Motivating Programmers via an Online Community

Poul Henriksen
Computing Laboratory
University of Kent
p.henriksen@kent.ac.uk

Michael Kölling
Computing Laboratory
University of Kent
mik@kent.ac.uk

Davin McCall
School of Engineering and IT
Deakin University
davmac@deakin.edu.au

ABSTRACT

Motivation is one of the decisive factors in the process of learning to program. This paper describes the *Greenfoot Gallery*, a project created to increase motivation of beginning programmers. The Greenfoot Gallery is a community driven web site that allows the publication of programming projects created with the Greenfoot environment. In this paper, we especially describe and discuss community communication patterns via commenting on the site, and possible effects of these on student motivation.

Categories and Subject Descriptors

D.2.6 [**Software Engineering**]: Programming Environments - *Interactive environments*
K.3.2 [**Computers & Education**]: Computer & Information Science Education - *Computer Science Education*

General Terms

Human factors.

Keywords

Programming education, motivation, community, Greenfoot.

1. INTRODUCTION

The learning and teaching of programming remains a challenging topic in the field of computer science education. Far from being solved, the problems which we struggled with over the years seem to grow faster than the solutions we have offered to address them.

One of the major problems for college and university computing departments over the last eight years or so are the rapidly declining enrolment numbers in our courses. This has been a consistent trend internationally.

Various studies have attempted to analyse the cause of this development [2, 7], and the most consistent factors seem to point to a serious image problem: Computer science is perceived as boring, geeky and not intellectually challenging. People working in the computing industry are viewed as lonely, isolated individuals sitting alone in dark rooms in front of keyboards. Social interaction is not one of the associations that comes up in context of computing as a profession.

This means that the problem of declining interest in our discipline cannot be solved at university level. Students decide against engaging in computer science before they even come to us; an intervention is needed at a much younger age.

Thus, our problem of teaching programming has shifted and grown: Not only are technologies evolving under our feet, but we must now teach computing to younger students, who are originally less motivated, and make it interesting at the same time.

2. PROGRAMMING AT SCHOOL AGE

While computer science is much more than just programming, we firmly believe that programming is one very good entry path into our field. Programming has several immensely useful characteristics in terms of generating interest in beginners: It can be constructive, immediate, “hands-on”, fun, engaging and “real”, while being intellectually stimulating and instructive at the same time. Few other topics of computer science have these advantages.

This argument, however, must be approached with a certain amount of care: Programming does not always or automatically have these characteristics. It can also be introduced in ways that are dull, abstract, theoretical, and where students do not see any practical relevance or personal enjoyment.

In this paper, we shall discuss some challenges and solutions to the problem of teaching programming to a young (high-school age) audience, and especially concentrate on aspects of online communities around programming.

Learning programming can be supported in various way, which we can group into three broad classifications:

- We can provide support for the learning of programming, via educational means;
- we can simplify the mechanics of programming, via use of dedicated programming languages and other tools; and
- we can increase motivation.

All pedagogic interventions fall into one of these groups. Here, we shall discuss mostly the third category: increasing motivation.

One of our reasons to focus on this topic is that the shift in introductory programming teaching from university to high school has caused the greatest disruption in this category. Teachers at university level have a much easier starting point than school teachers – because they have a selective audience. University students have all (in some sense) made a conscious decision to be there. Some interest in the topic of study can be assumed (even though it is not always at a level or of a nature that we would consider ideal). By and large, the students are initially interested, and that makes it much easier for the teacher.

At high school, the situation is different. At the start of computer science or programming courses, many students have already decided that they are not interested in the subject. Prior experiences and hearsay often lead to negative attitudes in many pupils, even before the course starts. Thus, generating motivation is the first thing that is needed in such a course, and this is much

ITiCSE’08, June 30–July 2, 2008, Madrid, Spain.

more important than at university level (although increased motivation is not a bad thing there, either).

Various software tools exist that aim at introducing programming at school age in a highly motivating context. Greenfoot [4], which is at the centre of this paper, is on of them; Alice [1] and Scratch [6] are the other best known tools at this level.

All these systems (and many more) have been documented in the literature. In this paper, we aim to discuss one specific aspect of the Greenfoot project: the online programmer community, supported though the *Greenfoot Gallery* web site [3].

