire the spatial information is increased when environments lack well placed object landmarks. Accordingly, women have problems navigating environments that are complex by nature (such as a maze), however this does not mean that for other types of environments their navigational skills will suffer, or that if given enough exposure time their knowledge of the environment will not equal or exceed that of men. This theory is backed by Vila et al. (2002), who indicates that as exposure time in the environment increases, the navigational differences between the genders decreases.

2.4.2 - Knowledge and Experience

Experience and knowledge of the environment, as well as the training system used to navigate through that environment critically affects exposure time required to acquire spatial knowledge. Knowledge concerning the system, whether it is a desktop computer that allows for mouse and keyboard input, or an immersive device, can have a limiting effect due to an overload of mental tasks. This overload is described by Booth et al. (2000) and is explained to be a limitation to attention due to unfamiliar controls and interfaces. According to Booth et al. (2000) this occurs mainly because attention is divided when undertaking these tasks, which are required to navigate and perceive the information seen on the screen. More effort is required to understand and interact with the interface, therefore not enough attention is given to creating cognitive maps of the environment. In compensation, a longer exposure time is required.

More effort is also required if an environment is novel to the user (i.e. if they have never navigated through this type of architectural structure). In other areas of HCI, the ability of experts VS novices during navigation plays a critical role for interface design (Egan, 1988; Dix et al., 1993; Eberts, 1994).

Kuipers (1975), Brewer (2000) and Mania et al., (2005) all explain how experience with a certain type of environment gives rise to certain structures in human knowledge memory. These structures are called schemas and are formed in human memory due to past experiences. Schemas consist of perceptual information and language comprehension, and are invoked when interacting with new information. The required exposure time to learn an environment depends on the memory performance, which is in its turn influenced by the Schemas. These can be affected by the consistent items of the environment, i.e. whether items that are likely to exist in such an environment appear in the virtual representation, such as trees in a forest, and is named the consistency effect (Brewer and Nakamura, 1984). Another theory is called the inconsistent effect and argues that inconsistent items influence memory performance positively, such as a car in a forest (Lampinen et al., 2001). It is quite obvious that schemas are highly relevant to landmark potential and seem to indicate that regardless of user orientation skill, a person that has strong past experiences navigating through a certain type of environment, such as a forest, will be more likely to recognise various landmarks and create a cognitive map of the area faster than a person with no experience in navigating within such an environment.

Knowledge of the environment was considered to be a variable in the experiment of Darken and Banker (1998), who only selected experienced mountaineers for their experiment. Darken and Banker (1998), however, reported that the advanced mountaineers did not benefit from the 60 minute exposure time in the virtual environment, although they did benefit from using a map. They did not, to the best of our knowledge, test to see the overall orientation skills of the users that took part in this experiment. Instead Darken and Banker (1998) used participants that had considerable experience navigating through real wilderness using cues and maps. This does not mean, however, that these participants were experienced with the interaction system, or had high aptitude and orientation skills.

2.4.3 - Aptitude and Spatial Orientation Skills

Perhaps the most discussed individual user difference, especially in the area of spatial knowledge acquisition, is orientation skill. Most experiments testing for spatial knowledge acquisition attempt to keep orientation skill as consistent amongst the participants as possible (Witmer et al., 1996; Goerger et al., 1998; and Waller et al., 1998). It is obvious that research considers spatial orientation skill as being a very influential attribute during a variety of areas involving human-computer interaction, such as browsing and other visual tasks (Egan and Gomez, 1985; Gomez et al., 1986; Vicente et al., 1987; Stanney and Salvendy, 1995). There is strong evidence that individuals have different orientation abilities, which are simply biological in nature (O'Keefe and Nadel, 1978; Smith and Millner, 1981; Maguire et al., 1996; Maguire et al., 1999; Maguire et al., 2000). Other research points to the hippocampus area, which is placed in the centre of the brain, as being responsible for providing spatial memory (O'Keefe and Nadel 1978). The amount of used spatial memory increases when the amount of spatial information increases, e.g. when size of the environment increase (O'Keefe and Nadel 1978). Smulders et al. (1995) suggests that during certain seasons, navigation ability in some migrating mammals, leads to their hippocampus volume enlarging in size. This variation in hippocampus volume, however is not so extreme in humans.

To verify whether the volume of the hippocampus in humans stops growing, or whether it can in fact increase in size through training, Maguire et al. (2000) experimented on the navigational ability of taxi drivers against a control group of non-taxi drivers. Their results showed that the longer people rely on navigation, the larger the volume of right-hippocampus brain area. Of course, it would be very difficult to determine a person's orientation skill by looking at the size of their hippocampus. Instead, there are various spatial visualisation and orientation tests that can determine a person's orientation skill; such as the Guilford-Zimmerman orientation survey. Other tests also exist (such as spatial memory, and spatial scanning tests), but spatial orientation tests are thought to be more successful in determining a user’s ability to acquire spatial knowledge (Waller et al., 1998).

Although the orientation skill of a user is often thought to be the most critical individual difference, there is currently no proof in literature that it has the most impact on the required exposure time needed to acquire spatial knowledge. We already discussed that there are other individual differences that affect the navigational ability of a user, such as gender and experience. In this thesis, we will investigate how important orientation skill is to SKA. Research indicates that there are more individual user differences such as age and cognitive styles that affect navigation ability and can therefore make it more difficult.

2.4.4 - Age differences in Navigation

Although we discuss age in the chapter, it is not one of the individual user differences that will be considered in our research. This is because our research aims to contribute knowledge of VE training into the domains of military and emergency training, which exclude older and younger age groups. However, there are strong indications that age has an overall detrimental effect on navigation, and it is therefore necessary to discuss it in the literature chapter.

Age plays an important role in navigation due to an overall change in sensory abilities, as well as various knowledge and cognitive skills, which are developed through life (Cohen and Scheupfer, 1980; Mathews, 1992; Wilkniss et al., 1997; Pine et al., 2002). Hasher and Zacks (1979) suggest that spatial ability is an automatic process that does not demand further cognitive abilities, and therefore should not be affected by age. However, according to Cohen and Scheupfer (1980), pre-adolescents navigate through a novel environment just as well as adults if the knowledge that is being acquired is through procedural means, such as direct exploration. It is in fact the transfer of survey knowledge to procedural knowledge, through a medium such as a map that seems to prove difficult for pre-adolescent children, since the ability to navigate through the environment using abstract mental representations is developed in later stages of adolescence (Mathews, 1992). This seems to be associated with changes made within cortical association areas, such as the frontal, postrolandic temporoparietal and medial temporal cortices (Lipska et al., 1998). When trying to determine how effective procedural knowledge learning is from a virtual environment, when compared to survey knowledge acquired from more traditional methods such as maps, Cohen and Scheupfer (1980) theorise that pre-adolescents are burdened when navigating through a novel environment using survey knowledge and abstract mental information (i.e. from a map). Due to this, pre-adolescents may find it advantageous when learning in a more procedural manner.

Interestingly, it has been shown that as children grow and reach adolescence they seem to have comparable navigation skills to adults, and can transfer survey to procedural knowledge. An attempt to prove this was made by Pine et al. (2002) who found that when navigating through a virtual city, adolescents reached as many goals and learned the environment as quickly as adults. Pine et al. (2002), however, also found that when asked to recall information, such as label points of interest on a map, adults exceeded adolescents by a significant amount. It seems that adults have better ability when transferring procedural to survey knowledge. There does not seem to be any evidence, however, suggesting that adults perform better when transferring procedural knowledge obtained from VE training, to the real world.

As an adult reaches old age they suffer from various issues surrounding both their sensory and orientation skills. Salthouse et al. (1990) argues that as people get older, they find it increasingly hard to process new information, whilst trying to retrieve information from memory. According to Kirasic (2000), navigation becomes increasingly difficult as age increases due to declines in perceptual, cognitive and motor abilities. It seems that disorientation in spatial navigation becomes more and more frequent when a person exceeds the age of seventy, even if there is no sign of mental deterioration (Hunt and Waller, 1999). Research suggests that in terms of learning navigational space, older people find it increasingly hard to retrace routes and learn maps (Wilkniss et al., 1997), orientate with respect to other environmental objects (Aubrey and Dobbs, 1990), as well as make distance and direction judgments (Kirasic, 1991). One of the most difficult problems that older people face, when trying to develop cognitive maps of an environment during navigation, is the attention divide of focusing on physical tiredness and poor sensory input (Darroch et al., 2005). Because of this, the ability to acquire spatial knowledge through a medium such as a virtual environment, could in fact be beneficial to older ages, since it would help them learn novel environments without the risk of physical tiredness.

2.4.5 - Cognitive Styles

The concept of people adopting different strategies in order to solve problems and make decisions was first presented by Allport (1937) who presented cognitive styles as a person's preferred way of perceiving, remembering, thinking and problem solving. Since then, research has looked into cognitive styles, and has referred to them as persistent strategies adapted by individuals when faced with problem solving tasks (Robertson, 1985). In more detail, cognitive styles affect perceiving, remembering, organising, processing, thinking and problem solving (Liu and Ginther, 1999).

Although it is known that cognitive styles can greatly affect strategy and decision making during navigation, the effect they may have on SKA is still relatively unknown. Previous research presents some limited results, such as that by Doyle et al. (1998), who report that different cognitive styles affect time-compressed learning in a flight simulator. However there is strong evidence suggesting that cognitive styles affect navigation and other decision-making tasks on various application areas, such as: presentation of multimedia content (Ghinea and Chen, 2003); hypermedia navigation (Chen and Macredie, 2002); online learning (Graff, 2003); computer aided learning (Atkinson, 2001), and web directory interface design (Chen et al., 2005).

Many different learning strategies are consistently adopted by a user in order to solve a problem. Messick (1976) identified 19 different cognitive styles that users adopt, and Smith (1984) identified 17. However Schmeck (1988) grouped them up to form two distinctly different learning styles. The first is a more holistic learning style, which is referred to as field-dependent and seems to emerge from activity in the right hemisphere of the brain. The second is a more analytical learning style that is referred to as field-independent and seems to emerge from activity in the left hemisphere of the brain. This relates to the learning styles of holistic strategy VS the serialistic strategy as proposed by Messick (1994).

Witkin et al. (1977) states that field-dependent people are more passive when learning information. They prefer to learn information by focusing on the information as a whole, rather than breaking it down (Pask, 1976; Pask, 1979). On the other hand, field-independent users are more active when learning new information and prefer to acquire information in a serial fashion by breaking it down (Pask, 1976; Pask, 1979). The implication that this has on navigation can be seen in previous research on 'hypermedia navigation', which indicates that field-dependent users were more efficient when they had to take a more holistic strategy, and navigate using a map of the overall system. Field-independent users, on the other hand, benefited more from an analytical strategy, which included a depth-search of the entire system (Ford, 1995; Ford and Chen, 2001). If this holds true in virtual environment training, we hypothesise that field-independent individuals will perform better in complex environments, whilst field-dependent individuals will perform better in large environments. We base this hypothesis on the importance of navigational updating during navigation and learning. It seems that field-dependent individuals will take longer to sample more cues, but will increase their chance of using this excess of cues in order to avoid disorientation. Field-independent individuals on the other hand, will excel in the process of dead-reckoning during navigation (as is suggested by their more analytic approach); therefore holding an advantage in more complex environments. We also hypothesise that field-independent individuals would benefit more from the procedural learning that is offered during navigation of the environment, whilst field-dependent individuals would probably benefit more from traditional learning methods such as maps and briefings. This hypothesis is supported by the results of Goodenough (1976) and Witkin et al. (1977), who state that field-independent people sample more relevant cues to solve a problem, while field-dependent people tend to sample more irrelevant cues to the current problem. This could also imply that in terms of landmarks, which are considered cues for navigation, field-independent users will benefit more from informative landmarks than field-dependent users.

2.5 The Interaction Between Various Environmental Factors and Individual User Differences with Respect to Spatial Knowledge Acquisition

One of the objectives of this thesis is to contribute towards a better understanding of how and why particular types of environments prove more challenging to various individuals. Overall, it seems that regardless of the type of environment, most people will be constrained according to their individual user differences. However, in some cases there seems to be a trend for low performance in specific environments. For example, females seem to perform more poorly in environments which are void of essential landmarks. Previously we argued that some individual differences are more affected by specific environmental properties. If we can somehow measure these properties, we could help a VE trainer predict the required exposure time for a particular user to acquire spatial knowledge from an environment. This would then lead on to a better understanding of just how efficient SKA is when acquired from different types of virtual environments for different types of individuals.

By summarising what we have discussed so far, and by closely examining the research presented in the previous section (2.4), we can conclude that some identified individual user differences ‘react’ more strongly with specific environmental factors. The reaction between these properties seems to critically influence the overall navigational complexity of the environment. Figure 2.2 illustrates the various interactions between the individual user differences and environmental factors, taken in the literature, as discussed in this chapter. Figure 2.2 was developed by the author in order to accommodate for the significance of various individual user differences during navigation when particular environmental properties are present. The top three items are the environmental factors, the items on the bottom represent the individual user differences, and are placed within their relative cognitive categorisation as discussed by Darken et al. (1999). The arrows represent a relationship. This relationship can be thought as one item affecting another. For example, we can see from the diagram that whether gender will affect performance depends on the landmark potential of the environment. Orientation skill on the other hand affects performance regardless of the type of environment since it has a relationship with all the environmental factors.

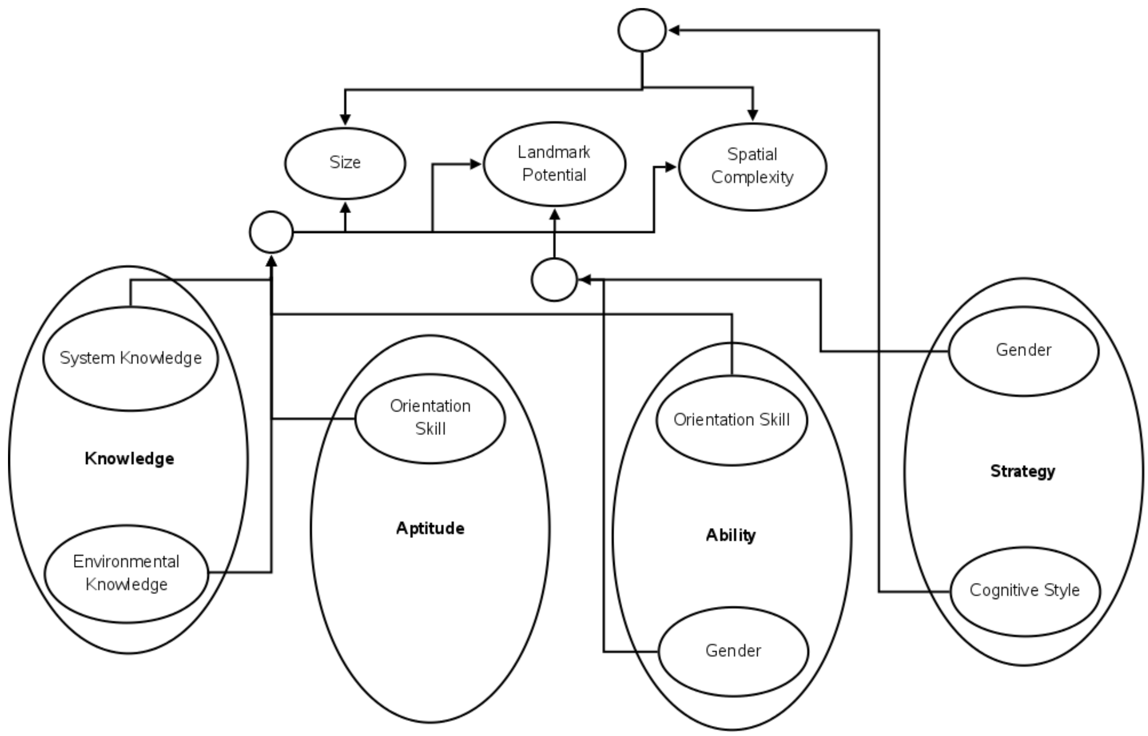


Figure 2.2 – The stages of the information-processing model affected by environmental factors and individual user differences

By closely examining figure 2.2 we can see that most individual user differences impact on all types of environments. The only exception seems to be gender and cognitive styles. This diagram is extracted from supporting literature and is therefore hypothetical in nature. In the next chapters, we aim to show that the model presented in figure 2.2 is indeed valid and correct, and also to quantify how much individual user differences and environmental factors impact the exposure time during SKA.

2.6 Summary

This chapter presented a variety of literature suggesting that SKA through virtual environment navigation is feasible, but is influenced by the exposure time given to a user to learn the environment. It was deducted through a comparison of previous studies, that individual cognitive and biological differences impact on the navigational skill of the user, and lead to higher exposure time requirements for

SKA. These individual user differences affect various skills such as knowledge, aptitude, ability and strategy, and have been identified as gender, age, orientation skills, knowledge/experience and cognitive styles. However, individual differences are not the only properties that affect exposure time requirements. It is obvious that various structural factors of an environment may render the environment more or less navigationally complex, and therefore influence the exposure time needed to acquire spatial knowledge as well (Darken and Peterson, 2001). These environmental factors are: size, spatial complexity, and landmark potential.

Early indications (see figure 2.2) show that most identified individual user differences impact exposure time for all types of environments. The only exceptions, as stated, are gender and cognitive styles. Gender seems to be a critical factor, only when navigation is taking place in low-landmark maze-like environments, whilst our hypothesis is that cognitive styles will only play a significant role in large, and complex environments.

Finally, we argue that in order to understand how exposure time is affected by various environmental factors and individual user differences, it is important to understand how and how much they impact the time required when acquiring spatial knowledge from a virtual environment. Knowledge of required exposure time will contribute to numerous domains, including military and emergency training. Moreover, it will also help facilitate VE designers in their understanding of the environmental factors that affect different users, and therefore support the use of aids and cues to avoid frustration and disorientation.

Chapter 3 - Methodology

3.1 Introduction

The previous chapter introduced the environmental properties and individual user differences that seem to be important to the required exposure time that a person needs in order to acquire spatial knowledge. Unlike previous experiments on spatial knowledge acquisition (SKA) this thesis does not attempt to determine whether SKA is feasible. Such investigations, as seen in chapter two, have yielded a variety of results, yet all agree that SKA from a virtual environment is feasible if given enough exposure time. Therefore the aim is to contribute towards a better understanding of how various individual differences and environmental factors impact on the exposure time requirements needed for a person to acquire spatial knowledge from a VE. The previous chapter discussed the role of individual differences and environmental factors, and their impact on the required exposure time of a user in order to acquire spatial knowledge from an environment. This chapter aims to present a feasible methodology, that will facilitate the thesis to tackle this problem.

The key individual differences that have been identified and will be investigated are: cognitive style; orientation skill; gender; environmental knowledge; and system knowledge. Although this list of individual user differences was identified in chapter two, it is most likely not exhaustive. However, running an exhaustive experimental process, that will identify all the key differences that have not already been discussed in the current literature is outside the scope of this project. Instead we decided to test the importance of the already identified individual user differences, and provide these results as an important stepping stone for future research in the area of SKA. We need to adopt a suitable method for grouping participants according to their individual differences during the experimental process. Various tests can be used to filter the participants in order to satisfy the requirements of the experimental process. By closely examining methodologies used in previous investigations, within the domains of: spatial knowledge acquisition; individual user differences; and VE training, a fitting methodology may be adopted or adapted from an array of related work.

Table 3.1 – Previous individual differences research using more than one interface

Research Paper	Individual Differences under investigation	Number of interfaces used	Number of participants used
Jennings et al. 1991	Spatial ability, verbal ability, field dependence, short term memory, thinking/feeling	Five database retrieval systems	24
Darken and Cevik 1999	Spatial Ability	Two interfaces used, an urban VE and an open ocean VE	30 participants
Chen 2000	Visual Memory and associative memory in the first study. Spatial ability and associative memory in the second study.	One interface for the first study (virtual environment), two interfaced (spatial and textual)	10 for first study, 12 for second
Hurder and Juvina 2004	Spatial ability, episodic memory, working memory, internet expertise	Three interfaces, three websites (two on personal finance, one online store)	30 participants

3.2 The Process of Selecting the Participants

In order to understand how exposure time required by the user changes according to their individual user differences, we can look at the structure of the methodologies that were adopted by previous research on individual user differences, and identify any appropriate patterns (Table 3.1).

All previous research papers in this domain begin by selecting a numbers of participants. They then place them into groups according to their individual differences, which are determined through the use of pre-tests. They create the appropriate interfaces (environmental virtual spaces) and have the participants undertake the experiments. Finally some post-tests may be administered if an interesting relationship is found between environmental factors and individual user differences; or between the user differences themselves. This experimental process flow is commonly used, and seems relevant as the foundation for the construction of an experimental methodology.

Since the thesis focuses on discovering the importance of five individual user differences (i.e. gender, orientation skill, cognitive style, system knowledge, and environmental knowledge), the experimental process requires five filters. A high level description of the process of finding and filtering the participants is as follows: i) begin by selecting a large numbers of participants; ii) run pre-tests; iii) place them into groups according to their individual differences; iv) create appropriate interfaces for experimentation; v) have the participants undertake the experiments; finally, vi) some post-tests may be administered if a relationship is found between certain interfaces and individual user attributes, or between the individual differences themselves. This methodology is summarised in figure 3.1.

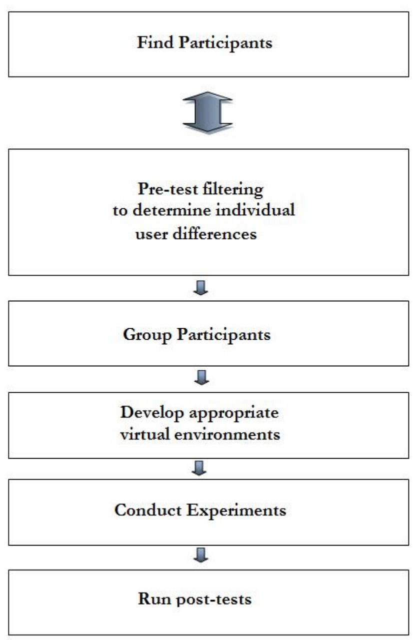


Figure 3.1 – Illustrative summary of methodology

Each square in figure 3.1 represents a task. Each task has a certain requirement that needs to be satisfied before the next task is started. Table 3.2 presents the requirements for each task.

Table 3.2 – How tasks from the discussed process can correspond to the requirements of this thesis.

Task	Research requirement of this thesis
Find Participants	A number of participants are required for the experimental process, in order to extract the impact of their individual user differences and environmental factors on SKA.
Run pre-tests to determine individual differences	The experimental process will need to examine all individual differences separately.
Group Participants	In order to correctly determine the impact of a certain individual user attribute, we ensure control of all others by grouping participants.
Create Interfaces	Four environments must be created in order to ensure that all the environmental factors are studied individually.
Conduct the Experiments	Participants will conduct the experiment on all environments.
Run post-tests	If at any point, the experiment process reveals an interesting relationship between individual differences, or environmental factors then post-tests may be required to further reveal these relationships.

In summary, this thesis will adopt the process displayed in figure 3.1, which has been justified in previous experiments (shown in table 3.1) as a suitable experimental methodology in this domain. The first task that needs to be performed is participant filtering, which is discussed in the next section.

3.3 Choosing the appropriate tests for participant filtering

This section will present a solution for investigating each required experimental property, whilst at the same time nullifying the effects of any other external variable that may interfere with the results. Ultimately this section demonstrates how each participant can be placed into an appropriate participant group through pre-testing. Table 3.3 presents the filters used for each individual difference.

Table 3.3 – Participant filters

	Gender	Cognitive Style	Orientation Skill	Environmental Knowledge	System Knowledge
Filter	Questionnaire	Cognitive Style Analysis test	Guilford-Zimmerman Spatial Orientation Survey	Questionnaire + a certain amount of training in the environment before the experiment begins	Questionnaire + Mouse dexterity test

The following sub-sections justify why the particular filters were used for the participant selection process.

3.3.1 – Gender

We will be using a pre-test questionnaire in order to determine a participant's gender. This method is common, and was also used in a similar study by Waller et al. (1998).

3.3.2 – Cognitive Style

Chen et al. (2005) compared a variety of tools to determine the cognitive styles of their participants. They found that the best test was the CSA (Cognitive Styles Analysis) test. The main alternative test, named the Group Embedded Figures Test proposed by Witkin et al. (1971) has several problems, e.g. levels of field dependence are inferred from poor field independence performance (Ford and Chen, 2001). The CSA is made up of three sub-tests. All sub-tests require the participant to react to a statement or question by tapping “true” or “false”. The first test aims at separating verbalisers from imagers and contains questions which require the participants to choose whether a written statement is true or false. The second sub-test asks the participants to determine whether a pair of complex geometrical objects are identical, which measures field dependent (holist) capacity. The third sub-test is made up of a primitive 3D object such as a square or triangle and a complex shape, the participant must judge whether the simple shape is contained in the complex one (Riding and Grimley 1999). This test measures field-independence (analytic). Figure 3.2 and figure 3.3 show screenshots of the CSA test.

Figure 3.2 – Questions aimed to separate verbalisers from imagers

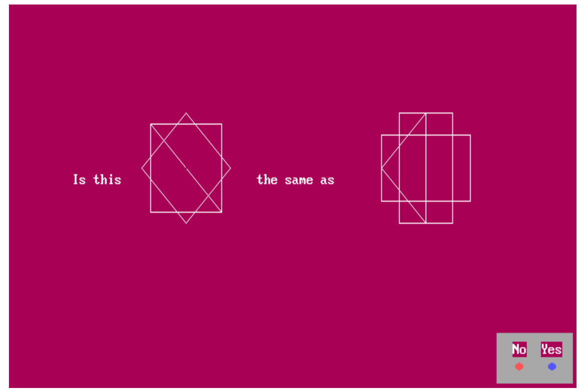
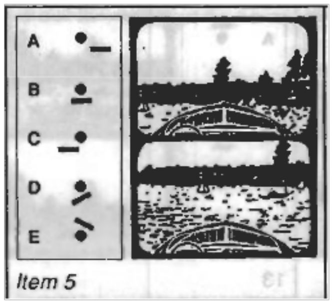


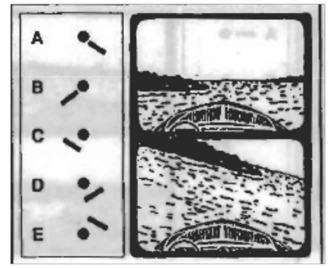
Figure 3.3 – Questions that measure field-dependence

3.3.3 - Orientation Skill

As with previous research in the domain of spatial knowledge acquisition, this thesis will follow the standard way of testing the participant's orientation skills by requiring that they take the Guilford-Zimmerman (GZ) orientation survey as part of the filtering process. This test comprises of various pictures of a boat along with multiple choice answers. Each picture is in a different angle and the users must imagine in what direction the boat moved, as seen in figure 3.4 (Tan et al., 2003). As a result of this test participants may be separated into those of high aptitude skill, and those of low aptitude skill.



The boat has moved left and down, so the correct answer is C



The boat has moved to the left, downwards and changed angle anti-clockwise. Therefore B is the correct answer

Figure 3.4 – Example image questions from the GZ test (Guilford and Zimmerman, 1948)

This test was very difficult to find, since the original publisher is no longer in business, and other publishers did not have a copy of it. Even the British library does not seem to have a copy of this test, and yet it is used as the standard orientation test in the majority of journals and conference papers relating to SKA. One of the key researchers in the area of SKA, who's work is very frequently referenced in this thesis, (Dr Darken) was good enough to provide us a copy of the test.

At first glance the answering system used in the Guilford-Zimmerman test seemed very confusing. This was confirmed, since it took a long time to train participants on how to use the test properly. Regardless, the GZ test is used by the majority of researchers in the domain of SKA. Accordingly, it was important for us to use the same test in order to avoid any variable change that could occur when another test is used (e.g. external factors that could have come into play such as intelligence or perception).

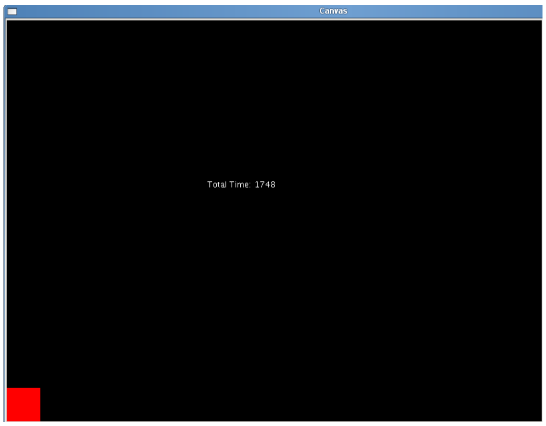
There is some contradictory evidence as to whether psychometric tests, such as the GZ-test, can accurately predict a person's navigational ability (Chase and Chi, 1981; Moeser, 1988). The reason for this debate, relies on the understanding that these tests emphasise on the capacity and processing ability of a person's working memory, whilst environmental knowledge takes much longer, and therefore emphasises long-term memory (Chase and Chi, 1981; Moeser, 1988). However, there is very strong evidence that psychometric tests can significantly predict orientation skill. Thorndyke and Goldin (1983) showed that participants with higher orientation scores, also performed significantly higher in navigation experiments than participants with low orientation scores. There is also some suggestion that orientation scores from tests can better predict SKA from a VE than a real environment. The hypothesis for this, is that people learning from a VE rely more on small-scale visuospatial abilities (i.e. interacting with the screen using the mouse), and therefore see the interaction with the desktop VE as a small-scale interaction rather than a full scale navigation (Chance et al., 1998; Klatzy et al., 1998).

3.3.4 – Environmental Knowledge

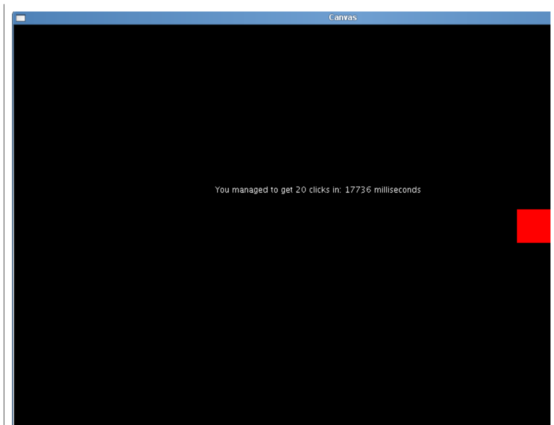
Environmental knowledge is the only individual user difference that can be experimentally altered. In fact, environmental knowledge implies that a participant has experience navigating in a particular environment. Therefore, by allowing a group of participants some “training” time before the actual test begins, an increase of environmental knowledge can be established. This thesis proposes selecting a group of participants, with all the other individual differences controlled for, and giving them five minutes experience in all environments before starting the experiment, in order to give them environmental knowledge.

3.3.5 – System Knowledge

The system knowledge of the participants was tested and controlled through both the use of a questionnaire, and a mouse dexterity test. The dexterity test was developed in Java, using a simple canvas and a few squares (boxes). This program, which has a simple graphical interface, measured the time it takes the participant to click on a box that appeared in a random corner of the screen. Twenty clicks are measured in total, and the total time is given. The participants who declared that they hardly used a computer in the questionnaire, did indeed take almost twice the time to click on the boxes as the ones that stated that they were experts in using a computer. Two screenshots of the dexterity test can be seen in figure 3.5. The code for the developed mouse dexterity test can be found in Appendix B.



The "action" screen, shows the current time and a red square that the user has to click on. Text reads "Total Time 1748"



The end screen. Presents the user with his/her dexterity score. Text reads "You managed to get 20 clicks in 17716 milliseconds"

Figure 3.5 – The Dexterity test

After the pre-tests were complete, participants were filtered down from 100 to 48 people – 8 participants in each participant user group. Table 3.4 presents the participant groups in detail.

Table 3.4 – Participant Groups

	Gender		Cognitive		Orientation		Environmental Knowledge		System Knowledge	
	M	F	A	H	L	H	L	H	L	H
Control	X			X		X	X			X
Gender		X		X		X	X			X
Cognitive Style	X		X			X	X			X
Orientation Skill	X			X	X		X			X
Environmental Knowledge	X			X		X		X		X
System Knowledge	X			X		X	X		X	

(M – Male; F – Female; A – Analytic; H – Holist; L – Low; H – High)

The filter process and the development of these groups, ensured the lack of confounding and external variables, which may influence the results of the experimentation process. External variables would be detrimental to the research, since the scope of the experiments is to conclude on the importance of each individual difference separately.

3.4 Controlling the Environmental Factors

Chapter two discussed the importance of environmental factors on the exposure time required to acquire spatial knowledge from a Virtual Environment. In order to control for these specific factors, different environments must be created that regulate all environmental factors, ensuring that all factors are consistent for all groups. Table 3.5 demonstrates the different types of environments that were developed in order to satisfy the requirements of the experimental process. Environment size is measured in pixels, where every pixel is approximately 60 centimetres in the 3D world. That means that an environment of 256*256 pixels equates to a virtual space of 154 metres squared.

Table 3.5 – Environment Types

	Size		Complexity		Unique Landmark Frequency	
	H	L	H	L	H	L
Large Environment	X			X	X	
Complex Environment		X	X		X	
Low Landmark Environment		X		X		X
Control Environment		X		X	X	

(H – Large/Complex/High; L - Small/Normal/Low)

These environments will be used as a test bed for the experimental process. Before we discuss the actual maps, we will present a definition of the three states (normal, large/high and small/low). After defining these attributes, the thesis will present a way of developing both the maps, and the 3D physics and rendering engine required for the experimental test bed.

3.4.1 - Altering the size

There appears to be no formal definition of what a large scaled environment is as opposed to a small scaled one. Some research has been carried out to investigate large scaled environments, since it is thought that they increase the memory requirements and can increase the difficulty of acquiring spatial knowledge from the environment (Vinson, 1999; Darken and Sibert, 1996). Various solutions are then proposed, such as adding more visual cues and aids, or adding more landmarks. Although these experiments have not compared the difference between a large scale environment and a smaller scaled environment in terms of required exposure time, a reasonable approach to solve this could be to increase the size of the VE without altering any other independent variables. We have already discussed that there does not seem to be a formal definition of what a large scaled environment is. Therefore, this thesis has merely split the size of the environments into three categories; large scale, medium, and small. Since a decision had to be made that both distinguished these environments from each other, but that also did not require too much time for a user to create cognitive maps (otherwise this would have been very burdensome on the the experimental process, and is outside the scope of the thesis). We decided to have 256*256 pixels as the largest size of the environment, and 128*128 as the normal size. In the domains that this research aims to contribute to (i.e. military and emergency training), the environments can range from large scaled outdoor environments, to smaller indoor environments such as buildings. Attempting to cover all types of environments is outside the scope of this thesis. Instead, we focus more on indoor environments where we believe this research will hold most value. Therefore, the two sizes are quite practical since they can easily represent buildings as big as 75 metres squared for the 'normal' environment, and as big as 154 metres squared for the 'large' one.

3.4.2 - Altering the landmark potential

As discussed in chapter two, Stankiewicz and Kalia (2004) present how landmark potential is comprised of visibility, consistency and how descriptive it is to the user. Altering these values renders a landmark more or less useful, and can have a negative impact on learning; especially if the landmark does not persist in the environment i.e. moves to another location (Newman et al., 2007). There is some uncertainty as to what can be considered a landmark, however Stankiewicz and Kalia (2004) argue that there are in fact two types of landmarks; structural and object. Structural landmarks are things such as T-Junctions, dead ends and other informative structures.

Object landmarks are objects such as trees and paintings that do not determine the environmental geometry but are instead simple objects that can be used as navigational aids. The way of ensuring that usable landmarks are present in the environment is presented by Vinson (1999), who provides information on how to add landmarks and make navigation less complex. Therefore the landmark potential can be controlled by the number of unique object landmarks available during navigation. Again, there is no formal definition determining what a high or low amount of unique object landmarks would be in a particular environment. We decided to have two states, a very low state of four object landmarks, and a very high state of 16 object landmarks. We already discussed how research has shown that well placed object landmarks make a huge difference in the navigational complexity of an environment, therefore, by increasing the amount of landmarks four times, we would expect to see some significant variance in result.

3.4.3 - Altering the Spatial Layout Complexity

Spatial layout complexity is the number of objects, such as walls, that obstruct the sight of a user from various reference points (e.g. visible landmarks). A common and simple way of altering the complexity seems to be the use of walls to obstruct the user's vision in a virtual maze. This technique was used by a variety of researchers (Kalia and Stankiewicz, 2007; Schnabel and Kyan, 2001; Marsh and Smith, 2001). This thesis will also adopt this method as it allows for rapid experimentation and easier testing of success by using draft drawing maps. Table 3.6 summarises how the environmental properties must be modified to allow experimentation.

Table 3.6 – Summary of actions necessary to modify the environmental factors

Size	Increase the overall environment size by altering the size of the 2D map. Either 128*128 or 256*256
Landmark potential	Vary the number of unique object landmarks (4 or 16). Adding more structural landmarks is not an option as this is controlled by the actual geometry.
Spatial layout complexity	Increase the amount of corridors, therefore obstructing navigational references .

3.5 Testing the Acquisition of Knowledge

In order to determine whether the participants actually acquired the spatial knowledge Darken and Banker (1998) took the participants to the real world terrain that the VE was representing. Goerger et al. (1998) also used the real world as the test, and took measurements of time to negotiate the path and the number of errors made during each leg of the route.

This thesis aims to contribute towards a better understanding of the properties that affect the required exposure time as a result of individual user differences and environmental factors. In light of this, it is thought that if enough exposure time is allowed, the user should first acquire landmark and procedural knowledge, and if enough time is given during navigation, that knowledge will be converted to survey knowledge (Siegel and White, 1975). This is important for determining whether a participant has in fact acquired spatial knowledge or not. It is theorised that once a user has acquired survey knowledge, they should be able to sketch a draft of the environment. This method was used by Waller et al. (1998) who had his participants construct a map of the environment they were navigating. He then compared the distances of landmarks on their sketch maps to the actual distances. This difference was named the map error, and the smaller this value was, the more knowledge a participant had acquired.

Our experimental process follows a commonly used methodology for determining how successfully a participant has acquired spatial knowledge. Furthermore, we attempt to significantly lower the required exposure time for all participants, by creating environments with right-angled corridors. This removes the need for the participants to apply a large amount of heuristics during the development of cognitive maps. People tend to apply various heuristics to their cognitive maps, thereby creating distortions (Tversky, 1993). These include alignment heuristics, which implies that participants may remember locations as more aligned than they actually are (Tversky, 1981); rotation heuristics, which may lead participants to place locations more vertically or horizontally than they actually are (Tversky, 1981; Chase, 1983; Lloyd and Heivly, 1987; Lloyd, 1989); and angular heuristics, which may lead participants to make angles more right-angled than they actually are (Byrne, 1979; Moar and Bower, 1983; Sadalla and Montello, 1989; Hirtle and Mascolo, 1992). Our experiments remove a significant cognitive load from participants, and, although our methodology will not be suitable for measuring the complexity of outdoor environments which are commonly not right-angled and neatly developed, It can still apply to the vast majority of indoor man-made environments.

Therefore the environment must be made up of corridors with 90 degree angles that are placed horizontally and vertically. The environment size must not be too large, as a large size will take too long to learn and may cause frustration to the participants. Finally since the dependent variable is exposure time, there must exist a way of knowing when a user has acquired adequate spatial knowledge, and if this is not the case (e.g. the user has not actually acquired the knowledge), a mechanism to resume the current state must exist. To accomplish this, in their experiments, Waller et al. (1998) used draft sketch maps to determine whether a participant has acquired spatial knowledge of the environment. Bearing and distance estimations were used to determine if the participant had in fact acquired knowledge of the space successfully. Absolute knowledge transfer (conversion from procedural to survey knowledge) was tested in the Waller et al. (1998) experiment by asking the participants to draw a line to a landmark from their starting position on a draft map. This was also seen in the Foreman et al. (2003) experiment that required the participants to pin-point landmarks on a draft map. The experiment process used for this thesis follows a similar method. The participants must pinpoint the quadrant position, within the current corner of a corridor, of the landmarks on a sketch map. This determines whether the landmark is placed in the correct corridor, and approximately the correct place. If a participant fails to correctly place the landmark, they carry on with the navigation training until they manage to position all the landmarks onto the map correctly. The time taken during navigation will be recorded, and if at any point the participant thinks that they are ready to complete the draft map the timer pauses, only to resume again if the participant fails to demonstrate that they have acquired spatial knowledge of the environment.

3.6 Hypotheses

As discussed in chapter two, findings in SKA indicate that exposure time appears to be a key influencing factor (Witmer et al., 1996; Darken and Banker, 1998; Waller et al., 1998; Goerger et al., 1998; Darken and Peterson, 2001). This thesis therefore considers exposure time as the dependent variable or “effect” when researching SKA. The independent variables are considered to be the environmental factors, such as size and complexity, as well as individual user attributes, such as orientation skill, that affect the exposure time required to acquire spatial knowledge. Chapter two presented (figure 2.3) the stages of the information-processing model affected by environmental properties and user attributes. This leads us to the following hypotheses:

- **H0:** When the size of the environment increases, more objects and area are processed. This leads to a greater information load to be perceived, understood and used in decisions (O'Keefe and Nadel, 1978). This will increase the required exposure time to acquire spatial knowledge.
- **H1:** When the amount of object landmarks increases, navigation difficulty decreases (Vinson, 1999). This leads to a decrease in required exposure time.
- **H2**: When spatial layout complexity is increased, more reference objects are obstructed (Kalia and Stankiewicz, 2007). This leads to an increase in required exposure time.
- **H3:** For a user with a large amount of system knowledge there is less attention divide during exploration (Booth et al., 2000). This leads to a better perception and understanding of the space, and can be a basis for better decision making during navigation.
- **H4:** A user with high environmental knowledge will have more schemas to use as a reference due to experience in navigating within that particular environment and will therefore have a better understanding of the environment and make better decisions (Mania et al., 2005). This will lower required exposure time.
- **H5:** A user with high aptitude/orientation skills will have more spatial memory and better orientation than a user with low aptitude/orientation skills and will therefore acquire knowledge faster. This will result in them requiring less exposure time (O'Keefe and Nadel, 1978; Smith and Millner, 1981; Maguire et al., 1996; Maguire et al., 1999; Maguire et al., 2000).
- **H6:** Female users will require greater exposure time than male users in environments with fewer landmarks, but should require the same exposure time in environments with high landmark potential (Sandstrome et al., 1998).
- **H7**: Field-dependent (holistic learning style) users may take longer to acquire procedural knowledge in complex environments, due to their more passive approach to learning, which leads them to learn more irrelevant

information (Pask, 1976; Pask, 1979), but will perform better in large environments for the same reason.

In order to prove or disprove these hypotheses an environment test bed was created. This development allowed us to investigate the required properties and their effect on SKA. The environment was named the SKAR (Spatial Knowledge Acquisition Research) engine.

3.7 Rapid Development of the maps using SKAR

A maze environment, similar to the one developed by Waller et al. (1998) was required for the experimental process. The reason for using a maze environment is mostly related to development time, experimental consistency, and reducing perception complexity, since It has been theorised that people remember angles as right angles during the development of cognitive maps of an environment (Gillner and Mallot, 1998). The environment will be populated with a variety of landmarks that serve as navigational aids for the testing phase. During testing, participants will be asked to point to the landmark on a paper map representation of the virtual environment. Such experimental methods were used and justified by both Waller et al. (1998), and Foreman et al. (2003).

In order to investigate the impact of single environmental factors on SKA, the created environments are required to control all other factors that may influence the results of the experiment. Table 3.6 displays relative information about each environment that must be developed for the experimental process. All these environments were created using the SKAR engine.

The SKAR engine itself was designed using a storyboarding technique. This technique is useful for brain storming ideas at an early stage of development, and is commonly used for rapid designing of interactive software (Hearst et al., 1998). The storyboarding focused on the draft development of environment maps, which were developed according to the requirements presented in table 3.5. Once the design was completed, an appropriate programming language was chosen to implement the engine. The SKAR engine was developed using Java3D, which is a higher-level set of Java APIs, that use OpenGL and Direct3D (depending on the version of the Java3D SDK). The reason that a custom made 3D space development environment was created, rather than using an off-the-shelf package was that SKAR allows the rapid construction of dynamic 3D environments using 2D bitmaps with gray-scale landscape information and colour-coded bitmaps with landmark potential, while also allowing us to time our participants correctly using a built-in timer which can be paused and resumed accordingly. In the experience of the author, most 3D development packages have a steep learning curve, which is unnecessary for this thesis. We wanted to create some maze-like environments quickly, and be able to modify them whenever we found it necessary without having to use heavy 3D rendering tools and packages. We also noticed that there is no formal way of creating environments for SKA research. So we aim to create a more unified tool that will allow researchers to easily pass spatial maps to each other. The SKAR engine is very simple, and very easy to use, since it allows the designer to visualise and create the maze using a top-down 2D view in any 'paint' application.

The first step is to create a terrain map. This map is a 2D bitmap, which contains the formation of the environment. Using paint, we can create a very simple maze such as the one in figure 3.6.

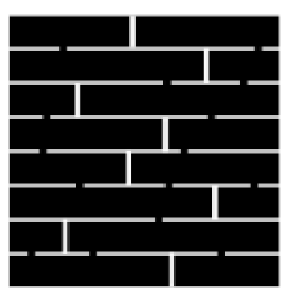


Figure 3.6 – A simple terrain map

This map is very simple, and has only one degree of complexity (n = 1). We measure complexity by looking at the amount of walls blocking the view in every corridor. Each corridor in this case has one wall blocking the horizontal space from one side to the other. The first thing the SKAR engine does, is pick out any pixel without the absolute value of R = 0; where R = red. In computer graphics, pixel colouring happens by altering the red, green, and blue values. The SKAR engine only reads the red value and ignores the other two. However, for the sake of convenience, we add absolutely white pixels to make the walls (which are RGB = 256,256,256). If the SKAR engine finds anything above R = 0, it will lift the floor at the pixel's position according to the value of the pixel. To make a small slope we could add a pixel with value RGB = 20,20,20. To add a wall, we would use RGB = 256,256,256 (which is the colour white). We are only interested in maze-like environments, so the bitmap image in figure 3.6 enables us to quickly make a 3D maze. If we use this image in SKAR, we will get the environment seen in figure 3.7.

Part of the requirements for the SKAR engine is to add landmarks. The process of landmark placement is done separately from the terrain map. In order to place landmarks on the actual map, the engine reads a separate bitmap file which contains pixels with a certain colour coding. For example, the RGB pixel 255,0,0 (Absolute red), can represent a rock object in the environment. Figure 3.8, shows a typical landmark bitmap file, while figure 3.9 shows us the result.

