Making University Education more like Middle School Computer Club: Facilitating the Flow of Inspiration

Alexander Repenning
University of Colorado Boulder
Department of Computer Science
Boulder, CO 80303
(303) 492-1349, 001
ralex@cs.colorado.edu

Ashok Basawapatna
University of Colorado Boulder
Department of Computer Science
Boulder, CO 80303
(720) 838-5838, 001
basawapa@colorado.edu

Kyu Han Koh
University of Colorado Boulder
Department of Computer Science
Boulder, CO 80303
(303) 495-0357, 001
kyu.koh@colorado.edu

ABSTRACT

The way programming is currently taught at the University level provides little incentive and tends to discourage student peer-to-peer interaction. These practices effectively stifle any notion of a 'learning community' developing among students enrolled in university level programming classes. This approach to programming education stands in stark contrast to the 'middle school computer club' approach; As part of 10 years of research projects aiming to teach programming to middle school children, it is observed that middle school students in computer clubs freely share programming ideas, code, and often query one another and provide solutions to the various programming problems encountered. To enable these interactions at the university level, a novel online infrastructure has been developed over the past 6 years through use in the Educational Game Design Class at the University of Colorado Boulder. The culmination of the submission system, entitled the Scalable Game Design Arcade (SGDA), seems to foster the flow of ideas among students yielding an effective open classroom approach to programming education.

Categories and Subject Descriptors

K.3.2 Computer and Information Science Education

General Terms

Algorithms, Design, Human Factors,

Keywords

University Programming Education, Middle School Programming Education, Scalable Game Design, Open Classroom, peer-to-peer interaction, flow of inspiration, Computational Thinking.

1. INTRODUCTION

The current structure in most computer science classes at the university level resembles the so-called "Sage on the Stage" approach to learning [1]. A single lecturer in the front of the class talks to a group of students who are taking notes. Currently, the emergence of remote and on-demand class viewing obviates the need to physically be in a class wherein a teacher takes this 'Sage on the Stage' approach to teaching. This approach is further indicted by a recent study suggesting that students who on-demand remotely view these types of lectures actually retain more information and get a deeper understanding of the material as compared to students who physically attend the class [8]. Thus, the value of physically attending class is decreased when a teacher takes this approach. Moreover, the 'Sage on the Stage' approach to teaching makes no attempt to use the inherent characteristics of a student-filled classroom to enable a better learning experience. One could argue that with this teaching approach, a lecture wherein one student attends would be identical to a lecture wherein 50 students attend. Given that this is the case, it begs the question, why even have a physical classroom environment? This line of reasoning is unfortunate since the physical student-filled classroom environment lends itself nicely to teaching strategies that foster peer-to-peer student learning and the creation of a learning community within the classroom. Unfortunately, most undergraduate computer science classes make no attempt to create any kind of learning community.

In computer science classes, for example, save for a possible "computer lab", assignments are completed outside of class with little or no motivation or incentive for peer-to-peer interaction [2]. Furthermore, collaborations, sharing of ideas, and looking at fellow students' code is actually frowned upon and often considered cheating; for example, it is not uncommon for teachers to run automatic checks on assignments to ensure everyone's code is strictly their own. This has the unfortunate side effect of inhibiting the proliferation of ideas among students drastically reducing the opportunity for students to learn from one another. Since students are actively discouraged from looking at fellow classmates' work, they are not pushed to do more based on what their peers have accomplished on a given assignment [3]. This has a negative effect not only on individual student achievement, but also, reduces the level of work produced by the class as a whole over the duration of a given course [4]. Some classes try to allow for peer-to-peer interactions through group projects. However, in our experience and as documented by others, group projects often do not yield true group collaborations as students divide the work up and only work together to reassemble the project when everybody's individual part is complete [7]. Thus, most group projects only have limited peer-to-peer interaction and learning. The above issues motivate the following question: what changes can be made such that students effectively learn from and inspire one another in undergraduate computer science classes?

Surprisingly enough, an interesting solution approach to this problem presents itself in the seemingly chaotic structure of middle school computer clubs. For the last 10 years, as part of several NSF funded projects, middle school students were taught the fundamentals of programming using a rapid game prototyping environment called AgentSheets [5]. In our most recent project called Scalable Game Design[1], we are exploring computer science education through game design starting at middle school and moving along all the way to graduate school. In this context we have found that there are interesting educational phenomena taking place at the middle school level that are perhaps worth exploring for more advanced levels [6].

