Ready, Steady, Program! How Children can Teach Numeracy to a Robot (and Learn Coding) During STEM School Visit Events.

ABSTRACT

STEM topics are often perceived by secondary school students as boring, difficult and uninteresting. Therefore, the authors designed an event to challenge these perceptions. The opportunity was given during STEM School visits to Brunel University where the authors use a 55 minute event to attempt to convince pupils that STEM and in particular computing is fun. During an interactive sessions where students were encouraged to play with robots, they were gently introduced to the art of coding. The results were an increased confidence in their programming abilities and a better perception of STEM.

Keywords: coding, robots, STEM

INTRODUCTION

Many Western countries face shortages of STEM graduates. This shortage can partially be explained by a lack of engagement by students with STEM topics. These topics are perceived as difficult and not very interesting and these are often among the reasons often given to explain this general reluctance. One way to reverse this trend is to introduce STEM topics in a friendly way and at an early stage of the education. Several initiatives have been promoted to support this idea (see for example (STEM 2014) and (Pierre and Christian 2002)). As discussed in (Šorgo and Kocijančič 2004), computer driven activities are also considered part of a successful strategy for connecting pupils with STEM related topics.

One such example is a scheme at Brunel University where local schools have been encouraged to meet within the University environment. The STEM team brings together school pupils, undergraduate students and academics to develop a pupil understanding of STEM and explore their future potential opportunities. The hope is that children will develop some interest on STEM topics and realise the potential STEM can offer. For this purpose, regular one day STEM events have been organised at the University with the aim of increasing the student’s
familiarity with STEM topics. During these events secondary school pupils participated in various STEM related activities. In particular, one of these activities focuses on computing and the aim of this paper is to discuss this activity in more detail.

THE ROBOT EVENT STRUCTURE

The robot event can be broken down into six steps. Since school pupils are expected to have little or no programming experience then, the various steps discussed below are designed to gradually take them from no knowledge to the final stage of being able to write simple programs.

**Step1. Introduction.** Students are divided into groups and small robots such as the hexbug (Hex 2014) are given to each group. These robots have very limited sensing with a single response and without closed feedback control. For example, they can detect when their antennas run into something or alternatively they can sense loud noises. Thus, when they run into something or hear a loud noise, they make a quarter-turn clockwise. Otherwise, they walk forward in an approximately straight line. Students, by initially playing with these robots, will start to familiarise themselves with basic concepts of robotics, such as sensors detection and control, and they begin to explore the robot functionality. A simple task, such as making the robot move around an obstacle, is defined so that, students will understand concepts such as goal, planning, errors in a very practical way. After a few minutes, the activities will switch into the next phase.

**Step2. Scribbler.** A more complex robot is used during this phase (Scribbler 2014). This robot has several sensors and it is controlled via a laptop connected to Scribbler via a wireless connection. An basic user interface running in the laptop is then used to control the robot. The interface has been designed so that the robot can draw segments on a horizontal whiteboard, placed under the robot. A standard whiteboard marker is vertically attached to the robot shell so that the tip of the pen is in direct contact with the whiteboard. Therefore, when the robot moves, it draws lines on the whiteboard underneath. The user interface allows a student to select a number using the laptop keyboard. Once a number has been selected, the robot will execute a predefined sequence of moves in order to write the chosen number on the whiteboard. The aim of this activity is both to introduce students to the general concept of programming and to test the students ability to identify problems. Indeed, due to some (ad-hoc) program bugs and natural occurring robot real time inaccuracies (slippery wheels, uneven surface, etc.) the sequence of segments drawn on the whiteboard by the robot are likely to be only a resemblance of the number selected by the student. During the discussions following the outcome of this task, the student’s conclusion is that a better program is required and in the next step students are challenged to do so.

**Step3. Coding.** The challenge during this stage is to enhance the pupils confidence on their coding abilities. This is done by gradually introducing them to the art of programming. To write a better program than the one they just tested, students will initially use *Guido Van Robot* (GVR) (GVR 2014), a virtual
robot running on a computer. GVR uses minimalistic programming language providing just enough syntax to help students learn the fundamental concepts of programming (see Figure 1). GVR is a robot represented in a window by a triangle on the computer screen that moves around in a virtual world very similar to the whiteboard used in the previous step. GVR can also set dots (very much like drawing dots/lines on the real whiteboard).

The robot actions are completely guided by a program written by the user on a separate window. The interactive environment allows instant visual feedback since every time the student type a command it is possible to observe the robot reactions (i.e. the triangle movements). There are only three basic instructions: move, turnleft, putbeeper. The virtual robot (i.e. the triangle on the screen) will respectively move one step forward if the move command is typed, turn left 90 degrees if the turnleft command is used and put a circle when putbeeper is used. In other words, the GVR language is very intuitive and with a syntax very similar to natural language.

The simple graphic and reduced cognitive load make very easy and straightforward for children to familiarise with the environment and they can confidently control the robot within minutes of being introduced to this application. At any given time, the list of created instructions can either be executed step by step (by clicking on the "Step" button, see Figure 1) or in block ( by clicking on the "Execute" button, see Figure 1).