3. ONLINE PROGRAMMING COMMUNITIES

With the dominance of the Web as the preferred internet interface, most communities are now web based (often with RSS and email integration), but still rely on essentially text-based discussion forums. Countless of these forums now exist, for every general or specific topic of programming, and for any experience level.

Specifically, we are interested in the motivational aspect of community. While the main aim of many general programming communities is the provision of technical help, this is only one of two main aims of our Greenfoot Gallery community, the other being motivation. In the case of the Gallery, motivation is achieved by facilitating recognition and encouragement through community interaction.

Community interaction has been proven to be a strong motivator for many teenagers today. Web communities such as Facebook, MySpace and Orkut are among the most popular places on the Web, and many young people spend considerable time and effort on interacting with communities through those sites. The communities here are often friends of the users, known to them from the physical world, so it is not clear that the enthusiasm easily transfers to other communities formed around different focal points. However, other sites form communities of strangers, who meet to discuss specific topics. Examples are Slashdot, Reddit, Digg, countless blogs and fan sites and Google groups, etc. Prior acquaintance does not seem to be a requirement for successful community interaction.

The two most directly relevant systems for comparison to Greenfoot – Alice and Scratch – both provide online support for their user communities as well.

Scratch has a community site similar to the one described in this paper, where users can publish their projects, and others can leave comments [8]. The Alice community forum does not have the ability to upload Alice projects directly. It is, instead, a traditional text-based discussion forum.

For Greenfoot, both exist: There is a traditional online discussion forum in addition to the Greenfoot Gallery, which allows publication of projects and discussion.

4. GREENFOOT

Greenfoot is a learning environment to support the learning of programming (especially object-oriented programming). The language exposed to users is Java, and the system is highly specialised for the development of animated, two-dimensional graphical applications, such as games and simulations. Greenfoot is typically used with students from age 15 upwards.

Using built-in functionality, any project created with Greenfoot can easily and quickly be exported to the Gallery.

5. GREENFOOT GALLERY DESIGN

The Gallery was designed specifically to increase motivation and engagement in programming activities through interactions with a subject-specific community. The goals were to increase motivation for Greenfoot users, and potentially to initiate interest in programming in young people who are not yet programmers.

The fact that motivation and engagement are primary goals in our project leads to some important design decisions for the community site. The most important is the sharing of artefacts.

The Greenfoot Gallery is a public web site that hosts Greenfoot projects. Any Greenfoot user can create an account on the Greenfoot Gallery and then publish their Greenfoot projects.

While within the Greenfoot environment, projects are editable and playable. Once exported to the Gallery, projects are no longer editable, but they are executable for anyone looking at the site.

Thus, the Greenfoot Gallery functions as a kind of "YouTube for Greenfoot projects": It provides a showcase for publishing work and offers options for others to view, rate and comment.

5.1 Live program publication

The central aspect of the Gallery design is the facility to publish working versions of Greenfoot projects directly to the web site. Projects - such as games - can be played directly in the web browser with full functionality. No additional software installations (other than the prevalent Java browser plug-in) are necessary for users of the web site.

This functionality serves two distinct purposes:

- Making their work visible to others on the web is a highly motivating factor for students developing those programs. It can be seen by friends and family, but also by many other people on the web, and many students get a thrill out of receiving feedback from strangers on their work. Examples of student reactions are given below.
- Some young people may interact with the Gallery site initially for the purpose of playing the available games (as pure consumers). Through this, they may start thinking about starting to program these themselves, maybe because they wish to improve certain aspects of a game. The comments discussing implementation and a low hurdle to viewing and manipulating the source code encourage this. The Gallery site is too new to draw conclusions about whether or not this strategy is working, and no systematic investigation has been undertaken. This is an interesting question for future work.

Publication of a project may or may not include full source code of the project, at the choice of the publisher. Publishing the source code supports the second goal described above, and supports learning by example. The choice to withhold source code is given to support use of the Gallery for marked class projects where teachers do not want students to copy each other's solutions.

Students have to create an account on the Gallery to be able to publish projects. Creating an account is a quick and easy process and requires only a valid email address. The email address is not publicly visible on the web site, so users can choose to remain anonymous to the public (but not to the site administrators).

Projects can be played by anyone without creating an account.

5.2 Commenting

Once a project is published on the Gallery site, a separate page is created, where the live project, its author and time stamp and a description is displayed. On this page, other users can leave comments relating to this project.