A typical AgentSheets lesson for a given day involves students in front of a computer creating a game. However, because of the relaxed non-classroom atmosphere of an after-school club, students are able to run around to the computers of their peers. Often, upon a student viewing a fellow classmate's game, the following sequence takes place. First, the student notices the classmate's interesting idea or concept and asks that classmate "How did you do that?" The classmate then explains to the student how to accomplish the concept even displaying the "code" used to implement the idea. Social scaffolding, wherein more advanced kids help others, happens naturally in this environment. The two students converse providing feedback on the idea; then, the wandering student would go back to the computer integrating and updating the new code for her/his own purposes. The ability to traverse the class and openly share concepts between classmates enables a network capable of carrying inspiration and influence from one student to the next. All of this appears to work with little, if any, teacher input.

Moreover, this classroom infrastructure allows for viral ideas, wherein one student discovers a concept at first, and that concept soon spreads throughout the class [6]. This flow of inspiration is not limited to simple ideas; A highly sophisticated example of this flow appeared when a student was explained the concept of a collaborative diffusion approach to help him give agents in his game better AI. Though the equations associated with diffusion involve more complex math than is learned at this student's grade level, by the next time the group met, one week later, a group of students had started using diffusion equations in their own games and were explaining the concepts to others.

These middle school experiences motivate the implementation of a university level programming class that allows for all the same communication modes among students. Over the past 6 years an online cyber-infrastructure, through use in the Educational Game Design Class at the University of Colorado Boulder, has evolved to allow for these interactions. The culmination of this evolution is the newly developed Scalable Game Design Arcade (SGDA) and, thus far, the SGDA seems to have successfully allowed for many of the same types of interactions visible in middle school computer clubs.

Section 2 explains the Flow of Inspiration Principles derived from the middle school computer club experiences. Section 3 explains previous approaches to this problem finally arriving at the SGDA implementation. Section 4 provides initial findings of the SGDA, along with a brief discussion, including the results of an in-class questionnaire used to gauge the extent to which the SGDA enables student peer-to-peer learning.

2. FLOW OF INSPIRATION PRINCIPLES

In the limited time, infrastructure and physical space allotted for university programming classes, to integrate the benefits of middle school computer club learning, a different approach must be taken. Thus, the methods employed in the Educational Game Design Class involve transferring the interactions present in the computer club environment to one that works within the limitations of the classroom. Specifically, a successful implementation of this environment should support the following five Flow of Inspiration Principles.

Flow Of Inspiration Principles
These Principles should allow students to:
1) Display projects in a public forum
2) View and run fellow students' projects
3) Provide feedback on fellow students' projects
4) Download and view code for any project
5) Provide motivation for students to view, download, and give feedback on fellow classmates' projects

The first four specifications describe general infrastructure characteristics that enable in-class peer-to-peer interactions. The final point, wherein the class provides incentive for these interactions, is crucial for attaining the emerging behaviors present in the middle school computer club environment. This incentive is provided through an initial highly-scaffolded curriculum that gradually gets relaxed as the semester proceeds.

2.1 Educational Game Design Class

Each week, the Educational Game Design Class consists of a theoretical and practical part. The theoretical part discusses gamelet creation strategies including adding educational value to games, making games engaging, and reviewing computational thinking patterns [12,13]. The practical part allows students to apply this knowledge by creating gamelets. The Educational Game Design Class has an aggressive schedule assigning students to create one gamelet a week for 8 weeks. The remaining semester time is spent on a final project. The first 4 weeklong assignments are the same among all students. The next 4 weeklong assignments comprise a period known as "Gamelet Madness." Gamelet Madness forces students to think of and implement their own original simple educational game idea each week. Furthermore, a given student must create a gamelet that has no relation to a prior week's Gamelet Madness game she/he created. Generally, during Gamelet Madness, the students' submissions for a given week have very little relation to one-another. The final project involves students creating an educational gamelet and playtesting it at the local middle schools for feedback. Students

[1] scalablegamedesign.cs.colorado.edu

then use this feedback to improve their gamelet and present their findings to the class. With approximately 32 students in the class creating 9 games over the course of the semester, the Educational Game Design Class as a whole yields around 290 games by semester's end.