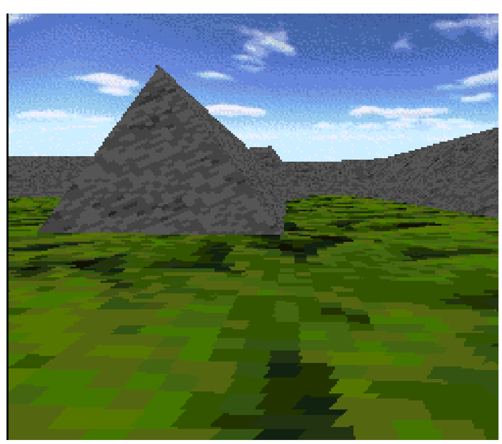


Figure 3.7 – 3D representation of the 2D map seen in figure 3.2

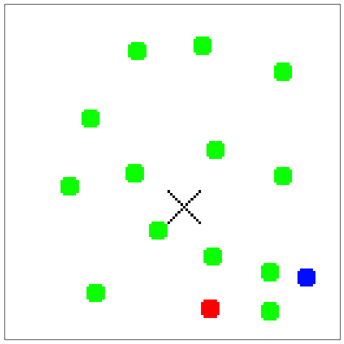


Figure 3.8 – Bitmap landmark map (x is the position of the participant, facing south-east)

Figure 3.8 – Result of the bitmap landmark map seen in figure 3.9

The SKAR engine is quite robust, since it can also parse hilly environments. Although our research focuses on maze-like environments, we believe that this is a nice feature that can help further research on SKA. We briefly demonstrate the results of a complex terrain map in figure 3.10, and the result of this map in figure 3.11. More information on the SKAR engine can be seen in appendix A.

Figure 3.10 – Hilly terrain map

Figure 3.11 – Result of the hilly terrain map as seen in figure 3.10

Four maps were developed in order to allow the creation of the four environments shown in table 3.5. These maps have been summarised in table 3.7.

3.8 The Experimental Process

After taking the pre-tests, the participants were placed into one of the six groups, one group for each individual difference that was tested against (8 participants per group). Four identical personal computers were used; each computer had the

SKAR engine set-up using one of the four maps. The participants were separated into four subgroups of two people per subgroup and each subgroup was assigned to a different environment initially, they then rotated so that all the participants went through each environment only once. This was done to avoid a learning curve during the experiments. Once a participant felt that they had ‘learned’ the environment, a paper map was handed out to the participant. This map was a printed copy of the terrain map for that environment. The participant was then asked to point to various landmarks on the paper map. If that landmark was within the correct quad sector of that room, the participants was deemed to have demonstrated that they had 'learned' the position of the landmark, otherwise they resumed navigation. Each participant had to point to four randomly selected landmarks in order to show that they had acquired spatial knowledge. To better demonstrate how this works, figure 3.12 shows how a participant would place a landmark on the draft map.

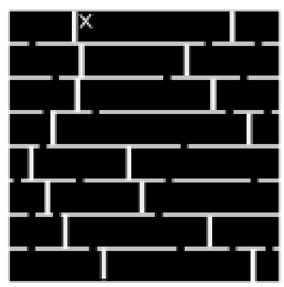
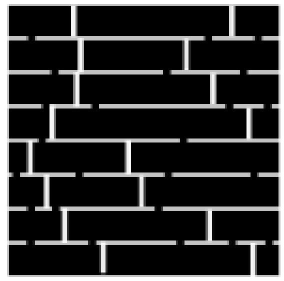
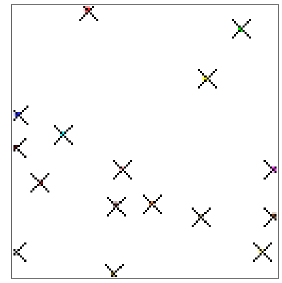
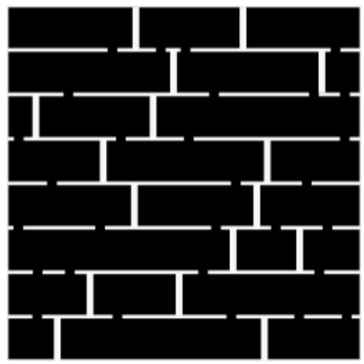
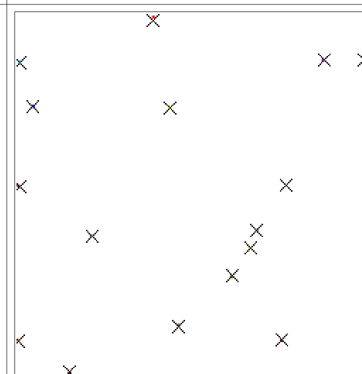
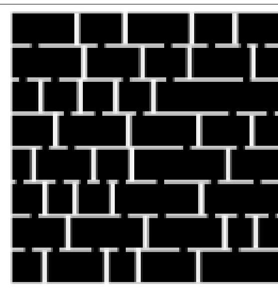
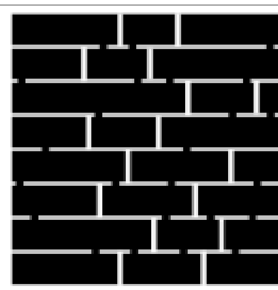
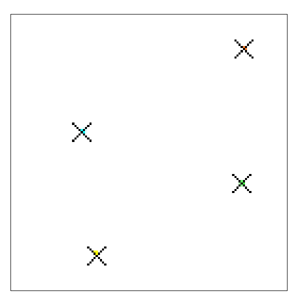


Figure 3.12 – Participant was asked to mark where the rocket is in the control environment. Notice how the 'X' is in the top-left quadrant of the room. The answer was therefore correct.

After the experiments, a post-questionnaire was handed out to the participants, which simply asked whether they had any trouble with the SKAR experiments in general. More details on the results of the experiments, as well as issues that were identified during the post-questionnaire can be seen in chapter four.

Table 3.7 – The four maps

Environment Type	Terrain map	Landmark map
Control environment: 16 unique landmarks, 128*128 pixel size, two obstructions per row, that means there are two rooms per row where each row is 8 pixels high.		
Large environment: 16 unique landmarks, 256*256 pixel size, two obstructions per row.		
Complex environment: 16 unique landmarks, 128*128 pixel size, four obstructions per row.		
Low landmark environment: 4 unique landmarks, 128*128 pixel size, two obstructions per row.		

3.9 Summary

This chapter presented the experimental process and methodology of the thesis. We discussed the design of the experimental process, as well as the implementation methodology adopted for the environmental test beds. We discussed how the SKAR engine, which is to be used as the environment test bed, was designed and developed. Moreover, we considered how the acquisition of spatial knowledge was tested during the experimental process. The results of the experiments, will now be discussed in chapter four, whilst certain issues that arose during the experiments, concerning the use of the GZ test, will be discussed in more detail in chapter five.

Chapter 4 - Experimental Results

4.1 Introduction

The thesis has so far discussed the environmental factors, and individual user differences that influence the total exposure time required for a user to acquire spatial knowledge from a VE. We have demonstrated an experimental methodology, and described the implementation of an environmental test bed that will be used to discover whether these individual user differences, and environmental factors, influence the exposure time required by various users. In summary, the individual differences were:

- Gender, which we hypothesised would have an impact on exposure time in landmark-poor environments.
- System knowledge, which we hypothesised would directly influence the navigational ability of a user due to lower attention divide, and therefore lower the required exposure time in all types of environments.
- Environmental knowledge, which we hypothesised would directly influence the navigational ability of a user, and therefore lower the required exposure time in all types of environments.
- Orientation skill, which we hypothesised would directly influence the navigational ability of a user by allowing for less disorientation, and therefore lower the required exposure time in all types of environments.
- Cognitive styles, where we hypothesised that field-dependent users would take a shorter time to acquire spatial knowledge from large environments due to their tendency to navigate in a holistic manner (and would therefore process more redundant information during navigation), whilst field-independent users would take a shorter time in complex environments (due to their more analytical approach to problem solving).

The environmental factors are the attributes that determine the structure of an environment. Previous research dictates that the factors that have an impact on spatial knowledge acquisition (SKA) are:

- Size, which we hypothesised would influence the required exposure time directly, since larger distances would have to be traveled by the user and therefore a larger area would have to be perceived and processed.
- Spatial complexity, which we hypothesised would influence the required exposure time directly, since more objects (object landmarks, geometrical landmarks) would not be perceived immediately by the user and would lead to loss of spatial awareness.
- Landmark potential, which we assumed would help navigation and therefore lower the required exposure time when a large frequency of unique landmarks were available in the environment.

The aim of this thesis is to understand whether various individual user differences, and environmental factors impact on the exposure time required by a person to acquire spatial knowledge from a VE. We would also seek to discover just how much impact the individual user differences and environmental factors have independently of each other. The previous chapter discussed the experimental process, which was used to collect data, in order to help us satisfy this aim, in this chapter we present the results of our initial experiments, and consider the statistical significance of the individual differences and environmental factors through statistical analysis.

4.2 Impact and Significance of the Individual User Differences

This section discusses the results acquired for each particular group (gender, orientation skills, cognitive style, as well as system and mental knowledge), and tests their significance when navigating in the four different environments. These environments have been presented in detail in chapter three, section 3.8, but to summarise:

- Control environment, used as a benchmark, for which the other three environments differ in only one factor. This environment allows us to determine the impact of each environmental factor separately. The control environment is

128*128 pixels of size (on the 2D map), has two obstructions per row, where each row is eight pixels high, and is populated with 16 unique object landmarks. Each pixel is equivalent to 60 centimetres of real space.

- Large Environment, which is four times the size of the control environment. See section 3.4.1 in chapter three for more details on measuring size.
- Complex environment, which has four obstructions (walls) per row where each row is eight pixels high. See section 3.4.3 in chapter three for more details on measuring complexity.
- Environment with fewer unique landmarks, which only had four unique object landmarks. See section 3.4.2 in chapter three for more details on measuring landmark potential.

The significance for each individual user difference is tested against the results acquired for the control group. The control group is made up of participants that had specific individual user differences, and therefore worked as a benchmark to help determine the impact of each user difference separately. The results of the control group have been clearly summarised into figure 4.1.

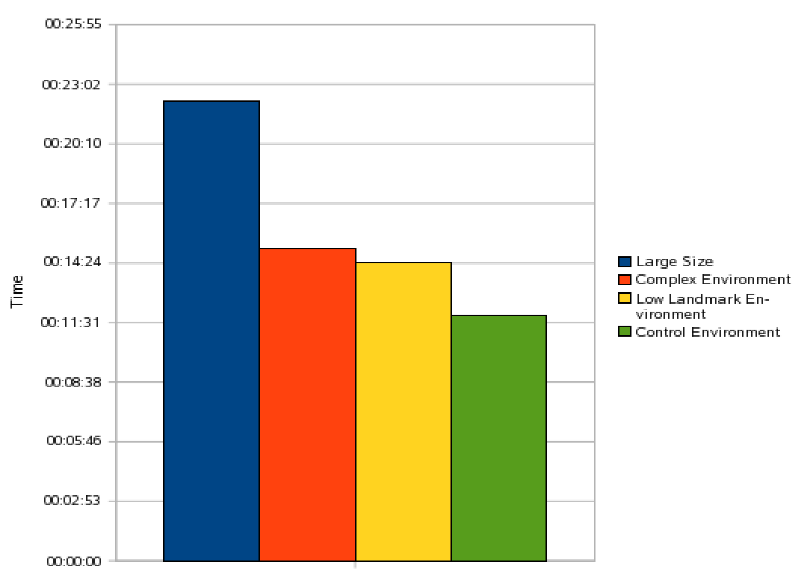


Figure 4.1 – Control Group

(Large size – 0:22:14; Complex Environment – 0:15:06; Low Landmark Environment – 0:14:24; Control Environment – 0:11:52)

In order to test whether the mean result taken from all the environments is significant, when compared to the control environment in respect to time, we kept the data only from the environments we wanted to check, and removed the data from the other two. To see whether the value of the large size environment is significantly different to the value of the control environment, we removed the low landmark and complex data and ran a univariate analysis of variance, with size as the independent variable, and time as the dependent variable. It seems that size has a significant impact on exposure time requirement {$F(1,1) = 395.913$, $P < 0.01$}. To see whether the value of the complex environment is significantly different to the value of the control environment, we removed the low landmark and large size environment data and ran a univariate analysis of variance, with complexity as the independent variable, and time as the dependent variable. Our results showed that complexity has a significant impact on exposure time requirement {$F(1,1) = 18.717$, $P < 0.01$}. Finally, to see whether the value of the environment, populated with a low amount of unique object landmarks, is significantly different to the value of the control environment, we removed the complex and large size environment data and ran a univariate analysis of variance, with landmark value as the independent variable, and time as the dependent variable. Our results showed that landmark value has a significant impact on the exposure time required for SKA to fully occur {$F(1,1) = 19.093$, $P < 0.01$}.

All the environmental factors have a significant impact on the control group during the process of SKA, and should therefore be considered during training in order to ensure that the correct amount of exposure time is given. If at any time, a single environmental factor changes, then the results show that this can have a significant impact on the exposure time required to acquire spatial knowledge. It is therefore vital that we measure a training environment properly, before attempting to train users and hope that they acquire spatial knowledge. Size, in our results, is the most significant, which might be as a result of increased locomotion duration, or burden on cognitive load. For the control group of participants, the impact of size is more significant than either the impact of an increased number of potential landmarks, or an increased spatial complexity. Univariate analysis of variance shows that the time taken in the large size environment is significantly different from the complex environment {$F(1,1) = 90.551$, $p < 0.001$}, and the low landmark environment {$F(1,1) = 181.222$, $p < 0.001$}. When determining whether complexity or landmark value are more significant to exposure time, we found no significant difference in terms of the burden on exposure time. Table 4.1 summarises the F-distribution, degrees of freedom, and significance.

Table 4.1 – F, df and P values for a within comparison of the different environments using the control group

	F	df	P
Large Size VS Complex Environment	90.551	1	< 0.001
Large Size VS Low Landmark Environment	181.222	1	< 0.001
Large Size VS Control Environment	395.913	1	< 0.001
Complex VS Low Landmark Environment	0.786	1	< 0.001
Complex VS Control Environment	18.717	1	< 0.001
Low Landmark VS Control Environment	19.093	1	< 0.001

4.2.1 – Significance of Gender

The difference in performance during navigation in a VE, as well as the difference in spatial awareness in a VE between users of different gender, has been an area of interest for quite some time. Bryden and Tapley (1977) as well as Petersen and Linn (1985) identified that females were less capable at orientation than males in a VE. Waller et al. (1998), found that females are particularly disorientated in a virtual maze, with large bearing errors, and have difficulties in drawing the maze that they just navigated through. Moffat et al (1998) reported that males learn a virtual maze faster than females, while Crook et al (1993) suggest that males learn a topographical map faster than females.

The gender group was made up of eight females, who scored similar scores in the pre-tests to the control group, so it differed only in respect to gender. In order to check whether the mean result taken from each environment is significantly different to the result taken from the control environment, we kept the data only from the environment we wanted to check, and removed the data from the other two. We looked at whether the large size environment is significantly different to the value of the control environment by removing the low landmark and complex data and running a univariate analysis of variance, with size as the independent variable, and time as the dependent variable. The results for size were similar to the control group, $\{F(1,1) = 214.390, P < 0.001\}$ and again, size was significant. To see whether the value of the complex environment is significantly different to the value of the control environment, we removed the low landmark and large size environment data and ran a univariate analysis of variance, with complexity as the independent variable, and time as the dependent variable. Our results show that complexity has a significant impact on the exposure time requirements, much like it did for the control group $\{F(1,1) = 17.768, P = 0.001\}$. In order to find whether the value of the low landmark environment is significantly different to the value of the control environment, we removed the complex and large size environment data and ran a univariate analysis of variance, with landmark value as the independent variable and time as the dependent variable. Our results show, much like they did for the control group, that landmark value has a significant impact on the exposure time required for SKA to occur {$F(1,1) = 144.399$, $P < 0.001$}. The only difference when looking at the in-between results of the gender group when compared to the control group, is that unlike the control group, the difference between the results taken from the complex environment and low landmark environment are significantly different {$F(1,1) = 41.395$, $P < 0.001$}. The results of the within comparison of the impact of gender within different environments have been summarised in table 4.2.

Table 4.2 – F, df and P values for a within comparison of the impact of gender within different environments.

	F	df	P
Large Size VS Complex Environment	89.979	1	< 0.001
Large Size VS Low Landmark Environment	16.704	1	= 0.001
Large Size VS Control Environment	214.390	1	< 0.001
Complex VS Low Landmark Environment	41.395	1	< 0.001
Complex VS Control Environment	17.768	1	= 0.001
Low Landmark VS Control Environment	144.399	1	< 0.001

We compared each result of the gender group from every environment, to the results of the control group from every environment in order to see if there is a significant difference between them. The results in comparison to the control group can be seen in Figure 4.2.

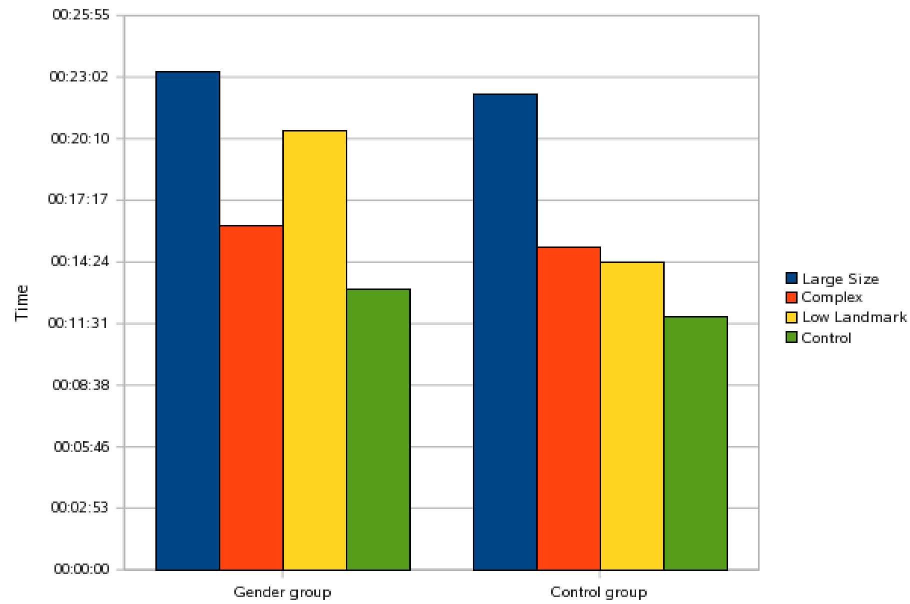


Figure 4.2 – Gender Group versus Control Group

(Large size – 0:23:19; Complex Environment – 0:16:04; Low Landmark Environment – 0:20:31; Control Environment – 0:13:05)

By running an independent samples T-test, with time as the test variable, and gender as the grouping variable, we found that the difference was only significant in the low landmark environment; with females performing much worse (Mean Difference 0:06:07; Std Err 0:00:37, $P < 0.001$). For the large size environment the results were not significantly different (Mean Difference 0:01:05; Std Err 0:00:39, $P = 0.117$). They were also not significantly different for complexity (Mean Difference 0:00:57; Std Err 0:00:50, $P < 0.276$). Finally, the results taken from both groups in the control environment showed they had strong trends, with women seemingly taking longer; however, the results were not experimentally significant (Mean Difference 0:01:13; Std Err 0:00:34, $P = 0.052$). More investigation is required to ensure that the significance level does not change if the sample size increases, or if a different control group is used.

The implications of our findings are that during environmental training, females should not require more training time than males, except in environments with a low amount of unique object landmarks (e.g. a desert; or smoke filled environment). This contributes much to the area of emergency and military training, since it is as an indication of how much additional exposure time a female soldier would require in a virtual representation of a certain landscape. In retrospect the findings agree with Sandstrome et al. (1998), that implied that females navigate more on landmarks than males, and indicate that perhaps the reason that females performed lower in the Waller et al. (1998) experiments was because the virtual maze had a low amount of unique object landmarks.

4.2.2 – Significance of Orientation Skill

In chapter two we showed that perhaps the most discussed individual difference relating to navigation is orientation skill (OS). A large variety of researchers have controlled OS by issuing pre-tests before the experimental process (Witmer et al., 1996; Goerger et al., 1998; and Waller et al., 1998). It seems, therefore, that spatial orientation skill is considered an influential attribute during navigation (Egan and Gomez, 1985; Gomez et al., 1986; Vicente et al., 1987; Stanney and Salvendy, 1995). In virtual environment research, spatial orientation is the ability of a user to orientate in space relative to objects and events, and be aware of self-location (Rebel, 1985). Furthermore, there is strong evidence that individuals have different orientation abilities, which are simply dependent on biological factors; such as a large hippocampus area (O'Keefe and Nadel, 1978; Smith and Millner, 1981; Maguire et al., 1996; Maguire et al., 1999; Maguire et al., 2000). The most widely used orientation test is the Guilford-Zimmerman orientation survey, although other test exist as well, such as the Eliot-Price Test (Eliot and Price, 1976) and the Stumpf – Fay Cube Perspectives test (Stumpf and Fay, 1983).

The OS group in this thesis differed to the control group, as they had the eight lowest scores in the Guilford Zimmerman orientation survey. There may be a problem with the scores, however, which we simply cannot ignore. An overwhelming amount of participants complained that the Guilford-Zimmerman orientation survery (GZ-test) was simply too difficult for them to understand and use. The source of these complaints was the extremely confusing dot-line answer key used by the GZ-test (as discussed in chapter three, section 3.3.3). This is a very serious problem, that may be responsible for false results, indicating that a cognitive load is applied that affects participant's understanding. This cognitive load may not rely on just orientation skill, but also other cognitive and biological abilities as well (possibly even including intelligence). If this is true, then not only are the results in this section unreliable, but also the results of every research paper that has used the GZ-test as a way of finding a participant's orientation skill as well. To further understand whether this is the case we had to create a new GZ-test, replacing the old dot-line answer keys with arrows. Chapter five discusses the new GZ-test in detail. At this point, we will presume that the GZ-test results are correct, until discussed in more detail in chapter five.

In order to test whether the mean result taken from all the environments is significant, when compared to the control environment in respect to time, we kept the data only from the environment that we wanted to check, and removed the data from the other two. To see whether the value of the large size environment is significantly different to the value of the control environment, we removed the low landmark and complex data and ran a univariate analysis of variance, with size as the independent variable and time as the dependent variable. The results for size were {$F(1,1) = 489.268$, $P < 0.001$}. This was similar to the control group, and was also found to be significant. To see whether the value of the complex environment is significantly different to the value of the control environment, we removed the low landmark and large size environment data and ran a univariate analysis of variance, with complexity as the independent variable and time as the dependent variable. Our results show that complexity has a significant impact on exposure time required much like it did for the control group {$F(1,1) = 27.116$, $P < 0.001$}. In order to find whether the value of the environment, populated with a low amount of unique object landmarks, is significantly different to the value of the control environment, we removed the complex and large size environment data and ran a univariate analysis of variance, with landmark value as the independent variable and time as the dependent variable. Our results show, much like they did for the control group, that landmark value has a significant impact on the exposure time required for SKA to occur {F(1,1) = 40.250, P < 0.001}. Although the results from all the environments in the low orientation skill group were much lower than those from the control group, the shape of the curve between the results does not change. This implies that a simple multiplier value may be used to correct the required exposure time given to users with a low orientation skill. The results of the within comparison of the impact of orientation skill within different environments have been summarised into table 4.3.

Table 4.3 – F, df, and P values for a within comparison of the impact of orientation skill within different environments

	F	df	P
Large Size VS Complex Environment	258.581	1	< 0.001
Large Size VS Low Landmark Environment	295.952	1	< 0.001
Large Size VS Control Environment	489.268	1	< 0.001
Complex VS Low Landmark Environment	0.286	1	= 0.601
Complex VS Control Environment	27.116	1	< 0.001
Low Landmark VS Control Environment	40.250	1	< 0.001

We compared each result of the orientation skill group from every environment, to the results of the control group from every environment in order to see if there is a significant difference between them. A visual representation of how these results compare to the control group can be seen in Figure 4.3.

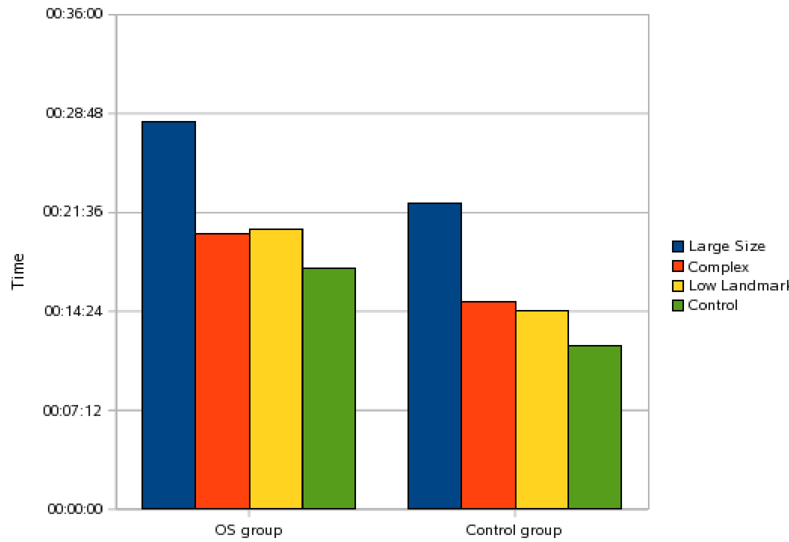


Figure 4.3 – OS group versus Control Group

(Large size – 0:28:09; Complex Environment – 0:20:06; Low Landmark Environment – 0:20:21; Control Environment – 0:17:33)

By running an independent samples T-test, with time as the test variable, and orientation skill as the grouping variable, we found that the difference was significant in all the environments, with participants of higher orientation skill greatly outperforming those with low orientation skill for the control environment (Mean Difference 0:05:41; Std Err 0:00:29, P < 0.001). For both the large size environment (Mean Difference 0:05:54; Std Err 0:00:30, P < 0.001), complexity (Mean Difference 0:04:54; Std Err 0:00:44, P < 0.001) and for low landmark environments (Mean Difference 0:51:56; Std Err 0:00:32, P < 0.001) the results were identified as being significantly different. This is an obvious indication that OS is a very important individual user difference, and is unfortunately not so easily trained. Accordingly, a user with low OS will require increased exposure time. This justifies our argument that a one-size fits all exposure time is not appropriate.

Although, studies have shown that orientation skill can be increased through training (Maguire et al., 2000), training this skill can be a long and burdensome process that is probably better dealt with in a real life situation by administering as many cues and navigational aids as possible to a person with a low OS skill, while at the same time allowing as much exposure time to the training environment as possible.

4.2.3 – Significance of Cognitive Style

A question posed in chapter two, was whether field-independent users would have better scores in a landmark-rich environment than field-dependent users. We initially hypothesised that field-independent users would generally acquire spatial knowledge faster than field-dependent users, due to their tendency to learn faster in a less-procedural way (such as that of traditional maps). The cognitive-style group was identical to the control group, except that participants had a verbal-analytical cognitive style.

To check whether the mean result taken from all the environments is significant when compared to the control environment in respect to time, we kept the data only from the environment we wanted to check, and removed the data from the other two. To see whether the value of the large size environment is significantly different to the value of the control environment, we removed the low landmark and complex data and ran a univariate analysis of variance, with size as the independent variable and time as the dependent variable. The difference between the large size environment and the control environment was significant {$F(1,1) = 970.860$, $P < 0.001$}. To see whether the value of the complex environment is significantly different to the value of the control environment, we removed the low landmark and large size environment data and ran a univariate analysis of variance, with complexity as the independent variable, and time as the dependent variable. The difference in results between the complex environment and the control environment were not significant {$F(1,1) = 0.128$, $P = 0.725$}, however it was significant between the complex environment and the low landmark environment {$F(1,1) = 21.863$, $P < 0.001$}. This differs completely to the control group, and implies that analytic-verbalisers have less trouble navigating in a complex environment, and therefore seem to hold an advantage in complex maze-like environments. Ultimately, the shape of the curve, showing the variation between results, is different to the control group since the results from the complex environment were much lower. The results of the within comparison of the impact of cognitive styles within different environments have been summarised into table 4.4.

Table 4.4 – F, df and P values for a within comparison of the impact of cognitive style within different environments

	F	df	P
Large Size VS Complex Environment	684.483	1	< 0.001
Large Size VS Low Landmark Environment	602.385	1	< 0.001
Large Size VS Control Environment	970.860	1	< 0.001
Complex VS Low Landmark Environment	21.863	1	< 0.001
Complex VS Control Environment	0.128	1	0.725
Low Landmark VS Control Environment	27.464	1	< 0.001

We compared each result of the cognitive style group from every environment, to the results of the control group from every environment in order to see if there is a significant difference between them. The way these results compare to the control group can be seen in Figure 4.4.

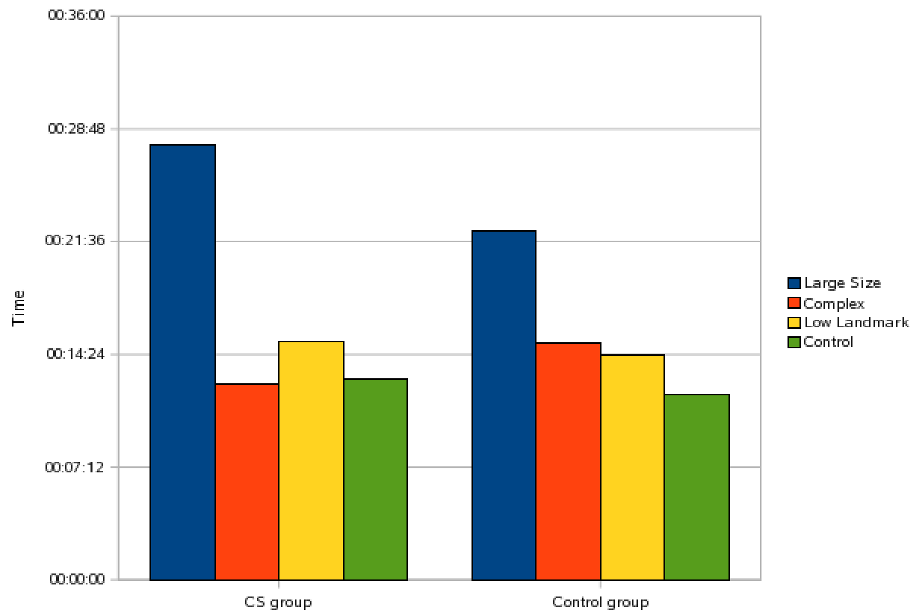


Figure 4.4 – Cognitive Styles Group versus Control Group

(Large size – 0:27:44; Complex Environment – 0:12:34; Low Landmark Environment – 0:15:14; Control Environment – 0:12:46)

By running an independent samples T-test, with time as the test variable, and cognitive style as the grouping variable, we found that the difference was not significant in the control environment (Mean Difference 0:00:53; Std Err 0:00:28, P = 0.084). For the large size environment, the results were significantly different to the control group (Mean Difference 0:05:19; Std Err 0:00:30, P < 0.001); with the visual-holistic users of the control group performing better than the verbal-analytical group. The results of the complex environment for each group were also significantly different (Mean Difference 0:02:32; Std Err 0:00:47, P = 0.006), with the verbal-analytical cognitive styles group performing better. Finally, there was not a significant difference between the results taken from the low landmark environment (Mean Difference 0:00:50; Std Err 0:00:34, P = 0.167), which implies that the number of landmarks does not significantly affect the results of people with these different cognitive styles.

These results prove our original hypothesis that field-dependent individuals (Holist-

Imagers) will always take longer to acquire spatial knowledge from a complex environment. Although there does not seem to be a single reason why field-independent users (Verbal-analytic cognitive styles group) scored better in the complex environment, we can make some assumptions, based on findings from literature. It seems that field-dependent people are more passive when learning information, and therefore prefer to learn by focusing on information as a whole, rather than breaking it down (Pask, 1976; Pask, 1979). By increasing the complexity of an environment, the number of available visual cues are decreased. This may interfere with the field-dependent user's passive learning, since the environment suddenly requires more procedural learning and needs to be 'broken down' during the creation of the cognitive map. Field-independent users are more inclined towards breaking down information, and seemingly have an obvious advantage in complex environments.

As to the question why field-dependent users performed much better than field-independent users in the large size environment, further research is required in order to better understand why this occurs. We hypothesise that it probably has to do with a difference in the available spatial memory between the two groups during spatial processing, updating and creation of the cognitive map. It may be that due to the ‘holistic’ approach of learning, as the user's mind is accommodated for the passive approach to spatial learning by allowing more memory for spatial processing. It may also, however, be related to the tendency that this group of users have towards learning through visual means. This implies that by manipulating the exposure time accordingly, we can now provide adequate training to both field-dependent, and field-independent individuals to acquire spatial knowledge. We now know that more exposure time is required by field-dependent users in complex environments, while more exposure time is required by field-independent users in large scale environments. These findings contribute significantly in the area of emergency and military training, since chances are that most trainees will differ significantly in their learning styles. Trainers need to be aware of the advantages and disadvantages of both learning styles as far as spatial learning is concerned.

4.2.4 – Significance of System Knowledge

The importance of experience and knowledge was discussed in chapter two, and has been shown to have a significant effect on learning (Booth et al., 2000). System knowledge, is the user's past experience and understanding of the training system. This includes the input and output devices, and in this thesis is considered the ease of use that participants experience whilst navigating in a virtual environment using a mouse and keyboard. Paas et al. (2003), stated that different interaction methods impose varying cognitive loads onto the user. If the system is novel or complex, the user's ability to complete a certain task (in this case to acquire knowledge of the space in the VE) can be constrained through the fragmentation of the already very limited available working memory (Cooper, 2004). This is because the user will have to focus on learning other tasks at the same time (such as understanding how to use the keyboard to move). Accordingly, low system knowledge results in a mental overload via a process called 'attention divide' (Booth et al., 2000). The 'system knowledge' group was made up of participants that had little or no knowledge of computers, and had the lowest scores on the mouse dexterity test which was created to test a participants ability with the system's input devices.

To check whether the mean result taken from all the environments within the system knowledge group is significant when compared to the control environments, we kept the data only from the environment we wanted to check, and removed the data from the other two environments. To see whether the value of the large size environment is significantly different to the value of the control environment, we removed the low landmark and complex data and ran a univariate analysis of variance, with size as the independent variable and time as the dependent variable. The difference between the large size environment and the control environment was significant {$F(1,1) = 35.981$, $P < 0.001$}; despite the fact that the differences between the scores from these two environments where smaller than the differences from the two in the control group. To see whether the value of the complex environment is significantly different to the value of the control environment, we removed the low landmark and large size environment data and ran a univariate analysis of variance, with complexity as the independent variable, and time as the dependent variable. The difference in results between the control environment and both the complex environment {$F(1,1) = 5.907$, $P = 0.029$} and the low landmark environment {$F(1,1) = 7.656$, $P = 0.015$} were significant, although those in the system knowledge group had a lower significance than those in the control group. Finally, the difference between the complex environment and the low landmark environment was found not to be significant, as was the case with the control group{$F(1,1) = 0.011$, $P = 0.919$}. Hence, the shape of the curve is similar to the control group, although the differences in scores are slightly smaller between the environments. The results of the within comparison of the impact of system knowledge within different environments have been summarised in table 4.5.

Table 4.5 – F, df and P values for a within comparison of the impact of system knowledge within different environments

	F	df	P
Large Size VS Complex Environment	11.646	1	0.004
Large Size VS Low Landmark Environment	15.007	1	0.002
Large Size VS Control Environment	35.981	1	< 0.001
Complex VS Low Landmark Environment	0.011	1	0.919
Complex VS Control Environment	5.907	1	0.029
Low Landmark VS Control Environment	7.656	1	0.015

We compared the results of the system knowledge group (in every environment), to the results of the control group in order to see if there is a significant difference between them. The way these results compare to the control group can be seen in Figure 4.5.

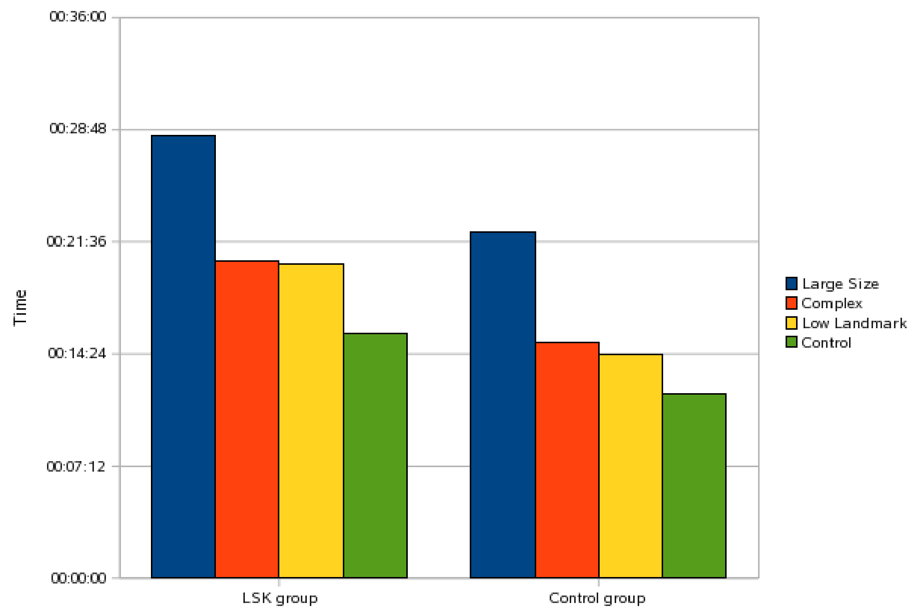


Figure 4.5 – Low System Knowledge Group versus Control Group

(Large size – 0:28:28; Complex Environment – 0:20:21; Low Landmark Environment – 0:20:09; Control Environment – 0:15:45)

It is interesting to note that the shape of the two data sets in figure 4.5 are very similar. This implies that either the users should be trained on the system before VE training, or have longer exposure time in the system.

By running an independent sample T-test, with time as the test variable, and system knowledge as the grouping variable, we found that the difference in scores was very significant between the system knowledge group, and the control group in the control environment (Mean Difference 0:03:53; Std Err 0:01:09, $P = 0.005$). The time taken for the control group was significantly different to the time taken by the system knowledge group in both the large size environment (Mean Difference 0:06:14; Std Err 0:01:51, $P = 0.005$), the complex environment (Mean Difference 0:05:15; Std Err 0:01:48, $P = 0.007$), and low landmark environment (Mean Difference 0:5:45; Std Err 0:01:14, $P < 0.001$). The results confirm to us the importance of system knowledge. All the results taken from the four environments were significantly different to the results from the control group in those same environments. We have therefore shown, that system knowledge, the most easily trainable skill, significantly impacts the required exposure time for all environments, which strongly support the findings of Booth et al. (2000).

Participants with a low level of system knowledge required significantly longer exposure times in all environments, when compared to the control group. The implication of this on navigation training, is that people with less experience of a particular system or device interface will have a significant disadvantage when using this system or interface to acquire spatial knowledge from a VE; when compared to those people who are more educated in the use of the system. Accordingly, people with lower experience with the system or interface will require more exposure time to ensure that they are able to acquire the same level of spatial knowledge.

4.2.5 – Significance of Environmental Knowledge

As with system knowledge, environmental knowledge is also an individual difference that relates to knowledge and experience. Research suggests that a user may find it more difficult to navigate through a novel environment, than one with characteristics with which they are familiar. Of course, learning to navigate through a certain type of environment (such as mountainous terrain) can be learned, however it requires time and training. Training can be costly, or time consuming, therefore understanding the impact of this skill is important to understand the required exposure time when assimilating information from a VE. The importance of environmental knowledge has been tested in research areas such as interface design; where experiments identify the difference of experts versus novices during navigation (Egan, 1988; Dix et al., 1993; Eberts, 1994). In our experiments, the environmental knowledge group was made up of participants that had a five minute 'play' in every single environment (the order was randomly allocated to them) before the experimental test duration started. In other words, the participants spent five minutes in the control environment, five minutes in the large environment, five minute in the complex environment, and five minutes in the low landmark environment before the experiments actually started. Low environment knowledge participants were instructed to navigate through every environment with no description of the task being given to them. Subsequently, they were learning the environment in a passive manner, rather than an active one.

To check whether the mean result taken from all the environments, within the environmental knowledge group, is significant when compared to the control environment, we kept the data only from the environment we wanted to check, and removed the data from the other two. To see whether the value of the large size environment is significantly different to the value of the control environment, we removed the low landmark and complex data and ran a univariate analysis of variance, with size as the independent variable and time as the dependent variable. The difference between the large size environment and the control environment was significant {$F(1,1) = 233.593$, $P < 0.001$}. To check whether the value of the complex environment is significantly different to the value of the control environment, we removed the low landmark and large size environment data and ran a univariate analysis of variance, with complexity as the independent variable, and time as the dependent variable. The difference in results between the complex environment and the control environment were significant {$F(1,1) = 23.767$, $P < 0.001$}. The difference between the low landmark environment and the control environment was also significant, but only by a very small amount {$F(1,1) = 12.016$, $P = 0.04$}. Finally, the difference between the complex environment and the low landmark environment was not significant, as was the case with the control group{$F(1,1) = 0.861$, $P = 0.369$}. The results of the within comparison of the impact of environmental knowledge within different environments have been summarised in table 4.6.

Table 4.6 – F, df, and P values for a within comparison of the impact of level of environmental knowledge within different environments

	F	df	P
Large Size VS Complex Environment	102.164	1	< 0.001
Large Size VS Low Landmark Environment	104.870	1	< 0.001
Large Size VS Control Environment	233.593	1	< 0.001
Complex VS Low Landmark Environment	0.861	1	0.369
Complex VS Control Environment	23.767	1	< 0.001
Low Landmark VS Control Environment	12.016	1	0.04

We compared each result of the environmental knowledge group from every environment, to the results of the control group from every environment in order to see if there is a significant difference between them. The way these results compare to the control group can be seen in Figure 4.6.

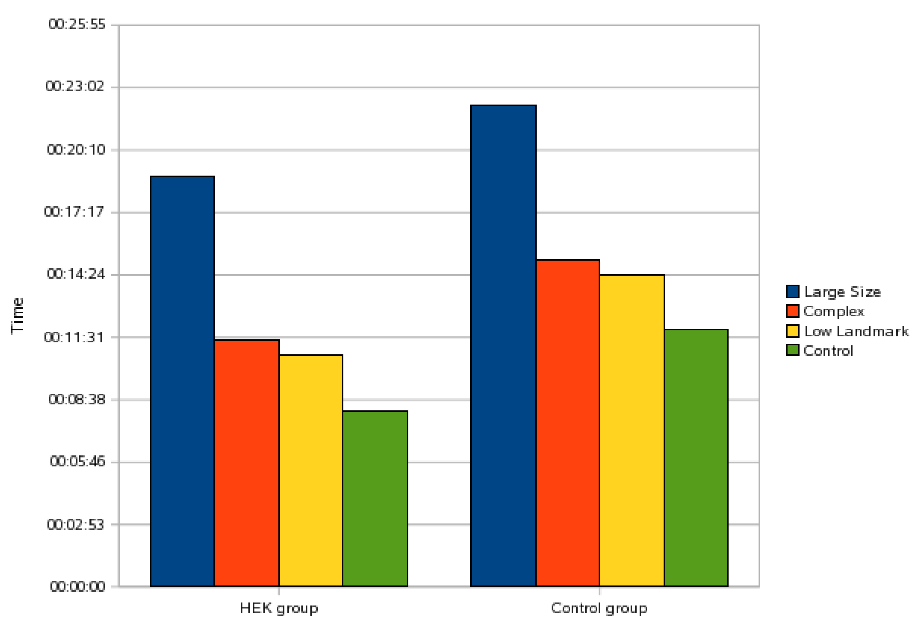


Figure 4.6 – High Environmental Knowledge Group versus Control Group

(Large size – 0:18:57; Complex Environment – 0:11:25; Low Landmark Environment – 0:10:42; Control Environment – 0:08:07)

To compare between the environmental knowledge group, and the control group, we conducted a separate independent samples T-test for each environment. This allowed us to determine whether the high level of environmental knowledge significantly impacts the time taken to acquire spatial knowledge from the different environments. With time as the test variable, and environmental knowledge as the grouping variable, we found that the difference in scores was significant in the control environment (Mean Difference 0:03:44; Std Err 0:00:29, $P < 0.001$). For the large size environment, the results of the environmental knowledge group were significantly different to the control group (Mean Difference 0:03:17; Std Err 0:00:39, $P < 0.001$). The results of the complex environment were also significantly different (Mean Difference 0:03:41; Std Err 0:00:49, $P = 0.001$). For the low landmark environment, the test showed a significant difference between those with high and low levels of environmental knowledge (Mean Difference 0:03:42; Std Err 0:00:44, $P < 0.001$). The difference in scores between users of high, and low levels of environmental knowledge were all significantly different. It is interesting that even though the extra exposure time was short, participants still showed a significant decrease in the overall time taken to acquire spatial knowledge from all environments.

The results suggest that experienced navigators within a certain type of environment will almost certainly have the competitive advantage when it comes to acquiring spatial knowledge. In fact, our research has shown that experts gain information faster from familiar terrain in the VE than novices, and therefore can benefit greatly from VE training. However, in all fairness, we must consider that if we add the five minutes of training time, to the results of those in the environmental knowledge group, then participants actually took around a minute longer than the control group in all the environments. Whether this is relevant or not depends on how much of the participants were actually forming a cognitive map of the environment during the more 'passive' navigation stage. Past research has shown that decision-making is the most important factor when learning the spatial layout of a VE (Bakdash et al., 2008), and hence, pointless navigation with a lack of instructions can greatly increase the time it takes for a person to acquire spatial knowledge from an environment.

4.3 Analysis of Results

The previous section presented results that show that certain individual differences impact the exposure time required to acquire spatial knowledge from different types of virtual environments. Our research aim was to find out whether and how individual user differences and environmental factors impact exposure time. The following list summarises the previous section of this chapter, and satisfies a part of the research aim.

- Gender has a negative impact on environments with a low amount of unique object landmarks if the navigator is female.
- Orientation skill has an impact on all types of environments (the lower the OS skill of a user, the more negative the impact).
- Cognitive style has an impact on both complex environments and large environments.
- System knowledge has an impact on all types of environments (the lower the system knowledge, the more negative the impact).
- Environmental Knowledge has an impact on all types of environments (The higher the environmental knowledge the more positive the impact).

What this shows, is that certain individual user differences, and environmental factors, do indeed have a significant impact on the exposure time required to acquire spatial knowledge. With these findings, we have satisfied the first part of the defined research scope. The second part of the research scope involves the comparison of each user difference and environmental factor with respect to their relative importance.

In truth some of the results were not surprising, these results simply confirm the conclusions drawn from previous research. There were however some results that were perhaps unexpected due to a lack of indication from previous research. For example, this thesis discovered that participants in the verbal-analytic cognitive styles group acquired spatial knowledge faster than holist-imagers in complex environments. Verbal-analytic participants, however, took longer to acquire spatial knowledge in large sized environments than the holist-imager cognitive styles control group. To clarify the findings, the scores for all the environments between the different groups have been summarised in Table 4.7.