This commenting facility is, jointly with the publication of the projects, the most important community functionality. The comments serve various purposes: feedback is given on projects, technical questions are asked and answered, suggestions are made, and more. A more detailed analysis of comments is given below.

Thus, the comments support several distinct goals: Firstly, seeing comments on their projects (especially positive comments) greatly increases excitement and motivation for students. Secondly, the comment section can carry the traditional function of online forums to provide help with technical problems.

To increase the feedback for project authors, the number of page views for a project page is also displayed. Through this, authors can also see how many people have looked at their projects even if they did not leave a comment.

5.3 Ratings

Users can also rate projects on the Gallery. The purpose is again to maximise the amount of feedback and motivation that authors get out of the site.

Different rating systems were considered. We decided against a numeric rating and opted for an "I like it" button that users can select for each given project. This way, projects can be singled out positively by community members, and negative judgement simply manifests itself in not selecting this button. Thus, the harshest experience possible for authors is to be ignored, which we consider an acceptable form of feedback.

Each project page presents a count of the number of people who "like it" as an approximate popularity rating.

5.4 Front page presentation

The front page of the Greenfoot Gallery consists of several important sections. Three of these sections present selected projects: The top of the page displays "showcase" projects (manually selected projects that site administrators have chosen to emphasize), below this are dynamically created lists of the most recent and most popular submissions.

All of these are intended to give each project author the chance to receive special attention by being listed on the site's front page. Especially the "most recent" list guarantees that every project is visible on the front page for at least a while.

The front page also includes a summary of the most recent comments on the site, a news section and a tag cloud.

5.5 Further functionality

Many other functionality elements were discussed and several additions are planned to increase the potential for interaction and engagement on the Gallery site, and to increase the potential of generating motivation. Some of these are discussed in Section 8.

6. OBSERVATIONS OF USE

The Greenfoot Gallery site was published in July 2008. Public visibility was initially low. Starting from September 2008, site visibility was increased by linking to it from other Greenfoot web sites and announcing it on Greenfoot related discussion groups.

Between then and mid January 2009, 544 users have signed up to the site, submitting 302 projects and leaving 1311 comments. The web site received approx. 34,000 visits (180,000 page views) by 19,400 absolute unique visitors[1]. The average page views per visit are 5.25 and the average time on site is 5 min, 23 sec. This demonstrates a significant interaction with the Gallery also by users who have not registered as users on the site.

The bounce rate (people who leave the site directly from the page where they entered it) is 33.85%. This supports the observation that most visitors enter into some interaction with the site.

6.1 User engagement

To investigate the frequency of different use patterns, we distinguish four user categories: Visitors, Consumers, Interactors and Creators. Descriptions and numbers of each are given in Table 1.

User category	Actions	Count
Creator	Submit projects	171
Interactor	Leave comments and ratings	18
Consumer	Create user account; no other creation; read and play	335
Visitor	Read and play; no account creation	18900

Table 1: Different user types (higher categories include actions of lower ones)

As expected, the number of visitors is far higher than those of the other categories. Among the users who have created accounts, it is encouraging to see a fairly high percentage of Creators (32.6%).

Creators often submit more than one scenario. The average number of scenarios per Creator is 1.8.

An interesting research question for the future is to investigate migration of individuals from one category into a higher one, and whether such migration can be actively supported.

6.2 Comments

In this paper, we are especially interested in the comments left for published projects, as this provides deeper insight into the different forms of interaction.

A project has a high chance of receiving comments. 81% of all projects have at least one comment by someone that is not the owner of the project. The average number of comments per project is 5.2.

We have analysed all comments left for any scenario that was submitted or updated in a period of 6 weeks between 10 December 2008 and 21 January 2009 (70 scenarios in total).

The first surprising result was that we have found no negative comments. As administrators, we expected to have to occasionally remove inappropriate comments from the site, but this has not been the case. Not only were there no offensive postings in the six months of the site's existence, but the analysis also showed no negative (purely critical or discouraging) comments left for any project.

[1] Data collected with Google Analytics.

When projects were uninteresting or poorly written, the worst that happened was to receive no comments. But even incomplete or poorly written projects often received feedback encouraging improvement of the scenario. The following comment for a project that simply allowed movement of a character is typical:

> *I like the way the animations work. Is this going to be some kind of platform game? It'd look much better if you had to get to the star by jumping on bricks and avoiding enemies rather than just holding the arrow keys down until you get there :)*

We are not certain about the cause for this positive tone on the site, but offer some speculation below.