As mentioned above, the first four homework assignments of the Educational Game Design Class are weeklong wherein all students create the same games: Frogger, Sokoban, Centipede and The Sims. These games start simple but get increasingly complex as students learn more sophisticated computational thinking patterns [6]. Though individually-created student games may look different, the main programming patterns used to solve various in-game problems are the same. Thus, in Frogger, for example, if a student wants to program the truck colliding with the frog, the implementation structure is virtually identical among all students in the class. This situation provides the motivation to look at fellow students' projects to see how they went about solving a particular problem. Furthermore, since everyone is working on the same project, students want to look at other projects to compare with their own assignment and integrate interesting things fellow classmates may have done into their own project. As the class progresses, this scaffolding is removed as students work on the more individualized assignments of Gamelet Madness and the final project; By this point, the precedent of viewing other students' projects and using their interesting ideas has hopefully been established.

3. EVOLUTION OF CYBERLEARNING INFRASTRUCTURE

One possible way to replicate the Flow of Inspiration witnessed at middle school computer clubs within a university classroom is to employ a cyberlearning infrastructure. But what is Cyberlearning? In the recent "Fostering Learning in the Networked World: The Cyberlearning Opportunity and Challenge. A 21st Century Agenda for the National Science Foundation, June 2008" report [10] the National Science Foundation defines Cyberlearning to be

> "[...] networked computing and communications technologies to support learning. Cyberlearning has the potential to transform education throughout a lifetime, enabling customized interaction with diverse learning materials on any topic—from anthropology to biochemistry to civil engineering to zoology. Learning does not stop with K–12 or higher education; cyberlearning supports continuous education at any age."

Cyberlearning infrastructures take many forms. In the classroom cyberlearning infrastructures, like Poogle for example, have been used to add social elements and publicly share solutions to homework assignments [11]. Google Code Search[2], on the other hand, is a cyberlearning infrastructure that helps developers seek out specific programming code for use in their projects. For our purposes, the cyberlearning infrastructure should enable the middle school computer club interactions. The evolution of a cyberlearning infrastructure that accomplishes the Flow of Inspiration Principles is described in the systems that follow. The shortcomings of prior systems with respect to the Flow of Inspiration Principles motivate the creation of the Scalable Game Design Arcade.

[2] http://www.google.com/codesearch

3.1 Individual Homework Email Submissions

Initially, in 2003, before attempting to recreate the Flow of Inspiration Principles, individual students in the Educational Game Design Class would email the grader their assignments. This is the classic structure of most university level programming classes; none of the Inspiration Principles are met. The only project feedback is given by the class grader who probably only reviews each assignment briefly. Students do not view one another's projects and cannot see how other students went about solving problems. Nor can students give feedback on a fellow classmate's project. One advantage to this system is that cheating is more easily discovered as similarities between student-code should not occur if students are not viewing each other's code or garnering illegal input from fellow students.

3.2 Group Projects

Allowing for increased student peer-to-peer interactions necessitate a shift from the email submission system. To this end, in 2004 and 2005, the Educational Game Design Class employed group projects as a way to encourage student interaction. It was thought that group projects would foster all the specifications above and create an open classroom environment among all the group members. However, this did not occur; instead, students would break up projects and delegate tasks. Interaction, for the most part, was restricted to when the group members met back up to reassemble the project. Within this divide and conquer approach the student goals are disparate; thus, there is very little advantage in peer interaction. In short, the group project system was found to be a poor peer learning model because there exists little intrinsic motivation to learn from other students. From this experience it became clear that to get the emergent interaction advantages inherent in the middle school computer club, students must be working on individual projects but with facilitated and encouraged community interaction. One way to accomplish this within the limitations of a university class involves creating a cyber-infrastructure.

3.3 Gallery Organizer Repository of Projects

The Gallery Organizer Repository of Projects (GORP, 2006) infrastructure used an online project posting submission system; this system allowed students of the Educational Game Design Class to put their projects online and freely view fellow student projects [9]. Furthermore, students could comment on and easily download other students' projects. This cyber-infrastructure obtained limited success in each of the above Principles laid out. Students in theory could display, view, provide feedback and download other students' projects. However, students could not rate projects in a simple manner. Furthermore, students in general would not leave comment feedback upon viewing a project; it was discovered that the main reason for this was that the overhead of leaving a simple comment, logging in, meant that the student had to go through more trouble than it was worth yielding the practice nonexistent for more casual comments. There was also existed no method of tracking who downloaded and viewed a project so there was no way to evaluate how well or even if the peer-to-peer interaction was taking place, though there was informal evidence to suggest that it was. The shortcomings of GORP motivated the current cyber-infrastructure, the SGDA.