Table 4.7 – Summary of scores for the various groups in each environment

	Large Size	Low frequency of unique landmarks	High Complexity	Control Environment
Control Group	22:14	14:24	15:06	11:52
Gender Group	23:19	20:31	16:04	13:05
Environmental Knowledge Group	18:57	10:42	11:25	8:07
System Knowledge Group	28:28	20:09	20:21	15:45
Orientation Skill Group	28:09	20:21	20:06	17:33
Cognitive Styles Group	27:44	15:14	12:34	12:46

We can understand just how much of an impact the individual user differences, and environmental factors, have on the exposure time when compared to each other. The table shows that, overall, the most detrimental user difference when compared to the control group was orientation skill, with a total average mean difference from the control group of five minutes and 38 seconds of required exposure time to acquire spatial knowledge. Second came system knowledge, with a mean difference of five minutes and 16 seconds. Third was environmental knowledge, with an average mean difference of three minutes and 36 seconds. Fourth was cognitive styles, with one minute and 58 seconds, and the last was gender, with a mean difference of one minute and 31 seconds. We have seen what individual user differences impact the most on required exposure time for all environments as a whole. We now need to look at which individual user difference impacts all environments the most. Table 4.8 summarises the most influential user differences for each particular environment, as seen in table 4.7.

Table 4.8 – User differences that require the most exposure time in each environment

Large Environment	Low frequency of unique landmarks	High complexity	Normal "control" environment
System Knowledge	Gender	System Knowledge	Orientation Skill

These findings indicate, that the individual user difference that impacts on exposure time more often than once is system knowledge. This particular individual user difference has the greatest impact on both the large sized environment, and the control environment. This is fortunate, since it is also the easiest individual difference to train. One of the objectives of this thesis is to discover how much each individual user difference, and environmental factor, impacts on the required exposure time to acquire spatial knowledge. We now know from our analysis that, overall, orientation skill impacts the most on SKA in terms of required exposure time to acquire spatial knowledge from a VE. We also know that system knowledge is most frequently an issue during training. Because of our findings, we now discuss how we created a set of guidelines that will help VE trainers better understand what sort of exposure time would be required for a certain group of individuals during training. It is possible to create a set of multipliers, that a VE trainer can apply after running the pre-tests, and grouping the participants appropriately. For example, how much exposure time is needed to ensure that a male analytic-verbaliser, who otherwise has a high orientation skill, is an expert with the mouse and keyboard, and is a novice to mountainous terrain, needs to acquire spatial knowledge from a complex environment of a certain size? This is the type of question we initially answer, and although we can only make vague predictions for a small subset of possible users and environments, we feel that the findings and results of this section will show that user-defined exposure times in VE is predictable. Therefore, the goal of our set of guidelines is to predict the exposure time requirement during VE training, for any user type, in any environment. To be able to create a set of multipliers, we have to work out the ratios of each group and every environment compared to the control group. The table of multipliers is available in table 4.9. A trainer can then apply these ratios, to acquire the required exposure duration. So a female with a low OS skill in a low landmark environment, will have to apply both the 1.42 multiplier because she is female, and the 1.41 multiplier because she has a low orientation skill.

Table 4.9 – Multipliers that could be used to "predict" the training time required by various groups

	Large	Complex	Low amount of landmarks	Control
Female trainee	Same as a control trainee	Same as control trainee	Apply a 1.42 multiplier	Same as control trainee
Trainee with low orientation skill	Apply a 1.27 multiplier	Apply a 1.33 multiplier	Apply a 1.41 multiplier	Apply a 1.48 multiplier
Trainee with verbal-analytic cognitive style	Apply a 1.25 multiplier	Apply a 0.83 multiplier	Same as a control trainee	Same as a control trainee
Trainee with high environmental knowledge (experienced in navigating in that particular type of environment)	Apply a 0.85 multiplier	Apply a 0.76 multiplier	Apply a 0.74 multiplier	Apply a 0.68 multiplier
Trainee with low system knowledge (Not experienced in using the training device)	Apply a 1.28 multiplier	Apply a 1.35 multiplier	Apply a 1.4 multiplier	Apply a 1.33 multiplier

It is clear that we cannot confidently answer a complex questions such as: How does a analytic-verbaliser female, with low system knowledge and a high environmental knowledge fare in a large sized and complex environment. To answer such a question will require a lot more work, and we will need to consider it in the future. For now, we assume that the multipliers stack, however additional research is required to further investigate the use of this method. For the sake of clarity, we created a tree diagram (figure 4.7), which a VE trainer can follow in order to apply the appropriate multipliers according to the trainee's individual user differences, and the particular environment.

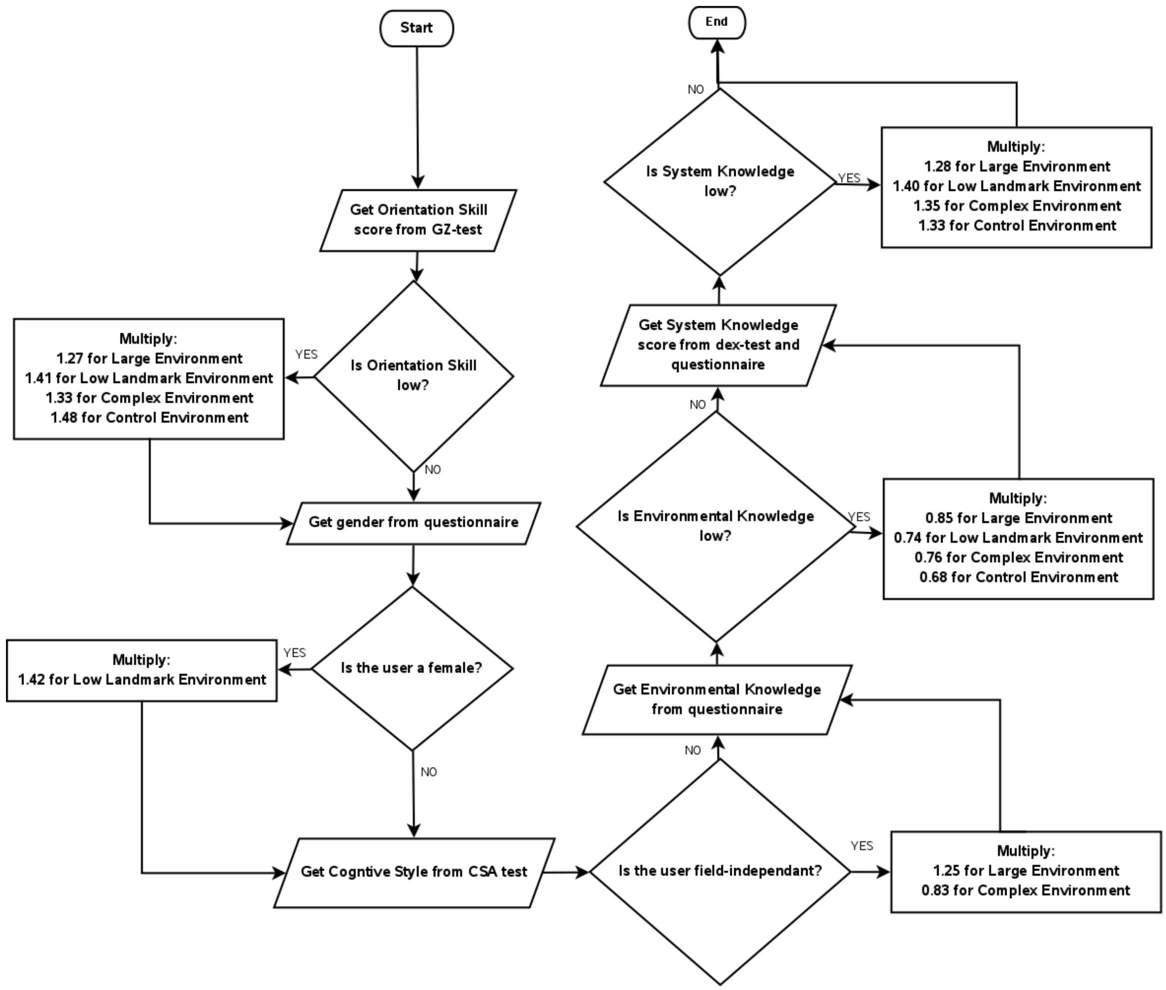


Figure 4.7 - Flowchart used as a reference when applying multipliers

Figure 4.7 allows us to relate our research findings to a real world example. For example, let us presume we have a person training in a large environment, who is a: female, an analytic-verbaliser, with a low OS skill, a high system knowledge and a low environmental knowledge. This participant would require:

1.27 because the trainee has a low OS skill.

1.25 because the trainee is an analytic verbaliser

1.27*1.25 = 1.5875 of the control time (i.e. the time of the control group profile).

In our control environment, participants required a training duration of 18 minutes. Our set of basic guidelines, suggests that the exposure time required for this particular trainee, in the defined environment, would therefore be at least 29 minutes.

Accordingly in this chapter, we have taken some small steps towards satisfying our research aim. We now know, to a certain degree, how much the individual user differences impact on the required exposure time when compared to each other; and we even have a tree of multipliers that can help us in predicting this exposure time.

4.4 Summary

This chapter has contributed towards a better understanding of how individual differences and environmental factors can impact exposure time required to acquire spatial knowledge. The experiment yielded a plethora of results, which will now be summarised:

Overall, orientation skill is the most influential skill, in terms of the mean total time taken throughout the environments. This was most obvious in the control environment. Orientation skill is important throughout, however, it is difficult to train. System knowledge seems to be the second most important skill, and is fortunately easy to train, as it is a matter of getting accustomed to the training interface. It is also relatively easy to measure, using a mouse dexterity test and a questionnaire. Since more and more people are getting accustomed to using a computer, one may predict that this would lower training times significantly. Environmental knowledge is also important throughout, in all environments, those with experience in a type of environment will have an advantage over those with no experience. Research suggests that this skill can be trained over time. Females have a serious disadvantage when it comes to learning from environments that are low in the number of unique landmarks. Whether this can change after training, is something that could be looked at in future work. Of course, theoretically, a female with high environmental knowledge, and high system knowledge, could out-perform a male with low environmental and system knowledge. Field-independent users will acquire spatial knowledge faster from a complex environment, however field-dependent users have the advantage in large environments. These learning styles are formed through life, so it is unlikely that a learning style will change through training.

Although our research has shown the impact of individual user differences in various environments. We have not yet clearly shown the impact of environmental factors on the required exposure time that a participant will need in order to acquire spatial knowledge. We do not know how much each environmental factor impacts on exposure time when compared to each other. The data collected in this chapter is not enough to help us answer that question. We know, for example, that large size impacts more on exposure time than complexity, but what we do not know is what the rate of change for the exposure time is as size gradually increases. At what point, if ever, does complexity actually become more burdensome than size? Just how many landmarks can we use to help the user before they too add to the environmental complexity by simply becoming visual obstructions? Since all our environmental factors were set, rather than dynamic, we cannot answer these questions. However, these are the questions we aim to answer in chapter six.

During experimentation, we identified possible problems with the GZ test participant score, which related to the cognitive load required to process the confusing dot-line answer key. If cognitive load is not related to just orientation skill, but also other cognitive and biological abilities as well, then not only are the results in this chapter unreliable, but also the results of every research paper that has used the GZ-test as a way of finding a participant's orientation skill. Accordingly, in the following chapter we investigate this issue, with the aim of testing and validating whether or not the GZ is appropriate for categorising orientation skill in participant groups.

Chapter 5 - The Electronic Guilford-Zimmerman Orientation Survey

5.1 Introduction

The Guilford-Zimmerman orientation survey test (GZ test) is a key test in the domain of spatial knowledge acquisition (SKA) (Guilford and Zimmerman, 1948). It is commonly used to determine a user's orientation skill. The GZ test is used by the majority of SKA research, as discussed in chapter two, even though alternatives exist; such as the Eliot-Price Test (Eliot and Price, 1975) and the Stumpf – Fay Cube Perspectives test (Stumpf and Fay, 1983). In order to ensure that the results of our tests are consistent with other research, we also decided to use the GZ test. In the previous chapter, we identified a problem with the GZ test. The GZ test was used to determine the participant's orientation skill (OS), in order to filter them into groups of low OS and high OS. The participants consistently complained that the directional system on the bottom of every question on the test was far too complicated and confusing. Most participants were frustrated as they believed that their scores did not reflect the actual results, simply because it took them a long time to answer the question (even though the answer may have been obvious) due to the very confusing answering system. We soon identified two serious issues when using the GZ test. Firstly, it was difficult to train participants in how to use the test and the answering system. Second, participants complained that the dash and dot directional system, which was on the bottom of every question, was far too complicated and confusing. In the domain of SKA, most studies control orientation skills by using pre-test experimental filters to categorise participants into groups with consistent OS scorings (Witmer et al., 1996; Goerger et al., 1998; Waller et al., 1998) . If, however, we question the ability of these pre-test filters, we place into doubt much of the research in this domain.

The GZ test comprises of pairs of images relating to the movement of a boat. The top image is the starting position of the boat and the bottom image is the finishing position of the boat. Each pair of images shows a shift in the position of the boat (position and/or angle) and the user's job is to determine how the boat has moved and, using a dot and line direction system, determine which of the given multiple-choice answers is correct (see figure 5.1 and 5.2). In order to explain that the boat moved down and to the left, they would have to select the key "_•"; since the line is below the dot and the dash (representing movement) is to the left of the dot (representing the original position). To explain the reverse (up and to the right), we would use the sign "•‾". If the movement was to the left and down, with an angle tilt to the left (as in figure 5.2), we use the key "⟋•". After a short explanation of the dot and dash direction system, users are asked to take the test. The GZ test lasts for 10 minutes and consists of up to 60 questions. Results are determined by summing up all the correct answers, summing up all wrong answers and then dividing the sum of wrong answers by four and subtracting this from the sum of correct answers. This result allows us to separate participants into groups relating to those of higher or lower aptitude. An OS does not directly relate to either high or low orientation ability, but instead test results are used to categorise relative participant groups.

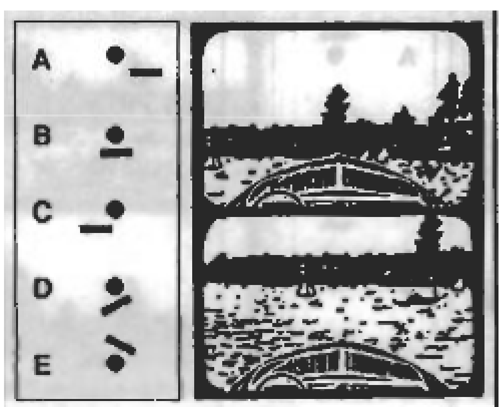


Figure 5.1 - The boat has moved down and then left (_•).

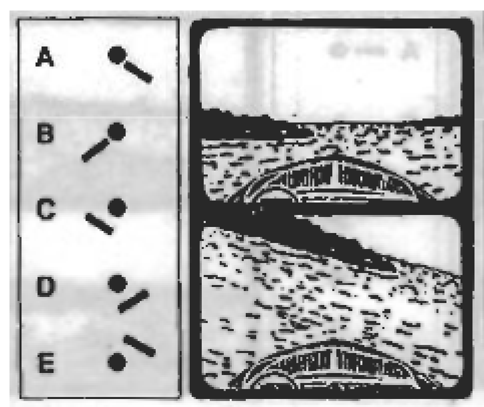


Figure 5.2 - The boat has moved to the left, downwards and has rotated to the left (⟋•).

Most participants in our experiments showed little problems with determining the movement of the boat, however, as was discussed earlier, the majority of participants found the answering system too complicated and confusing, and were unable to express the movement direction effectively. 98% of participants in our experiment believed that their orientation skill score did not truly reflect their ability; with participants often understanding the new orientation of the boat (for example: a shift down, a turn to the left, and an anti-clockwise rotation to the left, see figure 5.2), yet were confused when interpreting the symbol expressing this movement "i.e. ". This has seemingly lead to an increase in cognitive load, and if that was the case may have distorted the scores in a way that the test no longer reflects the participant's orientation skill, but rather a mixture of a cognitive skills. In literature it has been stated that, when a user interacts with an object purposely -that is to say, knowing that they are trying to achieve something during this interaction- they are constrained by a certain amount of working memory (Cooper, 2004). As the amount of cognitive elements required for this interaction increases, so does the amount of working memory (Chandler et al., 1996). The theory behind the measurement and understanding of cognitive load is called the Cognitve Load Theory (CLT) (Paas et al., 2003). Research in many fields takes CLT into account, and it is considered very important. Some fields include navigation in hypermedia systems, learning, and even marketing (Sweller, 1988; Iding et al., 2003; Dewitte et al., 2005; Yousoof et al., 2007). For the GZ test, the amount of working memory required during the interaction with the test by the participants is intrinsic to the question and answer system. This type of cognitive load (intrinsic cognitive load) is explained in detail by Paas et al. (2003). However, in short, it simply means that different materials impose a different amount of cognitive load on a user, and this load can not be reduced fully using a detailed walk through. Instead, the materials need to be altered in order to lower the cognitive load. In respect to the GZ test, the cognitive load on the participants imposed by the complicated answering system can only be reduced to a certain degree through comprehension, as a result of explanation. The only alternative way to reduce the cognitive load is to use an alternative answering system.

5.2 Problems with the traditional Guilford Zimmerman test

Since such a high percentage of participants raised a concern about problems with symbolic interpretation, we hypothesised that the results of the original GZ test did not reflect just the participant's orientation skill, but instead reflected a mixture of orientation and cognitive skills. This mixture of cognitive skills increases the cognitive load, and it has been speculated, that as the amount of cognitive load during a task increases, so does the mental effort required to solve that task (Xie and Salvendy, 2000). This is an issue for concern, because it is theorised that working memory is in fact extremely limited in capacity (Miller, 1956; Price and Catrambone, 2004), but also in duration (Peterson and Peterson, 1959). This theory is called the Cognitive Load Theory (CLT) and as a key research topic it focuses on discovering how different working instructions are used when the human working memory is constrained (Sweller, 1988; Sweller, 1994). It is therefore critical that we validate whether cognitive load is negatively impacting GZ scores. To do this we must ensure that cognitive factors are minimised during the orientation survey test. This can only be achieved by simplifying the interaction technique, or we risk cognitive overload, which may negatively impair a participant's ability to complete the test. However, if by simplifying the directional system, the results change, then it will be obvious that whatever caused the cognitive load, also played a major role during the GZ test. Accordingly, in order to test the impact of the dot-line system on user OS scores, a new directional system had to be created, which would avoid the “confusing” dot and line system of the old test (yet ensure that all other factors are kept as consistent as possible). In light of this, we decided to produce an electronic version of the test, which simply substitutes the complex dot and line system with a more simple type of interaction, using arrows.

5.3 The Online Electronic Test

By replacing the ‘complicated’ dot and line symbol interpretation with a less complex type of interaction, we are able to determine whether cognitive load, as a result of symbol interpretation, significantly impacted GZ scores (Yousoof et al., 2007). A new directional system, using Java Server Pages and JavaScript, was developed. Instead of combining all movements into a single symbol, the user was asked to identify each directional change separately by clicking arrow buttons relating to each movement. Figure 5.2 shows the movement of a boat to the left and down (as a result of pitch), with an angle tilt to the left. In the paper-based GZ test the user would have have to interpret this movement and match it with the single symbol “⟋•”. In the electronic GZ test (see figure 5.3), the user is required to click the relevant arrow buttons separately “↓←↰”; they can then click “ok” to move to the next question. If a mistake was made during the data entry process, the user is able to click the ‘clear’ button. The order that they choose to press the arrow keys is not important, as long as the movement results in the appropriate end position of the boat. In other words “↓←↰“, is in fact the same as “↰←↓“. Although the user interaction is clearly different, all other aspects of the test were kept consistent to try and reduce confounding experimental variables. The online electronic test used the same paired pictures as the traditional paper test; however multiple choice solutions were not made available to users.

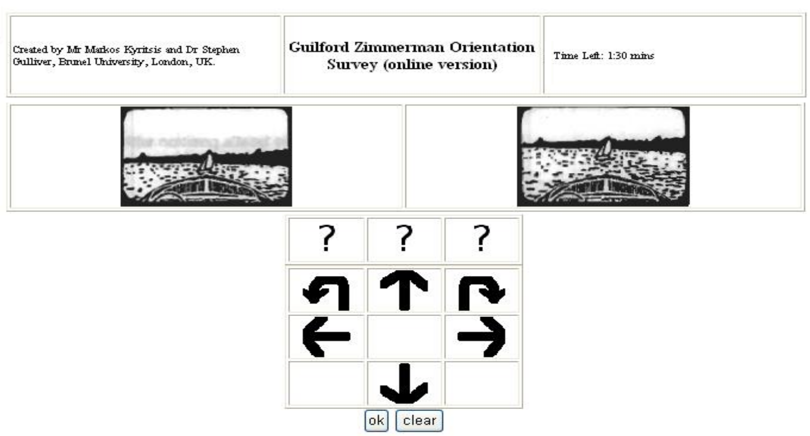


Figure 5.3 – Screen shot of the Guilford-Zimmerman online electronic survey

The ten minute time limit associated with the old test also exists in the electronic version, and the scoring mechanism is also consistent with the older version.

5.4 Taking the Test

In order to see whether a change in the answering system of the test would significantly alter the scores of participants, we asked participants from our previous experiments; discussed in chapters three and four; to take the new test. Our previous experiments grouped people with a low OS into one group, and a high OS into another group (the control group). We used the same groups for the new GZ test. After taking the original test, the participants of these groups were never shown their scores, and were never advised on which questions they had answered correctly. Furthermore, the new test was taken by the participants three months after they sat through the previous test. Both of these factors ensured that any improvement in orientation score was not seen to be as a result of participants remembering picture pairs, but as a result of improved user interaction (i.e. reduced cognitive load). All other individual differences were kept as equal as possible, i.e. the only difference was measured OS. Both groups retook the test under exactly the same conditions as the original GZ test. This helped us find the importance of orientation skill on SKA, allowing the results of these two groups to be compared. The same scoring system was used in order to ensure that no external variables altered the test results. All the participants reported that the test was a lot easier to understand, and there were no complaints made about other aspects of the new GZ test.

Comparison between the two sets of tests allowed us to determine the impact of cognitive load on participant orientation skill scores. Results showed that all participants scored much higher (almost twice as much) when taking the electronic GZ test than when undertaking the original GZ test. The average mean score rose from 17 to 30 points (see figure 5.4). T-test analysis showed that the type of test (original / electronic) was significantly responsible for altering user orientation score ($P < 0.001$). Accordingly, our results show that the paper-based GZ (using dot and line interaction) reflects not only user orientation skill, but a mix of user orientation and other factors. The interesting thing about our findings is that the impact on participant scores remains relative, independent of participant. Participants who scored the lowest on the original test, also scored the lowest with the new test, and participants that scored the highest remained on top. This is important, as it shows that the paper-based GZ test, despite additional factors, still functions as an effective OS pre-test filter. An error in the GZ test's ability to categorise participants would place into doubt the findings of much research, including the findings in our chapter four. Thankfully, however, although cognitive factors do interfere with the outcome of the test, the original test interfered with everyone's OS score in a similar way; lowering the scores for everyone, and not just for a select group of people.

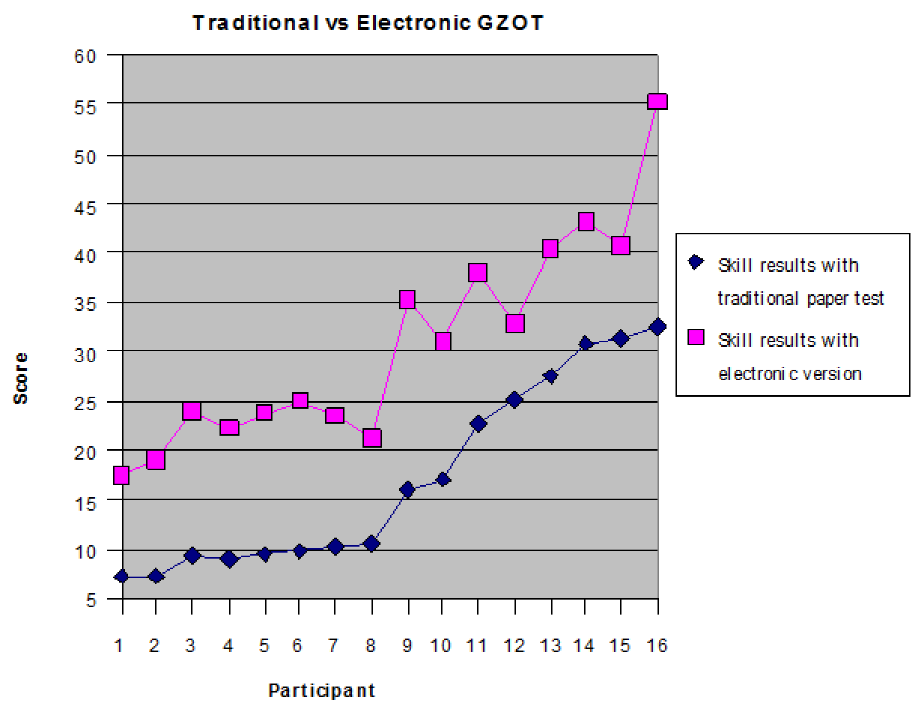


Figure 5.4 – Line graph showing the linearity between the two sets of results

The question however remains as to which cognitive factors were present in the paper test, and were no longer present once the directional system changed. Answering that question is not the primary focus of this particular thesis, but should be instead considered as future work.

5.5 Possible Contribution of the Electronic GZ-test

As was discussed in section 5.1 the problems that arose with the Guilford-Zimmerman test were:

- Difficulty obtaining the test.
- Difficulty understanding the answering system, and training participants to use it; therefore increasing training time for all participants.
- Confusing key-answering system, putting a higher cognitive load on all participants.

These problems were addressed with the creation of an electronic test. The next three sub-sections discusses how the electronic version provided a better solution to researchers who may require the use of the GZ test in future research.

5.5.1 – Obtaining the Test

We faced considerable problems obtaining the paper-based GZ test, and were surprised to find that a copy was not even available in the British Library. The electronic version is web-based, and can be accessed via www.newgztest.com. All the participant has to do is click on start, and the test begins. This is a very convenient method of accessing the test, that was otherwise almost impossible to find. Ultimately, any PC with an Internet connection and a browser can now access the test. Also, the old test is now available in the appendix of this thesis, which will be added to the British library.

5.5.2 – Training Participants

The original test took more than ten minutes at times in order for a participant to even understand the dot-line key answering system. The new test took less than five minutes for almost all participants to understand how it works. This should help future researchers lower training time as even half of the original training time is sufficient.

5.5.3 – Clarity of Test Lowers Cognitive Load

As well as significantly increasing participant's OS results, the new test also better reflected participant's orientation skill, as it largely removed or at least minimised the impact of external factors (such as other cognitive abilities on participant scores). Participants also did not show signs of frustration when trying to determine the correct answer key for the movement they already understood.

5.6 Conclusion

We identified in this chapter that the original Guilford-Zimmerman orientation test suffered from a number of problems. Firstly, we faced considerable problems obtaining the paper-based GZ test. Secondly, participants had difficulty in understanding the answering system, which resulted in participant frustration with the experiment. Lastly, and perhaps most importantly, the GZ test was shown not to reflect only user orientation skill, but rather a mixture of user orientation and cognitive skills. Fortunately, all problems were addressed by developing an electronic version of the GZ test and comparing this against the original. Firstly, since the new electronic GZ test was designed to be web-based, it is easy to disseminate. We have made the new electronic GZ test publicly available at www.newgztest.com. In the future, instead of having the same problems obtaining the paper-based test, researchers can access the electronic equivalent test online. Secondly, the electronic GZ test significantly reduced participant confusion, and hence user frustration with the test. The original test required more than ten minutes training time, before a participant started to understand how the dot-line key system worked. It took only a few minutes training for participants to understand the new arrow method of interaction. Finally, the test results of the new electronic GZ test were significantly higher for all participants, implying that the interaction methods used in the new test minimised the impact of external factors on OS scores. Results showed, however, that the negative interaction of the paper-based GZ test is consistent for all participants (see figure 5.4). This means that although there was a significant cognitive load, the load similarly affected all participants, and therefore did not interfere with participant categorisation. Participants with lower orientation skills in the original test also scored lower in the new test, which means that both the old and new electronic GZ test are still valid for use as a pre-experimental test for categorisation of participant orientation ability.

Despite suffering numerous interaction problems, which can largely be overcome by using the newly developed electronic GZ test, the traditional test has been validated as an effective test for categorisation of participant's relative orientation skill. This is a critically important result, as it supports the findings of research in the domain of SKA that depends on this test, but also supports our results in chapter four, and verifies that they are applicable to the domain of SKA.

Now that the findings in chapter four, concerning user individual differences, have been effectively validated, in chapter six we will attempt to quantify the importance and impact of the three environmental factors (size, spatial complexity, and landmark potential) during the process of acquiring spatial knowledge from a virtual environment.

Chapter 6 - Investigating the Impact of the Environmental Factors on the Required Exposure Time a User Needs to Acquire Spatial Knowledge

6.1 Introduction

In chapter four we showed that various environmental factors impact the exposure time required by a user during navigation, for the purpose of acquiring spatial knowledge. We argued that the three main environmental factors commonly found in virtual environments (VE) that influence navigation are: size, complexity and landmark potential. We also showed how users, with a certain individual difference (cognitive or biological in nature) either benefit from a certain type of environment (e.g. analytic verbalisers have an advantage over holist imagers in complex environments), or be burdened by it (e.g. female users have more difficulty navigating in environments with a low amount of unique landmarks). Although chapter four results contribute towards a better understanding of how individual user differences impact on spatial knowledge acquisition (SKA), they do not provide a clear indication of the impact of environmental factors. We can tell from the results in chapter four, that all factors are critically important when considering navigational complexity, especially for particular types of users. However, with the current data set, we cannot begin to predict the importance of these factors. In the previous experiments, we were satisfied with two states for each environmental factor, i.e. high and low. This was sufficient when we tried to discover the impact of individual user differences on the different environments, but did not indicate the impact of the environmental factors when compared to each other, and, as each factor increases in value (whether that means frequency and amount for landmarks and complexity, or simply space for size). In other words, we have not yet discovered the impact of the environmental factors on the process of SKA, and have not been able to show whether there are any trends or patterns involved when a particular environmental factor is changed within an environment. Therefore, this chapter attempts to discover, through experimentation, the importance of the three environmental factors (size, spatial complexity, landmark potential). It also provides a set of guidelines that will help VE trainers predict the required exposure time that a user will need, depending on these factors.

6.2 Obtaining the Missing Data

In chapter four, through a series of experiments, which involved participants navigating and learning different types of environments, we discovered how individual user differences influence the exposure time required for users acquiring spatial knowledge from these environments. Although the findings are interesting, and provide a good understanding concerning individual differences, the current results do not give us any insight on the importance of environmental factors in the process of spatial knowledge acquisition. This is because in the previous experiments, environmental factors were simply bipolar experimental variables, which were changed from a low state to a high state, and vice-versa. This does not help us understand how exposure time is affected as these factors change between these states. To achieve this, we need more environments, which only differ slightly from each other, as compared to the huge differences between the environments and the control environment, seen in chapter four. This will allow us to understand the rate of change on exposure time, and help us predict the impact of these factors on the exposure time requirements for a larger array of environments. In reality, the task of being able to predict the required exposure time for any specific user in any possible environment is extremely difficult (if not impossible). The variables involved when deciding how to measure complex, or large, or landmark rich, are many. To even consider the value one would give, to complexity for example, is a gigantic task (since it is rather hard in a non-flat and non-maze-like environment to predict things such as visual obstructions). We nevertheless strive to bring as much information to the fields of VE training as we possibly can. For now we start by providing a set of guidelines for more flat and manageable maze-like environments.

So far, we have discussed what data was missing from chapter four. We will now review and discuss what we found so far concerning the importance of environmental factors on the process of SKA.

6.2.1 - Importance of Size

Overall, size seems to have the most impact on the exposure time required by a user to acquire spatial knowledge from a VE. When we ran a univariate analysis of variance on the results obtained from all of the user groups, we found that compared to the other environments (control environment, complex environment, low landmark value environment), the overall time taken to acquire spatial knowledge was always significantly more than any other environmental variable. This is actually quite sensible, since not only does an increase in size mean higher spatial memory requirements (O’Keefe and Nadel, 1978; Stankiewicz and Kalia, 2004), but also means that the user has to spend more time to get from one place to another. We also ran independent samples T-tests between results of the different participant groups, in a large environment. We found that the only individual user difference that made an impact on the exposure time required specifically for large environments was cognitive styles. Analytic-verbalisers are burdened by the huge information, and therefore perform significantly worse than holist-imagers.

6.2.2 - Importance of Complexity

The importance of complexity on SKA was measured by testing all user groups in both a very complex environment and the control environment, and then comparing the scores. When we ran a univariate analysis of variance on the results obtained from the user groups, we found that compared to the other environments (control environment, large environment, low landmark value environment), the overall time taken to acquire spatial knowledge was always significantly less than the large environment, always significantly more than the control environment, but never significantly different to the low landmark environment. The exceptions to this rule, were: the gender group, which showed that females performed significantly worse in the low landmark environment compared to the complex environment; and the cognitive styles group (analytic-verbalisers), which we showed performed significantly better in the complex environment than the control and low landmark environments. We hypothesised that a complex environment calls for more analytical breakdown of procedural spatial knowledge, and therefore is better tackled by a person of an analytic learning strategy (i.e. analytic -verbaliser).

We ran independent samples T-tests between results of the different participant groups, in a complex environment. The results from this test indicated that the only individual user difference that made an impact on the exposure time required specifically for complex environments was cognitive styles. Analytic-verbalisers who are more suited to the analytical breakdown of information during the acquisition of knowledge from a complex environment, perform significantly better than holist-imagers.

6.2.3 - Importance of Landmarks

The importance of landmark potential on SKA was measured by taking the time it took for users to acquire spatial knowledge in an environment almost void of landmarks, and comparing those times to the time taken from the control environment, which had a high amount of unique object landmarks. When we ran a Univariate analysis of variance on the results that we obtained from all user groups, we found that, compared to the other environments, the overall time taken to acquire spatial knowledge was always significantly less than the large environment, almost always significantly more than the control environment, but almost never significantly different to the complex environment. The only exceptions were the gender group and the cognitive styles group. We found that the female participants performed significantly worse in the low landmark environment than in the complex environment, whilst the analytic verbalisers performed significantly better in the complex environments. Our results supported previous literature, such as Sandstrome et al. (1998) and Moffat et al. (1998), which indicate that female VE trainees are only significantly burdened in low landmark environments, e.g. such as deserts.

We ran independent sample T-tests between results of the different groups in a low landmark environment. The results from this test indicate that the only individual user difference that made an impact on the exposure time required specifically for low landmarks environments was gender.

To better understand the impact that environmental changes have on required exposure time, a new set of experiments was needed, which will be considered in the following section.

6.3 Experimental Methodology

In order to find how a change in the environmental factors impact exposure time, the experimental design must be similar to the design discussed in chapter three. Accordingly, we start by finding the participants for the experiments. In chapter four we used a set of pre-tests in order to filter the participants, and grouped them according to their individual user differences. For these experiments we used the same pre-tests, in order to ensure similar user differences. However, rather than separating participants into groups, we used one group of eight people with scores similar to the control group defined in chapter three. This was done in order to ensure that no other external variables influenced our results. The following sub-sections discuss the experimental methodology adopted for this new set of experiments.

6.3.1 - Finding Participants for the New Set of Experiments

The previous experiments focused on the impact of individual user differences, rather than the impact of environmental factors on SKA. In this chapter we will dig deeper into the impact of environmental factors, and are therefore not focusing on variation caused by the individual user differences. We still, however, need to ensure that the new experimental control participants have the same individual user differences as the control group in the previous experiments, otherwise the comparison of results will not be possible. This is especially important, as we have shown that user differences have a high impact on SKA. The demograph of the control group must therefore be of the type: male, holist-imager, with good orientation skill, a high level of system knowledge, and a low level of environmental knowledge. The pre-tests used in this chapter are summarised in Table 6.1. For further details on the pre-tests, see chapter three.

Table 6.1 – Participant filters

	Gender	**Cognitive Style**	**Orientation Skill**	**Environmental Knowledge**	**System Knowledge**
Filter	Questionnaire	Cognitive Style Analysis test	Guilford-Zimmerman Spatial Orientation Survey	Questionnaire and a certain amount of training in the environment before the experiment begins	Questionnaire and Mouse dexterity test

Twenty-two volunteers were again filtered using the pre-tests, and a new group of eight people, with similar traits to the control group were selected.

Chapter three discussed the creation of four different maze environments. Each environment was linked to a specific environmental factor by distinguishing that factor from the control environment, while keeping all other factors the same. It is clear that the environments we already created will not suffice alone for the new experiments. Instead, rather than having a 'low' and 'high' amount of a particular environmental factor, we will need to see stepped progression from 'low' to 'high'. Since we are trying to look at the way exposure time is affected when we change the environmental factors gradually, rather than using extreme low and high values, we will need to supplement the existing ones to facilitate stepped change. In chapter three we used four environments for our experiments, the control environment; the large environment; the complex environment; and the low-landmark environment. Table 6.2 summarises these environments. The left column contains the environment types, while the right column discusses the environmental properties. These properties are: the amount of unique object landmarks in the environment, the size in pixels (where each pixel is 60 cm), and the obstructions per row (meaning the amount of walls in every one of the eight segments that make up the height of the map).

Table 6.2 – Table of environments required for the previous experimental process

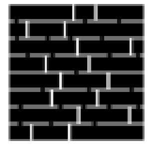
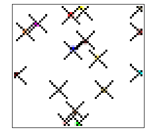
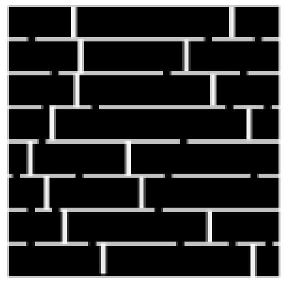
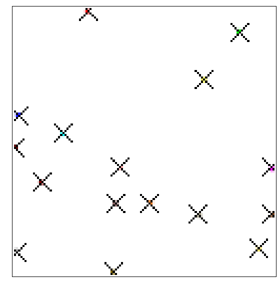
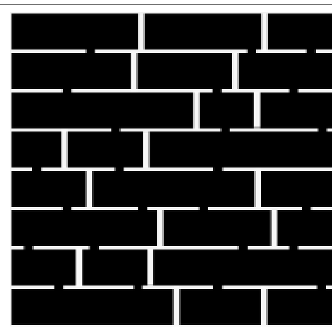
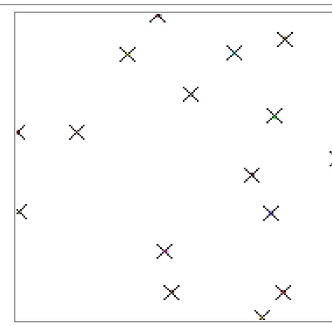
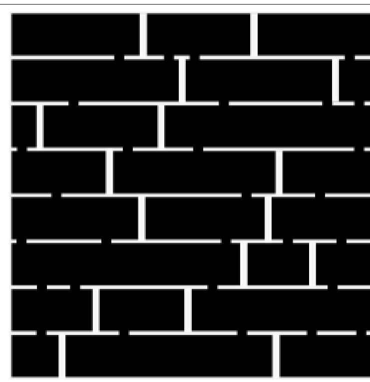
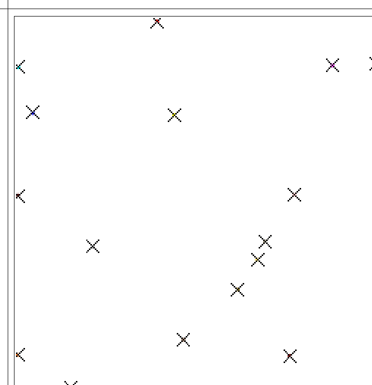
Environment Type	Properties
Control environment	16 unique landmarks, 128*128 pixel size, two obstructions per row, that means there are two rooms per row where each row is per 8 pixels.
Large Size	16 unique landmarks, 256*256 pixel size, two obstructions per row.
Complex Layout	16 unique landmarks, 128*128 pixel size, four obstructions per row.
Low landmark potential (frequency of unique landmarks)	4 unique landmarks, 128*128 pixel size, two obstructions per row.

6.3.2 - Making Environments to Further Investigate the Impact of Size

Size represents the raw space available for navigation. In chapter three we used a large size of 256*256 pixels for the 'large size' environment; and a small size of 128*128 pixels for all the other environments (including the control environment). In our experiments, we measured size in pixels, since we used 2D maps in order to construct the 3D environments. Although we do not use real world representations for our maps, we can still deduct the real world equivalent of the size in metres. This was accomplished by recording the time it took for a user to walk from one side of the environment to the other in a straight line. A special map was used for this, which was void of any obstacles.

Two more environments were created in order to examine the importance of size, as it increases. We already have the control environment, which was 128*128 pixels in size, and the large size environment, which was 256*256. We also need a really small environment, which will be 64*64, and a medium-sized environment, which will be 192*192, allowing stepped increases of 64 pixels per environment. The reason that we do not create a very large environment (over 256*256), as the experiments would take far too long, and may end up frustrating the participants, negatively impacting feedback. Table 6.3 summarises the environments to be used in order to investigate the impact of size on the process SKA.

Table 6.3 – Environments used to further investigate the impact of size

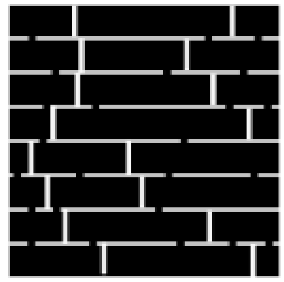
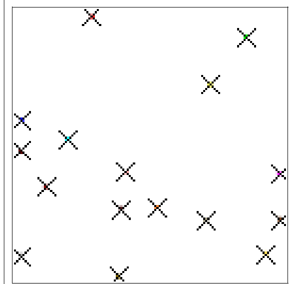
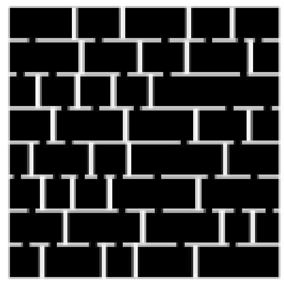
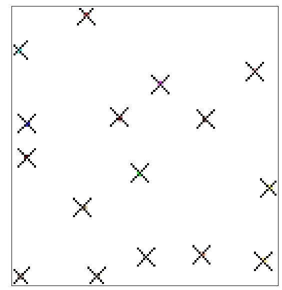
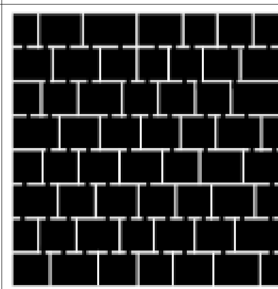
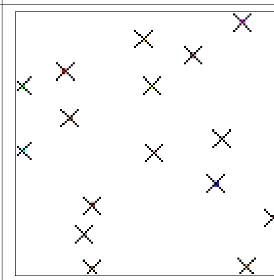
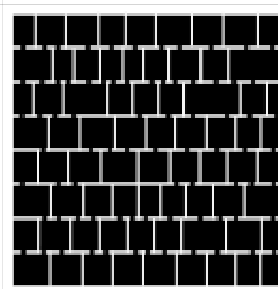
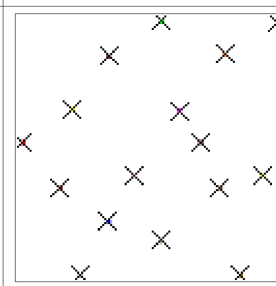
Environment type	Properties	Terrain Map	Landmark Map
Small Size	16 unique landmarks, 64*64 pixel size, two obstructions per row.		
Normal Size (used as control group)	16 unique landmarks, 128*128 pixel size, two obstructions per row, that means there are two rooms per row where each row is per 8 pixels.		
Medium Size	16 unique landmarks, 192*192 pixel size, two obstructions per row.		
Large Size	16 unique landmarks, 256*256 pixel size, two obstructions per row.		

6.3.3 - Making Environments to Further Investigate the Impact of Complexity

Complexity represents the environmental structure in which a user must navigate. In our experiments we control complexity by adding more corridors to an environment, therefore creating visual obstructions to key navigational aids (such as object landmarks). In chapter three we used a high complexity for the 'complex' environment, and a low complexity for all the other environments (including the control environment). In order to understand how the gradual increase in complexity impacts required exposure time, we will need to create more environments.

Two more environments were created in order to examine the importance of complexity as it increases. From the first set of experiments, we used the control environment, which was made up of two obstructions per row, and the complex environment, which was made up of eight obstructions per row. An obstruction is simply a wall separating two parts of a corridor (row). There are always eight corridors in each environment. We will create two environments with values between the high and low states, so one environment will have four obstructions per row, whilst the other will have six. Table 6.4 summarises the environments that will be used in order to investigate the impact of complexity on the process of SKA.

Table 6.4 – Environments used to further investigate the impact of complexity

Environment type	Properties	Terrain Map	Landmark Map
Very low complexity (used as control group)	16 unique landmarks, 128*128 pixel size, two obstructions per row, that means there are two rooms per row where each row is per 8 pixels.		
Low Complexity	16 unique landmarks, 128*128 pixel size, four obstructions per row.		
Medium Complexity	16 unique landmarks, 128*128 pixel size, six obstructions per row.		
High Complexity	16 unique landmarks, 128*128 pixel size, eight obstructions per row.		