As a consequence, pupils naturally start using block of commands to implement sequence of actions. During this stage, a student is invited to the front stage to control the robot by typing commands on the keyboard with the computer monitor being connected to a projector. In this way, everybody can follow and contribute to the task and therefore interaction among all the students is natural and everybody is encouraged to participate to this activity. Indeed, with the help of the whole class pupils will then be able to define the correct sequence of GVR instructions to draw a number on the screen. For example, to draw the number I, a possible sequence (or a program as they will then realise) would be the following: move; putbeeper; move; putbeeper; move; putbeeper; move; putbeeper. As a consequence of this sequence, then the triangle (facing upward) on the screen will move and put circles in the desired shape (in this case, a straight sequence of circles that can be seen as the number 1). Once they are happy with the program they have produced, then the program (i.e. the sequence of actions they have created) will be saved into a file.

During this stage, students familiarised with the task of writing a program by trial and errors. They are quickly realise, for example, the need to alternate moving actions and drawing circles by observing that if an robot movement action is not followed by a putbeeper action, then the robot will move around but there will be no writing on the whiteboard.

The immediacy of the visual feedback on the screen therefore help them to gain confidence in coding and to increase their perception of being able to program and control the robot. The fact that they are using a simulated environment
FIG. 1. The GVR robot. Commands are typed on the right. The robot (represented as a triangle) is displayed on the left. In the example here, the full list of commands has been executed (by either clicking "Step" or "Execute") and the robot is in its final position after having drawn circles (putbeeper) and moved (move, turnleft).

will give them the confidence they can try different options with no physical repercussions on the robot, while the use of the screen being projected on the wall will encourage co-operation.

Step 4. Translating. At this point, the code written into the GVR minimalistic programming language will be translated into a (Python (Python 2014)) program that can be executed by the Scribbler robot. This is done by an ad-hoc program without pupils intervention. However, the produced (Python) program will be shown (using a standard text editor application) to the students. By comparing the Python program displayed on the monitor with the GVR program they have previously implemented, they can see the outcome of their activities and have a basic insight of computer program structures.

Step 5. Execution. Finally, students will be invited to volunteer to launch the (Python) program. This program will be executed by the Scribbler robot. They will be able to monitor that the same sequential order as defined in the GVR program during Step 3 is now executed by the real robot (the Scribbler). That is, pupils have been able to control the real robot by executing the GVR program they have produced and tested (and being translated into a Python one). During this stage, pupils assess the real effects of their actions by observing the robot action.

Step 6. Debriefing. During this phase of the event pupils discuss and identify challenges, to reflect on how to optimise their work and discuss further problems. Discussion on examples of where programs are used in everyday de-
vices.

DISCUSSIONS

The duration of the robot event and the fact that each class will attend one event only, are some of the key restrictions to consider when designing this type of activities. Indeed the challenge is to use a short amount of time to introduce various complex concepts at once. Nevertheless, the duration of the event is such that pupils generally stay focus and concentrated for the full session.

The (almost) lack of prior programming knowledge is another aspect to be taken into account. Therefore, while designing the event, it was decided to focus on two key aspects of programming: the fact that, at its core, computer programming can be considered as a form of problem solving and that coding is the process of formulating a description of a method (or a set of instructions) to achieve a goal. In this way, the bite-size approach adopted here can successfully aim at increasing the student’s perception of being able to program rather than aiming at teaching core programming concepts.

In turn, this reinforces their idea that programming is the convergence of three factors: problem solving awareness, the ability to develop suitable strategies to tackle the identified problem and the acquisition of the technical knowledge necessary for the task. The hope is that, once their coding confidence has been raised in this way, they would be more inclined to engage with more formal aspects of programming (i.e. core programming aspects) in future. In other words, by hinting that programming can be seen as the ability to build and implement a plan, then it is possible to convince them that the art of coding can be reduced into breaking down programming into something that can be managed by simply following the three factors mentioned above. However, taking the students through this process in a very practical and realistic way is crucial. Indeed, the final demonstration of the robot moving on the whiteboard as they had originally planned is the concrete evidence that they naturally possess these qualities. Their confidence would also being reinforced by the fact that, even with very little or no prior preparation and knowledge of the topic treated, they have still been able to manipulate and handle coding aspects without major problems.

It also must be noted that often towards towards the end of Step4, students inquired about how to make the program more efficient. For example, they started thinking of strategies to avoid or minimise repetitions. That is, once they felt confident with the basic task, they spontaneously started thinking of core programming strategies such as looping to optimise their implementation.

EVALUATION

About 30 robot events have been run so far with about 15-20 school pupils attending each session. In general pupils have managed to finish the assigned task successfully. They expressed great satisfaction in being able to both improve and control the robot behaviour. Their participation and engagement during the event has been very enthusiastic. A qualitative analysis of questionnaire feedback obtained from the students after their visit has also been showing positive
responses. However, more quantitative evaluations are needed to assess the outcome in a more reliable way to confirm these initial results.

Plans could also include collecting data about their future academic choices to establish any possible correlation. These outcomes have also been supported by the feedback from accompanying teachers, suggesting that more involvement of the teachers during the event could be beneficial for both the event and the teachers.

Finally, an expanded version of this event based on involving more than one school at once and adding a competitive element is currently being developed at Brunel University by the authors (AdoptaBot 2014).
REFERENCES