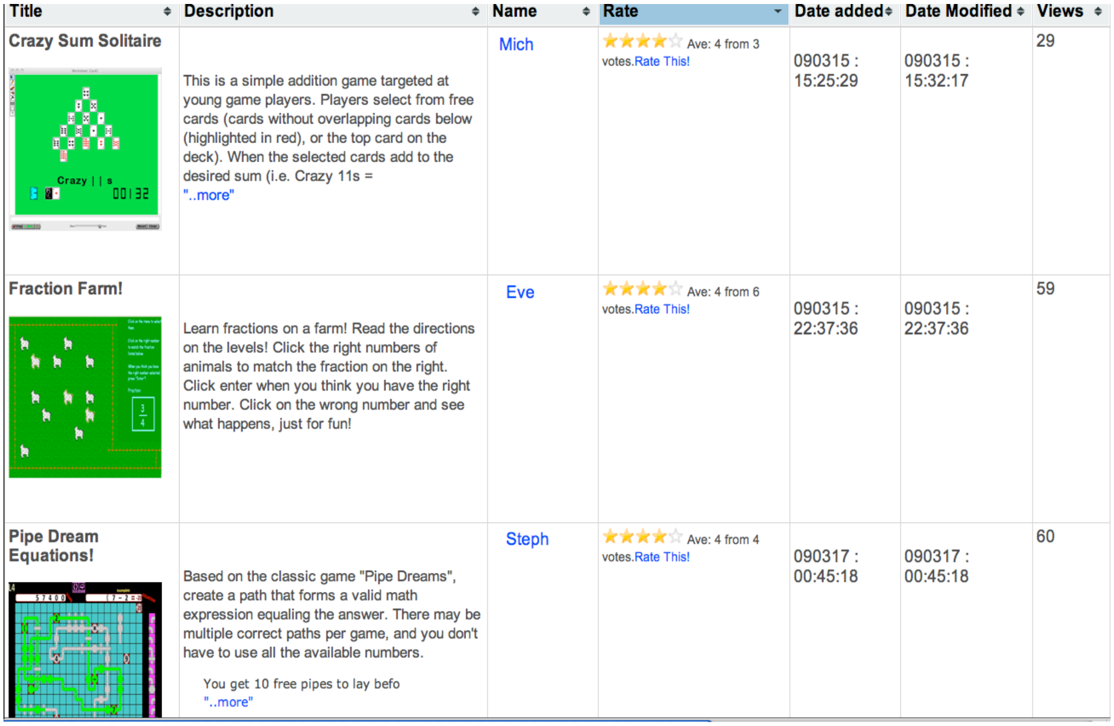

Figure 1: Scalable Game Design Arcade with thumbnail, author description, rating, and submission time. Clicking on the Author's name link for any of these projects allows one to play the game, download code, rate, and add comments.

3.4 Scalable Game Design Arcade

The Scalable Game Design Arcade (2009) extends the GORP online submission interface to facilitate user feedback. The user does not have to log in to leave feedback on given student's submission. Furthermore, a one to five rating system provides an easy way to appraise projects and organize by projects that got the best user feedback. This, in theory, makes it easy for students to look and be inspired by the best projects and allows a simple way to provide feedback. Embedded in the rating system and the ability to organize by ratings is the Youtube/Facebook mentality which allows students to gravitate towards classmate's work who they individually find interesting as well as work that the class as a whole finds interesting. Furthermore, just like with Youtube, students can sort according to many different criteria including views, star rating, and name. Students are encouraged to submit their work early and often so that the feedback received can be used in the creation of subsequent game iterations. The user does, however, have to login to download a program enabling the tracked flow of ideas from one user's game to another. It also discourages cheating as directly copying a fellow student's work can easily be detected.