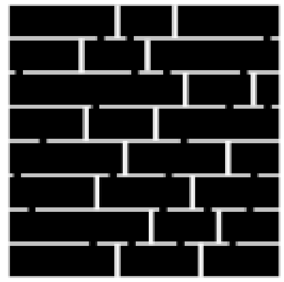
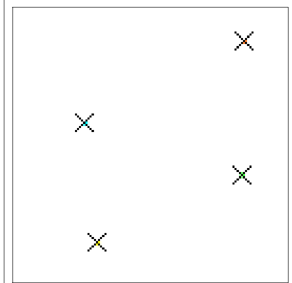
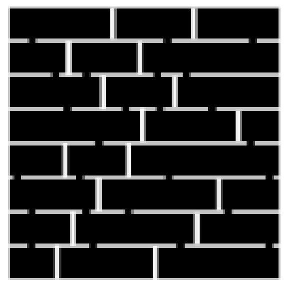
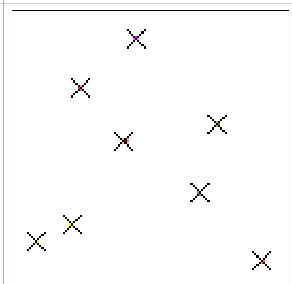
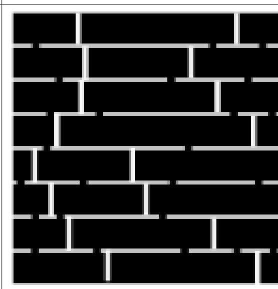
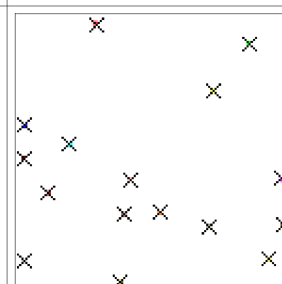
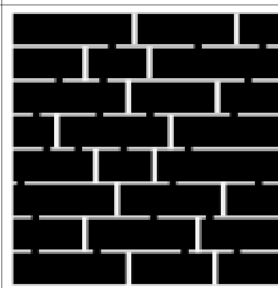
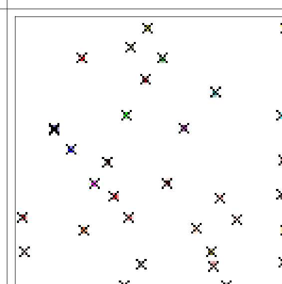
6.3.4 - Making Environments to Further Investigate the Impact of Landmarks

During our review of literature relating to environmental factors and navigation, we found that landmarks play a significant role in user navigation. These landmarks can be both structural landmarks (i.e. belong to the geometry of the landscape, such as a T-junction), and object landmarks (trees, rocks, houses, etc.). Structural landmarks, are, as one would expect, very difficult to control and measure. Object landmarks on the other hand, are not. Therefore, for the sake of simplicity, our experiments used object landmarks when calculating landmark value. Landmark potential is measured by uniqueness, clarity, and frequency. In our literature, we showed that importance of landmarks depends on their uniqueness (the less the same landmark is encountered, the higher its value of importance), how obvious they are (low fidelity landmarks are not as good as high fidelity ones), and finally how many landmarks are readily available. In chapter three we used four unique landmarks for the 'low landmark' environment, and sixteen unique landmarks for all the other environments (including the control environment). In order to understand how the gradual increase in the amount of object landmarks impacts on exposure time, we will need to create more environments.

Two more environments were created in order to examine the importance of landmarks as they become more frequent. From the original experiments, we used the control environment, which was made up of sixteen unique object landmarks, and the low-landmark environment, which was made up of four unique object landmarks. In addition, we created two environments with one valued between the current high and low state (i.e. eight objects), and one higher than the high state (i.e. thirty-two objects). So one environment had eight unique object landmarks, whilst the other had thirty-two. Table 6.5 summarises the environments that will be used in order to investigate the impact of landmarks on the process of spatial knowledge acquisition.

With these new environments, we could gain a better understanding of how the gradual change of environmental factors impacts required exposure time, and therefore have a better understanding of how long it would take someone to learn the environment, with consideration to the value of these environmental factors. As soon as the environments were completed, the experiments took place. The next section discusses the actual experimental process.

Table 6.5 – Environments used to further investigate the impact of landmarks

Environment type	Properties	Terrain Map	Landmark Map
Low frequency of unique landmarks	4 unique landmarks, 128*128 pixel size, two obstructions per row.		
Medium frequency of unique landmarks	8 unique landmarks, 128*128 pixel size, two obstructions per row.		
Normal frequency (used as control group)	16 unique landmarks, 128*128 pixel size, two obstructions per row, that means there are two rooms per row where each row is per 8 pixels.		
High frequency of unique landmarks	32 unique landmarks, 128*128 pixel size, two obstructions per row.		

6.4 Experimental Process

The experiments were run on seven PCs supporting OpenSuse Linux. The computers had a Geforce 8400 graphics card that was capable of hardware acceleration for the OpenGL version of the Java3D SDK. The PCs also had the SKAR engine running with a different environment on each. All participants used each PC in turn, and were asked to navigate through the environment until they could draw a map of it. Once a participant felt that they had acquired spatial knowledge from an environment, a paper map was handed out to that participant. In order to demonstrate that spatial knowledge was indeed acquired, the participants had to point to landmarks on the paper map. If that landmark was within the quad sector of that corridor, they had demonstrated that they had 'learned' the position of the landmark, otherwise they resumed navigation. A log was kept of their actions, including the amount of time they stopped and resumed, as well as their total time.

6.5 Further Insight on the Importance of Size

In order to gain a better understanding of the impact of size on exposure time that a user needs to acquire spatial knowledge, we conducted our experiments on four different environments with varying size. The environments started at 64*64 pixels (which represents a real world size of 38 metres squared), and increased by 64*64 pixels gradually, in order to reach a size of 256*256 (which represents a real world size of 153 metres squared). Our previous experiments already indicated that size was of critical importance to exposure time, but the new experimental findings allow us to understand how the gradual manipulation of size affects exposure time. Ultimately, the results should provide VE trainers with a set of rough guidelines concerning how much exposure time a user will require during training. The mean time taken for the participants to acquire spatial knowledge in all four 'size testing' environments is shown in table 6.6.

Table 6.6 – Results obtained from the environments used for investigating the impact of size.

Environment Type	Result
Small Environment	10:01
Normal Environment (control)	12:01
Medium Environment	13:51
Large Environment	23:47

It is interesting to note, that although the time taken by the control group in these experiments differed slightly from the original experiments, the difference was not statistically significant. This implies that the filtering process worked well. To investigate whether the between-environment difference of size is significant in our new experimental data, we ran a univariate analysis of variance, with size as the independent variable, and time as the dependent variable. We found that the differences overall are significant ($F(1,3) = 145.510$, $P < 0.001$). We also ran post-hoc tukey tests, in order to see whether the differences in results between the environments are significantly different to each other. We found that the difference between the small environment (64*64), and the normal environment (128*128) is significant (Mean Difference 0:02:20; Std Err 0:00:42, $P = 0.014$). The difference between the normal environment, and the medium environment, however, was not significant (Mean Difference 0:01:30; Std Err 0:00:42, $P = 0.175$). The difference between the medium environment, and the large environment is very significant (Mean Difference 0:09:56; Std Err 0:00:42, $P < 0.001$).

It is interesting that our results show that within a small environment, people acquire the knowledge of space significantly faster than a medium-sized environment. As size increases, however, there appears to be a grey area, where it seems that adjustments to size does not negatively or significantly impact required exposure times. If we consider the fact that 128*128 pixels = 16384 pixels, and that 192*192 pixels = 36864 pixels, then the later is 2.25 times larger. After 192*192, however, the difference in time varies significantly, implying that the cognitive load on the user increases greatly. Figure 6.1 illustrates the gradual change in time, as the environment increases in size.

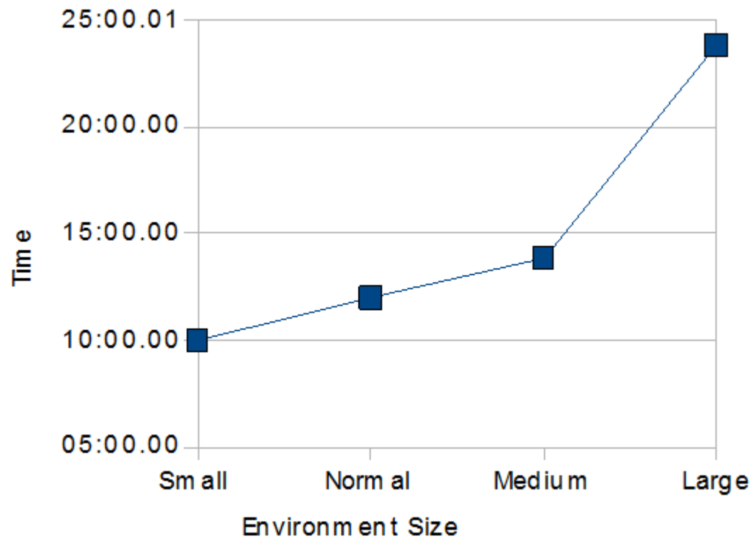


Figure 6.1 – Line graph showing the effect of size on required exposure time.

This graph allows us to look at the basic rate of change, in an attempt to gain a better understanding of how growth occurs in terms of time, as size increases. The line begins as a straight rate of change. This can be seen by drawing a line through initial points on the x-axis. Unfortunately the graph is only linear to a certain point. Beyond this point it will be harder to predict rate of change with changes in the environmental size. Up to the normal environment size, all lines intercept in more than two points. After that, however, they only intercept on two points at any time. We know that the average rate of change will be greater than 0 and that the value of the rate changes between points. Accordingly, we use linear interpolation to roughly predict the required exposure time for environments between 64 * 64 pixels, and 256*256 pixels. Linear interpolation, is a common technique used to find a value between two known points. The equation for finding linear interpolation is: $y = y_a + ((x - x_a)(y_b-y_a))/(x_b-x_a)$. For example, if we would like to know how long it will take a group of people to acquire spatial knowledge in an 96*96 environment, we would convert everything to seconds and pixels (for ease), and use the formula with variable values as described below:

y_a = 601 (time {sec.} taken for first environment)

*x = 9216 (total space in 96*96)*

*x_a = 4096 (total space in 64*64)*

y_b = 721 (total time {sec.} in second environment)

*x_b = 16384 (total space in 128*128)*

Therefore, y = 601+((9216 - 4096)(721 – 601))/(16384-4096) = 601 + ((5120)(120)/12288 = 651 = 10 minutes and 51 seconds. So for an environment of 96*96 pixels, with environmental characteristics similar to the control (two obstructions per row, and 16 unique object landmarks), a person will require at least 10 minutes and 51 seconds to acquire spatial knowledge. If we convert these figures to a real world size, then 96 pixels are 57.6 metres (1 pixel = 0.6 metres). So for an environment of 3317 square metres, with control characteristics, the total time it will require for a person fitting the control group prerequisite (male, holist-imager, with good knowledge of the system, and no previous knowledge of the environment) is at least 10 minutes and 51 seconds. Linear interpolation allows us to 'map' certain bands of pixel size to a quick time multiplier value. Equations, seen in table 6.7, allow us to quickly calculate (compared to the control group time) a multiplier that relates to the relative impact of size on exposure time requirements.

Table 6.7 – Converting from pixel to time factor for size

Pixel (x)	**y value**	**Time factor t(x)**
If 64 < x <= 128	y = (601 + (x-64)120)/64	t(x) = 0.8336 + 0.0026(x-64)
If 128 < x <= 192	y = (721 + (x-128)110)/64	t(x) = 1 + 0.0026(x-128)
If 192 < x <= 256	y = (831 + (x-192)596)/64	t(x) = 1.1526+0.0013(x-192)

6.6 Further Insight on the Importance of Complexity

To better understand the impact of complexity on the exposure time required for a user to acquire spatial knowledge, we conducted experiments with four different environments of varying complexity. The least complex environment consisted of two obstructions. This complexity was increased in linear step, with two additional obstructions per row for each additional environment. The most complex environment had eight obstructions per row (see table 6.4 for more details on the maps). The mean time taken for the participants to acquire spatial knowledge in all four complexity environments is shown in table 6.8.

Table 6.8 – Results obtained from the environments investigating complexity.

Environment Type	Result
Very low Complexity (control)	12:01
Low Complexity	12:21
Medium Complexity	13:28
High Complexity	16:23

Our previous findings, from the experiments detailed in chapter four, showed us that complexity affects all users (although significantly less for verbal analytics who thrive in such environments). To investigate the between environment significance of size in our new experimental data, we ran a univariate analysis of variance, with complexity as the independent variable, and time as the dependent variable. We found that the differences overall are less significant than size, but significant nonetheless ($F(1,3) = 29.780$, $P < 0.001$). We also ran post-hoc tukey tests, in order to see whether the differences in results between the environments differ significantly. We found that the difference between the very-low complexity environment (two obstructions), and the low complexity environment (four obstructions) was not significant (Mean Difference 0:00:20; Std Err 0:00:30, $P = 0.913$), however it was significant when we compared the very low complexity environment to the medium complexity environment (six obstructions per row) (Mean Difference 0:01:27; Std Err 0:00:30, $P = 0.039$). The difference between the low complexity environment, and the medium complexity environment, was again not significant (Mean Difference 0:01:07; Std Err 0:00:30, $P = 0.153$), but was significant when we compared the low complexity environment to the high complexity environment (eight obstructions per row)(Mean Difference 0:04:01; Std Err 0:00:30, $P < 0.001$). Finally, the difference between the medium complexity environment, and the high complexity environment is very significant (Mean Difference 0:02:54; Std Err 0:00:30, $P < 0.001$).

It seems that a gradual change in complexity does not interfere as much with the user's ability to acquire spatial knowledge, as does a change in size. At a specific point, like the size, the time required to learn the environment increases greatly. We saw that up to six obstructions per row (when adding an extra two obstructions at each step in complexity), the changes in required exposure time was not significant. After six obstructions, however, the time difference became very significant. We hypothesise that this is to do with the available spatial memory allocated for navigation. It seems that once all available memory is allocated for navigation, the navigator becomes overwhelmed by the level of complexity, and required exposure time increases greatly. Whether this hypothesis holds true, is something we would like to look at in the future. Figure 6.2 illustrates the gradual change in time, as the environment increases in complexity.

Again, as with size, the graph is an increasing, non-linear function. However, the rate of change is close between low-level values. We can work out a function for the graph, which will allow us to predict values along the line. To work out rate of change, we use differentiation (r = dy/dx).

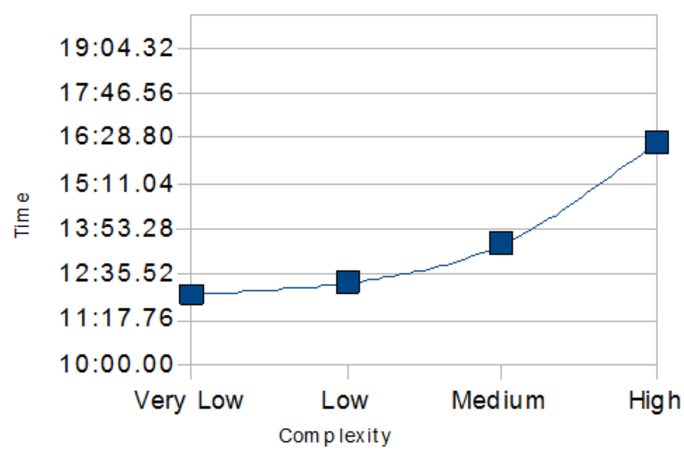


Figure 6.2 – Line graph showing the effect of complexity on required exposure time

As before, for the sake of convenience, we convert all values to seconds. The rate of change is not linear, and neither is the shape of the graph. We can not know, from the current set of data, whether in the long term this graph will turn into a tangent, a parabola, or a logistic graph. Our results, although bound by the experiment, allow us to calculate the impact of stepped complexity between a high state (8 obstructions per row), and a very low state (2 obstructions per row); which was not possible from the results of our experiments in chapter four. To work out in-between value, we can use linear interpolation. The equation for finding linear interpolation is given by $y = y_a + ((x - x_a)(y_b - y_a))/(x_b - x_a)$. For example, if we would like to know how long it will take a group of people to acquire spatial knowledge in a maze-like environment, of 128*128 pixel size, with three obstructions per row, we need to convert the graph points for two and four obstructions into seconds, and use the formula as before: $y = 721 + ((3 - 2)(741 - 721))/(4-2) = 731 =$ *12 minutes and 11 seconds*. So for an environment of 128*128 size, 16 unique object landmarks, and three obstructions per row, a person will require at least 12 minutes and 11 seconds to acquire spatial knowledge. This would significantly help military and emergency trainers predict required exposure times for personnel, at least in small maze-like environments (such as buildings). Such findings contribute towards both a better understanding of how SKA is influenced by various environmental factors, but also provide a guideline to VE trainers, which can be used to help avoid undesirable dis-orientation in complex environments. This is especially important in situations where global positioning systems are not available (such as in buildings). As it stands, our findings contribute to smaller indoor environments, however further work will guarantee a broader spectrum of environments. Linear interpolation allows us to 'map' certain bands of complexity (obstructions per row) to allow us to determine a quick time multiplier. Equations, seen in table 6.9, allow us to quickly calculate (compared to the control group time) a multiplier that relates to the relative impact of complexity on exposure time requirements.

Table 6.9 – Converting from complexity to time factor.

Pixel (x)	**y value**	**Time factor t(x)**
If 2 < x <= 4	y = (721 + (x-2)20)/2	t(x) = 1 + 0.014(x-2)
If 4 < x <= 6	y = (741 + (x-4)67)/2	t(x) = 1.0278 + 0.0046(x-4)
If 6 < x <= 8	y = (808 + (x-6)175)/2	t(x) = 1.1207+0.1214(x-6)

The following example considers the issue of mapping our definition of obstructions per row onto a real world space. Real environments are not so conveniently mapped into mazes. We already discussed how a natural terrain has many more features that contribute towards its complexity. However, even for indoor environments, complexity can not be measured as easily as in our experiments. We can hypothesise on a method that can help this measurement using a similar format to our own measurements. Figure 6.3 illustrates the map of a warehouse.

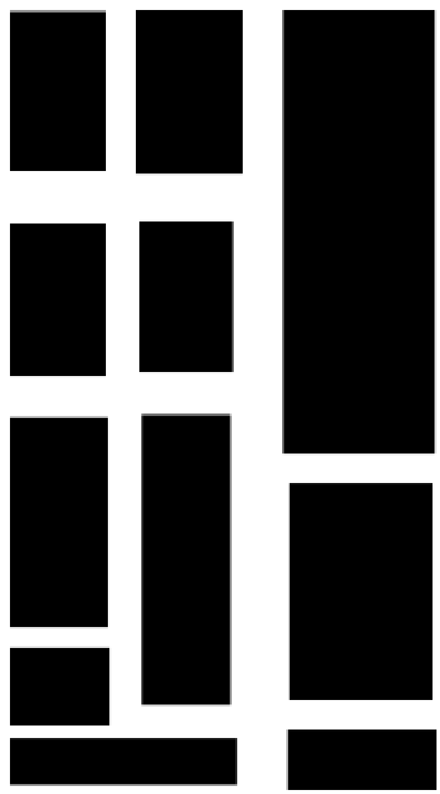


Figure 6.3 – Possible layout of a real life warehouse. Although it differs from our maps significantly, there is still a way to calculate its complexity

The illustration has eleven rooms that are separated by walls. Our method of measuring the complexity so far has been to count the amount of vertical walls separating each row. Our experiments always used eight rows, however if we break the environment into three parts, we can see that the first column has five rows, the second has four, and the third has three. The average amount of rows is four. The average amount of walls between these rows is two. Hence, we know, from our results, that the overall complexity of this map is even smaller than the "very-low" complexity map from our experiments. We can therefore safely assume, that for a group of participants with control like qualities, it should take less than 12 minutes for them to acquire spatial knowledge.

6.7 Further Insight on the Importance of Landmarks

To better understand the importance of well placed and unique object landmarks on exposure time, which a user needs to acquire spatial knowledge, we conducted our experiments on four different environments with varying amounts of landmarks. The environment most void of landmarks only had four in a total size of 128*128 pixels. We then doubled this amount each time, to reach an environment of 32 unique object landmarks. Our previous findings have already indicated that well placed object landmarks can greatly benefit the navigator; especially if the user is female. Our new findings aim to help us understand whether the gradual increase in landmarks benefit the user, and if so, in what way. The mean time taken for the participants to acquire spatial knowledge in all four 'landmark testing' environments is shown in table 6.10.

Table 6.10 – Results obtained from the environments used to investigate the impact of landmark usability.

Environment Type	Result
Small frequency of unique landmarks (4 landmarks)	16:15
In-between frequency of unique landmarks (8 landmarks)	13:06
Normal environment (control, 16 landmarks)	12:01
Large frequency of unique landmarks (32 landmarks)	12:17
Very large frequency of unique landmarks (64 landmarks), added as a post test	12:32

It is interesting to observe that after the normal environment (16 object landmarks), the time taken to learn the environment was actually longer, although a univariate analysis of variance shows no significance. To investigate the in-between environment significance of landmark frequency in our new experimental data, we ran a univariate analysis of variance, with landmark frequency as the independent variable, and time as the dependent variable. We found that the differences overall are less significant than size and complexity, but significant nonetheless ($F(1,3) = 23.499$, $P < 0.001$). We ran post-hoc tukey tests, in order to see whether the differences in results between the environments are significantly different to each other. We found that the difference between the small frequency landmark environment (four landmarks), and the in-between environment (eight landmarks) was very significant (Mean Difference 0:03:09; Std Err 0:00:32, $P < 0.001$). The difference between the in-between environment, and the normal environment (16 landmarks) was not significant (Mean Difference 0:00:45; Std Err 0:00:32, $P = 0.528$), however a significance was identified between the normal environment and the small frequency landmark environment (Mean Difference 0:03:54; Std Err 0:00:32, $P < 0.001$). The difference between the normal environment, and the large frequency landmark environment is not significant (Mean Difference 0:00:03; Std Err 0:00:32, $P = 0.999$), neither is the difference between the large frequency landmark environment when compared to the in-between environment (Mean Difference 0:00:48; Std Err 0:00:32, $P = 0.458$). Only the difference between the large frequency landmark environment, and the small frequency landmark environment was found to be significant (Mean Difference 0:04:58; Std Err 0:00:32, $P < 0.001$).

The increase in time between normal and large frequency landmarks, even though not statistically significant, is intriguing. The implication of increased time suggests that adding too many landmarks, even if they are unique, can contribute to a higher complexity, thus making it more difficult for a person to learn the environment. To investigate this further, we decided to run an additional test in an environment with 64 object landmarks. We found that the time required to learn the environment was even longer, with participants scoring a mean time of twelve minutes and thirty two seconds (see table 6.10 - very large frequency of unique landmarks). When we ran this through a univariate analysis of variance, and used post-hoc tukey test, we found that the results compared to the previously large frequency landmark environment (32 objects) was not significantly different (Mean Difference 0:00:33; Std Err 0:00:34, P = 0.991). We hypothesise, that the most probable reason why the landmarks stopped decreasing the overall required exposure time, is the cognitive load applied during navigation on the user's spatial awareness. A large increase in the amount of objects in the environment will ultimately lead to a burden on the perception stage. More objects will mean more visual processing. This is usually outweighed by the usefulness of these objects for navigation, but after a certain point, when most of these objects are not even used for navigation, they simply become a burden, therefore ceasing to work as navigational aids.

Our results show that users tend to use as many landmarks as possible, even if that means spending more time than is necessary. For our environment of 128*128 pixels (approximately 75 metres squared), it seems that eight object landmarks were sufficient. Adding more landmarks did not help the navigator. The implications of this contribute to both the area of military training, as well as emergency training, since these results indicate that soldiers will probably not have problems transferring knowledge into the real world if environments have at least some obvious landmarks. The results also contribute to VE design, since they indicate that to make an environment easy to remember, one does not have to flood it with unique landmarks, in fact by doing so a designer would increase the complexity of the environment. Instead, we suggest that for an environment of approximately 75 metres squared, which is not complex (two walls per row), a designer does not need to place more than 16 unique object landmarks in the maze environment. We hypothesise, that as far as the amount of landmarks is concerned, this is probably directly affected by both the size of the environment, and its complexity, since adding more physical obstructions will of course lead to visual obstructions between the navigator and possible object landmarks. Figure 6.4 illustrates the gradual change in time, as the environment is populated with more landmarks.

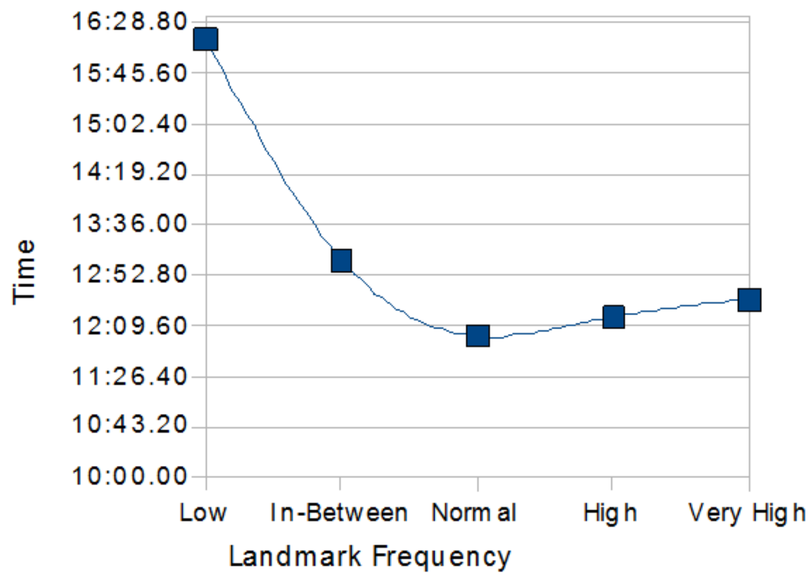


Figure 6.4 – Line graph showing the effect of landmarks on required exposure time.

To better understand the nature of the graph, we would need to create a new set of experiments, which look at how landmarks affect exposure time in different sized environments, and environments of different complexity. We would need to do this by increasing the size and complexity gradually, and running the landmark experiment for every environment using the same group of participants.

Although this is experimentally impractical, we can however use linear interpolation in order to predict the result of values situated between these points. Much like we did for size and complexity. For example, to find how much 24 object landmarks impact exposure time, in a control environment, we should add values for 16 and 32 landmarks (in seconds) into the interpolation equation as before: $y = y_a + ((x - x_a)(y_b - y_a))/(x_b - x_a)$. This would give us, $y = 721 + ((24 - 16)(737-721)/(32-16) = 726.5$ *= 12 minutes and 6.5 seconds. This would mean, that in an environment of 5898.24 square metres, with two obstructions per row, and 24 unique object landmarks, the total exposure time required to acquire spatial knowledge would be approximately 12 minutes, and six seconds.* Our results also allow us to map the landmarks between four and sixty four object landmarks to a time factor as seen in table 6.11.

Table 6.11 – Converting from landmarks to time factor

Pixel (x)	y value	Time factor t(x)
If 4 < x <= 8	y = (975 + (x-4)189)/4	t(x) = 1.3523 + 0.066(x-4)
If 8 < x <= 16	y = (786 + (x-8)65)/8	t(x) = 1.0902 + 0.0113(x-8)
If 16 < x <= 32	y = (721 + (x-16)16)/16	t(x) = 1+(x-16)/721
If 32 < x <= 64	y = (736 + (x-32)15)/32	t(x) = 1.0208+0.0007(x-32)

6.8 Using the Results to Predict Exposure Time

Our results do not allow us to predict the required exposure time for every combination of individual user differences and environments. To achieve an exhaustive set of tests is beyond the scope of this thesis. However, our results have contributed towards satisfying our main aim, and can be used to make some basic predictions for real life training and the development of basic training and design guidelines. When combined with the knowledge we obtained from our earlier experiments presented in chapter four, we begin to gain a good understanding of how time shifts accordingly for various individuals, and particular environments. With our cumulative knowledge of chapter four and this chapter, we created a diagram that can help us predict the required exposure time to acquire spatial knowledge for various individual differences, and environments. This diagram is shown in figure 6.5.

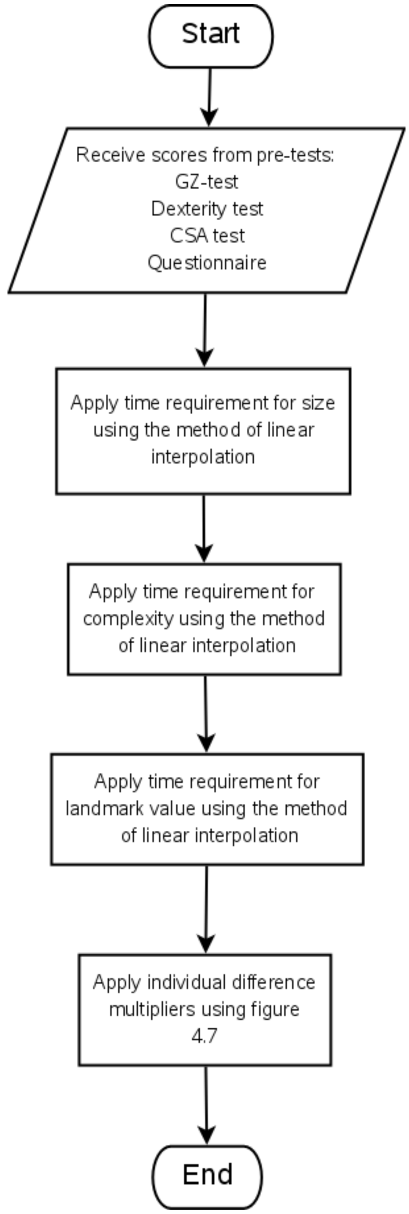


Figure 6.5 – Diagram illustrating method of predicting exposure time required to acquire spatial knowledge during VE training.

By following figure 6.5, we will now attempt to solve a real-world problem. Let us assume that we need to predict the required exposure time for a female trainee who needs to acquire spatial knowledge from a specific building. Let us assume that the building is an empty warehouse, with hardly any object landmarks, with six walls per row (on average), and is 80 metres square in size. We begin by looking at table 6.6 and checking the exposure time requirement for the control group in the control environment. This shows us that if this was an environment with environmental “traits” similar to the control environment, a person with the profile of our control group would require 12 minutes of exposure time in order to acquire spatial knowledge. This however is not the case. We therefore take the following necessary steps:

- Administer pre-tests - which defines the participant as being similar to the control group in the experiments in chapter 4, in all aspects of individual difference, with the exception of gender.
- We look at table 6.6 once more, and find the two exposure time durations, which relate to sizes just above and below the size of this particular environment. By applying linear interpolation we get an expected exposure time of +2 minutes to the total time.
- We then look at table 6.8, and notice that for six obstructions per row we will have to apply +1.5 minutes to the total time.
- We also need to count the amount of landmarks available. In this example we stated that the building is an empty warehouse, with only a few visible landmarks. From the results shown in table 6.10, we know that the difference between a normal amount of landmarks and a low amount, in terms of exposure time, was around three minutes. So we add +3 minutes to the total required exposure time.
- If we add up the weights, we can see that if this participant had similar individual user differences to our control group, they would require at least 12+2+1.5+3 = 18.5 minutes in order to acquire spatial knowledge from this environment to a degree that would allow them to transfer the knowledge into the real world.
- Because the participant differs from our control group in gender, we must also apply the appropriate gender multiplier from chapter four (see figure 4.7). In the case of female trainees, we apply the 1.42 multiplier, since we stated that this environment has a low number of landmarks. The resultant time is 26.27 minutes (18.5 x 1.42). We therefore assume that in this particular type of environment, this particular trainee will require over 26 minutes of exposure

time to acquire spatial knowledge.

These results may not be perfect, but can serve as a good guide. It is hoped that future experimentation will add and increasingly inform this process; providing an even better understanding of how exposure time changes when the values of the various environmental factors increase beyond our current limit.

6.9 Conclusion

In chapter four we discussed how individual user differences contributed towards a change in the required exposure time that a user needs in order to acquire spatial knowledge from an environment. In this chapter, we have looked at how the environmental factors affect the exposure time. This gives an overall understanding of how the process of spatial knowledge acquisition is affected during navigation on a cognitive level, which factors influence this process, and how different people may find themselves better suited to some environment types than others.

This chapter provided a variety of results. We found that size plays a major role during navigation. The time taken for users to acquire spatial knowledge in a small sized environment, and a medium sized environment are significantly different. This is also the case between a medium sized environment and a large sized environment. However, we showed that the difference in time taken between an environment of 128*128 pixels, and 192*192 pixels, was not significantly different. This result shows that users are able to process sizes up to 192*192 without facing cognitive overload, however beyond this size significant changes in required exposure time were identified. We also showed that using linear interpolation, we can now predict the total exposure time required for environments of up to 256*256 pixels in size, which relates to a space of approximately 150 metres squared (a considerable urban space).

We found that the differences in required exposure time between the different levels of complexity in an environment are not significant until the environment reaches a certain level of complexity. Again, we hypothesise that this has to do with limits in spatial memory, which allows up to a certain complexity. We showed how with linear interpolation we can predict the total exposure time required for environmental complexity of up to 8 obstructions per row.

Finally, we found that addition of landmarks only supports SKA, until a certain point. We showed that eight landmarks were enough in an environment of 128*128 pixels, with two obstructions per row. When we added more than eight landmarks, the differences in scores were not significant. In fact, after a certain point, the time taken actually increased, although not significantly so. We hypothesise that users only need a certain number of landmarks, depending on the size and complexity of the environment. We also showed how with linear interpolation we can predict the total exposure time required for environments with up to 64 unique object landmarks.

These findings contribute towards a better understanding of how different environments require different exposure times for a user during VE training. This is important for military training, and other types of emergency training, since it can help coordinate the required exposure time in advanced, but can also ensure that users of the VE will not find themselves disoriented during a real life crisis. Our findings indicate that all environmental factors are critical when considering the navigational difficulty of a VE. We found that as size increases, the cognitive load on the user is considerable, but only after a certain point (after 128*128 pixels, or approximately 75 metres squared). We found that as complexity increases, the cognitive load on the user is again considerable, but only after a certain point (in an environment of approximately 75 metres squared that seems to be above six obstructions per row (each row is a unit which is used to count the distance between all the horizontal walls of the environment). Finally we found that landmarks only help to a certain degree. More than a certain number of landmarks will not help navigation, and if there are no landmarks, navigation becomes very complex. In combination with the results found in chapter four, these findings are very useful to all domains of VE training.

Chapter 7 - Conclusion

7.1 Research Domain and Aim

Virtual environment (VE) training is an area of interest for a variety of application domains. Areas such as military training, emergency training and rehabilitation (Kim at al. 2007), workplace safety training (Lin et al., 2002), have all considered ways of using VE training to support real world interaction. Training knowledge of space is also an area of interest, and many researchers have reported that spatial knowledge acquisition (SKA) is feasible when spatial layout is learned within a VE (Peruch et al., 1995; Rossano and Moak, 1999; Ruddle et al., 1997; Waller et al., 1998; Wilson, et al., 1997). SKA research is a specific area of VE training, which attempts to clarify whether the ability to train in a VE can be applied to the acquisition and transfer of spatial information. Our research in chapter two, indicated that learning in such environments depends on both personal individual user differences, such as gender differences, and environmental factors such as the size of the environment. These factors and individual differences affect the exposure time that a person needs in an environment in order to gain spatial knowledge, and hence be able to transfer that knowledge into the real world. Chapter two, figure 2.1 summarises how both external stimuli, and individual user differences interact with the process of spatial knowledge acquisition.

Our research measures users acquiring spatial knowledge from a VE, and aimed to understand the impact that individual user differences and environmental factors have on exposure time requirements. Our work highlights specific individual groups that have problems when navigating in particular environments, and identifies the impact this has on required exposure time. In order to satisfy this aim, we constructed a series of objectives that will now be discussed in more detail.

7.1.1 - Objective One: Individual User and Environmental Differences

Our first objective was to discover which individual user differences and environmental factors impact the exposure time a user needs in order to acquire spatial knowledge from an environment. In chapter 2 we showed that the individual user differences, which affect the required exposure time are: gender; orientation skill; cognitive styles; previous knowledge of that type of environment; and previous knowledge of the system used in virtual environment (VE) training. Old age was also an issue, but was not looked at during the research, both because it has little contribution to military and emergency training, which were the main application domains, but also since the abilities that are influenced from the age groups can be measured using tests (e.g. older ages have a decrease in orientation, which can be measured using the Guilford – Zimmerman Orientation Survey). Chapter 2 also revealed that the environmental factors that contribute to navigational complexity, and therefore affect exposure time are: the size of the environment; the geometrical complexity of the environment; and the amount of unique object landmarks in the environment. In addition, we showed how the factors, and individual user differences interact with each other as seen in the model shown in chapter two, figure 2.2. The top layer of the model contains the environmental factors which affect SKA. The middle layer contains the individual user differences that affect SKA, the bottom layer contains the cognitive category that the middle layer belongs to, while the right column shows the actual process of SKA. We hypothesised in this diagram that cognitive styles only have a significant impact on complex and large environments.

7.1.2 – Objective Two: Discover Weighted Importances

Our second objective was to discover, through experimentation, the weighted importance of all the individual user differences and environmental factors on the process of SKA. In order for this to be accomplished we needed a two step process. First, we found multipliers that could be applied to the exposure time according to the participant's individual user differences. This was done in chapter four. Then we found variations (both positive and negative) in the exposure time requirements according to the environmental factors. This was shown in chapter six. These multipliers were discovered through experimentation. The experimental design was discussed in chapter three, which presented a set of tools and mechanisms for filtering and categorising participants into experimental groups. In order to find a participant's cognitive style, the CSA test was used. To find their orientation skill the Guilford-Zimmerman test (see appendix D.2) was used. The participant's experience with using computer systems was tested using a dexterity test, which was specifically created for this purpose (see Appendix B).

Since there was no standard tool that was used for the creation of environments for SKA research, we created our own 3D test bed. Chapter three describes the development of the Spatial Knowledge Acquisition Research (SKAR) engine. The SKAR engine was used to create the 3D environments required for our experimentation (both chapter four and chapter six). The SKAR engine allows a non-VE architect to quickly design and develop 3D environments using simple 2D bitmaps (see Appendix A for the source code). The results of the experiments are discussed in more detail in section 7.2.

7.1.3 - Objective three: VE Training Guidelines

Our third objective was to provide a set of guidelines that can help VE trainers consider the amount of training time required by an individual person, depending on the environment and their particular cognitive and biological differences. By analysing the results of chapters four and six, we created a diagram (see chapter six, figure 6.5) that can be used as a trainer's guide to predicting the required exposure time during training. By applying the appropriate multipliers, a VE trainer can predict the required exposure time that a particular user will require in a particular VE. This will help ensure that the trainees will not suffer from disorientation in critical emergency situations, which could possibly be life-threatening. Although our work did not intend to cover all types of VE space, but is instead limited to maze-like building environments, we believe it provides a contribution to the areas of military and emergency training, by clearly demonstrating that a one-size fits all exposure time (for variation in both user and environment type) is inappropriate and potentially dangerous to certain user groups.

We have demonstrated that all three research objectives were satisfied in this thesis. By satisfying the thesis aim, this research provides several contributions, which will be discussed in the following section.

7.2 Thesis Contributions and Findings

By satisfying our research aim, we have contributed towards a better understanding of how SKA works. We now know that it is in fact greatly dependent on both the individual user differences, and the environmental factors. We also know that we can measure these user differences and environmental factors, and predict to a satisfactory degree how much time an individual will require to learn the environment to the point that they can use this knowledge in the real world. Before we discuss our contributions, we will begin by presenting a summary of our findings.

The results of our experiments were shown and discussed in chapter four and chapter six in full detail, however, to summarise, we found that: orientation skill was the most influential individual user difference; system knowledge was the second most influential user difference; environmental knowledge was the third most influential user difference; female users had trouble navigating in environments with a low number of landmarks; and field-independent users have a significant advantage when navigating in complex environments, while field-dependent users have a significant advantage when navigating in large environments.

Our results give us insight concerning the importance of individual user differences on SKA (the first part of objective two, as discussed in section 7.1.2). However, in order to create any fundamental theories on the importance of variation in environmental factors (the second part of objective two) we needed to create transitional environments. The original environments in the first experiment were bipolar (i.e. variables changing from a low state to a high state , or vice-versa). This was fine when investigating the impact of individual user differences on extreme environment variations, but did not indicate the impact of stepped variation in environmental type (whether that means frequency and amount for landmarks and complexity, or simply space for size). We therefore ran additional experiments, only this time we created a linear range of environments, allowing us to test more than two bipolar states of environmental factors. Using the process of linear interpolation, this new set of experimental data allowed us to predict the exposure time required by participants when acquiring spatial knowledge in a particular environment within a certain variable range. Although we are limited by the range of variables considered in the experiment, the findings of our research are still very significant. In addition, results, although bound by environmental limitations, provide a stepping stone for further research. The importance of each environmental factor is discussed in detail in chapter six, however, to summarise, we found that: size significantly increases the required exposure time that a user needs to acquire spatial knowledge from a VE. An initial increase in size (between 64 pixels squared and 128 pixels squared) significantly impacts exposure time. No significant variation occurs until size is increased to 256 pixel squared, which suggests the existence of limitations in user cognitive capacity; complexity becomes a significant factor in terms of exposure time only after we increase this factor to at least six obstructions per row, before that the differences in exposure time was not significant; finally, adding more unique landmarks only helps lower the exposure time up to a certain point, after that adding more landmarks does not make a significant difference.

The results we found from the experiments allow us to both prove and disprove a set of hypotheses that were presented in chapter three.

- **H0:** We hypothesised that: 'When the size of the environment increases, more physical space must be processed. This leads to a greater information load being perceived, understood and used in decision making (O'Keefe and Nadel, 1978), which will increase the required exposure time to acquire spatial knowledge.', and we showed in our experiments that this is in fact correct.
- **H1:** We hypothesised that: 'When the amount of well placed object landmarks increases, navigation difficulty decreases (Vinson, 1999). This leads to a decrease in required exposure time', and we showed that this is correct until a certain amount of landmarks is exceeded, then the complexity stops decreasing and actually slightly increases (although not significantly so).
- **H2**: We hypothesised that: 'When spatial layout complexity is increased, more reference objects are obstructed (Stakienwikz et al., 2001). This leads to an increase in required exposure time.', and we showed that this is correct.
- **H3:** We hypothesised that: 'For a user with a large amount of system knowledge there is less attention divide during exploration (Booth et al., 2000). This leads to a better perception and understanding of the space, and can be a basis for better decision making during navigation.', and we showed that our hypothesis was correct.
- **H4:** We hypothesised that: 'A user with high environmental knowledge will have more schemas to use as a reference due to experience in navigating within that particular environment and will therefore have a better understanding of the environment and can make better decisions (Mania et al., 2005). This will lower required exposure time.', and we showed that this was correct.
- **H5:** We hypothesised that: 'A user with high aptitude/orientation skills will have more spatial memory and more orientation than a user with low

aptitude/orientation skills and will therefore will acquire knowledge faster. This will result in them requiring less exposure time (O'Keefe and Nadel, 1978; Smith and Millner, 1981; Maguire et al., 1996; Maguire et al., 1999; Maguire et al., 2000).', and our experiments showed that this is in fact true.

- **H6:** We hypothesised that: 'Female users will require greater exposure time than men in environments with fewer landmarks, but should require the same exposure time in environments with high landmark potential (Sandstrome et al., 1998).', and our experiments showed us that this is true.
- **H7**: We hypothesised that: 'Field-dependent (holistic learning style) users may take longer to acquire procedural knowledge in complex environments, due to their more passive approach to learning, which leads them to learn more irrelevant information (Pask, 1976; Pask, 1979), but will perform better in large environments for the same reason.', and we found that this was also true. We showed that field-dependent individuals require more exposure time in complex environments, but require less exposure time in large environments.

The process of satisfying our aim has brought us a variety of contributions, which will now be discussed.

- To the best of our knowledge, this is the first piece of work to actually group the various individual user differences and environmental factors, and to create a link between them. A new model was created (chapter two, figure 2.3 or figure 7.2) that models the literature described in chapter two, and shows which individual user differences interact with what environmental properties. We found from our results in chapter four, that the model was correct:

    - Women have a disadvantage when trying to acquire spatial knowledge from low landmark environment.
    - Verbal-analysers are better when navigating in complex environments,

but worse when navigating in large-sized environments.

  - Our experiments showed that low system knowledge had a significant impact on all environments, and caused a much higher exposure time requirement.
  - Our experiments showed that high environmental knowledge had a significant impact on all environments, giving participants with high environmental knowledge an advantage. This resulted in a lower exposure time requirement to acquire spatial knowledge.
  - Our experiments showed a very significant difference in scores for all environments between the low orientation skill group, and the high orientation skill group. The low orientation skill group required a significantly longer exposure time in order to acquire spatial knowledge.
- At the moment little or no information is given about the tools used to create 3D environments used in spatial knowledge research. This results in the 3D solution differing in many ways, which can impact the application of results (size is rarely given, complexity is not discussed, the amount of object landmarks are not discussed). We developed a tool that allows researchers to rapidly create an environment using 2D bitmaps. We have provided the source code for the SKAR engine (see appendix A), as this could help the VE training community adapt a universal tool when dealing with VE design (maps can be shared, objects created, etc.). By developing an easy to use experimental tool, we hope to provide open-source standardisation, which future researchers can consistently use if they intend to work in the research domain. Such a tool minimises conflict between separate research and would reduce errors when comparing VE models; as it is possible to simply send VE maps to other researchers trying to expand on the subject domain.
- The first part of objective two was to determine how individual user differences impact exposure time requirements when acquiring spatial knowledge from a VE. This is what we found:
  - The most influential user difference, in terms of total time taken

throughout the experiments, was orientation skill. This is quite unfortunate since orientation skill is very difficult to train.

  - Knowledge, both of the system, and the environment, is also very important. System knowledge, which is a user's experience with the training system is very easy to train, environmental knowledge (a person's experience with the particular type of environment) is not as easy, however it is not as difficult to train as orientation skill.
  - Females have a serious disadvantage when training in environments that are low in the number of unique landmarks (whether this can be trained, is something we would need to look at in the future).
  - Field-independent users (analytic-verbalisers in our experiments) will acquire spatial knowledge faster from a complex environment, however, field-dependent users (holistic-imagers in our experiments) have the advantage in large environments. These learning styles are formed through life, it is unlikely that a learning style will change through training.

The second part of objective two, was to determine how variation in environmental factors impacts the exposure time requirements when a user is acquiring spatial knowledge from a VE. What we found was:

- Size has a major impact on spatial knowledge acquisition. The time taken for users to acquire spatial knowledge in a small sized environment, and a medium sized environment were significantly different. This was also the case between a medium sized environment and a large sized environment. The time taken for users to acquire spatial knowledge in a small sized environment, and a medium sized environment were significantly different. It seems that at first the load on spatial memory increases required exposure time. As load is slowly added, however, there is a non-linear increase in exposure time requirements, suggesting that there is a grey area, where the differences in cognitive load do not significantly impact exposure time. As size moves towards 192 pixel squared, the burden of size quickly becomes too much for the user to handle. Therefore, it seems that at first the load on spatial memory is so small, that it takes almost no effort for the user during training. After that however, there is a grey area, where the differences in cognitive load are not significant until a certain point is reached. Then the burden is simply too much for the user to handle.

- Increase in complexity does not mean a gradual increase on cognitive load. Instead, the impact on the user is not significant until it reaches a certain point (above six obstructions per row). At this point the burden increases significantly and the user becomes disorientated.
- Increase in the amount of landmarks is beneficial only up to a certain point, after which the difference they make on navigational complexity is not significant. In fact, there seems to be an indication that at some point, too many landmarks will probably interfere with SKA. We hypothesise that this is because a large increase in the amount of objects in the environment will ultimately lead to a burden on the perception stage. More objects will mean more visual processing. This is usually outweighed by the usefulness of these objects for navigation, but after a certain point, when most of these objects are not used for navigation, they simply become a burden. The optimal number of landmarks is probably relative, and changes according to the size and complexity of the environment, although more work is needed in order to confirm this point.
- We created a new electronic version of the Guilford-Zimmerman orientation survey, since our participants complained that the answering system was far too complicated. This new version decreased the overall cognitive load of the complex answering system, and also decreases the overall training time it takes for someone to understand the test. Fortunately, our results showed that the negative interaction of the paper-based GZ test is consistent for all participants. This means that although there was a significant cognitive load, the load similarly affected all participants, and therefore did not interfere with participant categorisation. Participants with

lower orientation skills in the original test also scored lower in the new test, which means that both the old and new electronic GZ test are still valid for use as a pre-experimental test for categorisation of participant orientation ability. The test is readily available online, and is open source (see appendix D for the source code).

