Invoking girls in programming

by VAL CLARKE
Division of Cognitive Science and Psychology, Deakin University

Girls are under-represented in both formal and informal computing activities. Their interest can be stimulated by teaching programming skills using project work focused on the modification of pre-constructed games programs or the creation of new games programs. Games oriented activities have intrinsic appeal and immediate relevance, making class activities fun, exciting and student directed. The use and availability of pre-constructed games are discussed and the steps involved in introducing turtle graphics programming through the creation of simple games are described.

Teachers, school administrators and parents are concerned about girls’ lack of participation in many computing activities (The Computing Teacher, April 1984, April 1985). Boys outnumber girls in enrolments in secondary level computing courses in the U.S. (Kolata, 1984), the U.K. (E.O.C., 1983) and Australia (Firkin, 1984). At tertiary level, Australian data show that more men than women are enrolled in computing courses (Kay et al., 1986; Porter and Pirie, 1986) and more men plan to pursue their computing studies to honours and postgraduate levels (Clarke and Chambers, 1987).

Girls are also under-represented in non-school computing activities. Girls are less likely to enrol in computing summer schools and computing camps (Muir and Hess, 1984); this gender based disparity increasing as the cost or the grade level of the course increases. Girls are less likely to belong to computer clubs (E.O.C., 1983) and to own and use home computers (Acorn, 1983). Where girls do participate, their interests lie in word processing and office applications, whilst boys are more likely to enrol in programming courses (Lockheed and Frakt, 1984). If there is any truth in the claims that involvement in computing activities can increase children’s thinking skills (Papert, 1980) the increased participation of girls in a broad range of computing activities becomes an important goal.

Research shows that girls are more likely to choose and enjoy technical and scientific subjects if the material is presented in a context which emphasises its social relevance (Head, 1982; Omerod, 1979). Whereas boys will solve a problem because it provides a challenge, girls need to see its relevance to everyday life. One way to extend girls’ interests in computers from word processing and office applications into programming is to provide programming activities with obvious social relevance. Games provide an excellent medium to achieve this end. Games also have an intrinsic interest.

Children can be introduced to the idea of writing games programs either through the provision of tool kits of pre-constructed games, or they can be introduced to programming concepts through the creation of their own games programs.

Tool Kits
Students are given a tool kit of procedures designed to develop specific skills and provide programs which can be examined and explored. Initially, the programs are treated as “black boxes”. They are used to learn particular ideas or develop particular sets of skills. For example, they may be designed to develop familiarity with the computer and a particular set of commands from a given programming language, as well as to develop a defined set of skills or mastery of specific concepts. Once children are completely familiar with the use of the programs, and their understanding of the computing language has reached an appropriate level, the “black box” procedures can be opened, displayed and studied. Children can examine the way the code is written, modify it to make it more consistent with their own needs or interests, extend it, or use ideas derived from it to develop other programs. As exploration of these tool kits continues, students develop an understanding of the construction of them, which, in turn, creates an interest in exploring the capacities of the language and its applications to other activities.

This method of learning is essentially formalising a style of learning which is evident throughout the non-classroom computer culture. Many computer addicts have their initial contact with computers through a wide variety of arcade games (Reinecke, 1983: Keisler, Sproull and Eccles, 1983, 1984). They play these games extensively until, becoming bored with the pre-constructed games, they seek ways to crack the code and modify the original programs. Through code breaking and piracy, many computer users gain a fairly comprehensive understanding of the operation of both the hardware and the...
software. However, only a section of society engages in this experience. Most arcade games are designed by middle-class, white, male programmers to appeal to their middle-class, white sons (Edwards, 1985). This leads to the development of programs which appeal to males and to the creation of a male image of computers, computing and video arcades, as well as to the development of computing skills among this particular section of society (Kiesler, Sproull and Eccles, 1983, 1984). This male image can be neutralised, and the benefits of these learning opportunities retained, by the provision of similar learning situations, designed to appeal to a wider audience, and made available to a broader spectrum of students. This is achieved by designing tool kits which are equally appealing to both girls and boys, and which involve activities that can be integrated into the school program. Such a tool kit has been developed to introduce primary school children to Logo turtle graphics.

**Thinking with Logo**

Our initial games (Clarke, V.A. and Chambers, S.M., 1985) were designed to introduce children to the introductory Logo commands, FORWARD, BACK, LEFT, RIGHT, PENUP, PENDOWN and LOAD, and to develop particular skills such as an understanding of the turtle’s left and right turns, estimation of distances in turtle steps, estimation of angles and the use of X, Y co-ordinates. These games also provide relatively simple programs which the children can modify and explore.

Games designed to introduce the Logo commands provide a simple drawing (or target) on the screen and ask the player to move the turtle to the target or around the target. For example, the computer might display a drawing of a lake and ask the player to move the turtle into the lake or “take the turtle for a swim”. If the lake and the turtle are randomly positioned, primary school children will repeatedly play the game, trying to improve their performance, until they can use one turn and one move to reach the lake.

Alternatively, they can be presented with a path and asked to move the turtle along the path to the front door of a house. Children can compete with themselves to find the smallest number of commands needed to reach the goal.

To encourage children to estimate angles accurately they can play “target”. The screen displays a randomly positioned target and instructs the player to pretend it is a shooting event. The player tries to hit the target using a single turn and a single move. If the player is successful a bell rings. If the player is unsuccessful, the turtle can be returned to the starting position, enabling the player to try again, using the previous attempt to guide the next estimate.