4. RESULTS

Figure 1 depicts a screenshot of the Scalable Game Design Arcade. The SGDA Website consists of a list of game descriptions including a thumbnail depicting the game, descriptive text describing the game along with some additional meta-information such as number of views. On the left of Figure 1 is the "Title" column which contains a thumbnail screenshot of each project along with a title. Next to the thumbnail, in the "Description" column, is a brief synopsis of the game, game storyline, and game-play instructions written by the student. To the right of the description, in the "Rate" column, is the ratings fellow students give the game out of 5 stars along with how many students star rated the game; in the column to the right of the rating, is the "Date Added" column which shows the time the project was first submitted. The next column to the right, the "Date Modified" column, displays the time of the most recent game submission. Finally, the "Views" column furthest to the right displays the number of times the project has been viewed by people.

If a student clicks on the Author's name link in the 'Name' column, it allows the student to run the game, leave a comment, and download the code; Figure 2 depicts this. The projects are run as a java applet in the browser itself. However, as mentioned above, to download another student's code, the student must first log in.

In terms of accomplishing the five Flow Of Inspiration Principles, initial indications from students using the SGDA are promising. In order to get a better understanding of how students use the SGDA and to what extent, if any, the SGDA enables the Flow of Inspiration Principles, a questionnaire was given to the class. The following sections will analyze and discuss the results of this questionnaire.

Figure 2: A detailed game description window wherein a student can run the project, download the project code, star rate, and comment on the project.

4.1 Student Questionnaire

The online questionnaire administered to the class consisted of 23 questions, 15 open-ended and 7 multiple choice. Replies were anonymous, voluntary, and students could skip any question they did not want to answer. Responses from 20 students were collected, and the open-ended answers were categorized to discover any trends among students. On all open-ended questions students could give multiple answers, thus, the percentages of all categorized answers to open-ended questions can exceed 100%.

The questionnaire is meant to answer two questions. The first question is to what extent does the SGDA accomplish point five of the Flow of Inspiration Principles; namely, given that students can view, play, download and comment on classmates' projects, how often does this online peer-to-peer interaction actually occur. The second question investigates the following: to the extent that the Flow of Inspiration Principles are being integrated into the class, is there indications that the quality of work is increasing? It should be noted that we are not proving a causal relationship between the quality of work and the Flow of Inspiration Principles based on the questionnaire; however, we are trying to establish if there is an attempt on the student's part to improve their work because of the SGDA. The following result sections are organized in terms of Flow of Inspiration Principles 2-4 in order to see how well the SGDA is accomplishing motivating these principles (essentially how well the SGDA is accomplishing Flow of Inspiration Principle 5; it should be noted that Flow of Inspiration Principle 1 is essentially accomplished by the existence of the SGDA online submission system). In each section we attempt to answer the two questions presented above for a given Flow of Inspiration Principle.

4.2 Viewing and Running Fellow Students' Projects

As shown in Figure 2, running a fellow classmate's project can be done in the browser via clicking the "Run" link on the project page. Students were asked to estimate how many times per week they played a classmates' game on their own volition (not as part of an assignment). 95% (19/20 students) of students answered that they played at least one classmate's game a week and 85% (17/20) played 2 or more classmate's games a week with most students, 45% (9/20), playing exactly 2 games a week. Figure 3 is a graph that shows what benefits students hoped to gain when they played classmates' games on the SGDA.

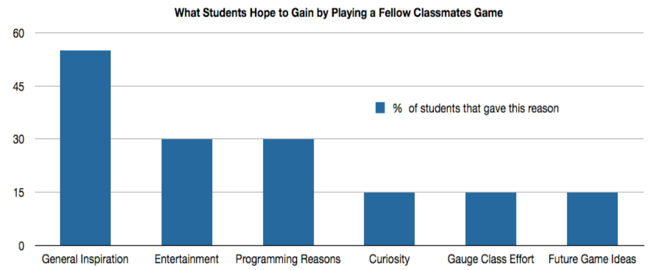

Figure 3: Graph that shows what benefits students hoped to gain from playing SGDA games.

Most students, 55% (11/20), mentioned general inspiration as the reason for playing classmates' projects. 30% (7/20) of the responses mentioned specific programming reasons for playing a fellow student's project. These include, among other things, looking for in-game graphics ideas, game play ideas and in-game agent interactions. Furthermore, 30% of students played games for pure entertainment purposes. Finally, 15% (3/20) of students mentioned they played games out of curiosity for what their fellow classmates were turning in and to gauge class effort for a given assignment. The following quote is representative of how many students answered this question:

"When I play a game created by someone else, I look for interesting ideas and design, and hope to find some inspiration for my own games. I also hope to be entertained for a while by their game."