- We created a set of guidelines that can help a VE trainer predict the exposure time required for a user to acquire spatial knowledge from a particular environment. More specifically we found a variety of multipliers that can be applied for particular users, and grouped them in chapter four, figure 4.7, which we also present in this chapter for the sake of clarity (figure 7.6). Next we found the appropriate multipliers for the various environmental factors, and, when combined with the model in chapter four, provided a guide that allows VE trainers to predict as accurately as possible the required exposure time for each participant, depending on their individual differences, and the particular environment. The multipliers for the individual user differences can be seen in chapter four, figure 4.7, whilst the time requirements for different types of environments can be taken from tables 6.6, 6.8 and 6.10 (although linear interpolation may be required in order to predict a particular environment if it doesn't fit in the ones shown in the tables).

7.3 Limitations of our research

Our research has contributed to the domain of VE training in many ways. It has provided an insight on the importance of control, when applying training techniques to individuals. It has helped us understand which cognitive and biological abilities are most important to acquiring space from a VE, and has shown us strong evidence of how the difficulty that people face when creating cognitive maps changes according to the environmental factors they encounter during navigation. In that respect, this thesis guides future research to a more empirical approach when handling VE training, and calls for more experimental control, and better use of methodologies in future SKA research. However, there are still a lot of limitations. At the moment, we have focused our research to a very specific environment type. Only maze environments were considered. We have not been able to predict the exposure time for environments made up of factors beyond the range of our test bed (although prediction can occur if time is taken for a particular user with control-like qualities, and multipliers are then applied to the rest of the users, regardless of the environment), instead we rely on linear interpolation in order to provide a multiplier for each environmental factor within a certain range. This range was 64*64 – 256*256 pixels for size, 2 – 8 points of complexity, and 4 – 64 landmarks. We also have not been able to prove that our guidelines for predicting exposure time are accurate. We assume that knowledge transfer has occurred, because participants were able to accurately position landmarks on a draft map, we have not actually observed them apply this knowledge to a real environment. In order to achieve this, we would have to spend a significant amount of time and resources transferring a real world space into a VE test bed, and then having participants actually navigate in the actual space after the appropriate training time is given. Moreover, because our thesis follows an empirical approach, we have only looked at the individual user differences and environmental factors that currently stand out in the literature. By adopting a different approach, perhaps using mathematical modeling, we might find even more factors that impact on SKA. Finally, our results would have been more reliable if we had more participants. Ultimately, we believe that many of these limitations can be dealt with in future work.

7.4 Future work

Although there are various limitations in our research, there is optimism in our work as well, for we have shown that spatial knowledge acquisition is feasible if enough exposure time is given. Our findings are a good starting point for a very large research area.

At times, we felt we must conduct further research either to solidify our work, or to expand our knowledge on the subject. Future research must consider a number of issues, which we will discuss in more detail:

- Our work only considers a very small sample of the possible combinations of user differences an individual may have. We understand the importance of each individual user difference while controlling the rest, but in the real world, people tend to differ in more than one cognitive ability (and indeed our participant filtering showed us that this is the case). For example, we know how constrained a female user, with high system knowledge, low environmental knowledge, field-dependent cognitive style and high orientation skill will perform in an environment; but we do not know what will happen if she also has a field-independent cognitive style, and has low system knowledge (since this differs in more than one aspect to the control group). We hypothesise that the multipliers are applied sequentially. Therefore, we simply apply all multipliers to the control time. We base our guide on this speculation, and must reinforce this in the future by further experimentation.
- Our work currently focuses on maze-like environments, we will need to expand this to other types of environments, and present suitable methodologies that can also control complexity in these environments.
- More work is needed to discover the effect that the environmental factors have on exposure time outside our current range. This can be done by running experiments with factors greater than the current maximum.
- We know that landmarks can significantly benefit a person navigating in a particular environment. Our results showed that after a certain number, they actually decrease a person's navigational ability (although not significantly). More work is required to understand the effect that an overpopulation of landmarks has on a participant's cognition.
- An experiment must be conducted in a representation of a real world environment, and tests must be held to see if the training guidelines apply. This will help prove that knowledge transfer did in fact occur with the

predicted exposure time for a particular participant.

Ultimately, we believe that once all these problems are tackled, we will greatly reinforce, and expand our current guide to VE training. We hope that this guide will greatly contribute to a variety of application areas by providing a suitable methodology concerning the filtering of participants, the creation of environments, and the prediction of exposure time requirements during the VE training process.

References

Allport G.W. (1937). Personality: A psychological interpretation. New York: Holt & Co.

Asthmeir P. Feiger W. & Muller S. (1993), Virtual design: a generic VR system for industrial applications. Computer and Graphics, 17. 671 – 677.

Astur R. S. Ortiz M. L. & Sutherland R. J. (1998). A characterization of performance by men and women in a virtual Morris water task: A large and reliable sex difference. Behavioural Brain Research, 93(1-2), 185-190.

Atkinson S. (2001). New Media in Technology Education. Proceedings PATT-11 Conference, 8-13.

Aubrey J.B. Dobbs A.R. (1990) Age and sex differences in the mental realignment of maps. Experimental Aging Research 16 (3), 133-139.

Birren J. E. & Livingston J. (1985). Cognition, stress, and aging. Englewood Cliffs, NJ: Prentice Hall.

Bliss J.P. Tidwell P.D. & Guest M.A. (1997). The Effectiveness of Virtual Reality for Administering Spatial Navigation Training to Firefighters. Presence, 6(1). 73-86.

Booth K. Fisher B. Page S. Ware C. & Widen S. (2000). Wayfinding in a virtual environment. Graphics Interface.

Brewer W. F. & Nakamura, G.V. (1984). The nature and functions of schemas. Handbook of social cognition, 1, 119-160.

Brewer W.F. (2000). Bartlett's concept of the schema and its impact on theories of knowledge representation in contemporary cognitive psychology. In Saito (Ed.), Bartlett, culture and cognition, 69-89, Psychology Press

Brown H.D. Lahar C.J. Mosley J.L. (1998). Age and gender-related differences in strategy use for route information: A “map-present” direction-giving paradigm. Environment and Behavior, 30, 123-143.

Butler D.L. Acquino A.L. Hissong A.A. and Scott P.A. (1993). Wayfinding by

Newcomers in a Complex Building. Human Factors, 35(1) 159-173.

Bryden S.M. and Tapley M.P. (1977) An investigation of sex differences in spatial ability: mental rotation of three-dimensional objects. Canadian Journal of Psychology 31(3), 122-30

Byrne R.W. (1979) Memory for Urban geography. Quarterly Journal of Experimental Psychology 31, 147-154.

Chase W.G. (1983) Spatial representations of Taxi drivers. Acquisition of Symbolic skills. New York, Plenum.

Chandler P. and Sweller J. (1996) Cognitive Load While Learning to Use a Computer Program. Applied Cognitive Psychology.

Chase W.G. & Chi M.T.H. (1981). Cognitive skill: Iimplications for spatial skill in large-scale environments. In J.H. Harvey (Ed.). Cognition, social behavior, and the environment (111-135). Hillsdale, NJ: Erlbaum.

Chen S.Y & Macredie R.D. (2001). Cognitive styles and hypermedia navigation: Development of a learning model. Journal of the American Society for Information Science and Technology, 53(1), 3-15.

Chen S.Y. Magoulas G.D. & Dimakopoulos D (2005). A Flexible Interface Design for Web Directories to Accommodate Different Cognitive Styles. Journal of the American Society For Information Science and Technology, 56(1), 70–83.

Chen J.Y.C. & Thropp J.E. (2007) Review of Low Frame Rate Effects on Human Performance, Systems, Man and Cybernetics, Part A. IEEE Transactions on , vol.37, no.6, pp.1063-1076

Clark S.A. & Wong B.L.W. (2000). QTVR Support for Teaching Operative Procedures in Dentistry. People and Computer XIV. Usability or Else! Proceedings of HCL 2000 , London.

Cobb S.U.G. & D'Cruz, M. D. (1994) First UK national survey on industrial application of virtual reality. VR News, 3.

Cohen R. and Schuepfer T. (1980). The representation of landmarks and routes.

Child Dev. 51: 1065-1071

Cooper. G. (2004) Research into Cognitive Load. Theory and Instructional Design at UNSW.

Crook T.H. Young J.R. and Larrabee G.J. (1993) The influence of age, gender and cues on computer-simulated topographic memory, Developmental Neuropsychology, 9, 41-53.

Darroch I. Goodman J. Brewster S. and Gray P. (2005) The Effect of Age and Font Size on Reading Text on Handheld Computers. Proceedings of IFIP INTERACT05: Human-Computer Interaction pp. 253-266.

Darken R.P. Sibert J.L. (1996). Wayfinding strategies and behaviours in large virtual worlds. Proceedings of ACM CHI96, 142-149.

Darken R.P. & Banker W.P. (1998). Navigating in Natural Environments: A Virtual Environment Training Transfer Study. Proceedings of VRAIS '98, 12-19.

Darken R.P, Allard T, & Achille L.B (1999) Spatial Orientation and Wayfinding in Large-Scale Virtual Spaces II. Presence, Vol 8. No 6, December.

Darken R.P. & Peterson B. (2001). Spatial Orientation, Wayfinding, and Representation. Handbook of Virtual Environment Technology. Stanney, K. Ed.

Dewitte S. Pandalaere M. Briers B. Luk W. (2005) Cognitive load has negative after effects on consumer decision making KUL. Departement toegepaste economische wetenschappen, Leuven, K.U.Leuven DTEW research report vol:45

Dix A. Finlay J. Abowd G. & Beale R. (1993). Human computer interaction. New York: Prentice Hall.

Doyle S. Dodge M. & Smith A. (1998). The potential of web-based mapping and virtual reality technologies for modeling urban environments. Computers, Environment, and Urban Systems 22(2): 137-155.

Eberts R.E. (1994). User interface design. Englewood Cliffs, NJ: Prentice Hall.

Egan D.E. & Gomez L.M. (1985). Assaying, isolating, and accommodating individual differences in learning a complex skill. In R.F. Dillon (Ed.), Individual differences in cognition (pp. 173–217). Orlando: Academic Press.

Egan D. E. (1988). Individual differences in human-computer interaction. In M. Helander (Ed.), Handbook of human-computer interaction (pp. 543–568). Amsterdam, Netherlands: North Holland

Egsegian, R., Pittman, K., Farmer, K., & Zobel, R. (1993). Practical applications of virtual reality to firefighter training. In Proceedings of the 1993 Simulations Multiconference on the International Emergency Management and Engineering Conference (pp. 155–160). San Diego, CA: Society of Computer Simulation

Eliot and Price (1976) Convergent and Discriminant Validities of Two Sets of Measures of Spatial Orientation and Visualization. Educational and Psychological Measurement, 35

Feldman A. & Acredolo L.P. (1979). The effect of active versus passive exploration on memory for spatial location in children. Child Development, 50, 698-704.

Fisk A.D. & Rogers W.A. (1991). Toward an understanding of age-related memory and visual search effects. Journal of Experimental Psychology, 120, 131–149.

Ford N. (1995). Levels and types of mediation in instructional systems: An individual difference approach. International Journal of Human–Computer Studies 43, 243–259.

Ford N. & Chen S.Y. (2001). Matching/mismatching revisited: An empirical study of learning and teaching styles. British Journal of Educational Technology, 32(1), 5–22.

Foreman N. Stanton D. Wilson P. & Duffy H. (2003). Spatial Knowledge of a Real School Environment Acquired From Virtual or Physical Models by Able-Bodied Children and Children With Physical Disabilities. Journal of Experimental Psychology. 9(2), 67–74.

Gallistel C.R. (1990). The organisation of learning. Cambridge M.A: The MIT Press.

Ghinea G. & Chen S.Y. (2003). The impact of cognitive styles on perceptual distributed multimedia quality. British Journal of Educational Technology 34(4) 393–406.

Gillner S. & Mallot H.A. (1998). Navigation and acquisition of spatial knowledge in a virtual maze. Journal of Cognitive Neuroscience, 10, 445-463.

Goillau P. Woodward V. Kelly C. & Banks G. (1998). Evaluation of virtual prototypes for air traffic control – the Macaw technique. M. A. Hanson, Contemporary Ergonomics, London, Taylor and Francis.

Goerger S. Darken R. Boyd M. Gagnon T. Liles S. Sullivan J. & Lawson J.P (1998). Spatial Knowledge Acquisition from Maps and Virtual Environments in Complex Architectural Spaces. Proceedings of the 16th Applied Behavioral Sciences Symposium, 22-23 April, U.S. Air Force Academy. Colorado Springs, 6-10.

Golledge R.G. Dougherty V. & Bell S. (1995). Acquiring Spatial Knowledge: Survey versus Route-Based Knowledge. Unfamiliar Environments Reginald Annals of the Association of American Geographers, 85(1), 134-158.

Gomez L.M. Egan D.E. & Bowers C. (1986). Learning to use a text editor: Some learner characteristics that predict success. Human-Computer Interaction, 2, 1–23.

Goodenough D. (1976). The role of individual differences in field dependence as a factor in learning and memory. Psychological Bulletin, 83, 675–694.

Gouteux S. Spelke E. (2001): Children's use of geometry and landmarks to reorient in an open space. Cognition 81 119-148.

Graff M. (2003). Cognitive Style and Attitudes Towards Using Online Learning and Assessment Methods. Electronic Journal of e-Learning, 1(1).

Guilford J.P. Zimmerman W.S. (1948). The Guilford-Zimmerman Aptitude Survey. Journal of Applied Psychology, 32, 24--34.

Hasher L, & Zacks R.T. (1979). Automatic and effortful processes in memory. Journal of Experimental Psychology, Vol. 108: 356-388

Hegarty. M. & Waller, D. (2004). A dissociation between mental rotation and perspective-taking spatial abilities. Intelligence, 32, 175-191.

Hermer L. & Spelke E. (1996). Modularity and development: the case of spatial reorientation. Cognition, 61, 195-232.

Hertzog C. (1989). Influences of cognitive slowing in age differences in intelligence. Developmental Psychology, 25, 636–651.

Hirtle S.C. Mascolo M. F. (1992) The Heuristics of Spatial cognition. Proceedings of the 13th annual conference of the cognitive science society. Hillsdale, N: Erlbaum pp. 629-634

Howes A. Jones D. Payne S. and Ruddle R. (1999). Navigating Discontinuous Virtual Worlds. Workshop on Spatial cognition in real and virtual environments at the Max Planck Institute, Tuebingen.

Hunt, E. & Waller, D. (1999). Orientation and wayfinding: A review. ONR Technical Report.

Riding M. Stelovsky J. & Cros M. (2003). Cognitive Load and Adaptive Instructional Designs for Computer-based learning. IADIS International Conference e-Society

Kalia A. & Stankiewicz B.J. (2007). Acquisition of Structural Versus Object Landmark Knowledge. Journal of Experimental Psychology: Human Perception & Performance, 33(2), 378-390.

Kim J. Kim K. Kim D.Y. Chang W.H. Park C. Ohn S.H. Han K. Ku J. Nam S.W. Kim I.Y

Kim .I. (2007) CyberPsychology & Behaviour. February 1, 10(1): 7-15.

Kirasic K.C. (1991) Spatial Cognition and Behavior in Young and Elderly Adults: Implications for Learning New Environments, Psychology and Aging 6; 10-18.

Kirasic K.C. (2000) Age differences in adults' spatial abilities, learning environmental layout, and wayfinding behavior. Spatial Cognition. Computing. 2:117–134.

Koh G. Wiegand T.E. Garnett R. Durlach N. Cunningham B.S. (1999). Use of

Virtual Environments for Acquiring Configurational Knowledge about Specific Real-World Spaces. Preliminary Experiment, 8(6), 632-656.

Kuipers B.J. (1975). A frame for frames: Representing knowledge for recognition. In D. G. Bobrow & A. Collins (Eds.), Representation and understanding: Studies in cognitive science. New York: Academic Press.

Lampinen J. Copeland S. Neuschatz J. (2001). Recollections of things schematic: rooms schemas revisited. Cognition, 27, 1211-1222.

Lee C.H. Liu A. Castillo S. Bowyer M. Alverson D. Muniz G. Caudell T.P. (2007). Towards an Immersive Virtual Environment for Medical Team Training, Studies in Health Technology and Informatics, 125, 274-279.

Lin F. Lan Y. Duffy V. & Su (2000) Developing Virtual Environments for Industrial Training. Information Sciences. Informatics and Computer Science: An International Journal, 140(1), 153-170.

Lipska B.K. al-Amin H.A. & Weinberger, D. R. (1998). Excito-toxic lesions of the rat medial prefrontal cortex: Effects on abnor mal behaviors associated with neonatal hippocampal damage. Neuropsychopharmacology 19: 451– 464.

Liu Y. & Ginther D. (1999). Cognitive Styles and Distance Education. Online Journal of Distance Learning Administration, 2(3).

Lloyd R. Heivly C. (1987) Systematic distortions in urban cognitive maps. Annals of the Associations of American Geographers 77, 191-207

Lloyd R. (1989) Cognitive maps: Encoding and decoding information. Annals of the Association of American Geographers 79, 101-124

Lynch K. (1960). The Image of the city. The MIT press.

Mania K. Robinson A. & Brandt K.R. (2005). The effect of memory schemas on object recognition in virtual environments. Presence: Teleoperators and Virtual Environments archive, 14(5), 606 – 615

Maguire E.A. Frackowiak R.S.J & Frith C.D. (1996). Recalling Routes around London: Activation of the Right Hippocampus in Taxi Drivers. The Journal for

Neuroscience, 17(18), 7103-7110.

Maguire E.A. Burgess N. & O'Keefe J. (1999) Human spatial navigation: cognitive maps, sexual dimorphism, and neural substrates. Current Opinion in Neurobiology 1999, 9:171–177.

Maguire E.A. Gadian D.G. Johnsrude I.S. Good C.D. Ashburner J. Frackowiak R.S.J. & Frith C.D. (2000). Navigation-related structural change in the hippocampi of taxi drivers. Proceedings of the national academy of science (USA), 97(8)

Marsh T. Smith S. (2001) Guiding user navigation in virtual environments using awareness of virtual off-screen space. User Guidance in Virtual Environments: Proceedings of the Workshop on Guiding Users through Interactive Experiences- Usability Centered Design and Evaluation of Virtual 3D Environments. Volker Paelke, Sabine Volbracht(Editors). pg 149-154, Shaker Verlag, Aachen, Germany

Mathews M.H. (1992) Making Sense of Place: Children's Uncerstanding of Large-Scale Environments. Harvester Wheatsheaf, Hertfordshire, England.

Matthews M.D. Pleban R.J. Endsley M. R. & Strater L.G. (2000). Measures of infantry situation awareness for a virtual MOUT environment. Proceedings of the Human Performance, Situation Awareness and Automation: User-Centered Design for the New Millennium. Savannah, GA: SA Technologies, Inc.

Messick S. (1976). Individuality in learning. San Francisco: Jossey-Bass.

Messick S. (1994). The Matter of Style: Manifestations of Personality in Cognition. Learning, and Teaching. Educational Psychologist 29(3), 121-136.

Miller G.A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological Review, 63, 81-97.

Moar G. Bower H. (1983) Inconsistency in spatial knowledge. Memory and Cognition 11, 107-113.

Moffat S.D. Hampson,E. & Hatzipantelis M. (1998) Navigation in a "virtual" maze: sex differences and correlation with psychometric measures of spatial ability in humans. Evolution and Human Behaviour, 19, 73-78.

Montello D.R. (2005) Navigation. In P. Shah & A. Miyake (Eds.), The Cambridge Handbook of Visuospatial Thinking. Cambridge University Press, 257-294..

Moser S.D. (1988). Cognitive mapping in a complex building. Environment and Behavior, 20, 21-49.

Newman E.L. Caplan J.B. Kirschen P. Korolev I.O. Sekuler R. & Kahana M.J. (2007). Learning your way around town: How virtual taxicab drivers learn to use both layout and landmark information. Cognition, 104, 231-253.

O'Keefe J. and Nadel L. (1978) The hippocampus as a cognitive map. Clarendon Press, Oxford.

Pair J. & Rizzo, A. (2006) Cognitive Performance Assessment in Mixed and Virtual Environment Systems. In Virtual Media for Military Applications. Proceedings RTO-MP-HFM-136, Neuilly-sur-Seine, France.

Pask G. (1976). Styles and strategies of learning. British Journal of Educational Psychology, 46, 128–48.

Pask G. (1979). Final report of S.S.R.C. Research programme HR 2708. Richmond (Surrey): System Research Ltd.

Passini R. (1984) Wayfinding in Architecture. Van Nostrand Reinhold Company Inc. New York

Pass F. Renkl A. Sweller J. (2003) Cognitive Load Theory and Instructional Design: Recent Developments. Educational Psychologist, 38(1), 1–4

Peruch P. Vercher J. & Gauthier G. M. (1995). Acquisition of spatial knowledge through visual exploration of simulated environments. Ecological Psychology, 7, 1 – 20.

Peterson L. & Peterson M. (1959). Short-term retention of individual verbal items. Journal of Experimental Psychology, 58, 193-198.

Petersen A.C. & Linn M.A. (1985). Emergence and characterization of sex differences in spatial ability. Child Development, 56, 1479-1498.

Pine D.S. Grun J. Maguire E.A. Burgess N. Zarahn E. Koda V. Fyer A. Szeszko

P.R. &

Bilder R.M. (2002) Neurodevelopmental Aspects of Spatial Navigation: A Virtual Reality fMRI Study. NeuroImage. 15, 396-406

Price J.L. Catrambone R. (2004) Part-Whole Statistics Training: “NSDL: Towards Reusable and Shareable Courseware: Effects on Learning and Cognitive Load, CogSci, Chicago

Riding R.J. (1991). Cognitive styles analysis. Birmingham, UK: Learning and Training Technology.

Riding R.J. Grimley, M. (1999) Cognitive Style, Gender and Learning from Multi-media. British Journal of Educational Technology, 30, 1, 43 -- 56.

Riva G. & Gambrini L. (2000). Virtual reality in telemedicine. Telemedicine Journal, 6, 327 – 340

Robertson I.T. (1985). Human information-processing strategies and style. Behavior and Information Technology, 4 (1), 19-29.

Romano D. Brna P. and Self J. (1998). Collaborative decision-making and presence in shared dynamic virtual environments. In Proceedings of the Workshop on Presence in Shared Virtual Environments. BT Labs, Martlesham Heath.

Rosanno M. J. & Moak J. (1998). Spatial representations acquired from computer models: Cognitive load, orientation specificity, and the acquisition of survey knowledge. British Journal of Psychology, 89, 481-497.

Rose F.D. and Foreman N.P. (1999). Virtual reality. The Psychologist, 12, 551-554.

Rose F. D., Attree E. A., Brooks B. M., Parslow D. M., & Penn P. R. (2000) Training in virtual environments: transfer to real world tasks and equivalence to real task training Authors. Ergonomics 43(4). 494-511.

Ruddle R.A. Payne S.J. & Jones D.M. (1997). Navigating buildings in “desk-top” virtual environments: Experimental investigations using extended navigational experience. Journal of Experimental Psychology: Applied, 3, 143-159.

Sadalla E.K. Montello D.R. (1989) Remembering changes in direction.

Environment and Behaviour 21, 346-363.

Sandstrom N.J. Kaufman J. Huettel S.A. (1998). Males and females use different distal cues in a virtual environment navigation task. Cognitive Brain Research, 6, 351-360.

Salthouse T. Donald K. Saults S. (1990) Age, Self-Assessed Health Status and Cognition. Journal of Gerontology 45(4):P156-P160

Salthouse T.A. (1992). Why do adult age differences increase with task complexity? Developmental Psychology, 28, 905–918.

Schmeck R.R (1988). Learning strategies and learning styles Plenum Press.

Schnabel M.A. and Kvan T. (2001) A 3D Maze: Getting Lost in Virtual Reality. 5th Iberoamerican congress of digital graphic SIGRADI 2001 [Conference Proceedings] Conception (Chile) 21-23 November 2001.

Siegel A.W. & White S.H. (1975). The Development of Spatial Representation of Large- Scale Environments. In H.W. Reese (Ed.), Advances in Child Development and Behavior. New York: Academic Press

Silverman D. R. & Spiker V. A. (1997), Ancient wisdom—future technology. Proceedings of the Human Factor and Ergonomics Society, 41st Annual Meeting Albuquerque, New Mexico.

Smith, M. L., and B. Milner (1981) The role of the right hippocampus in the recall of spatial location. Neuropsychologia 19:781-793

Smith R.M. (1984). Learning how to learn. Milton Keynes: Open University.

Smith E. and Riding R. (2004). Cognitive style and instructional preferences. Instructional Science, Vol 27, No. 5, pp 355-371

Smulders T.V. Sasson A.D. & DeVoogd T.J. (1995) Seasonal variation in hippocampal volume in a food-storing bird, the black-capped chickadee, Journal of Neurobiology 27, pp. 15–25

Stankiewicz B.J. & Kalia A. (2004). Acquisition and Retention of Structural versus Object Landmark Knowledge When Navigating through a Large-Scale Space.

Journal of Experimental Psychology: Human Perception & Performance, 33(2), 378-390.

Stanney K.M. & Salvendy G. (1995). Information visualization; assisting low spatial individuals with information access tasks through the use of visual mediators. Ergonomics, 38(6), 1184–1198.

Stanney K.M., Mourant R.R. & Kennedy R.S. (1998). Human Factors Issues in Virtual Environments: A Review of the Literature. Presence, 7(4), 327–351.

Stanton D. Foreman N. Wilson P. Duffy H. & Parnell R. (2002) Use of virtual environments to acquire spatial understanding of real-world multi-level environments. Proceedings of the 4th International Conference. Disability, Virtual Reality & Assoc. Tech., Veszprém, Hungary,

Stumpf H. & Fay E. (1983). Schlauch guren—Ein Test zur Beurteilung des räumlichen Vorstellungsvermögens. Göttingen: Hogrefe

Sweller J. (1988) Cognitive load during problem solving: effects on learning. Cognitive Science, 12, 257-285.

Sweller J. (1994). Cognitive load theory, learning difficulty and instructional design. Learning and Instruction, 4, 295-312.

Tan D.S. Gergle D. Scupelli P. Pausch R. (2003) With similar visual angles, larger displays improve spatial performance. Proceedings of the SIGCHI conference on Human factors in computing systems, April 05-10, Ft. Lauderdale, Florida, USA

Thorndyke P. (1980). Performance models for spatial and locational cognition (R-2676-ONR). Washington, D.C.: The Rand Corporation.

Thorndyke P. & Hayes-Roth B. (1982). Differences in spatial knowledge acquired from maps and navigation. Cognitive psychology 14, 560-89.

Thorndyke P. & Goldin S.E. (1983). Spatial learning and reasoning skill. In H.L.Pick & L.P.Acredolo (Eds.), Spatial Orientation: Theory, research and application (195-217). New York: Plenum.

Tversky B. (1981) Distortions in memory for maps. Cognitive Psychology, 13, 407-433

Tversky B. (1993). Cognitive maps, cognitive collages, and spatial mental models. In A. V. Frank and I. Campari (Eds.) Spatial information theory: A theoretical basis for GIS. Berlin: Springer-Verlag.

Vila J. Beccue B. Anandikar S. (2002). The Gender Factor in Virtual Reality Navigation and Wayfinding. Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03).

Vicente K.J. Hayes B.C. & Williges R.C. (1987). Assaying and isolating individual differences in searching a hierarchical file system. Human Factors, 29(3), 349–359.

Vinson N.G. (1999) Design Guidelines for Landmarks to Support Navigation in Virtual Environments. Proceedings of CHI '99, Pittsburgh, PA.

Waller D. Hunt E. and Knapp D. (1998). The transfer of spatial knowledge in virtual environment training. Presence: Teleoperators and Virtual Environments, 7, 129-143.

Waller D. Knapp D. and Hunt E. (2001) Spatial Representations of Virtual Mazes: The Role of Visual Fidelity and Individual Differences. Human Factors: The Journal of the Human Factors and Ergonomics Society, Vol. 43, No. 1, 147-158

Weisman J. (1981). Evaluating achitectural legibility: Way-finding in the built environment. Environment and Behavior, 13, 189-204.

Whitaker L.A. (1996) Getting around in the natural world. Ergonomics in Design, 4, 3, 11-15.

Wilkniss S. M. Jones M. Korel D. Gold P. & Manning C. (1997) Age-related differences in an ecologically based study of route learning. Psychology and Ageing 12(2) 372-375

Wilson P. N. Foreman N. & Tlauka M. (1997). Transfer of spatial information from a virtual to a real environment. Human Factors, 39(4), 526 - 531.

Witkin H.A. Oltman P.K. Raskin E. & Karp S.A. (1971). A manual for the group embedded figures test. Palo Alto, CA: Consulting Psychologists Press.

Witkin H. A. Moore C. A. Goodenough D.R. and Cox P.W. (1977) Field-dependent and field independent cognitive styles and their educational implications. Review of Educational Research Vol. 47, pp 1–64.

Witmer B.G. Bailey J.H. & Knerr B.W. (1995), Training Dismounted Soldiers in Virtual Environments: Route Learning and Transfer. U.S. Army Research Institute for the Behavioral and Social Sciences

Witmer B.G. Bailey J.H. Knerr B.W. & Parsons K.C. (1996). Virtual spaces and real world places: transfer of route knowledge. International Journal of Human-Computer Studies, 45(4), 413-428

Witmer B.G. & W.J. Sadowski. (1998) Nonvisually guided locomotion to a previously viewed target in real and virtual environments. (Special Section: Virtual environments: Models, Methodology, and Empirical Studies). Human Factors, 40, 478--485.

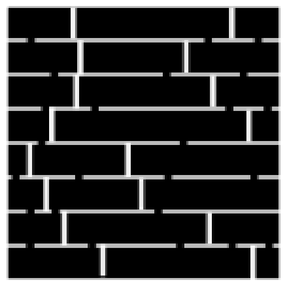
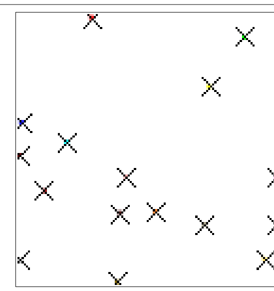
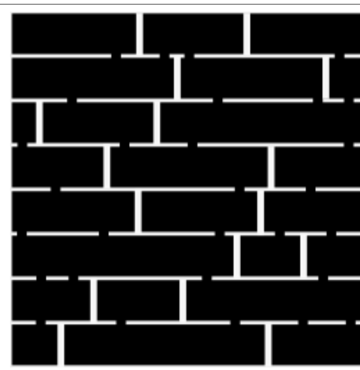
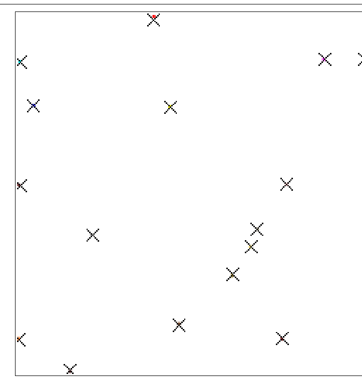
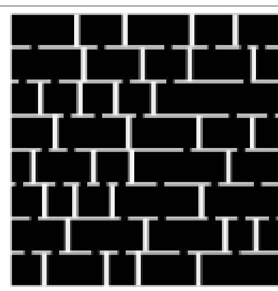
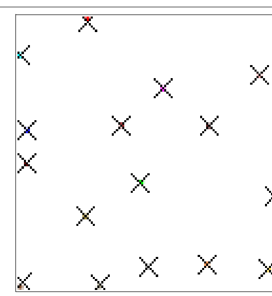
Xie B. & Salvendy G. (2000). Review and reappraisal of modeling and predicting mental workload in single- and multi-task environments. Work and Stress, 14 (1), 74-99

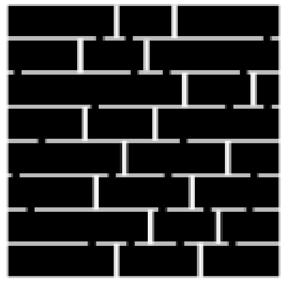
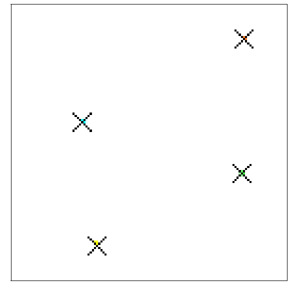
Yousoof M. Sapiyan M. & Kamaluddin K. (2007). Measuring Cognitive Load - A Solution to Ease Learning of Programming. Proceedings of World Academy of Science, Engineering and Technology Volume 20

Appendix A - SKAR ENGINE

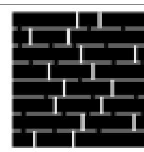
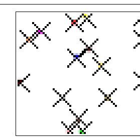
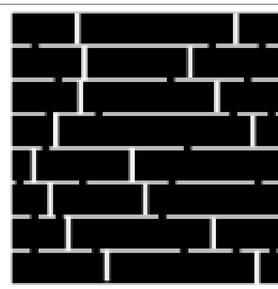
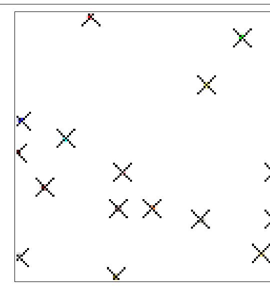
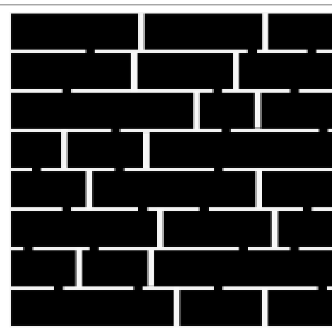
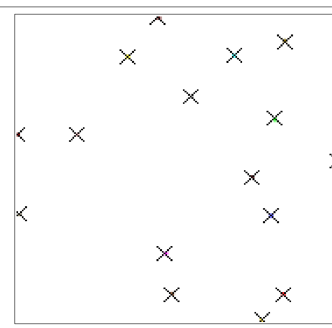
A.1 Maps

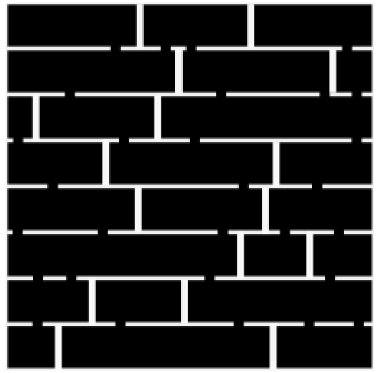
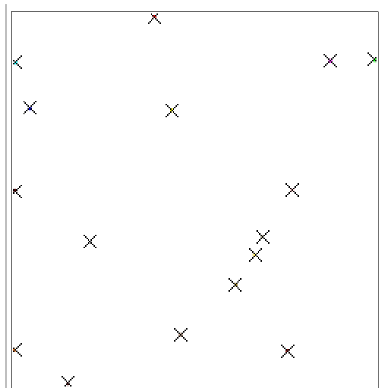
The environment maps below were used for the first experiment. Each map differs to the control map in only one factor. This allowed us to investigate the impact of each individual user difference for different environment types.

Environment Type	Terrain map	Landmark map
Control environment: 16 unique landmarks, 128*128 pixel size, two obstructions per row, that means there are two rooms per row where each row is 8 pixels high.		
Large environment: 16 unique landmarks, 256*256 pixel size, two obstructions per row.		
Complex environment: 16 unique landmarks, 128*128 pixel size, four obstructions per row.		

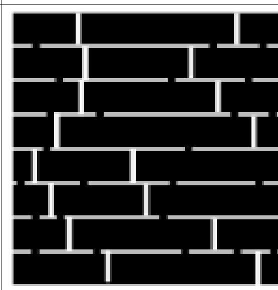
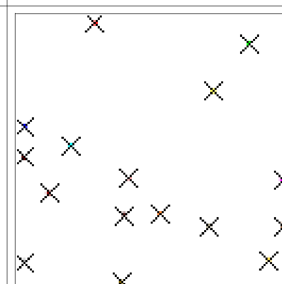
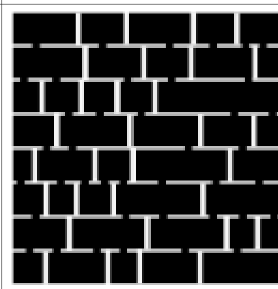
Low landmark environment: 4 unique landmarks, 128*128 pixel size, two obstructions per row.		

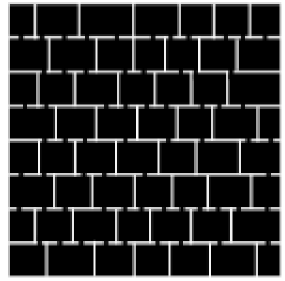
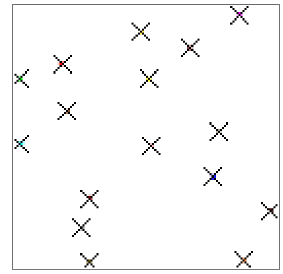
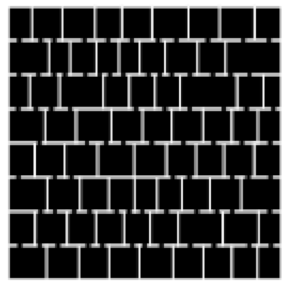
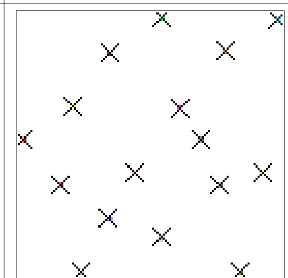
The maps below were used for the investigation of size in the second experiments. The map size started at 64*64 pixels and reached 256*256. This transition allowed us to investigate the importance of size on the process of SKA.

Environment type	Properties	Terrain Map	Landmark Map
Small Size	16 unique landmarks, 64*64 pixel size, two obstructions per row.		
Normal Size (used as control group)	16 unique landmarks, 128*128 pixel size, two obstructions per row, that means there are two rooms per row where each row is per 8 pixels.		
Medium Size	16 unique landmarks, 192*192 pixel size, two obstructions per row.		

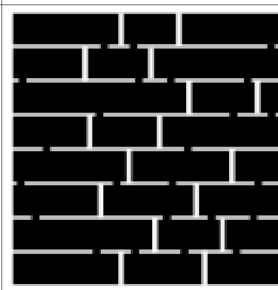
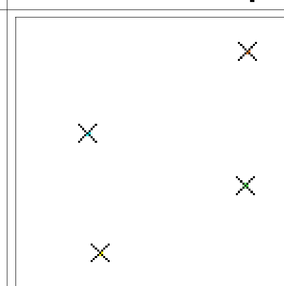
Large Size	16 unique landmarks, 256*256 pixel size, two obstructions per row.		

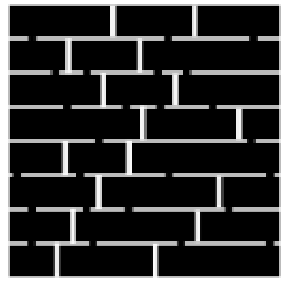
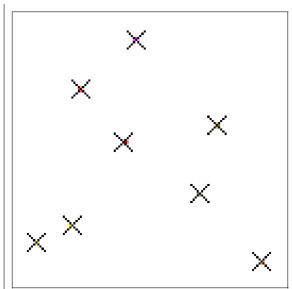
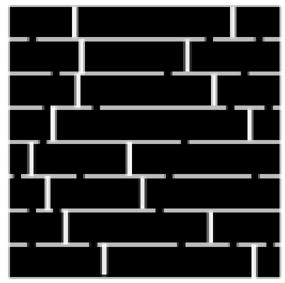
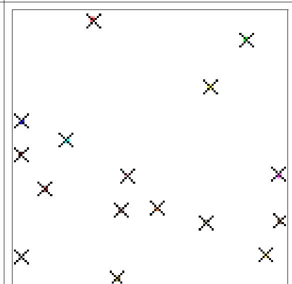
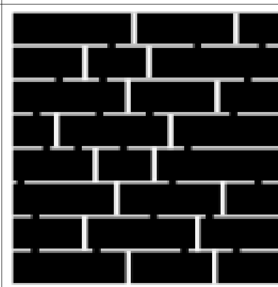
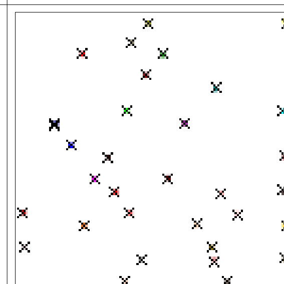
The maps below were used for the investigation of complexity in the second experiments. We started with a map of two obstructions per row, and reached eight obstructions per row. This transition allowed us to investigate the importance of complexity on SKA.

Environment type	Properties	Terrain Map	Landmark Map
Very low complexity (used as control group)	16 unique landmarks, 128*128 pixel size, two obstructions per row, that means there are two rooms per row where each row is per 8 pixels.		
Low Complexity	16 unique landmarks, 128*128 pixel size, four obstructions per row.		

Medium Complexity	16 unique landmarks, 128*128 pixel size, six obstructions per row.		
High Complexity	16 unique landmarks, 128*128 pixel size, eight obstructions per row.		

The maps below were used for the investigation of landmarks in the second experiments. The maps started with four obstructions per row, and grew to 32 obstructions per row. This transition allowed us to investigate the importance of landmark value on SKA.

Environment type	Properties	Terrain Map	Landmark Map
Low frequency of unique landmarks	4 unique landmarks, 128*128 pixel size, two obstructions per row.		

Medium frequency of unique landmarks	8 unique landmarks, 128*128 pixel size, two obstructions per row.		
Normal frequency (used as control group)	16 unique landmarks, 128*128 pixel size, two obstructions per row, that means there are two rooms per row where each row is per 8 pixels.		
High frequency of unique landmarks	32 unique landmarks, 128*128 pixel size, two obstructions per row.		

A.2 SKAR Description

The SKAR engine attempts to simplify the process behind the development of 3D virtual environments primarily made to accomplish experimental studies for spatial knowledge acquisition research in 3D virtual environment training. The development of such an environment had to follow a set of criteria obtained through research.

- The environment had to allow for its overall size to change (In order to evaluate the impact of size)
- The environment had to support landscape formation. This was done through

quad-vector geometry algorithms that read information from a contour map for environment development. This allows flexible environment development ranging from canyons to mountainous lake regions.

- The environment had to support a variety of objects to be imported that could be used as possible landmarks during navigation (In order to evaluate the impact of landmark potential). The landmarks had to be placed according to the height of the landscape.
- The environment had to support a fast and effective way of changing the landscape and landmarks without altering the code. This was achieved through the use of colour coding on 2D images.

The engine does NOT, at the current version support:

- Animations without code modification
- Manipulation of lights without code modification
- Collision with 3D sprite objects (as these are used mainly for populating)
- Adding or deleting objects from the list without code modification

The engine was built using the the Java3D API, that sits on top of OpenGL

A.3 Required and recommended tools

A.3.1 - Running and compiling

In order to run and compile the SKAR engine, the Java Virtual Machine needs to be installed. The latest version of Java3D is also required.

A.3.2 - Creating new Landscape and Landmark images

The 3D objects and images required by SKAR need to be placed in the images directory. Before manipulating the contour and landmark2 .png files, it is recommended that old files are backed up (the custom maps).

In order to develop the landscape you need software that can read and write in the .png format, and also allows for multiple layers (highly recommended, even though it is not required). My personal favourite is GIMP, which is also freeware, but Photoshop should do fine as well. You may also download terrain maps from the internet, and import them into SKAR. SKAR will do it's best to re-create the environment. A final note, the larger the contour map, the more memory is required for the landscape generation, and of course the longer it will take to create the landscape. We generally don't recommend sizes greater than 256*256 pixels. The same applies for landmarks, the more the landmarks, the more memory, the more burdensome the environment becomes on the processor, etc…

A.3.3 - Creating 3D sprites

Although new objects must be inserted programmatically, it is understandable that the pre-installed objects may not be enough or even compatible with certain experiments. In the event that new objects are inserted, the following rules must apply:

- All images must be in the .png format
- It is recommended that images have their height and width smaller than 200 pixels

Again the images may be created using GIMP/Photoshop etc…
A tool will be developed shortly, in order to allow importing of objects.

A.3.4 - Creating .3DS files

If there is a need to insert more 3D mesh objects programmatically, the models may be developed using 3DS MAX 7 and then exported as .3DS objects.

A.4 Running with the custom landmarks/ landscape/ settings

A.4.1- Running the environment

The engine comes with pre-installed landmark and landscape maps. There exists a compiled .exe version, which works in windows. Otherwise you can use java .-jar Skar3D.jar in a terminal.

A.4.2 - Changing the floor and landscape texture

You may change the default floor texture by changing the content in the file "grass2.png" found in the "res" directory, the default is set to a blue colour to represent water. You may also change the size of the image but keep in mind that too large a size will require more memory, while too small a size will decrease the quality. To change the texture for the landscape edit the "grass22.png" in the same way, height map support is now available, and as you may have noticed when running the engine, the landscape texture will overlap accordingly. As the the size of the heightmap increases, so does the detail of the environment, and, as usual, so does the memory and processor requirements. Here is a picture of the environment when using a height map.

A.5 Changing the background image

You may change the default background image by altering the content in the file “narrow.png”. Altering the size and layout is not recommended however.

A.6 Manipulating the Landscape and Landmarks

Changing the layout and the landmarks of an environment is simply a question of adding coloured pixels in two images.

