Design of Programming Activities for Children

Ylva Fernaeus
Department of Computer and Systems Sciences
Stockholm University, KTH
Forum 100, 164 40 Kista, Sweden
+46 8 6747468
ylva@dsv.su.se

ABSTRACT
This paper describes my specific research interests within the domain of children's programming within an animated programming environment.

INTRODUCTION
My PhD studies are funded by two research projects, and my thesis will hopefully fit somewhere in between the scope of these. The first project, Weblabs, is an EU project in which we will investigate new ways for European children aged 10-14 to collaboratively work with scientific ideas. Ken Kahn, the inventor of Toontalk (Kahn 1996), will have an active role within this project, and an important task will be to extend and adapt the functionality of Toontalk to fit this purpose. The second project, Spläsh, is a collaboration between DSV and SITREC (Stockholm international toy research center) at KTH, and has a focus on activities were children design, construct and modify their own and each others' computer games. Both projects build upon the work of the Playground project (Hoyles and Noss 2001), where a platform for simple game construction activities was developed in ToonTalk. Hence, my thesis will concern children creation of their own animated computer programs, such as simple games, animated fantasy worlds and simulated ecosystems.

Since I started my PhD studies one year ago, I have found certain issues that I would like to focus on within this research area. The purpose of this paper is to specify and give an overview of these matters.

PROGRAMMING AS A METHOD OF EXPRESSION
In the two projects that I am involved in, programming in itself is not the main focus of interest. In the Weblabs project, programs are sent between students in different countries as a method of communication. In the Spläsh project, children develop their own games. In both projects, what students are able to express with their programming activities, compared to making static drawings and written stories, is central. I would like this to run as a central theme in my thesis, i.e. focusing on what students manage to express with programming, rather than how well they manage to complete predefined programming tasks (see difference between *construction* and *creation*). Instead of focusing on children's learning, I would like to focus on what students are able to express with the tools that we provide, with a strong focus on the tools.

DESIGN OF EXAMPLE PROJECTS
Instead of starting to explore programming at the level of codes and algorithms, students may be introduced to the programming environment by using pre-built projects that can be explored and expanded in various ways (see for example Tholander, Kahn et al. 2002).

One of my interests concerns design features that invite for exploration and modification of programming elements in such example projects. The workings of a pong game, for instance, might be easy to grasp, but the game itself offers few obvious ways of improvements that are feasible for the novice programmer. Single-scene adventure games may invite for more modification, but students often get stuck at modifying graphics of the game and never dig into any actual programming. Multi-scene adventure games invite for more changes, but these can be too complex to experiment with as a first exercise. Hence, an important design feature of example games is that they should include potentials for improvements, in ways that are both technically feasible in the programming environment and meaningful for the game play (and for the students themselves). Another important feature is that changes that students make should imply some exploration of the underlying programming elements. All this might seem obvious, but many example projects used for this purpose lack these features, so it might still be a point in exploring these issues further. (see Jakob Tholander's thesis)

An Example Project
This section describes a simple game that we have used to introduce a group of Swedish fourth-graders (10 years of age) to the concepts of programming with ToonTalk. Forty children participated, of which none had any previous experience of programming. The sessions took place in the students' normal classroom with a researcher working with two children at a time during sessions of forty minutes.

The game (Figure 1), was designed so that the player uses the arrow keys to control a female character, and the task is to make her pass two bouncing Vikings in order to reach a food table. One of the Vikings moves significantly slower than the other. When reaching the food table, the computer makes a sound and when colliding with any of the Vikings the player is blown up in an explosion. We also left some space so users could add more objects to the game.

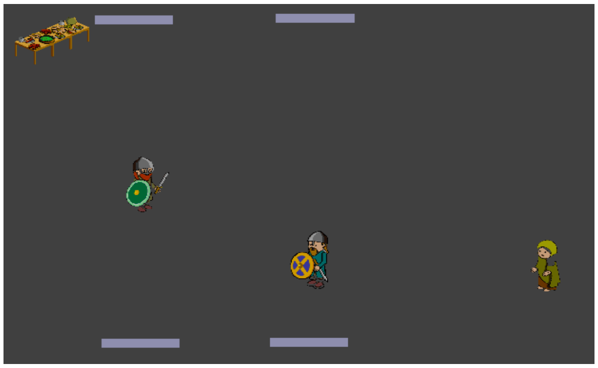

Figure 1. The Viking game

After having played with this game for a short time, most children complained that it was too easy to play and that it would become much better if it could be modified a little. Most pairs initially wanted to make changes to the functionality, but there were also children who started by exchanging sounds and graphic elements of the game. The students quickly seemed to realize that each object was controlled by a number of "behaviours", and that these could be moved, removed or modified in various ways. Most changes that students made required some exposure to the underlying structure of programming mechanisms. Although no real "programming" took place during the session, each pair of children ended up with a fairly unique game, which indicates that the game served its purpose in inviting students to act creatively within the environment and also learn something about programming.

DESIGN OF ACTIVITIES

Another research interest concerns ways of putting programming activities into practice in a more realistic setting. Example projects, as described above, are interesting in themselves but the social setting and the role of the teacher is also important. It is hard to tell what students should accomplish with the Viking game without a researcher introducing the task and coaching them along the way. Having one teacher for each pair of children is not realistic in a real classroom, and we need to study how activities could be designed for larger groups of students.

We are currently working with a group of eight students (which is still far from full class) trying out various ways of arranging programming activities on, as well as off, the computers in a classroom. On the computer, one challenge is to design tasks that are at an appropriate level of difficulty while still taking student's own interest into account. For example, it seems that students are more eager to engage in lower level programming after some exploration of the higher level aspects of program construction (exploring example projects, combining pre-built behaviours, designing and discussing the workings of an own system, etc). In order to get this to work in a real classroom, it could be worth considering development of tools that let students work more autonomously on these activities.

Outside the computer, we are working with role play activities, where we also try to concentrate on ways of understanding higher level aspects of programming mechanisms. For example, each student may represent a simple behaviour (e.g. "move spider down one step" and "move spider to top when reaching bottom") in a mock-up paper game that is laid out on the floor. By physically controlling the game objects, students get a richer experience of the parallel and distributed nature of animated systems, than if just passively studying the system running on the screen.

A more basic issue that we experiment with is how students should work with the computers. We are looking at what happens when students work individually or in pairs, if we make them move between a number of "activity stations" or if we have lectures or let them hold presentations on a large screen. Of course, different settings imply different kinds of activities, but one could also look at how the setting itself affect the work within a group.

Our roles as researchers/teachers is another issue that is impossible to detach from this topic. Some students seem to expect the teacher to instruct them with everything, or even make design choices for them, while we as researchers would like students to work more freely in the environment. The way we act affects the way students act, learn and express themselves, so just learning how to behave with the students is an important task in itself.

CONCLUSION

I have specified some issues that I would like to concentrate on for my licentiate thesis, even if there are many more issues that interest me within this research area. The main theme is that I will try and adopt a top-down, rather than a bottom-up, approach to programming, which means I will concentrate on the behaviour level, rather than algorithm level of programming. The focus will be on tools and activities that support such programming within a classroom-like setting.

REFERENCES

Hoyles, C. and R. Noss (2001). *Playground project description*. *2002*.

Kahn, K. (1996). ToonTalk - An animated programming environment for children. *Journal of Visual Languages and Computing* *7*(2): 197-217.

Tholander, J., K. Kahn, et al. (2002). *Real Programming of an Adventure Game by an 8 year old*. ICLS, Seattle.