As suggested earlier, these games programs can be used at three levels. At the first level the game is presented as a “black box” procedure. The children load the procedure from a disk, display the instructions on the screen and play the game using the introductory Logo commands. At the second level they take the lid off the “black box” and look at the procedures, making simple changes to colours, shapes and instructions, or they can take ideas from more sophisticated games to develop less sophisticated ones. It is not necessary for children to fully understand all aspects of the programming to make minor modifications. However, by successfully modifying existing procedures, children develop an interest in programming and feelings of competence and confidence in their own ability to use computers. At the third level, children create their own new games. Drawing on the ideas underlying the pre-constructed games and the knowledge acquired from playing with these procedures, children create new games which are then played by other groups of children.

**Creating Games**

Some teachers prefer to create their own games with their students rather than to use pre-constructed games. A project which focuses on the construction of a game provides a context in which teachers can introduce new programming concepts and commands in response to student’s needs. Commands and concepts which are introduced in response to student needs, rather than by curriculum requirements, are generally more readily learned and integrated with existing knowledge. Once children have become familiar with the introductory Logo commands, they can be divided into pairs or groups and asked to draw their own screen display, and position the turtle to develop their own simple game. They can suggest the shape of their own target, or the teacher can provide ideas. For example, they might be asked to draw a square to represent a swimming pool, or a house the turtle can visit. Working step by step in immediate mode, a group of children might draw a pool and position the turtle using a simple set of Apple Logo commands:

```
FORWARD 40
RIGHT 90
FORWARD 40
RIGHT 90
FORWARD 40
RIGHT 90
PENDOWN
LEFT 90
FORWARD 110
RIGHT 90
PENDOWN
```

After a child or group of children has drawn the game, another child can play it. Children can vary the size of the pool and the position of the turtle. If they have coloured screens, they can draw different coloured pools.

In response to a child’s question “Can I colour in my pool?” the teacher can introduce the REPEAT command and abbreviations, making the command look something like:
The greatest difficulty experienced by children drawing in immediate mode is that the commands must be re-typed each time a child plays the game. When children complain about the continual re-typing, the teacher can introduce the notion of procedures, showing how the game can be broken into sub-procedures, and how to save these procedures on disk.

For example, there might be a main procedure PLAY which is made up of the procedures POOL, which draws the pool; PLACE, which positions the turtle and DIRECTIONS, which tell the player what to do. The procedures could be:

```plaintext
SETPC 5
REPEAT 20 [FD 40 RT 90 FD 1 RT 90 FD 40 LT 90 FD 1 LT 90]
PU
LT 90
FD 110
RT 90
FD 180
PD
SETPC 1
```

Looking at the procedure TO POOL, CLEARSCREEN clears the screen and starts the turtle in the home position in the centre of the screen. SETBG 0 sets the background to black and SETPC 5 sets the pencolour to blue ready to draw the swimming pool. HT hides the turtle which makes it draw more rapidly. RT 90 FD 50 LT 90 positions the turtle. The REPEAT statement draws the pool. The procedure PLACE, lifts the pen, positions the turtle ready for the player to begin, puts the pen down, sets the pencolour to white and shows the turtle. DIRECTIONS instructs the player.

When children ask about variations to the game, they can be introduced to RANDOM. For example, the lines:

```plaintext
RT 90
FD 50
LT 90
```

in TO POOL could be changed to:

```plaintext
RT 90
FD RANDOM 50
LT 90
BK
RANDOM 60
```

Similarly, the commands in POSITION could take RANDOM inputs.

This game could be further developed to allow for a random selection of the colour, size and shape of the pool. Counters can be added to record and report the number of moves taken to complete the game. A bell can ring when the turtle reaches the target. Then all these programming ideas can be used to create more games.

With a little imagination, children can have fun creating their own games whilst learning new commands and concepts. For the sake of brevity a single example has been presented, but a number of games can be developed in parallel, each game being extended as new ideas are requested.

**Conclusion**

Whether teachers prefer to use pre-constructed games, to create their own games or to use both approaches, the use of games as a medium to teach programming should provide the context which makes the activity more attractive to girls. To the extent that teachers can introduce concepts in response to student needs or encourage children to explore and play with pre-constructed procedures, the learning becomes child-directed, making it more readily integrated with existing knowledge, and relevant to the child.
reflective. The emphasis is on an integration and synthesis of theory and practice over a period of time, rather than on each as discrete, preferred approaches.

- Negotiation as to what activities are important. Knowledge can be evaluated in cultural, political and personal terms. That is, Henry Lawson may occupy an important place in Australian literature, but have no attraction to a particular eleven year old. Computers may be important to Australia’s economic future, but be seen as irrelevant to an aspiring fifteen year old professional golfer. What really counts, is whether the activity makes sense to the individual.

- Negotiation as to how activities are to be conducted. The teacher does not withhold knowledge or information, but facilitates the investigation, preferably as a team member. If information is required to conduct an enquiry, it is provided if appropriate, or the means of finding it provided if appropriate. True enquiry occurs only when the teacher does not possess truth and must engage with students in experimentation, trial and error, reflection, hypothesising, analysing, risk taking, using imagination and creativity.

The concept of negotiation introduced above, involves the notion that students heighten their understanding when they are motivated to learn, when the issue being investigated is important to them and when the environment respects their capacities as learners. The degree of control that learners have over their own learning in the school setting is central to the fidelity of enquiry.

Continued overleaf