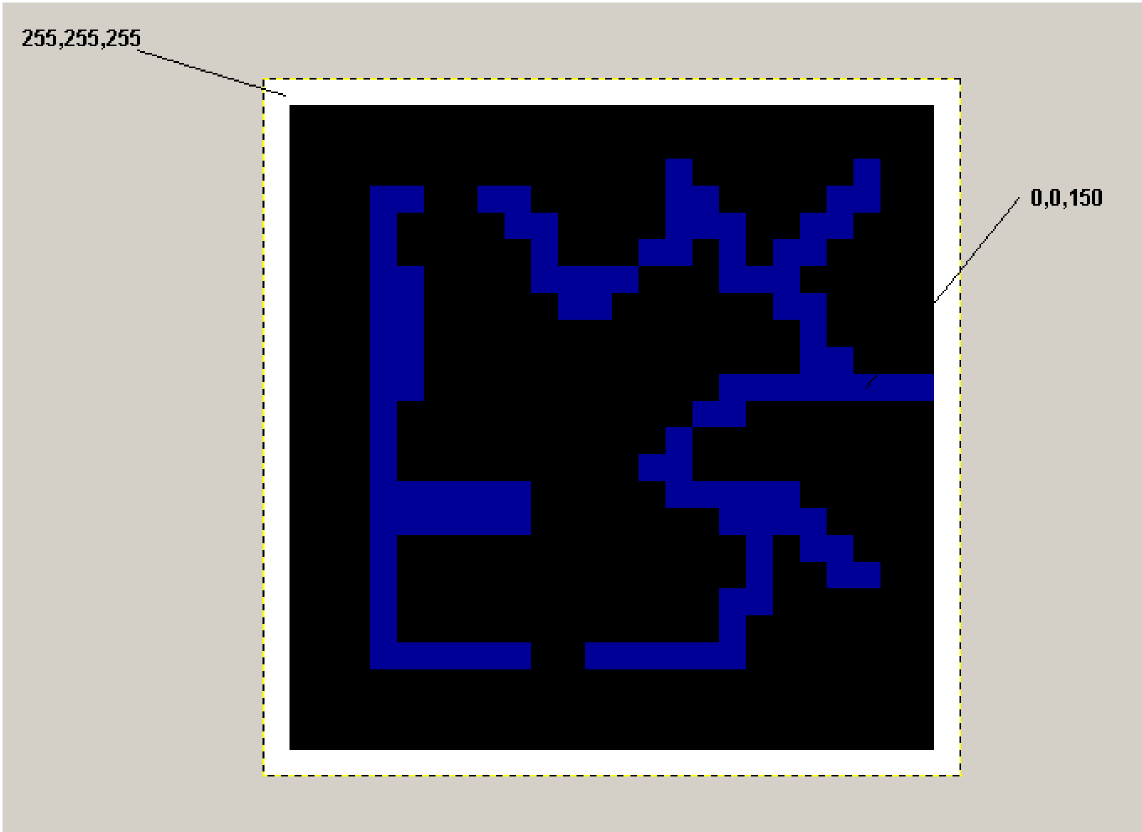
To best explain the process I will present a small tutorial on how to build a very simple map with a few trees and a house. First however it is necessary to explain how landscape quads are either increased or decreased in height. Each pixel is connected to a vertex in the environment.

In order to set the y-value of a vertex to be above the default, you must edit the pixel of the contour map and increase it’s value of the blue component. The other two components do not make a difference at all. So a pixel with value 0,0,255 is the maximum height you can have, and 0,0,0 or 255,255,0 or even 1,55, 0 would all be the lowest possible height, which is below sea level (under the floor).

A.7 Simple World Creation Tutorial

Use an image manipulation program to open “contour.png”. Select a black background and insert a size of 26*26. Keep in mind that the layout must always be 1:1.

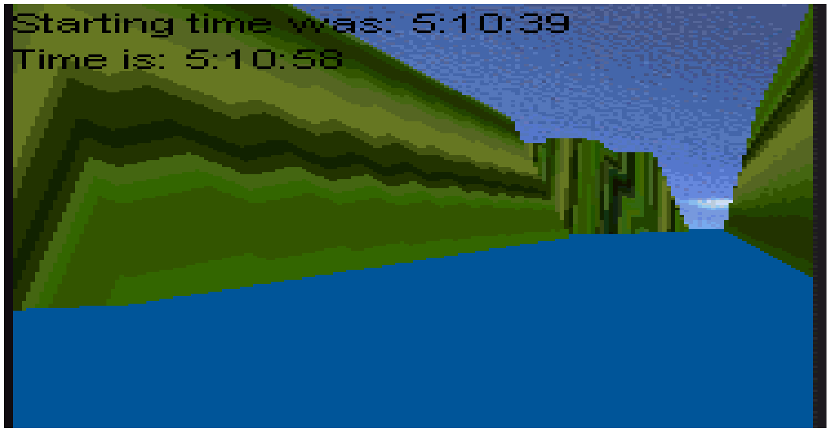
Now insert



Before saving make sure you **FLIP THE IMAGE VERTICALY.** This is necessary due to some geometric rotations in the program. Also absolutely make sure that the landscape is always an even number of pixels in width and height (2,4,6,8… 60).

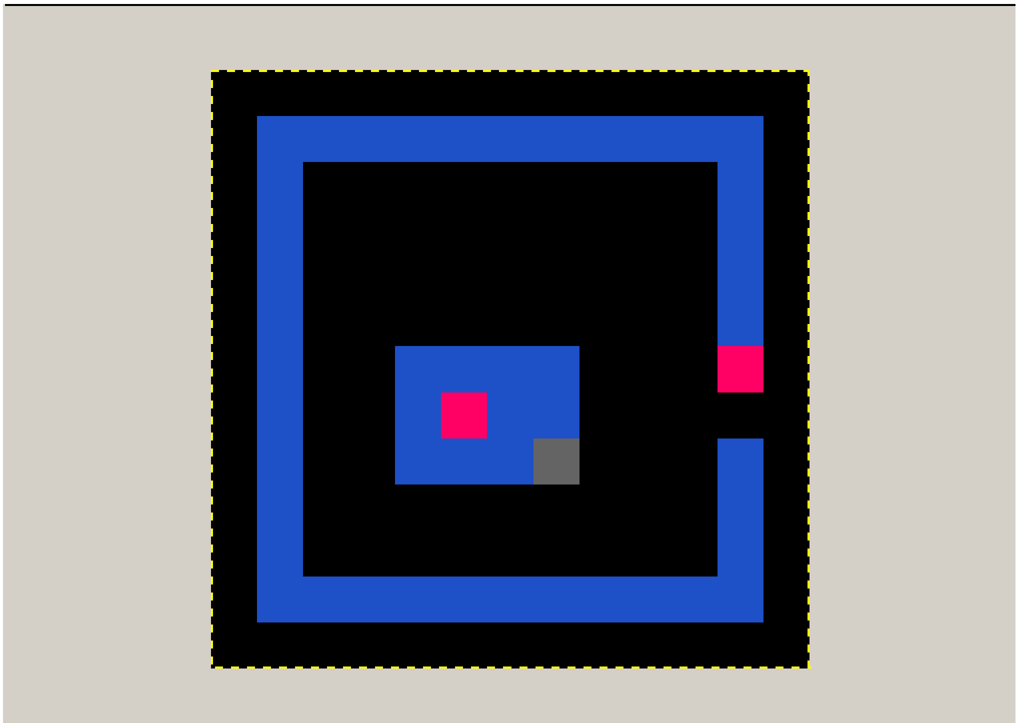
For now create a file in the “images” directory named landmarks2.png, the size must be 13*13 pixels, and it may either be black or transparent throughout.

Compile the application and you should see the environment you just made. Below is a screenshot of the empty simple environment.



At the moment the environment is empty, we will now add a few trees and two mesh objects. In order to add the landmarks you may either directly edit the landmarks2.png file, or add a new layer on the contour.png file and add the objects on top. You may then copy and paste this layer only in the landmarks2.png image, as to avoid objects “sinking” into the hills, you must be careful NOT to save over the existing contour image. Either way make sure you use the pixel RGB values 30,80,200 for the weeds, 255,0,100 for the house and 100,100,100 for the missile.

Your landmarks2.png should look like this:



We save the landmarks2.png (but not the contour.png) file, and compile. We should get an environment populated with bushes, a house, and a missile.

A.7.1 - Changing the collision mode

The SKAR Mobile Engine supports three types of collision; collision without slope ascension, collision with slope ascension, and pure slope ascension. The first disables the ability to ascend up the slopes, the second allows slope ascension but only if the slope is not too steep (as you would see in various games), and the third is absolute slope ascension. The SELECT button also plays an interesting role in all this, as for the second two it serves as a "jump" button, but for the first it allows the user to fly up and see the whole environment, to fly down the user need to press the num_key 1. This of course can be disabled and is explained below.

Use your favourite image editor to open the png file named 'floorcollide.png'. You will notice that the image is 1*3 pixels long. Each pixel corresponds to a switch for collision. Here is how it works.

Switch 0 – Controls type 1 collision (collide or do not collide with steep slopes).
Switch 1 – Controls type 2 collision (allow climbing of slopes)
Switch 2 – Controls jump/fly capability (disable or enable fly/jump).

To manipulate the switches, you must simply add 255,255,255 (pure white) to the pixel colour value. At the moment the default settings are: allow slope collision, allow slope climbing, disable jump/fly. This is the best setting for exploring an environment realistically, as it will allow the camera to climb a slope, but not a very steep one.

I suggest that you experiment with the switch combinations before moving into the next section which will demonstrate how a very complex terrain can be imported into the SKAR engine.

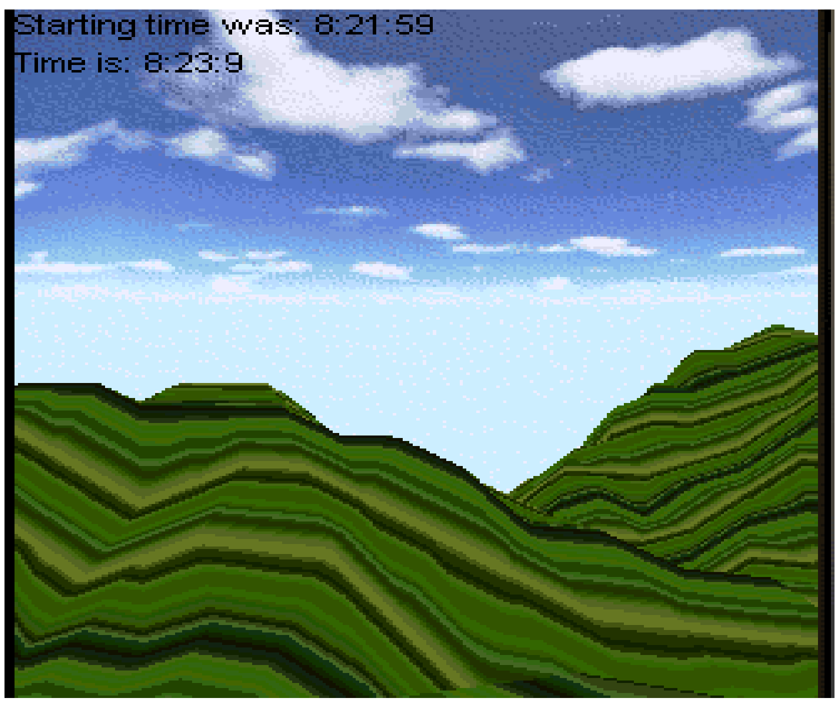
A.7.2 - Importing a terrain heightmap

Adding an actual landmass photographic contour map is easy when using SKAR.

As long as the contour map is in black and white, or differentiates heights through the blue component, then there shouldn't be a problem, you simply rename the map to contour.png, edit it and flip it vertically. You then create a landmarks2.png. What we will do is edit the switches so that the user can climb about the environment so he/she can take a look at the whole map. Edit the 'floorcollide.png' and set the switches in the following order:

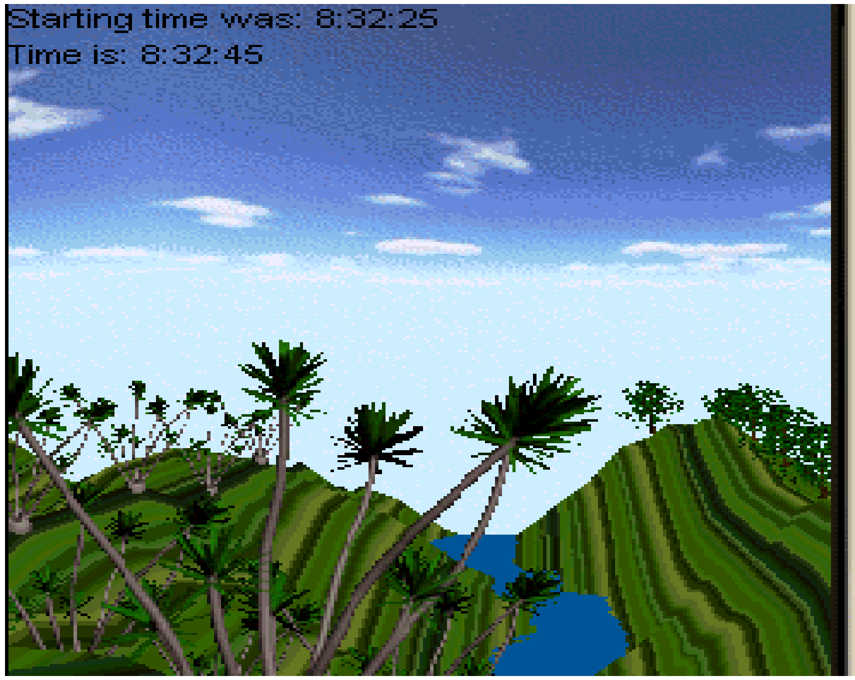
Switch 0: non-white
Switch 1 – non-white
Switch 2 – non-white

Here is a screenshot of the environment without landmarks:



Now we can populate the environment by adding trees, this can be do by editing landmarks2.png. A nice trick is to use the contour map, and place the landmarks in the appropriate positions. This should be done BEFORE flipping the map, if you have already flipped the map simply flip it back before adding the landmarks.
Since SKAR takes care of height placement we don't need to worry about the hills

“swallowing” the trees. Here is a snapshot of the populated environment.