Given that so many students hoped to be inspired by a classmate's game, it begs the question what effect, if any, does playing classmates' games on the SGDA have on a given student's project submissions. Asked if they had ever changed their project after playing a classmate's version of the game, 52.6% of the students said 'yes' (10/19 with 1 person abstaining). Of those 52.6%, when asked the reason for changing their game, 100% (10/10) replied that they wanted to improve their own game based on what they saw in their classmate's game. The following is a typical student quote:

"On the centipede gamelet i did notice other games made the centipede smarter , so i changed so it wouldn't get stuck in certain situations."

Furthermore, students were asked if a classmate's game they played from a previous week's assignment had ever inspired the current week's assignment; 57.9% (11/19) said yes. One gave the following answer:

"Yes, my Digital Logic gamelet was inspired by one of my classmates who had done a digital logic gamelet the week before. I got the idea of using logic circuits from their game and molded my own game out of it."

The questionnaire results indicate that students played other students' projects on a weekly basis, and thus, the SGDA is effective in enabling and motivating students to view and run, fellow student's projects. Moreover, after playing classmates' games, students often use the experience to try to improve their own game or as inspiration for a future game. The data shows strong evidence that the SGDA helps to increase the quality of work submitted as students are trying to improve upon what they turn-in based on what their peers have done.

4.3 Providing Feedback on Fellow Students' Projects

The SGDA provides two formal mechanisms for leaving feedback. The quickest way is to star rate a given project. The hope is that over time, the amalgamation of star ratings should yield a consensus as to what particular projects were liked and disliked by the class as a whole. Star ratings can be made anonymously.

Commenting is the other formal way students can leave feedback. Comments allow students to give a more in-depth critique, or a better description of things they liked. Students must give a "handle" to leave a comment which could be the student's real name but did not have to be.

The final way students can provide feedback is through in-class verbal feedback on a classmate's game. One could argue that this is not explicitly an SGDA feedback mechanism as students could provide verbal feedback in any class regardless of the existence of an online infrastructure. However, students must be able to play a classmate's project before having the ability to give verbal feedback, and the characteristic of playing any classmate's game is explicitly enabled by the SGDA. The fact that students have to submit their assignments publicly to the class enables in-class verbal feedback among students to exist. The questionnaire queried students about all three types of feedback.

When asked how many times this semester they had star rated a classmate's game on their own volition, 40% replied seven or more times (8/20); 80% of the class said 3 or more times (16/20). These results seem to indicate that there is motivation to use the star-rating functionality in the SGDA. When asked to give the reasons for star rating a classmate's game, the overwhelming majority replied that they star rated because they liked the game. Figure 4 is a graph that shows all the reasons students said they star rated games. 61.11% (11/18 with 2 abstaining) said they star rated when they liked the game;

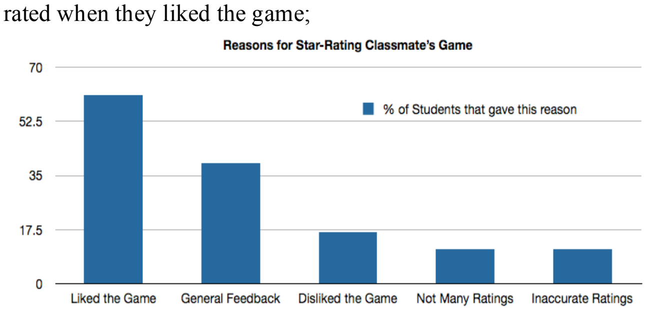

Figure 4: Reasons students gave for star-rating a classmate's game. Most people star rated when they liked a game.

38.89% (7/18) said they star rated as a general feedback mechanism. Only 16.67% (3/18) said that they star rated when they disliked a game. In the current state, most star ratings are only given by the subsection of the class which view the game positively. Therefore, if true, the SGDA star rating does not reflect a class consensus as to whether a given game is good or bad because the whole class is not involved in rating a particular game. One student wrote the following:

"Usually I leave a rating if it's a great game... not quite as often if I didn't think it was very good."