We evaluated the main purpose and content of each comment and created categories reflecting their focus. The categories chosen are: Encouragement, Technical discussion, Exchanging ideas, Sharing resources, and Miscellaneous.

6.3 Encouragement

44% of all comments were classified as *encouragement*[2]. The simplest form of encouragement consists purely of positive remarks about the project, e.g.:

> *Nice! It's really fun, and the graphics and movements are very nice too.*

Often, the feedback was specific about certain concrete aspects of the project:

> *Great job. I really like the whole game in general. No major problems, perhaps you could have added music. Great images and a nice GAME OVER at the end (which I happen to see alot)!!!*

This feedback was not only useful for the immediate author, but we observed cases where these remarks were picked up by other project authors to get ideas for improving their own projects.

Another form of encouragement includes suggestions for further improvement. This often went together with positive feedback:

> *Great idea for a game. The graphics are very nice now. I especially like the bubbles.*
>
> *Seems a bit hard to navigate from side to side. Maybe you could gradually slow the ship down when no key is pressed?*

Suggestions like these seem to have been successful in creating motivation for the authors. We observed numerous cases where these suggestions were subsequently implemented in the project.

Another interesting pattern we observed repeatedly was the development of a dialogue, where the project author replied to the user comments:

> *Thanks all for the suggestions. All the bugs have been ironed out and it runs quite smoothly.*
>
> *If anyone can think of any new creative ways of killing off the people or any other features let me know and I will try to implement them.*

This demonstrates directly the interaction between project development and community response.

6.4 Technical discussion

Another obvious category were comments asking for or offering technical help (37% of comments). This includes technical questions, bug reports, and technical advice.

Most interesting for us was that help was not only given as a result of direct questions (although that also happened), but frequently unprompted. It seems to have been a common pattern that users studied the source code of a published project, discovered bugs or possible ways to improve it, and offered their advice to the project author. The following comment illustrates this:

> *hey, i discovered this neat trick, java has a way of drawing a scaled image onto another one, so you can actually replace your enemy drawing code with this single line:*
>
> *im.getAwtImage().getGraphics().drawImage(en.getAwtImage( ), (int)(x1-l), (int)(150-l*0.9), (int)(2*l), (int)(2*l),null);*

6.5 Exchanging ideas

In the context of dealing with diverse skill levels of students in a class, it has been suggested that open assignments are a way to cope with this problem [5]: Good students can then develop additional project extension ideas to challenge them at their own level of ability.

We observe that many comments on the Greenfoot Gallery help to support this process by providing ideas for additional project extensions. 14% of the comments fall in this category. The following is a typical example:

> *Pistols shouldn't drop when you die: they clutter up the map a lot.*
>
> *Add a weapon "Sniper Rifle" with low fire rate but a laser line. (This can't be too hard, can it?)*
>
> *Add the option to have movement controls be relative to the direction faced instead of relative to the screen.*

Again, this process works both ways. Comments were also used by project authors to check ideas with users before implementing them:

> *Btw, how tall are your screens? Because right now the game has a resolution of 1024x800. I was thinking about lowering it to 600? Or would that not be enough?*
>
> *Thanks again for your help.*

6.6 Sharing resources

Some comments revolved around the sharing of resources for project development (3% of all comments). Sometimes the resource was a class or other unit of code from the project itself (typically initiated by the potential user of the resource). For example:

> *Hey, do you mind if I use your code for the midi player for my game?*

And the reply:

> *You can use it ;). Anyone can use my code as long as credit is given where it's needed.*

In other cases, the resource in question was external to the source code. In that case, it was sometimes offered by the author:

[2] Some comments contained elements of more than one category. In this case, their were counted in more than one category.

> *i wrote some tutorial in greenfoot in my blog (sorry all, Indonesian language ^_^")*
>
> *this, maybe useful, http://mandelag.blogspot.com/*

We also observed many cases of source code sharing where it was not explicitly discussed through the Gallery comments. Some classes were regularly reused across many different projects.

6.7 Miscellaneous comments

The remaining comments were either categorised as *Social talk* (unrelated to the project development, 2% of comments) or did not fit any of our categories (3% of comments).

7. PEDAGOGICAL CONSIDERATIONS

At this stage it is still impossible to draw firm conclusions from the data available. Especially, it is difficult to make definite statements about the effects of the gallery interactions on motivation and learning. The user group was self-selected, and it is not clear to what extent communication and interaction patterns can be transferred to, for example, a classroom setting.