A.8 Source Code for SKAR Engine

This section presents the source code of the SKAR engine, written in Java. We aim to strengthen this code to the point of making the software a useful standardised tool for VE research.

```
// Floor2D.java

package skar;

import java.awt.*;
import java.awt.image.*;
import java.awt.event.*;
import java.applet.*;
import com.sun.j3d.*;
import com.sun.j3d.internal.*;
import com.sun.j3d.loaders.*;
import com.sun.j3d.utils.geometry.*;
import com.sun.j3d.utils.universe.*;
import com.sun.j3d.utils.universe.*;
import javax.media.j3d.*;
import com.sun.j3d.loaders.objectfile.ObjectFile;
import com.sun.j3d.loaders.ParsingErrorException;
import
com.sun.j3d.loaders.IncorrectFormatException;
import com.sun.j3d.loaders.Scene;
import java.applet.Applet;
import java.awt.BorderLayout;
import java.awt.Frame;
import java.awt.event.*;
import com.sun.j3d.utils.applet.MainFrame;
import com.sun.j3d.utils.universe.*;
import com.sun.j3d.utils.geometry.ColorCube;
import javax.media.j3d.*;
import javax.vecmath.*;
import com.sun.j3d.utils.image.TextureLoader;
import com.sun.j3d.utils.behaviors.mouse.*;
import com.sun.j3d.utils.behaviors.keyboard.*;
import java.lang.Thread;
import com.sun.j3d.utils.picking.*;
import javax.imageio.*;
import java.io.*;
import ncsa.j3d.loaders.*;
import ncsa.j3d.loaders.load3ds.*;
import com.sun.j3d.loaders.objectfile.*;
import org.j3d.ui.navigation.CollisionListener;
import com.sun.j3d.utils.image.*;

// This class extends Shape3D and basically creates
a plane made of many squares to represent a
landscape.
// This is a lot like the LandGenerator of SKAR-
Mobile, however squares were used rather than
triangles.
// This is then converted to triangles.

public class Floor2D extends Shape3D{

  public Floor2D(int length,int[][] height) {
// Create the Texture Coordinates, not currently used
as Java3D provides an automated way of doing this
// Leaving it in as reference

float[] tex = new float[0];

for (int j = -length*10/2; j < length*10/2; j+=10) {
for (int i = -length*10/2; i < length*10/2; i+=10) {

float[] texbak = new float[tex.length];
for (int k = 0; k < tex.length-1; k++) {
texbak[k] = tex[k];
}

tex = new float[tex.length+8];

tex[tex.length-8] = 0;
tex[tex.length-7] = 0;

tex[tex.length-6] = 1;
tex[tex.length-5] = 0;

tex[tex.length-4] = 1;
tex[tex.length-3] = 1;

tex[tex.length-2] = 0;
tex[tex.length-1] = 1;
for (int h = 0; h < texbak.length-1; h++) {
tex[h] = texbak[h];
}
}
}
///// Create the Vertices of the Landscape. This is
basicaly a plane made up of an x amount
// of squares were x is defined by the image width.
float[] pts = new float[0];
    for (int j = -length*10/2; j < length*10/2; j+=10) {
for (int i = -length*10/2; i < length*10/2; i+=10) {

  float[] ptsbak = new float[pts.length];
  for (int k = 0; k < pts.length-1; k++) {
    ptsbak[k] = pts[k];
  }
pts = new float[pts.length+12];
pts[pts.length-12] = i;
pts[pts.length-11] = j;
pts[pts.length-10] = 0;
pts[pts.length-9] = i+10;
pts[pts.length-8] = j;
pts[pts.length-7] = 0;
pts[pts.length-6] = i+10;
pts[pts.length-5] = j+10;
pts[pts.length-4] = 0;

pts[pts.length-3] = i;
pts[pts.length-2] = j+10;
pts[pts.length-1] = 0;


for (int h = 0; h < ptsbak.length-1; h++) {
  pts[h] = ptsbak[h];
}



}
    }


// The strip count is required to explain the length of
each side
int[] stripCounts = new int[(pts.length/3)/4];
```

```
for (int l = 0; l < (pts.length/3)/4; l++) {
  stripCounts[l] = 4;
}

// Contour count can be used to define "holes" in the
environment. These could be used as caves
// At the moment all squares are drawn
int[] contourCount = new int[pts.length/4];
for (int m = 0; m < pts.length/4; m++) {
  contourCount[m] = 1;
}


// ADD THE PEAKS
// Take info from the rgb file. Set the y-value of the
coordinates accordingly by taking the blue value
// of each pixel. Therefore pixel 1,1 with blue value
100 will cause coordinate 1,1 to be raised 100/5
for (int j = 0; j < length; j++) {

for (int i = 0; i < length; i++) {

// Check to see whether there is a vector of that
offset
  int pixeldata = height[i][j];
  int blue = pixeldata & 0xff;
int readit = height[31][31] & 0xff;

  for (int n = 0; n < pts.length; n+=3) {
    if (pts[n] == i*10-length*10/2 && pts[n+1] == j*10-
length*10/2) {
      pts[n+2] = blue/5;
      }
  }

}
}

// Create a new GeometryInfo Object
GeometryInfo gInf = new
GeometryInfo(GeometryInfo.POLYGON_ARRAY);


// Set the coordinates for this object as previously
defined
gInf.setCoordinates(pts);

// Set the strip counts
gInf.setStripCounts(stripCounts);
// Convert to triangles. This enables faster rendering
gInf.convertToIndexedTriangles();

// NormalGenerator, this automaticaly creates
normals so shadows, roughness etc... can be added
NormalGenerator ng = new NormalGenerator();
ng.setCreaseAngle((float)Math.toRadians(44));
ng.generateNormals(gInf);

// Stripifier used to increase performance
Stripifier st = new Stripifier();
st.stripify(gInf);

// Finally set the geometry from the array of items
this.setGeometry(gInf.getGeometryArray());

  }
}


// MouseMove.java

package skar;

import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.media.j3d.*;
import javax.vecmath.*;
import com.sun.j3d.utils.behaviors.mouse.*;
import java.awt.*;


// This class extends MouseBehavior and was
created in order to overide a few issues:
// Uses a robot method to check the position of the
mouse and rotate accordingly
// without requiring the user to keep a mouse button
pressed. It also ensures that the
// mouse pointer is invisible and never leaves the
bounds of the screen.

public class MouseMove extends MouseBehavior {
    double x_angle, y_angle;
    double x_factor = .03;
    double y_factor = .03;
    private MouseBehaviorCallback callback = null;
private Robot r;

    public MouseMove(TransformGroup
transformGroup) {
        super(transformGroup);
    }

    /**
     * Creates a default mouse rotate behavior.
     **/
    public MouseMove() {
        super(0);
    }


    public MouseMove(int flags) {
        super(flags);
    }


    public MouseMove(Component c) {
        super(c, 0);
    }


    public MouseMove(Component c, TransformGroup
transformGroup) {
        super(c, transformGroup);
```

```
    }


    public MouseMove(Component c, int flags) {
        super(c, flags);
    }

    public void initialize() {
        super.initialize();
// Create the robot to control position
try {
r = new Robot();
}
catch (Exception e) {
  System.out.print("cannot init robot");
}
        x_angle = 0;
        y_angle = 0;
        if ((flags & INVERT_INPUT) ==
INVERT_INPUT) {
            invert = true;
            x_factor *= -1;
            y_factor *= -1;

        }

    }


    public double getXFactor() {
        return x_factor;
    }


    public double getYFactor() {
        return y_factor;
    }



    public void setFactor( double factor) {
        x_factor = y_factor = factor;
    }


    public void setFactor( double xFactor, double
yFactor) {
        x_factor = xFactor;
        y_factor = yFactor;
    }

    public void processStimulus (Enumeration criteria)
{
        WakeupCriterion wakeup;
        AWTEvent[] events;
         MouseEvent evt;


        while (criteria.hasMoreElements()) {
            wakeup = (WakeupCriterion)
criteria.nextElement();
            if (wakeup instanceof WakeupOnAWTEvent)
{
                events =
((WakeupOnAWTEvent)wakeup).getAWTEvent();
                if (events.length > 0) {
                    evt = (MouseEvent)
events[events.length-1];
                    doProcess(evt);
                }
            }

            else if (wakeup instanceof
WakeupOnBehaviorPost) {
                while (true) {
                    // access to the queue must be
synchronized
                    synchronized (mouseq) {
                        if (mouseq.isEmpty()) break;
                        evt =
(MouseEvent)mouseq.remove(0);
                        // consolidate MOUSE_DRAG events
                        while ((evt.getID() ==
MouseEvent.MOUSE_DRAGGED) &&
                               !mouseq.isEmpty() &&
                               (((MouseEvent)mouseq.get(0)).g
etID() ==
                                MouseEvent.MOUSE_DRAGGE
D)) {
                            evt =
(MouseEvent)mouseq.remove(0);
                        }
                    }
                    doProcess(evt);
                }
            }

        }
if (SKAR3D.buttonMove == true)
  wakeupOn (mouseCriterion);
if (SKAR3D.buttonMove == false)
  wakeupOn (new
WakeupOnAWTEvent(MouseEvent.MOUSE_MOVE
D));

// Keep the mouse constantly pressed. This tricks
Java3D into thinking the player is keeping the mouse
pressed
r.mousePress(InputEvent.BUTTON1_MASK);

    }

    // If the mouse is dragged then rotate, if it leaves
the bounds then reset the position
    void doProcess(MouseEvent evt) {
        int id;
        int dx, dy;

        processMouseEvent(evt);
            id = evt.getID();
            if ((id == MouseEvent.MOUSE_MOVED || id
== MouseEvent.MOUSE_DRAGGED)) {
                x = evt.getX();
                y = evt.getY();
```

```
            dx = x - x_last;
            dy = y - y_last;

            if (!reset){
                x_angle = dy * y_factor;
                y_angle = dx * x_factor;

                transformX.rotX(x_angle);
                transformY.rotY(y_angle);

                transformGroup.getTransform(currXform
);

                Matrix4d mat = new Matrix4d();
                // Remember old matrix
                currXform.get(mat);

                // Translate to origin
                currXform.setTranslation(new
Vector3d(0.0,0.0,0.0));
                if (invert) {
                    currXform.mul(currXform,
transformX);
                    currXform.mul(currXform,
transformY);
                } else {
                    currXform.mul(transformX,
currXform);
                    currXform.mul(transformY, currXform);
                }

                // Set old translation back
                Vector3d translation = new
                    Vector3d(mat.m03, mat.m13,
mat.m23);
                currXform.setTranslation(translation);

                // Update xform
                transformGroup.setTransform(currXform
);

                transformChanged( currXform );

                if (callback!=null)
                    callback.transformChanged( MouseB
ehaviorCallback.ROTATE,
                                    currXform );


            }
            else {
                reset = false;
            }


            x_last = x;
            y_last = y;

        }
        else if (id ==
MouseEvent.MOUSE_PRESSED) {
            x_last = evt.getX();
            y_last = evt.getY();
//      }
    }

            if (x > SKAR3D.screenwidth || x < 0) {
              x_last = SKAR3D.screenwidth/2;
              r.mouseMove(SKAR3D.screenwidth/
2,SKAR3D.screenheight);
            }

    }


    public void transformChanged( Transform3D
transform ) {
    }



    public void
setupCallback( MouseBehaviorCallback callback ) {
        this.callback = callback;
    }
}
```

```
package skar;

import java.awt.*;
import java.awt.image.*;
import java.awt.event.*;
import java.applet.*;
import com.sun.j3d.*;
import com.sun.j3d.internal.*;
import com.sun.j3d.loaders.*;
import com.sun.j3d.utils.geometry.*;
import com.sun.j3d.utils.universe.*;
import com.sun.j3d.utils.universe.*;
import javax.media.j3d.*;
import com.sun.j3d.loaders.objectfile.ObjectFile;
import com.sun.j3d.loaders.ParsingErrorException;
import
com.sun.j3d.loaders.IncorrectFormatException;
import com.sun.j3d.loaders.Scene;

import java.applet.Applet;
import java.awt.BorderLayout;
import java.awt.Frame;
import java.awt.event.*;
import com.sun.j3d.utils.applet.MainFrame;
import com.sun.j3d.utils.universe.*;
import com.sun.j3d.utils.geometry.ColorCube;
import javax.media.j3d.*;
import javax.vecmath.*;
import com.sun.j3d.utils.image.TextureLoader;
import com.sun.j3d.utils.behaviors.mouse.*;
import com.sun.j3d.utils.behaviors.keyboard.*;
import java.lang.Thread;
import com.sun.j3d.utils.picking.*;
import javax.imageio.*;
import java.io.*;
import ncsa.j3d.loaders.*;
```

```
import ncsa.j3d.loaders.load3ds.*;
import com.sun.j3d.loaders.objectfile.*;
import org.j3d.ui.navigation.CollisionListener;
import com.sun.j3d.utils.image.*;


/**
 * <p>Title: </p>
 * <p>Description: </p>
 * <p>Copyright: Copyright (c) 2005</p>
 * <p>Company: </p>
 * @author unascribed
 * @version 1.0
 */

public class SKAR3D extends Applet {

public  BoundingSphere bounds;
public static boolean buttonMove = true;
  private java.net.URL bgImage = null;
  private java.net.URL texImage = null;
  private java.net.URL texImage2 = null;

  private java.net.URL hillImage = null;
  private java.net.URL Tree1Image = null;
  private java.net.URL Tree2Image = null;
  private java.net.URL Tree3Image = null;
  private java.net.URL Tree4Image = null;
  private java.net.URL Tree5Image = null;
  private java.net.URL Tree6Image = null;
  private java.net.URL Tree7Image = null;
  private java.net.URL Tree8Image = null;
  private java.net.URL Tree9Image = null;
  private java.net.URL Tree10Image = null;
  private java.net.URL Tree11Image = null;
  private java.net.URL BushImage = null;
  private java.net.URL CartImage = null;
  private java.net.URL HouseTexImage = null;
  private java.net.URL HouseTex2Image = null;
  private java.net.URL HouseTex3Image = null;
  private java.net.URL BarnImage = null;
  private java.net.URL MarbleImage = null;
  private java.net.URL Metal2Image = null;
  private java.net.URL WierdImage = null;
  private java.net.URL Wierd2Image = null;


private Thread t;
private long mFrameDelay;
private Box textureCube;
private  SimpleUniverse simpleU;
private boolean Running = true;
private boolean started = false;
private ViewingPlatform ourView;
private Viewer camera;
private ViewerAvatar va;
private ColorCube cb;
private PickTool picker;
public BranchGroup scene;
private BranchGroup objRoot;
public BranchGroup objHills;
public BranchGroup objModels;
public BranchGroup objSprites;


private double radians;
private boolean negx  = false;
private boolean posx  = false;
private boolean negz  = false;
private boolean posz  = false;
private boolean stopmove = false;
private boolean accelerate = false;
private boolean inverted = false;
public   static TransformGroup viewTrans;
private BufferedImage Landscape;
private BufferedImage Landmarks;
private BufferedImage FloorCollide;

private int Landwidth;
private int Landheight;
private boolean noTriangulate = false;
private boolean noStripify = false;
private double creaseAngle = 60.0;
private Scene loadedScene = null;
private BranchGroup loadedBG = null;
private WakeupOnCollisionEntry wEnter;
private int[][] collx;
private boolean collided = false;
public final int BOX = 0;
public final int CONE = 1;
public final int SPHERE = 2;
public static int screenwidth;
public static int screenheight;
public boolean pickedsomething = false;
private TimerInterface canvas3D;
private int[][] contourheight;
private Point3d nextpt = new Point3d();
private Point3d currentheight = new Point3d();
public static boolean floorcollide = false;
public static boolean populated = false;
public static boolean begin = true;
public static boolean transfloor = false;
public int[] settings;
private int chosenscreen = 0;

private static Frame frame;


// SKAR3D is the main class. This class controls
things such as canvas and universe setup, object
adding and
// Manipulating, keyboard and mouse controls, etc...



  public SKAR3D() {

// Nothing to do here, so keep it blank

  }



// CREATE THE SCENEGRAPH
```

```
  public BranchGroup createSceneGraph() {
// Create the root of the branch graph
objRoot = new BranchGroup();
// Make it pickable (so rays can detect it)
objRoot.setPickable(true);
objRoot.setCapability(objRoot.ENABLE_PICK_REP
ORTING);
objRoot.setCapability(BranchGroup.ENABLE_COLLI
SION_REPORTING);
// objHills is another branchgroup which is used
purely for the landscape and water
objHills = new BranchGroup();


// Create the background and add the texture
(bakimage)
Background bg = new Background();
bg.setApplicationBounds(bounds);
BranchGroup backGeoBranch = new BranchGroup();
Sphere sphereObj = new Sphere(1.0f,
Sphere.GENERATE_NORMALS |
                    Sphere.GENERATE_NORMALS_IN
WARD |
                    Sphere.GENERATE_TEXTURE_CO
ORDS, 45);
Appearance backgroundApp =
sphereObj.getAppearance();
backGeoBranch.addChild(sphereObj);
bg.setGeometry(backGeoBranch);
objRoot.addChild(bg);

TextureLoader tex = new TextureLoader(bgImage,
                              new String("RGB"), this);
if (tex != null)
   backgroundApp.setTexture(tex.getTexture());



/// Create the Appearance for the water floor
//Create a new appeareance object
Appearance appo = new Appearance();
// Create a loader and set mipmaping as y_up (good
for Directx sdk)
TextureLoader loadero = new
TextureLoader(texImage2,TextureLoader.Y_UP,null);
// Create an image component
ImageComponent2D landImageo =
loadero.getImage();
// Create the texture
Texture2D tex2o = new
Texture2D(Texture.BASE_LEVEL, Texture.RGBA,
landImageo.getWidth(),landImageo.getHeight());
// Set the image to the texture
tex2o.setImage(0,landImageo);
// Enable the texture
tex2o.setEnable(true);
// Set a linear filter(avoids distortion, but required
more power)
tex2o.setMinFilter(Texture2D.MULTI_LEVEL_LINEA
R);
// Convert to a polygon so we can change culling. By
setting CULL_NONE we make the landscape
double-sided
PolygonAttributes pao = new PolygonAttributes();
pao.setCullFace(PolygonAttributes.CULL_NONE);
pao.setPolygonMode(pao.POLYGON_FILL);

// Set the polygon attributes
appo.setPolygonAttributes(pao);
// Set the texture
appo.setTexture(tex2o);
TextureAttributes texAttro = new TextureAttributes();
texAttro.setTextureMode(TextureAttributes.MODULA
TE);
appo.setTextureAttributes(texAttro);
// Create plane coordinates
Vector4f planeSo = new Vector4f(1f,0,0,0);
Vector4f planeTo = new Vector4f(0,1f,0,0);

// Create transparency (water effect)
TransparencyAttributes ta = new
TransparencyAttributes();
ta.setTransparencyMode(TransparencyAttributes.BL
ENDED);
// Set water transparency
ta.setTransparency(0.6f);
if (transfloor == true) {
// Add the transparency if requested by the user
appo.setTransparencyAttributes(ta);
}
// Add the texcoordinates
appo.setTexCoordGeneration(new
TexCoordGeneration(TexCoordGeneration.TEXTUR
E_COORDINATE_2,TexCoordGeneration.OBJECT_
LINEAR,planeSo,planeTo));

// Create THE water floor

Water water = new Water(Landwidth*2);
water.setAppearance(appo);
water.setPickable(true);
water.setCapability(water.ENABLE_PICK_REPORTI
NG);

// Rescale so it fits to the environment
Transform3D scaleo = new Transform3D();
scaleo.setScale(0.5);
TransformGroup tg33o = new
TransformGroup(scaleo);

// Lower it to y-value -1 (just above lowest possible
floor value)
Transform3D moveo = new Transform3D();
Vector3f v3o = new Vector3f(0,-1f,0);
moveo.set(v3o);
TransformGroup tgo = new TransformGroup(moveo);

// Rotate so it is placed under the camera's feet
Transform3D rotateo = new Transform3D();
rotateo.rotX(-Math.PI/2.0);
TransformGroup tg22o = new
TransformGroup(rotateo);
```

```
//tg.addChild(textureCube);

// Add leaf-node relationships and finaly add it to the
branchgroup objHills
tg33o.addChild(water);
tg33o.setCapability(tgo.ENABLE_PICK_REPORTIN
G);
tg33o.setPickable(true);

tg22o.addChild(tg33o);
tg22o.setCapability(tgo.ENABLE_PICK_REPORTIN
G);
tg22o.setPickable(true);
tgo.addChild(tg22o);
tgo.setCapability(tgo.ENABLE_PICK_REPORTING);
tgo.setPickable(true);
PickTool.setCapabilities(water,PickTool.INTERSECT
_COORD);
objHills.addChild(tgo);


/// CREATE THE TEXTURE FOR THE Landscape
//Create a new appeareance object
Appearance app = new Appearance();
// Add a texture to the landscape
Texture tex2 = new TextureLoader(texImage,
this).getTexture();

// Convert to a polygon and ensure that it is double-
sided
PolygonAttributes pa = new PolygonAttributes();
pa.setCullFace(PolygonAttributes.CULL_BACK);
pa.setPolygonMode(pa.POLYGON_FILL);

// Set the polygon attributes
app.setPolygonAttributes(pa);
// Set the texture
app.setTexture(tex2);
// Create texattribtes to define how object will handle
the textures
TextureAttributes texAttr = new TextureAttributes();
texAttr.setTextureMode(TextureAttributes.MODULAT
E);
app.setTextureAttributes(texAttr);

// Create plane coordinates, configured to stretch
over the plane
Vector4f planeS = new Vector4f(0.003f,0,0,0);
Vector4f planeT = new Vector4f(0,0.003f,0,0);


// Set the texture coordinates
app.setTexCoordGeneration(new
TexCoordGeneration(TexCoordGeneration.TEXTUR
E_COORDINATE_2,TexCoordGeneration.OBJECT_
LINEAR,planeS,planeT));


// Create THE Landscape and add it to the scene
graph.

Floor2D floor = new
Floor2D(Landwidth*2,contourheight);
floor.setAppearance(app);
floor.setPickable(true);
floor.setCapability(floor.ENABLE_PICK_REPORTIN
G);

// Scale so it fits on the landscape
Transform3D scale = new Transform3D();
scale.setScale(0.5);
TransformGroup tg33 = new TransformGroup(scale);

// Translate right under water floor
Transform3D move = new Transform3D();
Vector3f v3 = new Vector3f(0,-2f,0);
move.set(v3);
TransformGroup tg = new TransformGroup(move);

// Rotate so it sits under camera
Transform3D rotate = new Transform3D();
rotate.rotX(-Math.PI/2.0);
TransformGroup tg22 = new TransformGroup(rotate);

// Add all leaf-nodes and finally add to branchgroup
objHills
tg33.addChild(floor);
tg33.setCapability(tg.ENABLE_PICK_REPORTING);
tg33.setPickable(true);

tg22.addChild(tg33);
tg22.setCapability(tg.ENABLE_PICK_REPORTING);
tg22.setPickable(true);
tg.addChild(tg22);
tg.setCapability(tg.ENABLE_PICK_REPORTING);
tg.setPickable(true);
PickTool.setCapabilities(floor,PickTool.INTERSECT_
COORD);

objHills.addChild(tg);




// Add the branchgroups to objRoot
objRoot.addChild(objHills);
objRoot.addChild(objModels);
return objRoot;
} // end of createSceneGraph method of
HelloJava3Da
 //Initialize the applet
// INITIALIZE THE WORLD + OBJECTS
 public void init() {
  try {
   jbInit();
  }
  catch(Exception e) {
   e.printStackTrace();
  }
 }
 //Component initialization, loading sequence starts
here
 private void jbInit() throws Exception {
```

```
// Load floor settings (collideable floor, transparent
water floor)
    try{
    File f3 = new File("../images/floorcollide.png");
    FloorCollide = ImageIO.read(f3);
settings = new int[2];
settings[0] = FloorCollide.getRGB(0,0);
settings[1] = FloorCollide.getRGB(1,0);

if (settings[0] == -1)
  floorcollide = true;
if (settings[1] == -1)
  transfloor = true;

    }
    catch (Exception e ) {
      System.out.println("Cannot load floorcollide
settings");
    }



// Load landmark file
    try{
    File f2 = new File("../images/landmarks2.png");
    Landmarks = ImageIO.read(f2);

    }
    catch (Exception e ) {
      System.out.println("Cannot load Landmarks");
    }

// Load Landscape file
    try{
    File f = new File("../images/contour.png");
    Landscape = ImageIO.read(f);

    }
    catch (Exception e ) {
      System.out.println("Cannot load Landscape");
    }

////////// NOW RETRIEVE THE SIZE OF THE
LANDSCAPE
Landwidth = Landmarks.getWidth();
Landheight = Landmarks.getHeight();

contourheight = new int[Landwidth*2][ Landwidth*2];

for (int j = 0; j < Landwidth*2; j++) {

for (int i = 0; i < Landwidth*2; i++) {
contourheight[i][j] = Landscape.getRGB(i,j);
}
}

////////// SET THE BOUNDS OF THE SURROUNDING
BACKGROUND
    double boundsize = Landwidth;
    bounds = new BoundingSphere(new
Point3d(0.0,0.0,0.0), Landwidth*20);
    /// and add more images (such as background and
Sprites)

    java.net.URL bgurl = null;
    try {
        bgurl = new
java.net.URL("file:../images/bak.jpg");
    }
    catch (java.net.MalformedURLException ex) {
        System.out.println(ex.getMessage());
        System.exit(1);
    }
bgImage = bgurl;


      java.net.URL wierd2url = null;
      try {
          wierd2url = new
java.net.URL("file:../images/wierd2.png");
      }
      catch (java.net.MalformedURLException ex) {
          System.out.println(ex.getMessage());
          System.exit(1);
      }
   Wierd2Image = wierd2url;

      java.net.URL wierdurl = null;
      try {
          wierdurl = new
java.net.URL("file:../images/wierd.png");
      }
      catch (java.net.MalformedURLException ex) {
          System.out.println(ex.getMessage());
          System.exit(1);
      }
   WierdImage = wierdurl;


      java.net.URL Metal2url = null;
      try {
          Metal2url = new java.net.URL("file:../images/
metal2.png");
      }
      catch (java.net.MalformedURLException ex) {
          System.out.println(ex.getMessage());
          System.exit(1);
      }
   Metal2Image = Metal2url;


      java.net.URL Marbleurl = null;
      try {
          Marbleurl = new java.net.URL("file:../images/
marble.png");
      }
      catch (java.net.MalformedURLException ex) {
          System.out.println(ex.getMessage());
          System.exit(1);
      }
   MarbleImage = Marbleurl;


      java.net.URL HouseTex3url = null;
```

```
      try {
         HouseTex3url = new
java.net.URL("file:../images/stone.png");
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   HouseTex3Image = HouseTex3url;


   java.net.URL Barnurl = null;
   try {
      Barnurl = new
java.net.URL("file:../images/wood.png");
   }
   catch (java.net.MalformedURLException ex) {
      System.out.println(ex.getMessage());
      System.exit(1);
   }
BarnImage = Barnurl;



      java.net.URL HouseTex2url = null;
      try {
         HouseTex2url = new
java.net.URL("file:../images/metal.png");
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   HouseTex2Image = HouseTex2url;


      java.net.URL HouseTexurl = null;
      try {
         HouseTexurl = new
java.net.URL("file:../images/wood.png");
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   HouseTexImage = HouseTexurl;


      java.net.URL carturl = null;
      try {
         carturl = new
java.net.URL("file:../images/cart.png");
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   CartImage = carturl;


      java.net.URL bushurl = null;
      try {
         bushurl = new
java.net.URL("file:../images/bush1.png");
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   BushImage = bushurl;


      java.net.URL tree11url = null;
      try {
         tree11url = new
java.net.URL("file:../images/tree11.png");
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   Tree11Image = tree11url;


      java.net.URL tree10url = null;
      try {
         tree10url = new
java.net.URL("file:../images/tree10.png");
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   Tree10Image = tree10url;


      java.net.URL tree9url = null;
      try {
         tree9url = new
java.net.URL("file:../images/tree9.png");
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   Tree9Image = tree9url;


      java.net.URL tree8url = null;
      try {
         tree8url = new
java.net.URL("file:../images/tree8.png");
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   Tree8Image = tree8url;


      java.net.URL tree7url = null;
      try {
         tree7url = new
java.net.URL("file:../images/tree7.png");
```

```
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   Tree7Image = tree7url;

      java.net.URL tree6url = null;
      try {
         tree6url = new
java.net.URL("file:../images/tree6.png");
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   Tree6Image = tree6url;

      java.net.URL tree5url = null;
      try {
         tree5url = new
java.net.URL("file:../images/tree5.png");
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   Tree5Image = tree5url;

      java.net.URL tree4url = null;
      try {
         tree4url = new
java.net.URL("file:../images/tree4.png");
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   Tree4Image = tree4url;

      java.net.URL tree3url = null;
      try {
         tree3url = new
java.net.URL("file:../images/tree3.png");
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   Tree3Image = tree3url;


      java.net.URL tree2url = null;
      try {
         tree2url = new
java.net.URL("file:../images/dentro2.png");
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   Tree2Image = tree2url;

      java.net.URL tree1url = null;
      try {
         tree1url = new
java.net.URL("file:../images/dentro1.png");
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   Tree1Image = tree1url;



      java.net.URL texurl = null;
      try {
         texurl = new
java.net.URL("file:../images/grass2.png");
      }
      catch (java.net.MalformedURLException ex) {
         System.out.println(ex.getMessage());
         System.exit(1);
      }
   texImage = texurl;


   java.net.URL tex2url = null;
   try {
      tex2url = new
java.net.URL("file:../images/grass22.png");
   }
   catch (java.net.MalformedURLException ex) {
      System.out.println(ex.getMessage());
      System.exit(1);
   }
texImage2 = tex2url;



         java.net.URL hillurl = null;
         try {
            hillurl = new
java.net.URL("file:../images/rock.jpg");
         }
         catch (java.net.MalformedURLException ex) {
            System.out.println(ex.getMessage());
            System.exit(1);
         }
      hillImage = hillurl;


   System.out.print("Welcome to SKAR");


// set the layout
setLayout(new BorderLayout());
// create a craphics configuration
GraphicsConfiguration config =
SimpleUniverse.getPreferredConfiguration();
// Create a canvas3d (notice TimerInterface is called
```

```
as I have overided the postrender method in order to add
// 2D graphics on the 3D canvas
canvas3D = new TimerInterface(config);
add("Center", canvas3D);
add(canvas3D);
canvas3D.setFocusable(true);
canvas3D.requestFocus();

// Various standard commands to set the canvas such as getting the best cursor
Toolkit t = Toolkit.getDefaultToolkit();
Dimension d = t.getBestCursorSize(1,1);
Cursor no_cursor = t.createCustomCursor(new BufferedImage(d.width,d.height,BufferedImage.TYPE_INT_ARGB),new Point(0,0),"no_cursor");
canvas3D.setCursor(no_cursor);


screenwidth = getWidth();
screenheight = getHeight();





// Add a keylistener
canvas3D.addKeyListener(new KeyAdapter() {


public void keyReleased(KeyEvent e) {
  int keyCode = e.getKeyCode();


if ((keyCode == KeyEvent.VK_UP)) {
// Reset values
accelerate = false;
collided = false;
}
}

  public void keyPressed(KeyEvent e) {

if (canvas3D.currenttimeshow == true) {
  int keyCode = e.getKeyCode();
  if ((keyCode == KeyEvent.VK_Q)) {
// If escape screen is true then exit
  System.exit(0);
  }
else  if ((keyCode == KeyEvent.VK_B)) {
// if escape screen is true then resume
canvas3D.currenttimeshow = false;
canvas3D.postRender();
  }

}

if (canvas3D.currenttimeshow == false) {

    int keyCode = e.getKeyCode();


    if ((keyCode == KeyEvent.VK_ESCAPE)) {
// Bring up escape screen (end screen)
canvas3D.currenttimeshow = true;
canvas3D.postRender();

    }



else if ((keyCode == KeyEvent.VK_D)) {
// Pressing D will release the mouse so it can leave the boundaries of the screen
if (buttonMove == false)
  buttonMove = true;
else if (buttonMove == true)
  buttonMove = false;
}
else if ((keyCode == KeyEvent.VK_UP)) {
// Moves the user forward
  if (populated == true) {

// Get sin and cos of vector and find the resultant. This causes forward movement
  double convertang = radians*Math.PI/180.0;
  double movexx = Math.sin(convertang);
  double moveyy = Math.cos(convertang);
  Vector3d v3d = new Vector3d(-movexx*0.5,0,-moveyy*0.5);
  Vector3d v3db = new Vector3d(-movexx,0,-moveyy);

inverted = false;
// Check to see if there is a collision with floor or objects
checkray(v3db);
testMove(v3d);
  }
}

else if ((keyCode == KeyEvent.VK_DOWN)) {
  // Moves the user backwards

if (populated == true) {
  // Get sin and cos of vector and find the resultant. This causes back movement
  double convertang = radians*Math.PI/180.0;
  double movexx = Math.sin(convertang);
  double moveyy = Math.cos(convertang);
  Vector3d v3d = new Vector3d(movexx*0.5,0,moveyy*0.5);
  Vector3d v3db = new Vector3d(movexx,0,moveyy);

inverted = false;
// Check to see if there is a collision with floor or objects

checkray(v3db);
testMove(v3d);
}
}


else if ((keyCode == KeyEvent.VK_2)) {
// In start screen set the resolution to mode 1
```

```
if (populated == false) {
  chosenscreen = 1;
  populateWorld();

}
}

else if ((keyCode == KeyEvent.VK_3)) {
  // In start screen set the resolution to mode 2

if (populated == false) {
  chosenscreen = 2;
  populateWorld();

}
}

else if ((keyCode == KeyEvent.VK_4)) {
  // In start screen set the resolution to mode 3

if (populated == false) {
  chosenscreen = 3;
  populateWorld();

}
}


else if ((keyCode == KeyEvent.VK_1)) {

  // In start screen set the resolution to mode 0

if (populated == false && begin == true) {
  chosenscreen = 0;

populateWorld();
}

}

//// FLOAT if floorcollide is enabled
else if ((keyCode == KeyEvent.VK_0)) {
  if (populated == true) {

    // Get sin and cos of vector and find the resultant.
This causes forward up movement
  double convertang = radians*Math.PI/180.0;
  double movexx = Math.sin(convertang);
  double moveyy = Math.cos(convertang);
  Vector3d v3d = new Vector3d(-movexx*0.5,2.5,-
moveyy*0.5);
  Vector3d v3db = new Vector3d(-movexx,2.5,-
moveyy);

inverted = false;
checkray(v3db);
testMove(v3d);
  }

}

else if ((keyCode == KeyEvent.VK_RIGHT)) {
// Rotate right by using the right key
double rad = -0.05;
radians = rad;
  doRotateY();
  if (radians >=360.0f)
    radians -= 360.0f;
  if (radians <=-360.0f)
    radians += 360.0f;

}

else if ((keyCode == KeyEvent.VK_LEFT)) {
  // Rotate left by using the left key

double rad = 0.05;
radians = rad;
  doRotateY();
  if (radians >=360.0f)
    radians -= 360.0f;
  if (radians <=-360.0f)
    radians += 360.0f;

}


  }
}
}
);

// Create a scene by calling the createSceneGraph()
method. This should return objRoot
scene = createSceneGraph();



// SimpleUniverse is a Convenience Utility class
simpleU = new SimpleUniverse(canvas3D);

/// AMBIENT LIGHTS (NO SHADOWS), to add more
realism we would add at least 1 directional light to
simulate
// sunlight. However this would cause various
differences in the mobile and java3d version and so
has been
// left out completely

     PlatformGeometry pg = new
PlatformGeometry();

         // Set up the ambient light
         Color3f ambientColor = new Color3f(0.5f,
0.5f, 0.5f);
         AmbientLight ambientLightNode = new
AmbientLight(ambientColor);
         ambientLightNode.setInfluencingBounds(
bounds);
    pg.addChild(ambientLightNode);

    simpleU.getViewingPlatform().setPlatformGeom
etry( pg );
```

```
 // This moves the ViewPlatform back a bit so the
 // objects in the scene can be viewed.
//
simpleU.getViewingPlatform().setNominalViewingTransform();
ViewingPlatform vp = simpleU.getViewingPlatform();
Transform3D initpos = new Transform3D();

float a = -Landwidth*5+15;
float b = -Landheight*5+15;
Vector3f v3f = new Vector3f(a,2,b);
initpos.setTranslation(v3f);
TransformGroup tf = vp.getViewPlatformTransform();
// Set to pos 15,15
tf.setTransform(initpos);

// TransformGroup viewTrans controls rotations and
movements of the camera.
viewTrans =
simpleU.getViewingPlatform().getViewPlatformTransform();



//COLLISION SET FOR CAMERA

WakeupOnCollisionEntry wEnter  = new
WakeupOnCollisionEntry(vp);


// ADD THE FOG (not implemented yet)
//LinearFog fogLinear = new
LinearFog(-1,15.0f,30.0f);
//fogLinear.setInfluencingBounds(bounds);
//scene.addChild(fogLinear);


// ADD THE mouse behaviour. note how MouseMove
is used instead of MouseBehavior as it
// overides the stimulus and adds a robot so keeping
the mouse button pressed is not
// required and also hides the cursor and keeps it
within the bounds of the screen

MouseMove behavior2 = new MouseMove();

   behavior2.setTransformGroup(viewTrans);
   scene.addChild(behavior2);
   behavior2.setFactor(-0.02,0); // speed of rotation
   behavior2.setSchedulingBounds(bounds);


// Create objSprites and objModels branchgroups to
store landmarks
objSprites = new BranchGroup();
objModels = new BranchGroup();

// COMPILE THE SCENE (Background, FLOOR and
behaviours)
scene.compile();
simpleU.addBranchGraph(scene);

// Set the clip distance to 0.05 - 50
 View view = simpleU.getViewer().getView();
 view.setBackClipDistance (50);
 view.setFrontClipDistance(0.05);



 started = true;

  }



  //Get Applet information
  public String getAppletInfo() {
    return "Applet Information";
  }
  //Get parameter info
  public String[][] getParameterInfo() {
    return null;
  }
  //Main method
  public static void main(String[] args) {
// Set the main frame
 Frame frame = new MainFrame(new SKAR3D(),
320, 200);

  }


private void testMove(Vector3d theMove) {
// Take curent position and test whether camera is
colliding with a landmark
theMove.y = nextpt.z/2-currentheight.z/2;
  Transform3D t3d = new Transform3D();
  viewTrans.getTransform(t3d);
  Transform3D toMove = new Transform3D();
  toMove.setTranslation(theMove);
  t3d.mul(toMove);

checkcollisions(t3d);
if (collided == false && pickedsomething == false) {
  viewTrans.setTransform(t3d);

}

}
  private void doMove(Vector3d theMove) {
// Move the camera
Transform3D t3d = new Transform3D();
viewTrans.getTransform(t3d);
Transform3D toMove = new Transform3D();
toMove.setTranslation(theMove);
t3d.mul(toMove);
viewTrans.setTransform(t3d);
  }


public static Transform3D currentpos() {
// return the current camera position
  Transform3D t3d = new Transform3D();
viewTrans.getTransform(t3d);
return t3d;
```

```
}


private void doRotateY() {
// Rotate the camera (actually the scene is rotated...)
  Transform3D t3d = new Transform3D();
viewTrans.getTransform(t3d);
Transform3D toRot = new Transform3D();
toRot.rotY(radians);
t3d.mul(toRot);
viewTrans.setTransform(t3d);
}


private Appearance setApp(java.net.URL imagethis) {
// Create the appearance for the Billboards (3D
sprites such as trees).
  Appearance app3 = new Appearance();

  Texture tex3 = new TextureLoader(imagethis,
this).getTexture();

  app3.setTexture(tex3);
  TextureAttributes texAttr2 = new TextureAttributes();
  texAttr2.setTextureMode(TextureAttributes.MODUL
ATE);
  app3.setTextureAttributes(texAttr2);

// Add transparency so boundaries of the image are
left out
  TransparencyAttributes tra2 = new
TransparencyAttributes();
tra2.setTransparencyMode(TransparencyAttributes.B
LEND_ONE_MINUS_SRC_ALPHA);
tra2.setTransparency(0.0f);
 app3.setTransparencyAttributes(tra2);


return app3;
}

private void Model3D(float x, float y, float z, Scene
sc, java.net.URL imagethis, double scaling, boolean
rep) {
// Load a model from a file, transform it and place it.

Transform3D move2 = new Transform3D();
float a = -Landwidth*5;
float b = -Landheight*5;
Vector3f v3b = new Vector3f(x,y,z);
Transform3D move3 = new Transform3D();
//AxisAngle4d aa = new AxisAngle4d(90,1,1,0);
move2.set(v3b);
move3.rotX(-Math.PI/2.0);

TransformGroup tg2 = new TransformGroup(move2);
TransformGroup tg3 = new TransformGroup(move3);

BranchGroup BG = sc.getSceneGroup();
//BG.setPickable(true);
Shape3D hill = (Shape3D)BG.getChild(0);
//hill.setPickable(true);

// Create the appearance for the model
Appearance app2 = new Appearance();

TextureLoader texLoader = new
TextureLoader(imagethis,null);
Texture2D texture = (Texture2D)
texLoader.getTexture();
if (texture!= null)
  texture.setEnable(true);
app2.setTexture(texture);

// Create a bounding box both for increased
performance, but also for picking
BoundingBox boundbox = new
BoundingBox(hill.getBounds());
Point3d lower = new Point3d();
Point3d upper = new Point3d();
boundbox.getLower(lower);
boundbox.getUpper(upper);
double width = upper.x - lower.x;
double height = upper.y - lower.y;
// Create plance coordinates
Vector4f planeS = new Vector4f( (float)
(1.0/width),0.0f,0.0f,(float)(-lower.x/width));
Vector4f planeT = new Vector4f( 0,(float)
(1.0/height),0.0f,(float)(-lower.y/height));


// Generate tex coordinates
TexCoordGeneration texGen = new
TexCoordGeneration();
if (rep == false) {
texGen.setPlaneS(planeS);
texGen.setPlaneT(planeT);
}
else if (rep == true) {
texGen.setPlaneS(planeT);
texGen.setPlaneT(planeS);
}

// add texture attributes, convert model to polygon
and ensure double-sided filling
TextureAttributes ta = new TextureAttributes();
ta.setTextureMode(TextureAttributes.MODULATE);
PolygonAttributes pa = new PolygonAttributes();
pa.setCullFace(PolygonAttributes.CULL_NONE);
app2.setPolygonAttributes(pa);
app2.setTexCoordGeneration(texGen);
app2.setTextureAttributes(ta);

hill.setAppearance(app2);

// Add the leaf-node transformations and translations


tg3.addChild(sc.getSceneGroup());

Transform3D Scale = new Transform3D();
Scale.setScale(scaling);
TransformGroup tg4 = new TransformGroup(Scale);
```

```
tg4.addChild(tg3);

tg2.addChild(tg4);
tg2.setPickable(true);
PickTool.setCapabilities(hill,PickTool.INTERSECT_C
OORD);


objModels.addChild(tg2);

}


private void Addprim(int type,float x, float y, float z) {
// Create the 2 primitive landmarks, cone and sphere
(small peak and rock)

  // Create appearance object
Appearance app = new Appearance();
// INIT THE TEXTURE
Texture tex2 = null;
if (type == CONE) {
tex2 =  new TextureLoader(texImage,
this).getTexture();
}
else if (type == BOX || type == SPHERE) {
  tex2 =  new TextureLoader(hillImage,
this).getTexture();

}

app.setTexture(tex2);
TextureAttributes texAttr = new TextureAttributes();
texAttr.setTextureMode(TextureAttributes.MODULAT
E);
app.setTextureAttributes(texAttr);



if (type == BOX) {
// Create a box
Box chosentype = new Box(5f, 5f, 5f,
                    Box.GENERATE_TEXTURE_COOR
DS, app);

Transform3D move = new Transform3D();
Vector3f v3 = new Vector3f(x,y,z);
//move.transform(v3);
move.set(v3);
TransformGroup tg = new TransformGroup(move);
tg.addChild(chosentype);
tg.setCapability(tg.ENABLE_PICK_REPORTING);
tg.setPickable(true);
objRoot.addChild(tg);

}

else if (type == CONE) {
// Create a cone
Cone chosentype = new
Cone(5f,5f,Cone.GENERATE_TEXTURE_COORDS,
app);

Transform3D move = new Transform3D();
Vector3f v3 = new Vector3f(x,y,z);
//move.transform(v3);
move.set(v3);
TransformGroup tg = new TransformGroup(move);
tg.addChild(chosentype);
tg.setCapability(tg.ENABLE_PICK_REPORTING);
tg.setPickable(true);
objRoot.addChild(tg);

}

else if (type == SPHERE) {
// Create a sphere
Sphere chosentype = new
Sphere(2.5f,Sphere.GENERATE_TEXTURE_COOR
DS,app);

  Transform3D move = new Transform3D();
  Vector3f v3 = new Vector3f(x,y-1f,z);
  move.set(v3);
  TransformGroup tg = new TransformGroup(move);
  tg.addChild(chosentype);
  tg.setCapability(tg.ENABLE_PICK_REPORTING);
  tg.setPickable(true);
  objRoot.addChild(tg);


}

}

private void Sprite3D(float x, float y, float z,
Appearance app,float width,float height) {
// Method that creates 3dSprites with billboard
behaviour (e.g. - always face camera)
// create a box to host the tree image and add it to
the objRoot
Box Tree = new Box(width, 0.01f, height,
                    Box.GENERATE_TEXTURE_COOR
DS, app);
// Do not make it pickable as billboards are used
mostly for populatingand collision detection is not
desirable
Tree.setPickable(false);
// Transform, move and rotate
Transform3D moveb = new Transform3D();
Vector3f v3c = new Vector3f(x,y,z);
moveb.set(v3c);
TransformGroup tgb = new TransformGroup(moveb);

Transform3D movec = new Transform3D();
movec.rotX(-Math.PI/2.0);
TransformGroup tgc = new TransformGroup(movec);


TransformGroup tgd = new TransformGroup();
tgd.setCapability(TransformGroup.ALLOW_TRANSF
ORM_READ);
tgd.setCapability(TransformGroup.ALLOW_TRANSF
ORM_WRITE);
```

```
// Create a billboard behavior
Billboard bill = new Billboard(tgd);
bill.setSchedulingBounds(bounds);
bill.setAlignmentAxis(0,1,0);
// Add billboard behavior to branchgroup
objSprites.addChild(bill);
// add leaf-nodes
tgc.addChild(Tree);

tgd.addChild(tgc);

tgb.addChild(tgd);

objSprites.addChild(tgb);

}

private void checkray(Vector3d v3d) {
// Method creates a ray from the camera position and
shoots it forward. If an intersection occurs it causes
// the camera to stop moving in that direction
pickedsomething = false;
// Create the picker
  PickTool picker;
  picker = new PickTool(objModels);
// Picker mode
picker.setMode(PickTool.BOUNDS);

Transform3D t3d = new Transform3D();
viewTrans.getTransform(t3d);
Transform3D toMove = new Transform3D();
toMove.setTranslation(v3d);
t3d.mul(toMove);

Vector3d new3d = new Vector3d();
t3d.get(new3d);
Point3d p3d = new
Point3d(new3d.x,new3d.y,new3d.z);
Point3d p3db = new
Point3d(new3d.x*1.01,new3d.y,new3d.z*1.01);
picker.setShapeSegment(p3d,p3db);
PickResult picked = picker.pickClosest();
if (picked != null) {
if (picked.numIntersections() != 0) {
//  If something was picked then set pickedsomething
as true
pickedsomething = true;
}
}

// This section creates a ray from the heighest
possible point and shoots it down
// once an intersection occurs it takes the
coordinates. This enables the camera to
// transform to the intersected position and therefore
climb and descend slopes
// No gravity yet :)
Point3d p3dfloor = new
Point3d(new3d.x,60,new3d.z);
Point3d p3dbfloor = new
Point3d(new3d.x,0,new3d.z);
Vector3d v3dfloor = new Vector3d(0,-1,0);
Point3d intersectedat = new Point3d();
currentheight = new Point3d();
currentheight = nextpt;

PickTool pickerfloor;
pickerfloor = new PickTool(objHills);
pickerfloor.setMode(PickTool.GEOMETRY_INTERSE
CT_INFO);

pickerfloor.setShapeRay(p3dfloor,v3dfloor);
PickResult pickedfloor = pickerfloor.pickAny();

if (pickedfloor != null) {
if (pickedfloor.numIntersections() != 0) {
PickIntersection pi = pickedfloor.getIntersection(0);



// Attempt to get the intersected coordinates, if this
fails use previous ones (if for example the user exits
the
// landscape area)
try {
  nextpt = pi.getPointCoordinates();
if (floorcollide == true) {
if (nextpt.z > 2) {
  pickedsomething = true;
nextpt = currentheight;
}
}

}
catch (Exception e) {
nextpt = p3dfloor;

}

}
}

}

private double returnheight(double x, double z) {
// Method used at startup to get initial height position
for the camera and landmarks. A ray is shot at a
region
  // from the highest position. The intersected
coordinates are returned and the landmarks/camera
are placed
  // on those coordinates
Point3d landh = new Point3d();

Point3d p3dfloor = new Point3d(x,60,z);
Vector3d v3dfloor = new Vector3d(0,-1,0);

PickTool pickerfloor;
//pickerfloor = new PickTool(objHills);
pickerfloor = new PickTool(scene);
pickerfloor.setMode(PickTool.GEOMETRY_INTERSE
CT_INFO);

pickerfloor.setShapeRay(p3dfloor,v3dfloor);
```

```
PickResult pickedfloor = pickerfloor.pickAny();

if (pickedfloor != null) {
System.out.print("NOT NULL");
if (pickedfloor.numIntersections() != 0) {
PickIntersection pi = pickedfloor.getIntersection(0);

try {
landh = pi.getPointCoordinates();

}
catch (Exception e) {
}
}
}

return landh.z/2;

}


private void checkcollisions(Transform3D t3dt) {
// outdated method, keeping it here for future
reference
  Vector3d v3d = new Vector3d();
  t3dt.get(v3d);


int checkx = (int)v3d.x/10 + Landwidth/2;
int checkz = (int)v3d.z/10+ Landheight/2;

int a = (int)v3d.x + Landwidth*5;
int b = (int)v3d.z + Landwidth*5;

}

private void addlandmarks() {
// This method first extracts rgb information from
every pixel in the landmarks png image.
  // it then sets the appropriate landmark in the
appropriate coordinate.

// ADD THE LANDMARKS

int tempdata = Landmarks.getRGB(0,0);
System.out.print("Landmark Colordata is: " +
tempdata);
   for (int j = 0; j < Landheight; j++) {

   for (int i = 0; i < Landwidth; i++) {

/// EXTRACT COLOR INFORMATION FROM THE
FILE
  int colordata = Landmarks.getRGB(i,j);


// IF Tree1
if (colordata == -65536) {
// ADD  TREE1

// Create appearance for the tree
Appearance app3 = setApp(Tree1Image);

/// ADD THE TREE1
Sprite3D(-Landwidth*5+(i*10)+5,(float)(returnheight(-
Landwidth*5+(i*10)+5,-Landwidth*5+(j*10)+5))+3.1f,-
Landwidth*5+(j*10)+5,app3,4,6);


}

// IF TREE2
else if (colordata == -16776961) {
// ADD  TREE2

// Create appearance for the tree
Appearance app3 = setApp(Tree2Image);

/// ADD THE TREE2
Sprite3D(-Landwidth*5+(i*10)+5,(float)(returnheight(-
Landwidth*5+(i*10)+5,-Landwidth*5+(j*10)+5))+1.6f,-
Landwidth*5+(j*10)+5,app3,7,4);

}

// IF TREE3
else if (colordata == -13474586) {
// ADD  TREE3

// Create appearance for the tree
Appearance app3 = setApp(Tree3Image);

/// ADD THE TREEs3
Sprite3D(-Landwidth*5+(i*10)+5,(float)(returnheight(-
Landwidth*5+(i*10)+5,-Landwidth*5+(j*10)+5))+3.1f,-
Landwidth*5+(j*10)+5,app3,4,6);

}

// IF TREE4

else if (colordata == -8849931) {

// ADD  TREE4

// Create appearance for the tree
Appearance app3 = setApp(Tree4Image);

/// ADD THE four TREEs4
Sprite3D(-Landwidth*5+(i*10)+5,(float)(returnheight(-
Landwidth*5+(i*10)+5,-Landwidth*5+(j*10)+5))+3.1f,-
Landwidth*5+(j*10)+5,app3,4,12);


}

// IF TREE5

else if (colordata == -14752056) {
// ADD  TREE5

// Create appearance for the tree
Appearance app3 = setApp(Tree5Image);
```

```
/// ADD THE four TREEs5
Sprite3D(-Landwidth*5+(i*10)+5,(float)(returnheight(-
Landwidth*5+(i*10)+5,-Landwidth*5+(j*10)+5))+3.1f,-
Landwidth*5+(j*10)+5,app3,4,6);

}

// IF TREE6

else if (colordata == -14759736) {
// ADD  TREE6

// Create appearance for the tree
Appearance app3 = setApp(Tree6Image);

/// ADD THE four TREEs6
Sprite3D(-Landwidth*5+(i*10)+5,(float)(returnheight(-
Landwidth*5+(i*10)+5,-Landwidth*5+(j*10)+5))+3.1f,-
Landwidth*5+(j*10)+5,app3,4,6);

}

// IF TREE7
else if (colordata == -14764856) {
// ADD  TREE7

// Create appearance for the tree
Appearance app3 = setApp(Tree7Image);

/// ADD THE four TREEs7
Sprite3D(-Landwidth*5+(i*10)+5,(float)(returnheight(-
Landwidth*5+(i*10)+5,-Landwidth*5+(j*10)+5))+3.1f,-
Landwidth*5+(j*10)+5,app3,4,6);

}


// IF TREE8
else if (colordata == -14769976) {
// ADD  TREE8

// Create appearance for the tree
Appearance app3 = setApp(Tree8Image);

/// ADD THE four TREEs8
Sprite3D(-Landwidth*5+(i*10)+5,(float)(returnheight(-
Landwidth*5+(i*10)+5,-Landwidth*5+(j*10)+5))+3.1f,-
Landwidth*5+(j*10)+5,app3,4,6);

}

// IF TREE9
else if (colordata == -14775096) {
// ADD  TREE9

// Create appearance for the tree
Appearance app3 = setApp(Tree9Image);

/// ADD THE four TREEs9
Sprite3D(-Landwidth*5+(i*10)+5,(float)(returnheight(-
Landwidth*5+(i*10)+5,-Landwidth*5+(j*10)+5))+3.1f,-
Landwidth*5+(j*10)+5,app3,4,6);

}


// IF TREE10
else if (colordata == -14780216) {
// ADD  TREE10

// Create appearance for the tree
Appearance app3 = setApp(Tree10Image);

/// ADD THE four TREEs10
Sprite3D(-Landwidth*5+(i*10)+5,(float)(returnheight(-
Landwidth*5+(i*10)+5,-Landwidth*5+(j*10)+5))+3.1f,-
Landwidth*5+(j*10)+5,app3,4,6);

}

// IF TREE11
else if (colordata == -14785336) {
// ADD  TREE11

// Create appearance for the tree
Appearance app3 = setApp(Tree11Image);

/// ADD THE four TREEs11
Sprite3D(-Landwidth*5+(i*10)+5,(float)(returnheight(-
Landwidth*5+(i*10)+5,-Landwidth*5+(j*10)+5))+3.1f,-
Landwidth*5+(j*10)+5,app3,4,6);

}


// IF BUSHES
else if (colordata == -14790456) {
  // ADD  Bushes

// Create appearance for the Bushes
Appearance app3 = setApp(BushImage);

/// ADD THE five Bushes
Sprite3D(-Landwidth*5+(i*10)+3,(float)(returnheight(-
Landwidth*5+(i*10)+3,-Landwidth*5+(j*10)+3))-1,-
Landwidth*5+(j*10)+3,app3,1f,1f);
Sprite3D(-Landwidth*5+(i*10),(float)(returnheight(-
Landwidth*5+(i*10),-Landwidth*5+(j*10)+8))-1,-
Landwidth*5+(j*10)+8,app3,1f,1f);
Sprite3D(-Landwidth*5+(i*10)+8,(float)(returnheight(-
Landwidth*5+(i*10)+8,-Landwidth*5+(j*10)+7))-1,-
Landwidth*5+(j*10)+7,app3,1f,1f);
Sprite3D(-Landwidth*5+(i*10)+7,(float)(returnheight(-
Landwidth*5+(i*10)+7,-Landwidth*5+(j*10)+1))-1,-
Landwidth*5+(j*10)+1,app3,1f,1f);
Sprite3D(-Landwidth*5+(i*10)+3,(float)(returnheight(-
Landwidth*5+(i*10)+3,-Landwidth*5+(j*10)+5))-1,-
Landwidth*5+(j*10)+5,app3,1f,1f);

}

// IF cart
else if (colordata == -16711936) {
  // ADD  cart
```

```
// Create appearance for the cart
Appearance app3 = setApp(CartImage);

/// ADD THE cart
Sprite3D(-Landwidth*5+(i*10)+3,(float)(returnheight(-
Landwidth*5+(i*10)+3,-Landwidth*5+(j*10)+5))-1f,-
Landwidth*5+(j*10)+5,app3,2f,1f);

}

// IF House of wood
else if (colordata == -39836) {

Scene house = null;
   try {

     ModelLoader loader = new ModelLoader();

 house = loader.load("../images/house2.3DS");

   }
   catch (Exception e) {
   System.out.print("CANNOT LOAD FILE!");
   }



Model3D(-Landwidth*5+(i*10)+5,(float)(returnheight(-
Landwidth*5+(i*10)+5,-Landwidth*5+(j*10)+5))-2f,-
Landwidth*5+
(j*10)+5,house,HouseTexImage,0.06,true);




}

// IF House of metal
else if (colordata == -65436) {

  Scene house = null;
     try {

       ModelLoader loader = new ModelLoader();

   house = loader.load("../images/house2.3DS");

     }
     catch (Exception e) {
     System.out.print("CANNOT LOAD FILE!");
     }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2f,-Landwidth*5+
(j*10)+5,house,HouseTex2Image,0.06,true);


}

// IF House of Stone

else if (colordata == -16751361) {

  Scene house = null;
     try {

       ModelLoader loader = new ModelLoader();

   house = loader.load("../images/house2.3DS");

     }
     catch (Exception e) {
     System.out.print("CANNOT LOAD FILE!");
     }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2f,-Landwidth*5+
(j*10)+5,house,HouseTex3Image,0.06,true);


}



// IF BARN

else if (colordata == -10210816) {

  Scene house = null;
     try {

       ModelLoader loader = new ModelLoader();

   house = loader.load("../images/barn.3DS");

     }
     catch (Exception e) {
     System.out.print("CANNOT LOAD FILE!");
     }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2f,-Landwidth*5+
(j*10)+5,house,BarnImage,0.035,true);


}

// IF TUBE

else if (colordata == -10855846) {

  Scene house = null;
```

```
    try {

      ModelLoader loader = new ModelLoader();

   house = loader.load("../images/tube.3DS");

    }
    catch (Exception e) {
    System.out.print("CANNOT LOAD FILE!");
    }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2f,-Landwidth*5+
(j*10)+5,house,MarbleImage,0.035,false);


}

// IF CAGE

else if (colordata == -987126) {

  Scene house = null;
    try {

      ModelLoader loader = new ModelLoader();

   house = loader.load("../images/cage.3DS");

    }
    catch (Exception e) {
    System.out.print("CANNOT LOAD FILE!");
    }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2f,-Landwidth*5+
(j*10)+5,house,Metal2Image,0.01,false);


}

// IF Plane

else if (colordata == -16774416) {

  Scene house = null;
    try {

      ModelLoader loader = new ModelLoader();

   house = loader.load("../images/plane.3DS");

    }
    catch (Exception e) {
    System.out.print("CANNOT LOAD FILE!");
    }

  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2f,-Landwidth*5+
(j*10)+5,house,WierdImage,0.02,false);



}

// IF CAR1

else if (colordata == -12829575) {

  Scene house = null;
    try {

      ModelLoader loader = new ModelLoader();

   house = loader.load("../images/car1.3DS");

    }
    catch (Exception e) {
    System.out.print("CANNOT LOAD FILE!");
    }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);


}

// IF CAR2

else if (colordata == -16776963) {

  Scene house = null;
    try {

      ModelLoader loader = new ModelLoader();

   house = loader.load("../images/car2.3DS");

    }
    catch (Exception e) {
    System.out.print("CANNOT LOAD FILE!");
    }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);

}
```

```
// IF CAR3

else if (colordata == -9735538) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/car3.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);

}

// IF CAR4

else if (colordata == -16713479) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/car4.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);

}

// IF CAR5

else if (colordata == -4235328) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/car5.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);

}

// IF BUS

else if (colordata == -9643558) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/bus.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);

}


// IF MISSILE

else if (colordata == -10197916) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/missile.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);
```

```
}

// IF TANK

else if (colordata == -426883) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/tank.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);

}

// IF Garbage

else if (colordata == -393215) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/garbage.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);

}

// IF Bench

else if (colordata == -1710733) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/bench.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);

}

// IF Fountain

else if (colordata == -11427246) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/fountain.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);

}

// IF Statue1

else if (colordata == -2500351) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/statue1.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
```

```
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);

}
// IF Statue2

else if (colordata == -3213063) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/statue2.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);

}

// IF Statue3

else if (colordata == -8454658) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/statue3.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);

}

// IF Statue4

else if (colordata == -16722175) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/statue4.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);

}

// IF Statue5

else if (colordata == -5834246) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/statue5.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }

  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);

}

// IF lamp post

else if (colordata == -14028302) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/lamppost.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);
```

```
}

// IF pond

else if (colordata == -16755884) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/pond.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }

  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);
}

// IF tent

else if (colordata == -5917515) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/tent.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }

  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);
}

// IF METEOR

else if (colordata == -11141120) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/meteor.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }

  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);
}

// IF TIGER

else if (colordata == -16744319) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/tiger.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }

  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);
}

// IF ELEPHANT

else if (colordata == -16767449) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/elephant.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }

  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);
 }

// IF HORSE

else if (colordata == -14248480) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/horse.3DS");
```

```
    }
    catch (Exception e) {
    System.out.print("CANNOT LOAD FILE!");
    }

  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);
}

// IF PENGUIN

else if (colordata == -16764074) {

  Scene house = null;
    try {

      ModelLoader loader = new ModelLoader();

   house = loader.load("../images/penguin.3DS");

    }
    catch (Exception e) {
    System.out.print("CANNOT LOAD FILE!");
    }

  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);
}

// IF TV

else if (colordata == -16763906) {

  Scene house = null;
    try {

      ModelLoader loader = new ModelLoader();

   house = loader.load("../images/tv.3DS");

    }
    catch (Exception e) {
    System.out.print("CANNOT LOAD FILE!");
    }

  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);
}

// IF ROBOT

else if (colordata == -14471633) {

  Scene house = null;
    try {

      ModelLoader loader = new ModelLoader();

   house = loader.load("../images/robot.3DS");

    }
    catch (Exception e) {
    System.out.print("CANNOT LOAD FILE!");
    }



  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);
}

// IF APE

else if (colordata == -8809740) {

  Scene house = null;
    try {

      ModelLoader loader = new ModelLoader();

   house = loader.load("../images/ape.3DS");

    }
    catch (Exception e) {
    System.out.print("CANNOT LOAD FILE!");
    }

  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);
}

// IF Giant Insect

else if (colordata == -8618883) {

  Scene house = null;
    try {

      ModelLoader loader = new ModelLoader();

   house = loader.load("../images/insect.3DS");

    }
    catch (Exception e) {
    System.out.print("CANNOT LOAD FILE!");
    }


  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);
```

```
}

// IF WOMAN

else if (colordata == -14583687) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/woman.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }

  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);
}

// IF MAN

else if (colordata == -5589276) {

  Scene house = null;
      try {

        ModelLoader loader = new ModelLoader();

   house = loader.load("../images/man.3DS");

      }
      catch (Exception e) {
      System.out.print("CANNOT LOAD FILE!");
      }

  Model3D(-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2.1f,-Landwidth*5+
(j*10)+5,house,Wierd2Image,0.008,true);
}


// IF PEAK
else if (colordata == -12829441) {

Addprim(CONE,-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2f,-Landwidth*5+(j*10)+5);

}

// IF ROCK
else if (colordata == -1) {

Addprim(SPHERE,-Landwidth*5+(i*10)+5,(float)
(returnheight(-Landwidth*5+(i*10)+5,-Landwidth*5+
(j*10)+5))-2f,-Landwidth*5+(j*10)+5);

}



   }
   }

// Compile the branchgroups objModels and
objSprites. This was done after the scene since ray
picking works
   // only on branchgroups that are live.
   objSprites.compile();
simpleU.addBranchGraph(objSprites);
objModels.compile();
simpleU.addBranchGraph(objModels);

// Change the resolution accordingly, the resolution is
resized here in order to fix a strange D3D bug
// when adding the sprites

if (chosenscreen == 0) {
// Twice the phonesize 240*290
this.setSize(480,580);
}


if (chosenscreen == 1) {
this.setSize(640,400);
}
else if (chosenscreen == 2) {
this.setSize(800,600);
}
else if (chosenscreen == 3) {
this.setSize(1024,768);
}


}


private void populateWorld() {
// Populate the world
  canvas3D.initDate();
addlandmarks();
double convertang = radians*Math.PI/180.0;
double movexx = Math.sin(convertang);
double moveyy = Math.cos(convertang);
Vector3d v3d = new
Vector3d(movexx*0.001,0,moveyy*0.001);
Vector3d v3db = new
Vector3d(movexx*0.001,0,moveyy*0.001);

inverted = false;
checkray(v3db);
testMove(v3d);
populated = true;

}
```

```
}

// TimerInterface.java

package skar;

/**
 * <p>Title: </p>
 * <p>Description: </p>
 * <p>Copyright: Copyright (c) 2005</p>
 * <p>Company: </p>
 * @author unascribed
 * @version 1.0
 */
import java.awt.*;
import java.awt.image.*;
import java.awt.event.*;
import java.applet.*;
import com.sun.j3d.*;
import com.sun.j3d.internal.*;
import com.sun.j3d.loaders.*;
import com.sun.j3d.utils.geometry.*;
import com.sun.j3d.utils.universe.*;
import com.sun.j3d.utils.universe.*;
import javax.media.j3d.*;
import com.sun.j3d.loaders.objectfile.ObjectFile;
import com.sun.j3d.loaders.ParsingErrorException;
import
com.sun.j3d.loaders.IncorrectFormatException;
import com.sun.j3d.loaders.Scene;

import java.applet.Applet;
import java.awt.BorderLayout;
import java.awt.Frame;
import java.awt.event.*;
import com.sun.j3d.utils.applet.MainFrame;

import com.sun.j3d.utils.universe.*;
import com.sun.j3d.utils.geometry.ColorCube;
import javax.media.j3d.*;
import javax.vecmath.*;
import com.sun.j3d.utils.image.TextureLoader;
import com.sun.j3d.utils.behaviors.mouse.*;
import com.sun.j3d.utils.behaviors.keyboard.*;
import java.lang.Thread;
import com.sun.j3d.utils.picking.*;
import javax.imageio.*;
import java.io.*;
import ncsa.j3d.loaders.*;
import ncsa.j3d.loaders.load3ds.*;
import com.sun.j3d.loaders.objectfile.*;
import org.j3d.ui.navigation.CollisionListener;
import com.sun.j3d.utils.image.*;
import java.util.Date;

// THIS CLASS EXTENDS CANVAS3D. The reason
we need to extend it is so we can add 2D awt objects
on the 3D canvas.
// This displays time, menu information etc...

public class TimerInterface extends Canvas3D {
final java.text.DateFormat timeFmt =
java.text.DateFormat.getTimeInstance(java.text.Date
Format.MEDIUM);
public Date startdate;
public boolean currenttimeshow = false;
public TimerInterface(GraphicsConfiguration gcln) {
super(gcln);
}

public void initDate() {
startdate = new Date();
}

// The postRender() Canvas3D class has been
overided. 2D images and texts are added
public void postRender() {
  J3DGraphics2D g = getGraphics2D();
  g.setColor(Color.orange);

// First screen
if (SKAR3D.populated == false && SKAR3D.begin
== true) {
  g.setColor(Color.black);
  g.fillRect(0,0,1024,720);

g.setColor(Color.orange);

  g.drawString("PRESS '1' for 320*240",0,60);
  g.drawString("PRESS '2' for 640*480",0,80);
  g.drawString("PRESS '3' for 800*600",0,100);
  g.drawString("PRESS '4' for 1024*768",0,120);


  if (SKAR3D.floorcollide == false) {
     g.drawString("Floor colliding is disabled",0,160);
  }
  else g.drawString("Floor colliding is
enabled",0,160);

  if (SKAR3D.transfloor == false) {
     g.drawString("Floor is not transparent",0,180);
  }
  else g.drawString("Floor is transparent",0,180);

}

// If the user presses 'Escape'
if (currenttimeshow == false && SKAR3D.populated
== true){
  g.drawString("Started at: " +
timeFmt.format(startdate),0,20);
}

else if (currenttimeshow == true) {
g.setColor(Color.black);
g.fillRect(0,0,1024,768); // Max resolution for SKAR
engine
g.setColor(Color.orange);
g.drawString("Started at: " +
timeFmt.format(startdate),0,20);
  g.drawString("Finished at: " + timeFmt.format(new
Date()),0,40);
```

```
  System.out.print("\nStarted at: " +
timeFmt.format(startdate));
    System.out.print("\nFinished at: " +
timeFmt.format(new Date()));

  g.drawString("Press 'Q' to QUIT",0,60);
  g.drawString("OR 'B' to RESUME",0,80);

}
//System.out.print(timeFmt.format(new Date()));
g.flush(true);

}

}

// Water.java

package skar;

/**
 * <p>Title: </p>
 * <p>Description: </p>
 * <p>Copyright: Copyright (c) 2005</p>
 * <p>Company: </p>
 * @author unascribed
 * @version 1.0
 */

import java.awt.*;
import java.awt.image.*;
import java.awt.event.*;
import java.applet.*;
import com.sun.j3d.*;
import com.sun.j3d.internal.*;
import com.sun.j3d.loaders.*;
import com.sun.j3d.utils.geometry.*;
import com.sun.j3d.utils.universe.*;
import com.sun.j3d.utils.universe.*;
import javax.media.j3d.*;
import com.sun.j3d.loaders.objectfile.ObjectFile;
import com.sun.j3d.loaders.ParsingErrorException;
import
com.sun.j3d.loaders.IncorrectFormatException;
import com.sun.j3d.loaders.Scene;

import java.applet.Applet;
import java.awt.BorderLayout;
import java.awt.Frame;
import java.awt.event.*;
import com.sun.j3d.utils.applet.MainFrame;
import com.sun.j3d.utils.universe.*;
import com.sun.j3d.utils.geometry.ColorCube;
import javax.media.j3d.*;
import javax.vecmath.*;
import com.sun.j3d.utils.image.TextureLoader;
import com.sun.j3d.utils.behaviors.mouse.*;
import com.sun.j3d.utils.behaviors.keyboard.*;
import java.lang.Thread;
import com.sun.j3d.utils.picking.*;
import javax.imageio.*;
import java.io.*;
import ncsa.j3d.loaders.*;
import ncsa.j3d.loaders.load3ds.*;
import com.sun.j3d.loaders.objectfile.*;
import org.j3d.ui.navigation.CollisionListener;
import com.sun.j3d.utils.image.*;

// This class create a simple plane made up of four
coordinates. The size depends on the width of the
contour map.
// It is used for the creation of the bottom layer for
things such as water, grass, etc...

public class Water extends Shape3D{

  public Water(int length) {


///// Create the VERTICES
float[] pts = {-length*10/2,-length*10/2,0,
length*10/2,-length*10/2,0,
length*10/2,length*10/2,0,
-length*10/2,length*10/2,0};
int[]  stripCounts = {4};


// Create a GeometryInfo object
GeometryInfo gInf = new
GeometryInfo(GeometryInfo.POLYGON_ARRAY);


// Set the coordinates
gInf.setCoordinates(pts);

// Set the stip count
gInf.setStripCounts(stripCounts);
// Convert to triangles for faster rendering (2
triangles)
gInf.convertToIndexedTriangles();

// Create normals to add shadows and rougness
NormalGenerator ng = new NormalGenerator();
ng.setCreaseAngle((float)Math.toRadians(44));
ng.generateNormals(gInf);
// Stipify for performance gain
Stripifier st = new Stripifier();
st.stripify(gInf);

// Finaly set the info into the Shape3D object
this.setGeometry(gInf.getGeometryArray());

  }
}
```

Appendix B - Mouse Dexterity Test Source Code

This section presents the source code for the Mouse dexterity test, created for the use of filtering participants according to their system knowledge during the experiments.

```
// Canvas.java
package dextest;


import javax.swing.*;
import java.awt.*;
import java.awt.image.*;
import java.lang.Runnable;
import java.util.Random;
import java.awt.event.*;
import java.awt.Robot;


public class Canvas implements Runnable,
MouseListener, MouseMotionListener
{
    // instance variables - replace the example below
with your own
    private JFrame display;
    public JPanel canvas;
    private Graphics g;
    private Color backgroundColour;
    BufferedImage DB_Image = null;

    Graphics DB_Graphics = null;


    public Random random;

    private boolean createnew = false;

    private boolean isButtonPressed = false;

    private boolean hasstarted, hasended;

    private int clickno = 0;

    private long begintimer, endtimer, resulttimer;

    private int pos = 0;

    int w, h, x, y;

    int mx, my;

    int t = 0;
int width = 800;
    int height = 600;

    public Robot mouser;



private boolean Running = true;

    public Canvas(String title, int width, int height,
Color bgColour)
    {
random = new Random();
      try {
              mouser = new Robot();
      } catch (Exception e) {
      }

        display = new JFrame();
        canvas = (JPanel)display.getContentPane();
        display.setTitle(title);
        display.setSize(width, height);
        canvas.setBackground(bgColour);
        backgroundColour = bgColour;
g = this.display.getGraphics();


display.addMouseListener(this);
//        addMouseListener(this);
  display.addMouseMotionListener(this);
        this.
        hasstarted = false;
        hasended = false;
        clickno = 0;
        begintimer = 0;
        endtimer = 0;
        resulttimer = 0;


        Thread t = new Thread(this);
t.start();
    }


    // MOUSE EVENTS

    public void mouseEntered(MouseEvent e) {
            // called when the pointer enters the applet's
rectangular area
    }

    public void mouseExited(MouseEvent e) {
            // called when the pointer leaves the applet's
rectangular area
    }

    public void mouseClicked(MouseEvent e) {
            // called after a press and release of a mouse
button
            // with no motion in between
            // (If the user presses, drags, and then
releases, there will be
            // no click event generated.)
    }

    public void mousePressed(MouseEvent e) { //
called after a button is

// pressed down
            isButtonPressed = true;
mx -=4;
            my-=30;
            if (pos == 0) {
                    if (mx >= 0 && mx <= 50 && my >= 0 &&
my <= 50) {
                            successclick();
                    }
```

```
        } else if (pos == 1) {
            if (mx >= 349 && mx <= 400 && my >= 0
&& my <= 50) {
                successclick();

            }
        } else if (pos == 2) {
            if (mx >= 750 && mx <= 800 && my >= 0
&& my <= 50) {
                successclick();

            }
        }

        else if (pos == 3) {
            if (mx >= 0 && mx <= 50 && my >= 274
&& my <= 324) {
                successclick();

            }
        }

        else if (pos == 4) {
            if (mx >= 750 && mx <= 800 && my >=
274 && my <= 324) {
                successclick();

            }
        }

        else if (pos == 5) {
            if (mx >= 0 && mx <= 50 && my >= 548
&& my <= 600) {
                successclick();

            }
        } else if (pos == 6) {
            if (mx >= 349 && mx <= 400 && my >=
548 && my <= 600) {
                successclick();

            }
        } else if (pos == 7) {
            if (mx >= 750 && mx <= 800 && my >=
548 && my <= 600) {
                successclick();
            }
        }

        // "Consume" the event so it won't be
processed in the
        // default manner by the source which
generated it.
        e.consume();
    }

    public void successclick() {
        if (hasstarted == true) {
            if (clickno < 19) {
                createnew = false;
               mouser.mouseMove(400, 300);
                clickno++;
            } else
                hasended = true;
        } else {
            hasstarted = true;
            mouser.mouseMove(400, 300);
            begintimer = System.currentTimeMillis();
        }

    }

    public void mouseReleased(MouseEvent e) { //
called after a button is

// released
        isButtonPressed = false;
        e.consume();
    }

    public void mouseMoved(MouseEvent e) { // called
during motion when no

// buttons are down
        mx = e.getX();
        my = e.getY();
        e.consume();
    }

    public void mouseDragged(MouseEvent e) { //
called during motion with

// buttons down
        mx = e.getX();
        my = e.getY();
        e.consume();
    }

public void run() {

    while (Running == true) {

if (g != null)      {
      if (createnew == false) {
          pos = Math.abs(random.nextInt(8));
          // pos = 5;
          createnew = true;
      }

update(g);
}


          try
          {
            Thread.sleep(20);
          }
          catch (Exception e)
          {
            // ignoring exception at the moment
          }


    }
```

```
    }


    public void update(Graphics g) {

            // create the buffer if it does not exist
            if (DB_Graphics == null) {
                    DB_Image = new
BufferedImage(800,600,BufferedImage.TYPE_INT_
RGB);

                    DB_Graphics =
DB_Image.getGraphics();
            }
            // clear the buffer
            DB_Graphics.setColor(Color.white); // set the
background color
            DB_Graphics.fillRect(0, 0, 800, 600); // clear
the buffer with the

// background color

            // draw the current state on the buffer.
Replace this
            DB_Graphics.setColor(Color.black);
            DB_Graphics.fillRect(x, y, 800, 600);
            DB_Graphics.setColor(Color.white);

            if (hasstarted == false) {
                    DB_Graphics.drawString(
                                    "Welcome to the Kyritsis &
Gulliver Dex Test: ", 300, 250);
                    DB_Graphics.drawString(
                                    "The objective of the test is to
click on the", 300, 270);
                    DB_Graphics.drawString("red square as
soon as it appears.", 300,
                                    290);
                    DB_Graphics.drawString("Click on it
now to begin the test... ",
                                    300, 310);
            } else if (hasstarted == true) {
                    if (hasended == false) {
                            endtimer =
System.currentTimeMillis();
                            resulttimer = endtimer - begintimer;

                            DB_Graphics.drawString("Total
Time: " + resulttimer, 300, 250);
                    } else {
                            DB_Graphics.drawString("You
managed to get 20 clicks in: "
                                            + resulttimer + "
milliseconds", 300, 250);

                    }
            }

            if (createnew == true) {
                    // DB_Graphics.drawString("THE
RANDOM IS: " + pos, 150, 50);
                    // DB_Graphics.drawString("Mouse
Position is: " +
                    // mx+","+my,400,400);
                    DB_Graphics.setColor(Color.red);

                    if (pos == 0)
                            DB_Graphics.fillRect(0, 0, 50, 50);
                    else if (pos == 1)
                            DB_Graphics.fillRect(width / 2 - 50,
0, 50, 50);
                    else if (pos == 2)
                            DB_Graphics.fillRect(width - 50, 0,
50, 50);
                    else if (pos == 3)
                            DB_Graphics.fillRect(0, height / 2 -
25, 50, 50);
                    else if (pos == 4)
                            DB_Graphics.fillRect(width - 50,
height / 2 - 25, 50,
                                            50);
                    else if (pos == 5)
                            DB_Graphics.fillRect(0, height - 50,
50, 50);
                    else if (pos == 6)
                            DB_Graphics.fillRect(width / 2 - 25,
height - 50, 50,
                                            50);
                    else
                            DB_Graphics.fillRect(width - 50,
height - 50, 50, 50);

            }

            // copy the buffer to the canvas
//          DB_Graphics.drawString("posx: "+mx+"
posy: "+my,300,310);
            g.drawImage(DB_Image, 0, 0, null);
    }



    public Canvas(String title)
    {
        this(title, 600, 400, Color.white);
    }

    /**
     * Constructor for objects of class DisplayCanvas
with a default
     * background colour (white).
     * @param title  title to appear in Canvas Frame
     * @param width  the desired width for the canvas
     * @param height  the desired height for the
canvas
     */
    public Canvas(String title, int width, int height)
    {
        this(title, width, height, Color.white);
    }

    /**
```

```
     * Sets the canvas visibility and brings canvas to
the front of screen
     * when made visible. This method can also be
used to bring an already
     * visible canvas to the front of other windows.
     *
     * @param visible  boolean value representing the
desired visibility of
     * the canvas (true or false)
     */
    public void setVisible(boolean visible)
    {
        display.setVisible(visible);
        if(g == null)
            g = (Graphics2D)canvas.getGraphics();
    }

   /**
     * provides information on visibility of the Canvas
     *
     * @return  boolean value representing the
visibility of
     * the canvas (true or false)
     */
    public boolean isVisible()
    {
        return display.isVisible();
    }

    public void setSize(int width, int height)
    {
        display.setSize(width, height);
    }

    /**
     * waits for a specified number of milliseconds
before finishing.
     * This provides an easy way to specify a small
delay which can be
     * used when producing animations.
     * @param  milliseconds  the number
     **/

}

// MainClass.java
package dextest;

import javax.swing.UIManager;
import java.awt.*;

/**
 * <p>Title: </p>
 * <p>Description: </p>
 * <p>Copyright: Copyright (c) 2007</p>
 * <p>Company: </p>
 * @author not attributable
 * @version 1.0
 */

public class MainClass {
  boolean packFrame = false;

  //Construct the application
  public MainClass() {
    Canvas frame = new
Canvas("Canvas",1024,768,Color.black);
    //Validate frames that have preset sizes
    //Pack frames that have useful preferred size info,
e.g. from their layout
    //Center the window
    Dimension screenSize =
Toolkit.getDefaultToolkit().getScreenSize();
    frame.setVisible(true);
    frame.setSize(1024,768);
  }

  //Main method
  public static void main(String[] args) {
    try {
      UIManager.setLookAndFeel(UIManager.getSyste
mLookAndFeelClassName());
    }
    catch(Exception e) {
      e.printStackTrace();
    }
    new MainClass();
  }
}
```

Appendix C - Pre-Experimental Questionnaire for the SKAR project

- Name……………………………………………..
- Age………………………………………………..
- Gender…………………………………………….
- Do you consider yourself an experienced PC user?..
- How many times a week on average do you use the PC?.....................................
- Do you own a game console?...
- Do you consider yourself a hardcore gamer?...
- Are you confident when navigating with the mouse?..

Appendix D - Guilford-Zimmerman Orientation Survey and Web-based Version

D.1 Source Code for web-based version

This section presents the source code for the web-based GZ Orientation Survey written in PhP.

```
<!--index.php -->

<?php session_start(); ?>
<?php

if ($_GET["lo"] != '') {
$_SESSION['questionnum']=7;
   }

if(isset($_SESSION['questionnum']))
$_SESSION['questionnum']=$_SESSION['questionnum']+1;
else
  $_SESSION['questionnum']=8;

$question = $_SESSION['questionnum'];
$correct = 0;
$wrong = 0;

if ($_GET["corr"] != '')
$correct = $_GET["corr"];

if ($_GET["wrong"] != '')
$wrong = $_GET["wrong"];


$timenow = 600000;

if ($_GET["timenow"] != '')
$timenow = $_GET["timenow"];

?>



<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<style type="text/css">
<!--
.style1 {
	font-size: 14px;
	font-weight: bold;
}
.style3 {font-size: 10px}
-->
</style>
</head>

<body>
<table width="672" height="80" border="1" align="center">
  <tr>
    <td width="224" height="74"><span class="style3">Created by Mr Markos Kyritsis and Dr Stephen Gulliver, Brunel University, London, UK.</span></td>
    <td width="212"><div align="center" class="style1">Guilford Zimmerman Orientation Survey (online version)</div></td>
    <td width="214"><p class="style3">Time Left: <?php echo($timenow/1000/60);?> mins</p>    </td>
  </tr>
</table>
<p> </p>
<table width="671" border="1" align="center">
  <tr>
    <td><div align="center"><img src="<?php echo($question);?>a.png" id = "prim" value = <?php echo($question);?> width="144" height="99" /></div></td>
    <td><div align="center"><img src="<?php echo($question);?>b.png" width="145" height="99" /></div></td>
  </tr>
</table>
<p> </p>
<table width="200" border="1" align="center">
  <tr>
    <td><div align="center"><img src="none.png" value = '-1' id = 'ans1' width="40" height="40" /></div></td>
    <td><div align="center"><img src="none.png" value = '-1' id = 'ans2'  width="40" height="40" /></div></td>
    <td><div align="center"><img src="none.png" value = '-1' id = 'ans3'  width="40" height="40" /></div></td>
  </tr>
</table>
<table width="200" border="1" align="center">
  <tr>
    <td><div align="center"><img src="rotleft.png" style = "cursor:pointer" id = "b0" width="40" height="40" onclick="javascript: copy(0);" /></div></td>
    <td><div align="center"><img src="up.png" id = "b1" style = "cursor:pointer" width="40" height="40" onclick="javascript: copy(1);"/></div></td>
    <td><div align="center"><img src="rotright.png" style = "cursor:pointer" id = "b2" width="40" height="40" onclick="javascript: copy(2);"/></div></td>
  </tr>
  <tr>
    <td><div align="center"><img src="left.png" style = "cursor:pointer" id = "b3" width="40" height="40" onclick="javascript: copy(3);"/></div></td>
    <td> </td>
    <td><div align="center"><img src="right.png" style = "cursor:pointer" id = "b4" width="40" height="40" onclick="javascript: copy(4);"/></div></td>
  </tr>
  <tr>
    <td> </td>
    <td><div align="center"><img src="down.png" style = "cursor:pointer" id = "b5" width="40" height="40" onclick="javascript: copy(5);"/></div></td>
    <td> </td>
  </tr>
</table>
```

```
<!--<form id="form2" name="form2" method="post"
action=""><div align="center"><input type="submit"
name="button1" id="button1" value="OK" />-->
<div align="center"><input type="button" name =
"button1" value="ok" onclick = "javascript: check();">
<input type="button" name = "button2" value="clear"
onclick = "javascript: deleteall();">
</div>
  </label>
<label>Question: <?php echo($question);?
>/67</label> </form>
</body>
</html>

<script>
document.getElementById('ans1').value = "-1";
document.getElementById('ans2').value = "-1";
document.getElementById('ans3').value = "-1";

var timeoutat = <?php echo($timenow);?>;
function updatetime() {
timeoutat-=100;
if (timeoutat == 0)
endit();
}
setInterval('updatetime()', 100);


    var pos = 1;

    function endit() {
    var corr = <?php echo($correct);?>;
    var wro = <?php echo($wrong);?>;

window.location = 'results.php?
corr='+corr+'&wrong='+wro;

    }

    function copy (t) {
if (pos <= 3) {
if (t == 0) {
document.getElementById('ans'+pos).src =
'rotleft.png';
document.getElementById('b0').style.display =
'none';
document.getElementById('b2').style.display =
'none';
}

else if (t == 1) {
document.getElementById('ans'+pos).src = 'up.png';
document.getElementById('b1').style.display =
'none';
document.getElementById('b5').style.display =
'none';

}
else if (t == 2) {
document.getElementById('ans'+pos).src =
'rotright.png';
document.getElementById('b0').style.display =
'none';
document.getElementById('b2').style.display =
'none';

}
else if (t == 3) {
document.getElementById('ans'+pos).src = 'left.png';
document.getElementById('b3').style.display =
'none';
document.getElementById('b4').style.display =
'none';

}
else if (t == 4) {
document.getElementById('ans'+pos).src =
'right.png';
document.getElementById('b3').style.display =
'none';
document.getElementById('b4').style.display =
'none';
}
else if (t == 5) {
document.getElementById('ans'+pos).src =
'down.png';
document.getElementById('b1').style.display =
'none';
document.getElementById('b5').style.display =
'none';

}
document.getElementById('ans'+pos).value = t;
pos++;
}
    }

    function check() {
//var current =
document.getElementById('prim').value;
var current = <?php echo($question);?>;
var corr = <?php echo($correct);?>;
var wro = <?php echo($wrong);?>;

var covered = 'false';


if (current == '8') {
//covered = 'true';
for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == "5") {
covered = 'true';
}
else if (document.getElementById('ans'+k).value !=
"-1")
covered = 'false';
}

}
else if (current == '9') {
for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '3')
covered = 'true';
else if (document.getElementById('ans'+k).value !=
```

```
'-1')
covered = 'false';
}
}
else if (current == '10') {
for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '0')
covered = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered = 'false';
}
}
else if (current == '11') {
for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '3')
covered = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered = 'false';
}
}
else if (current == '12') {
for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '4')
covered = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered = 'false';
}
}
else if (current == '13') {
for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered = 'false';
}
}
else if (current == '14') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '4')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'5')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '15') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'3')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '16') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}

else if (current == '17') {
for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '4')
covered = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered = 'false';
}
}

else if (current == '18') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '2')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'5')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
```

```
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '19') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '0')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'5')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '20') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '0')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'3')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '21') {
for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '5')
covered = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered = 'false';
}
}
else if (current == '22') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'0')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '23') {
for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '2')
covered = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered = 'false';
}
}
else if (current == '24') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '0')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'1')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'3')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '25') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '4')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'5')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
```

```
}
else if (current == '26') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '3')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'0')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '27') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '5')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'4')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}

else if (current == '28') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'0')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '29') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'4')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '30') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'4')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '31') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '5')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'3')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'0')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '32') {
var covered1 = 'false';
```

```
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'3')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '33') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '5')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'0')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '34') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '5')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'4')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}

else if (current == '35') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '36') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'3')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'0')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '37') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '4')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '38') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';
```

```
for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'4')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '39') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '5')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '40') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'3')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'0')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}

else if (current == '41') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '5')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'3')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}

else if (current == '42') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'4')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}

else if (current == '43') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '5')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'3')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'0')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '44') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
```

```
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '45') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'4')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}

else if (current == '46') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'3')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '47') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'3')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'0')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '48') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '0')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'3')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}


else if (current == '49') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '5')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'4')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}

else if (current == '50') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
```

```
if (document.getElementById('ans'+k).value == '5')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '51') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '5')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'4')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}

else if (current == '52') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'4')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '53') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '54') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '5')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'0')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '55') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '5')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '56') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '5')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
```

```
'3')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '57') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '58') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '3')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'0')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '59') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'3')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'0')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '60') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'3')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'0')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '61') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '5')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '62') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'4')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
```

```
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '63') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '5')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'3')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '64') {
for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '0')
covered = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered = 'false';
}
}
else if (current == '65') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '4')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'0')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '66') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'true';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'0')
covered2 = 'true';
else if (document.getElementById('ans'+k).value !=
'-1')
covered3 = 'false';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}
else if (current == '67') {
var covered1 = 'false';
var covered2 = 'false';
var covered3 = 'false';

for (k = 1; k <=3; k++) {
if (document.getElementById('ans'+k).value == '1')
covered1 = 'true';
else if (document.getElementById('ans'+k).value ==
'4')
covered2 = 'true';
else if (document.getElementById('ans'+k).value ==
'2')
covered3 = 'true';
}
if (covered1 == 'true' && covered2 == 'true' &&
covered3 == 'true') {
covered = 'true';
}
}


if (covered == 'true')
corr++;
else
wro++;

if (current == '67' || current == '68')
window.location = 'results.php?corr='+corr;
else
window.location = 'index.php?
corr='+corr+'&timenow='+timeoutat+'&wrong='+wro;


}


   function deleteall() {
for (i = 1; i <=3; i++) {
document.getElementById('ans'+i).src = 'none.png';
}
for (j = 0; j <=5; j++) {
document.getElementById('b'+j).style.display =
'block';
}
pos = 1;
   }
```

```
</script>

<!-- results.php-->
<html>
   <script>
     function restart() {
     window.location = 'index.php?lo=1';
     }


   </script>

   <head>
       <meta http-equiv="Content-Type" content="text/
html; charset=UTF-8">
       <title>RESULT Page</title>
   </head>
   <body>

   <h1>RESULTS</h1>

   <br>
   <br>
   <br>

   <?php
   $correct = 0;
   $wrong = 0;

   if ($_GET["corr"] != '')
       $correct = $_GET["corr"];
   if ($_GET["wrong"] != '')
       $wrong = $_GET["wrong"];

$totalscore = $correct - ($wrong/4);
       ?>
   Your Orientation Skill score is: <?php
echo($totalscore);?>
   <form name="theform">
       <input type="button" id = "admin"
name="admin" value="restart" onclick = "javascript:
restart();"/>
   </form>
   </body>
</html>
```

D.2 Guilford-Zimmerman Orientation survey

The following test, is commonly used for determining a person's orientation skill. Although it is very commonly used, it is relatively difficult to find since the publisher is no longer in business. Therefore, we are providing it for future research in the appendix section of this thesis.

The Guilford-Zimmerman Aptitude Survey

Part 5/Spatial Orientation

Name ____________________ Date ______ Score ______ Sex: M F

INSTRUCTIONS.
This is a test of your ability to see changes in direction and position. In each item you are to note how the position of the boat has changed in the second picture from the original position in the first picture.

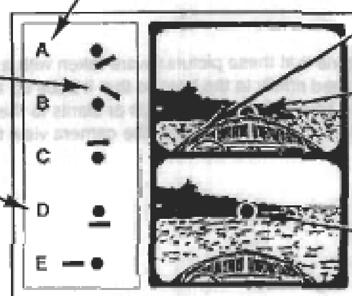
Here is Sample Item 1.

These are the five possible answers to the item.

These bars represent the boat's prow.

This is the correct answer. It shows that the prow of the boat has dropped below the aiming point.

(If the prow had risen, instead of dropped, the correct answer would have been C, instead of D.)



Sample Item 1

This is the prow (front end) of a motor boat in which you are riding.

This is the aiming point. It is the exact spot you would see on land if you sighted right over the point of the prow.

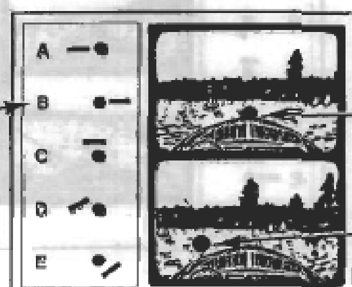
This is the same aiming point shown above. Note that the prow has dropped below it.

To work each item: First, look at the top picture and see where the motor boat is headed. Second, look at the bottom picture and note the CHANGE in the boat's heading. Third, mark the answer that shows the same change on the separate answer sheet.

Try Sample Item 2.

This also shows that the prow of the boat is to the right of the aiming point. So, *it is the correct answer.*

(If the boat had turned to the left, instead of to the right, the correct answer would have been A.)



Sample Item 2

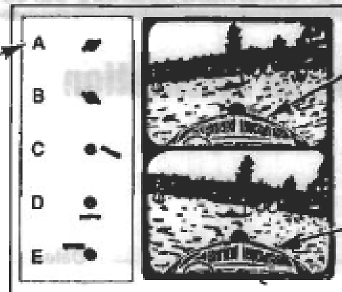
This is the aiming point.

This is the same aiming point. The motor boat is now headed to the right of it.

Consulting Psychologists Press, Inc., 3803 E. Bayshore Road, Palo Alto, CA 94303 0039
98 97 96 95 94 8 7 6 5 4

Now try Sample Item 3.

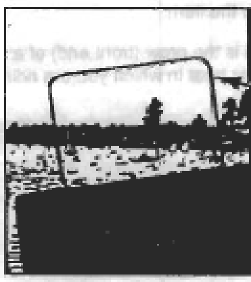
This is the correct answer. It shows that the motor boat changed its slant to the left, but is still heading toward the aiming point.



Sample Item 3

Here the motor boat is slanted slightly to the right. (Note that the horizon appears to slant in the opposite direction.)

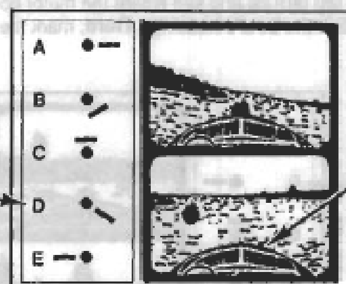
Here the boat has changed its slant toward the left. (To become level, the boat slanted back toward the right.)



Imagine that these pictures were taken with a motion picture camera. The camera is fastened rigidly to the boat so that it bobs up and down and turns and slants with the boat. Thus, when the boat tips or slants to the left (as in the lower sample in SAMPLE ITEM 3), the scene through the camera view finder looks slanted like this.

Look at Sample Item 4.

D is the correct answer. It shows that the boat changed its heading both downward and to the right; also that it changed its slant toward the right.



Sample Item 4

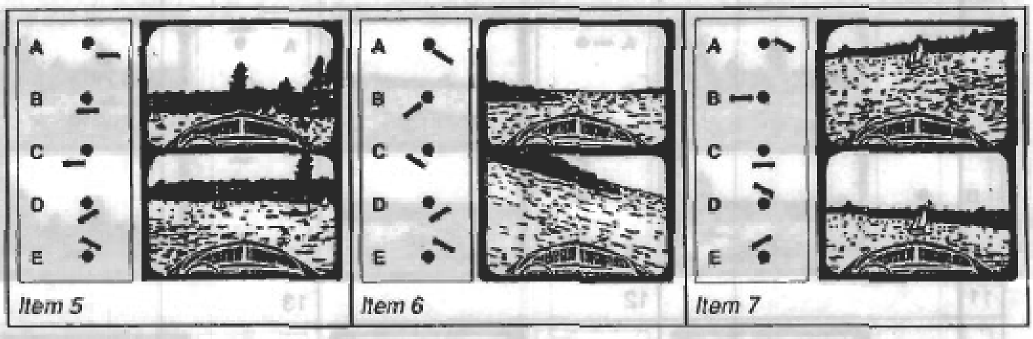
The prow of the boat has moved downward and toward the right. Also, it has changed its slant toward the right.

Now do Practice Items 5, 6, and 7. Record your answers on the separate answer sheet.

The aiming point is not marked in the test items. You must see the change in the boat's position without the aid of the dots.

To review:

First – Look at the top picture. See where the motor boat is headed.
Second – Look at the bottom picture. Note the change in the boat's heading.
Third – Mark the answer that shows the same change (in reference to the aiming point before the change).



C is the correct answer. The prow appears to have moved to the left and downward. It has not changed its slant.

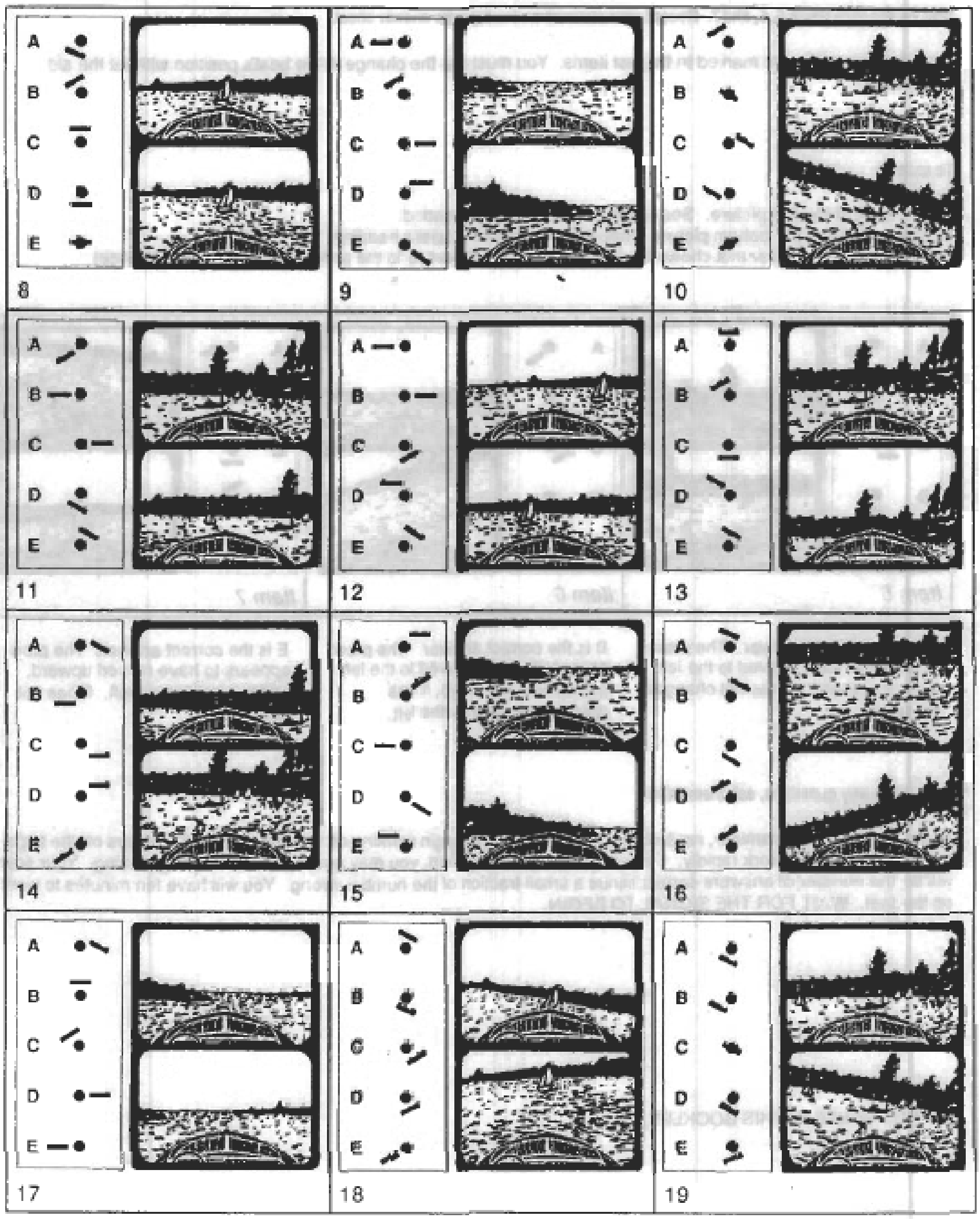
B is the correct answer. The prow appears to have moved to the left and downward. Also, it has changed its slant to the left.

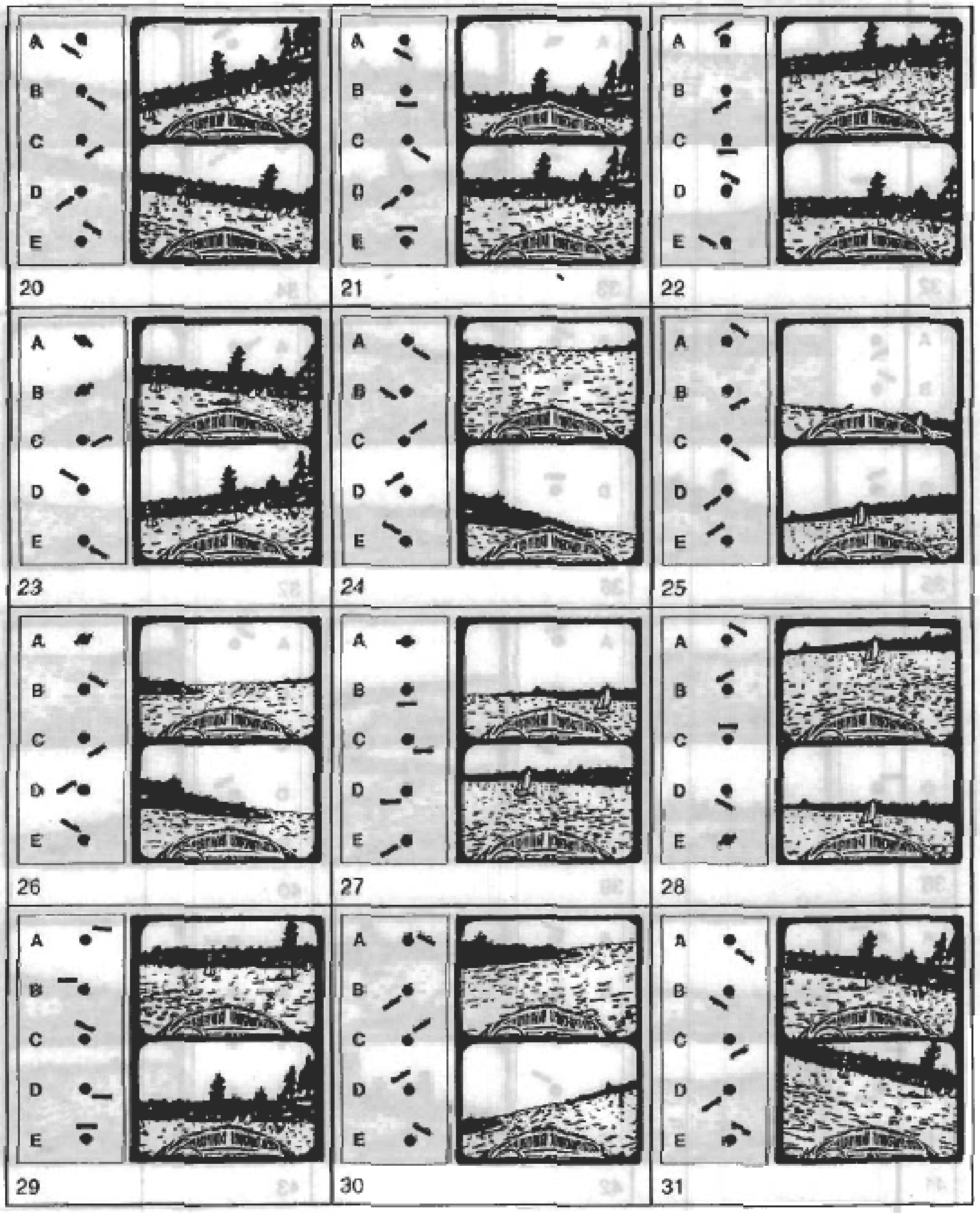
E is the correct answer. The prow appears to have moved upward, and to have tipped left. It has not turned.

If you have any questions, ask them NOW.

At the signal of the examiner, not before, turn the page and begin working on the test. Mark all answers on the separate answer sheet. Work rapidly. If you are not sure of any item, you may guess, but avoid wild guessing. Your score will be the number of answers correct minus a small fraction of the number wrong. You will have ten minutes to work on the test. *WAIT FOR THE SIGNAL TO BEGIN.*

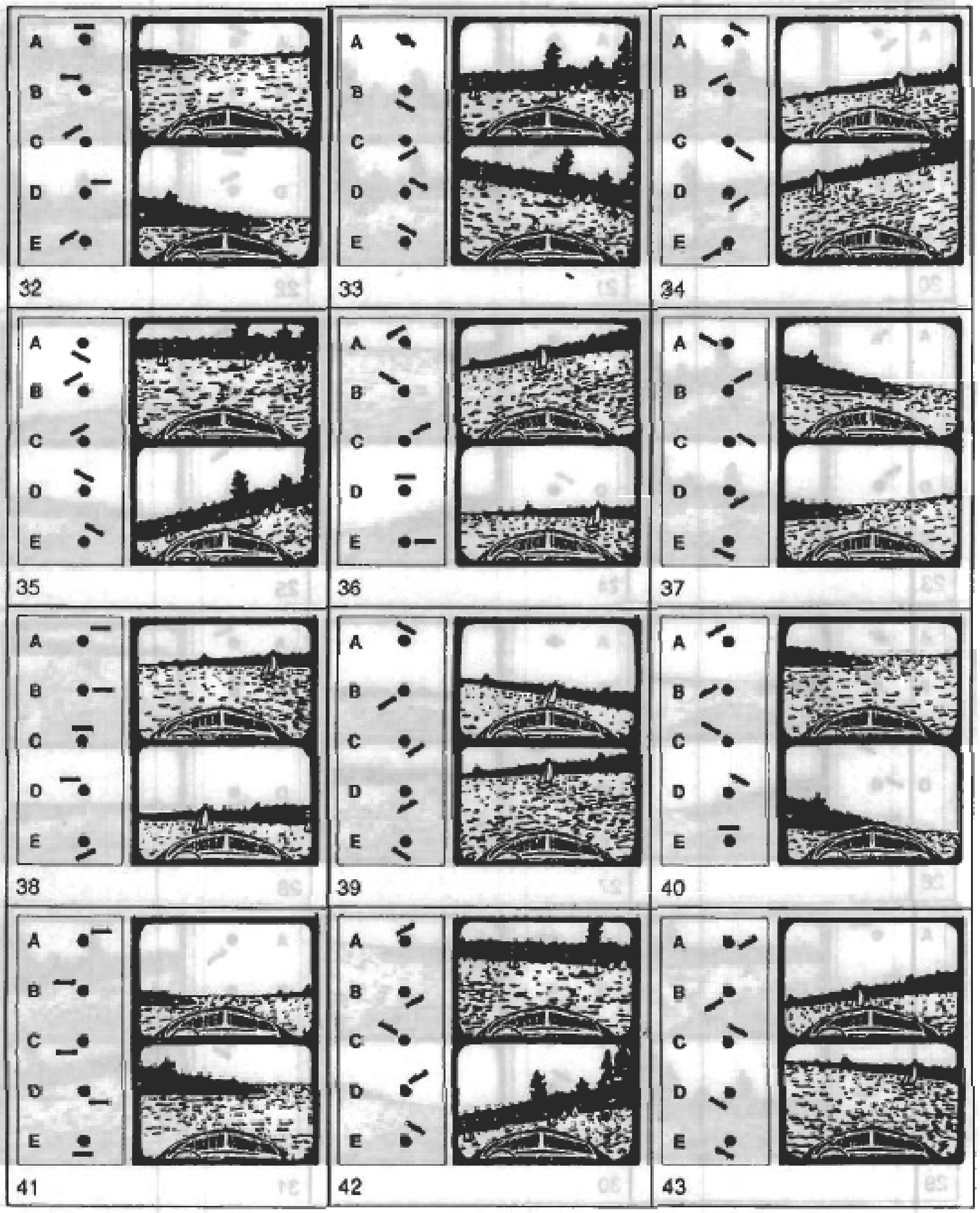
DO NOT WRITE IN THIS BOOKLET.


8
9
10
11
12
13
14
15
16
17
18
19

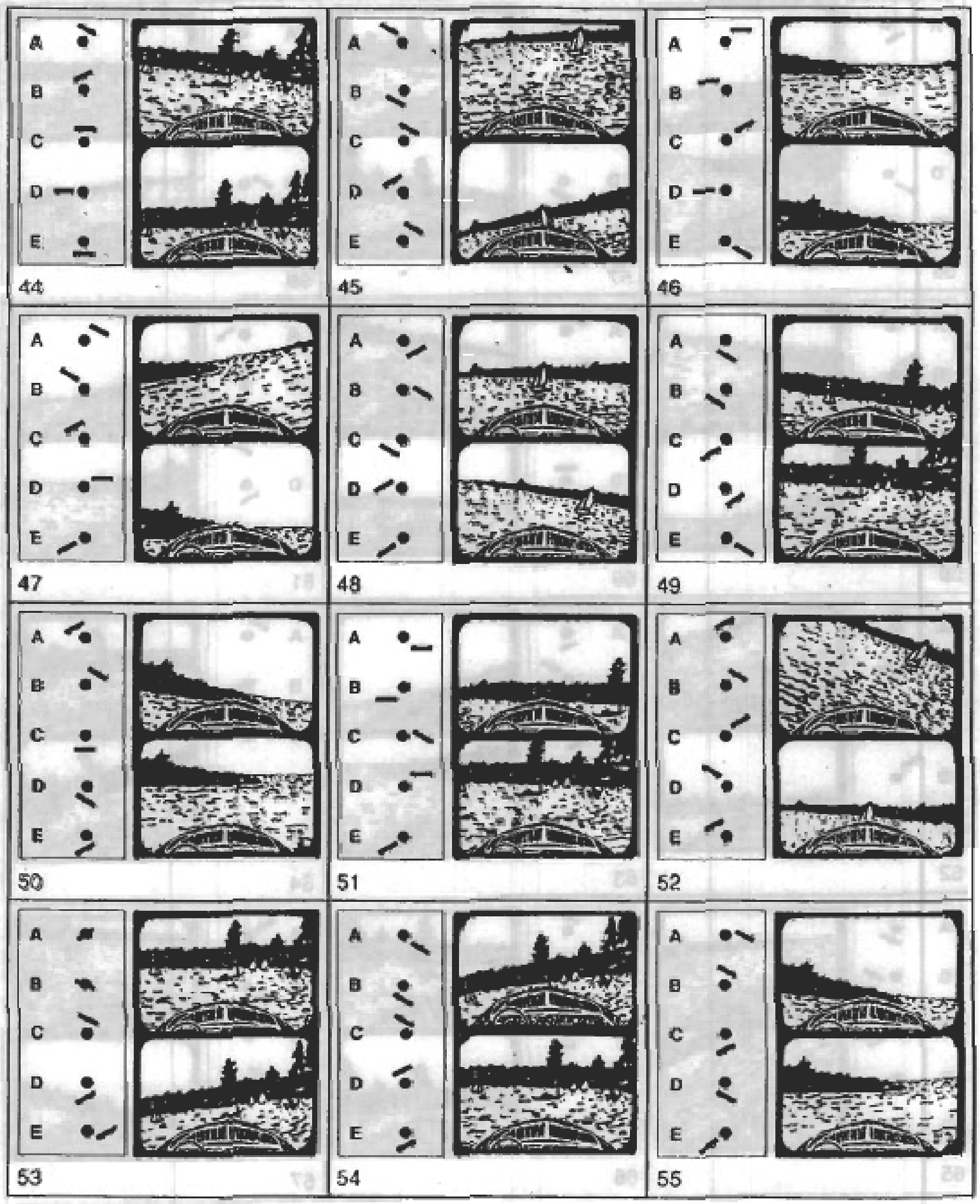

A B C D E
20
A B C D E
21
A B C D E
22
A B C D E
23
A B C D E
24
A B C D E
25
A B C D E
26
A B C D E
27
A B C D E
28
A B C D E
29
A B C D E
30
A B C D E
31

5

(Continue)

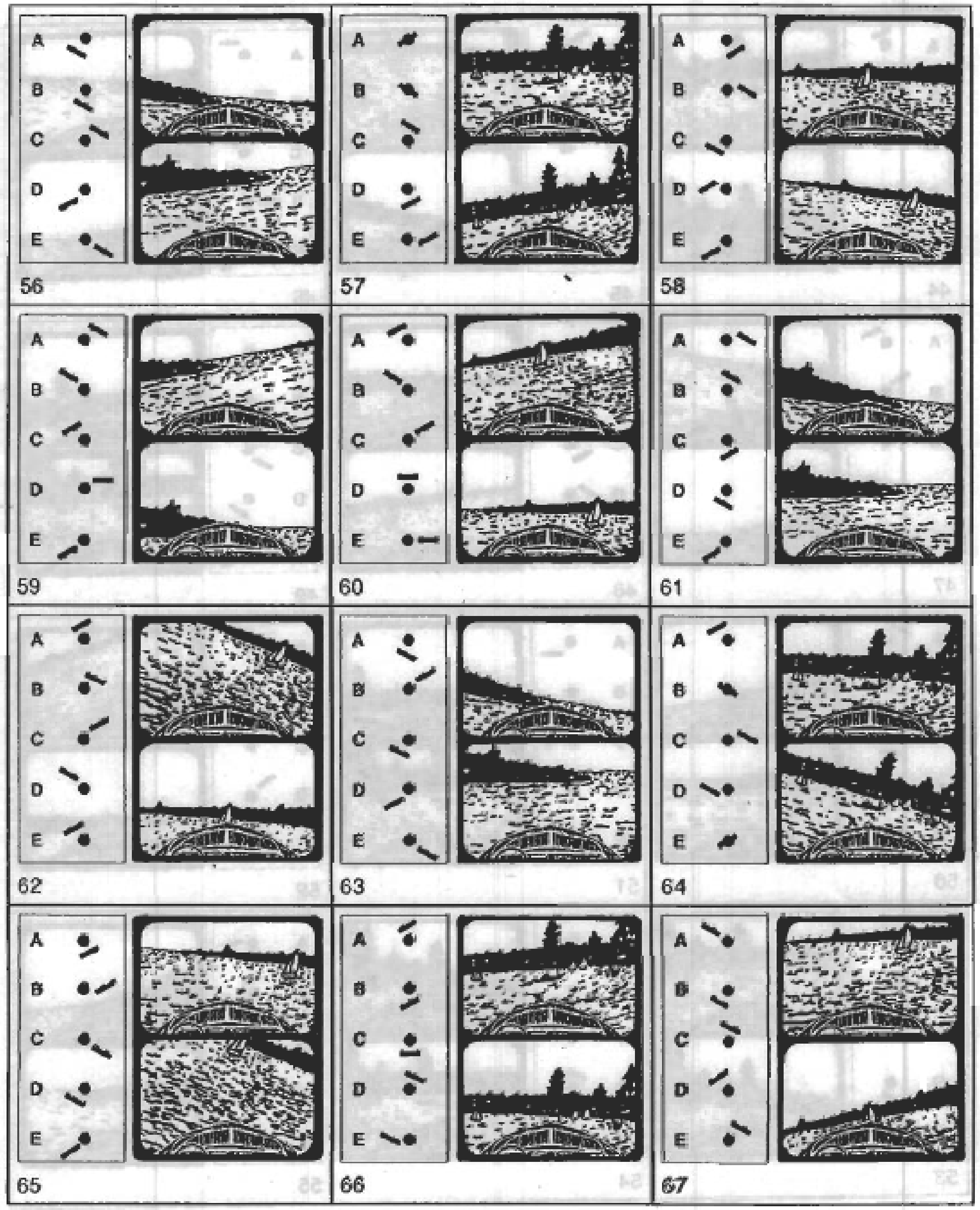


6



7

(Continue)



8

The Guilford-Zimmerman Aptitude Survey Part ____ **Title** ______________________ 0045

Name ______________________ Date __________ Sex ______ Number of years of school completed ______

Organization ______________________ Group __________ Job Title ______________ ☐ Applicant ☐ Employee

Scores: Rights ______ Wrongs ______ Formula Score ______

1 A B C D E	31 A B C D E	61 A B C D E	91 A B C D E	121 A B C D E
2 A B C D E	32 A B C D E	62 A B C D E	92 A B C D E	122 A B C D E
3 A B C D E	33 A B C D E	63 A B C D E	93 A B C D E	123 A B C D E
4 A B C D E	34 A B C D E	64 A B C D E	94 A B C D E	124 A B C D E
5 A B C D E	35 A B C D E	65 A B C D E	95 A B C D E	125 A B C D E
6 A B C D E	36 A B C D E	66 A B C D E	96 A B C D E	126 A B C D E
7 A B C D E	37 A B C D E	67 A B C D E	97 A B C D E	127 A B C D E
8 A B C D E	38 A B C D E	68 A B C D E	98 A B C D E	128 A B C D E
9 A B C D E	39 A B C D E	69 A B C D E	99 A B C D E	129 A B C D E
10 A B C D E	40 A B C D E	70 A B C D E	100 A B C D E	130 A B C D E
11 A B C D E	41 A B C D E	71 A B C D E	101 A B C D E	131 A B C D E
12 A B C D E	42 A B C D E	72 A B C D E	102 A B C D E	132 A B C D E
13 A B C D E	43 A B C D E	73 A B C D E	103 A B C D E	133 A B C D E
14 A B C D E	44 A B C D E	74 A B C D E	104 A B C D E	134 A B C D E
15 A B C D E	45 A B C D E	75 A B C D E	105 A B C D E	135 A B C D E

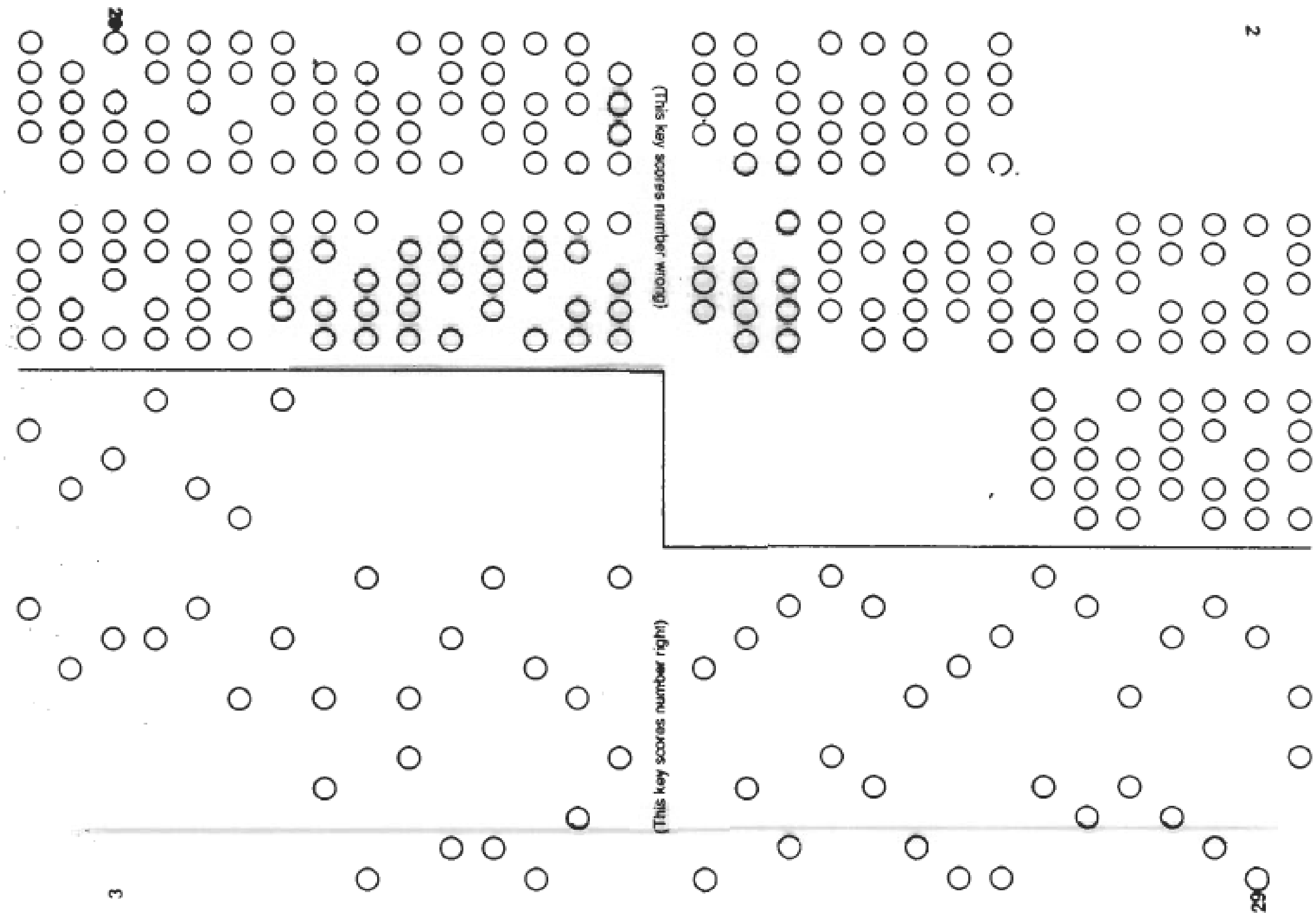
Be sure your marks are heavy and dark. Erase completely any answer you wish to change.

16 A B C D E	46 A B C D E	76 A B C D E	106 A B C D E	136 A B C D E
17 A B C D E	47 A B C D E	77 A B C D E	107 A B C D E	137 A B C D E
18 A B C D E	48 A B C D E	78 A B C D E	108 A B C D E	138 A B C D E
19 A B C D E	49 A B C D E	79 A B C D E	109 A B C D E	139 A B C D E
20 A B C D E	50 A B C D E	80 A B C D E	110 A B C D E	140 A B C D E
21 A B C D E	51 A B C D E	81 A B C D E	111 A B C D E	141 A B C D E
22 A B C D E	52 A B C D E	82 A B C D E	112 A B C D E	142 A B C D E
23 A B C D E	53 A B C D E	83 A B C D E	113 A B C D E	143 A B C D E
24 A B C D E	54 A B C D E	84 A B C D E	114 A B C D E	144 A B C D E
25 A B C D E	55 A B C D E	85 A B C D E	115 A B C D E	145 A B C D E
26 A B C D E	56 A B C D E	86 A B C D E	116 A B C D E	146 A B C D E
27 A B C D E	57 A B C D E	87 A B C D E	117 A B C D E	147 A B C D E
28 A B C D E	58 A B C D E	88 A B C D E	118 A B C D E	148 A B C D E
29 A B C D E	59 A B C D E	89 A B C D E	119 A B C D E	149 A B C D E
30 A B C D E	60 A B C D E	90 A B C D E	120 A B C D E	150 A B C D E

0027

The Guilford-Zimmerman Aptitude Survey
Part 5/Spatial Orientation

Scoring Formula: number right minus 1/4 number wrong



The Guilford-Zimmerman Aptitude Survey

Name ______________________ Date __________ Sex ______

Organization ______________ Group ______________ Job Title ______________

% tile	T Score	College Men VC	College Men GR	College Men NO	College Men PS	College Men SO	College Men SV	College Women VC	College Women GR	College Women NO	College Women PS	College Women SO	College Women SV	T Score	% tile
99.8	80			128		50	39	68	22	112		37		80	99.8
		67 64	26	120	72 70	46	36	63	20	103	72	34	24		
97.7	70	58	24	110	65	41	32	56	18	94	68	30	21	70	97.7
		51	21	100	60	36	28	50	15	85	63	26	18		
84.1	60	45	18	90	56	31	25	43	13	76	58	21	15	60	84.1
		38	15	80	51	26	21	37	11	67	53	17	12		
50.0	50	32	12	70	46	21	17	30	9	57	48	13	9	50	50.0
		25	10	60	41	15	13	24	6	48	43	8	6		
15.9	40	19	7	50	36	10	10	18	4	39	39	4	3	40	15.9
		12	4	40	31	5	6	11	2	30	34	0	0		
2.3	30	6	1	30	26	0	2	5	0	21	29	-5	-3	30	2.3
			-1	20	21	-5	-2			11	24				
0.2	20			15	17					4	21			20	0.2
Raw Scores		____	____	____	____	____	____	____	____	____	____	____	____		

Notes:

 0047

The Guilford-Zimmerman Aptitude Survey

Name ____________________ Date ________ Sex ______

Organization ______________ Group ______________ Job Title ______________

		College Men						College Women							
% tile	T Score	VC	GR	NO	PS	SO	SV	VC	GR	NO	PS	SO	SV	T Score	% tile
99.8	80	67		128	72	50	39	68	22	112		37		80	99.8
		64	26	120	70	46	36	63	20	103	72	34	24		
97.7	70	58	24	110	65	41	32	56	18	94	68	30	21	70	97.7
		51	21	100	60	36	28	50	15	85	63	26	18		
84.1	60	45	18	90	56	31	25	43	13	76	58	21	15	60	84.1
		38	15	80	51	26	21	37	11	67	53	17	12		
50.0	50	32	12	70	46	21	17	30	9	57	48	13	9	50	50.0
		25	10	60	41	15	13	24	6	48	43	8	6		
15.9	40	19	7	50	36	10	10	18	4	39	39	4	3	40	15.9
		12	4	40	31	5	6	11	2	30	34	0	0		
2.3	30	6	1	30	26	0	2	5	0	21	29	-5	-3	30	2.3
			-1	20	21	-5	-2			11	24				
				15											
0.2	20				17					4	21			20	0.2
Raw Scores															

Notes:

0047