However, some of the observations are interesting enough to formulate some hypotheses for future investigation.

- It seems that the ability to publish student work may increase motivation for students. This is indicated by a level of engagement with projects that was deeper than what is typically observed in student projects.
- The encouragement received on the web site may help to sustain engagement with the project for a longer time.
- The technical advice provided through the site may help teachers by offering additional sources of help.
- The exchange of ideas may help to challenge especially the good students.
- Sharing existing resources may produce programs that are better in visual appearance or interaction, thus allowing students to learn programming in a context that is closer to their existing expectations of modern software.

Overall, the interaction and communication patterns that have evolved between users point to the Greenfoot Gallery as a promising instrument for improving motivation and engagement in classroom computing courses, as well as for informal learning.

We plan to execute a more formal test of these hypotheses at a later time.

8. DISCUSSION AND FUTURE WORK

One striking observation of communication on the Gallery is the exclusively positive and polite tone of the comments. We speculate that this may have been caused by very slow initial growth. In the first two or three months, the number of subscribers was low, and the Greenfoot development team members presented a fairly high percentage of users on the site, thus able to set a tone on the site and implant culture and expectations. It will be interesting to see whether this carries on as user numbers grow.

Amongst plans for technical development of the Gallery, we consider the addition of personal scenario collections a high priority. This would allow a member of the gallery to create their own named collection of scenarios, into which they can invite scenarios developed by other members. The creator of the invited scenario can then choose to accept or decline the invitation. In a similar way, a scenario creator can ask to have their scenario added to an existing collection, and the collection owner can then accept or decline the addition.

Other than providing an alternative to the "I like it" mechanism for showing which collections a member personally appreciates, we envisage that themed collections could be created (such as a collection for simulations, for instance, or one for "platform" games). Collections might also be used to group scenarios for other reasons, such as to showcase scenarios developed by a particular school or group.

The addition of personal collections adds a new creative facet to the Gallery. It potentially allows non-programmers to become more involved by being able to create something that they "owned", rather than just commenting on other members' scenarios. A popularity system for collections, similar to that already provided for scenarios, would provide a motivation to carefully choose suitable scenarios and to maintain the collection over time.

Collections also have the potential to increase the interactivity between Gallery members. Creating a collection and populating it with scenarios naturally requires establishing a communication between the collection owner and the scenario creators.

Collections may also present an additional motivating factor for scenario developers – that is, to get their scenarios into various collections. Collection owners are free to make whatever demands they wish on the quality and nature of scenarios that are to be part of their collection; this may provide a challenge to scenario developers to meet such requirements, and give them a sense of satisfaction and accomplishment once their scenario has been accepted.

9. REFERENCES

[1] Cooper, S., Dann, W., & Pausch, R., *Alice: A 3-D tool for introductory programming concepts.* In Proceedings of the 5th Annual CCSC Northeastern Conference 2000, Ramapo, NJ, April, 28-29, 2000.

[2] Council of Professors and Heads of Computing (CPHC), *Investigation into the decline in BSc Computing/IT Applications to British Universities*; http://www.cphc.ac.uk/docs/cphc-admissions-trends-report.pdf , July 2006, accessed Jan 2009.

[3] *Greenfoot Gallery*. Web site at http://www.greenfootgallery.org, accessed Jan 2009.

[4] Henriksen, P. & Kölling, M., *Greenfoot: Combining object visualisation with interaction.* In Companion to OOPSLA'04, pages 73-82, 2004.

[5] Kölling, M. and Barnes, D., *Apprentice-based learning.* In Bennedsen, Caspersen, Kölling (Eds.), Reflections on the Teaching of Programming, Lecture Notes in Computer Science, Vol. 4821, pages 29-43. Springer, April 2008.

[6] Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., and Resnick, M., *Scratch: A Sneak Preview.* Second International Conference on Creating, Connecting, and Collaborating through Computing. Kyoto, Japan, pp. 104-109, 2004.

[7] Microsoft Corporation, *Developing the Future– A report on the challenges and opportunities facing the UK software development industry*, http://download.microsoft.com/documents/UK/developingthefuture/Developing%20_The_Future_07.pdf, accessed Jan 2009.

[8] Monroy-Hernández, A. and Resnick, M. *Empowering kids to create and share programmable media.* Interactions, March-April 2008.