This seems to indicate an area where the SGDA needs to be modified such that students are more willing to honestly rate games they find good and bad. A few students suggested that star ratings be updateable, meaning, that if a student initially gives a project a low star rating, if the project improves, the student could go back and replace the initial star rating. Currently, a student can only give one star rating for a given project, and it cannot be changed. Furthermore, other students suggested that it should be possible to tie a star rating explicitly to a comment such that if you give a low star rating you could couple it with an explanation as to why the star rating was so low; this is backed by the SGDA comment feedback data that will be presented next. This indicates that students would not mind being critical with star ratings as long as there exists a way to make the low star rating constructive criticism. The reluctance to leave a bad star rating is at least partly tied to the lack of an SGDA mechanism to explain the criticism and reward an improved project.

When students were asked how many times this semester they left a comment on a classmate's project, 25% (5/20) responded seven or more times, 75% (15/20) said they left a comment 2 or more times, and 85% (17/20) left at least 1 comment. Though there is room for improvement, this data seems to show that students are leaving comments on classmates' projects. Figure 5 shows the student responses when asked the main reasons for leaving a comment. Again, as with the star rating, students left comments because they liked a given game (35.29%, 6/17 with 3 abstaining); however, in contrast to the star rating, equally as many students left comments to suggest improvements (35.29% 6/17). As mentioned above, it appears that students are less reluctant to criticize as long as they can explain their criticism.

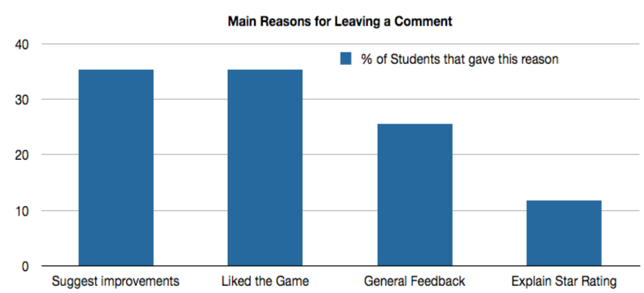

Figure 5: Reasons students left comments. Mostly students commented because they liked the game or to suggest improvements.

When students were asked whether they had ever updated a project based on comments received, 36.85% said 'yes' (7/19, 1 abstaining). The people that said 'yes' replied that usually the comment pointed to a specific problem with their project that they corrected. The following quote is typical of the 'yes' responses received:

"YES! On some of the earlier projects I submitted early, and the feedback told me it was too hard, or the interface was awkward. So I switched it up, went back and resubmitted."

On the other hand, 63.16% (12/19) answered 'no'; however, it should be noted that of the 'no' responses, at least 4 said that they were intending to use comments made on a previous game for a future game including game-remakes they were planning on doing for their final projects. Many of the 'no' responses were because students either commented after the project was due or the author of the project only looked at the comments post due-date.

When asked if comments left on previous projects influenced their current projects 38.89% said yes (7/18, 2 abstaining); 61.11% (11/18) said 'no'. Many of the 'no' responses refer to the fact that since the assignments are so different, it is hard to apply previous project comments to current projects. The following is a typical quote from a student who gave the 'no' response:

"No, since the gamelets are so different, the comments usually only apply to that project."

Given that this is the case, it makes sense that the students who are remaking old games for their final project plan on taking older comments into consideration. Students are commenting and, to a certain extent, using the comments to improve their projects. However, it seems like the SGDA has to increase motivation for pre-deadline commenting on projects. Moreover, it might be helpful to send an email to the author such that if a comment is made before the deadline, the project author sees it before turning the project in rather than after the deadline. Furthermore, 57.14% of students (8/14, 6 abstaining) said they would comment more if they had more time. The following is a typical quote from these answers:

"If I had more time!. . . When I spend 15-20 hours (from start to finish) putting a game together and submitting it, and am faced with another game due in 5-6 days, it's tough to motivate myself to go play a bunch of other folks' games. . ."

Since students feel the workload is so large, it seems promising that comments are occurring using the SGDA, and students are using these comments to update their own games. A workload change or phased development wherein students can leave comments after an initial project phase might allow more time for comments to be made and an opportunity for comments to readily be used in project development.

When asked if other students had provided verbal in-class feedback on a project (under their own volition, not as part of an assigned discussion), 75% said 'yes' (15/20). Furthermore, when asked if they had ever changed their project because of in-class feedback received 53.5% said 'yes' (8/15, 5 said 'no' to the last question). Interestingly, it seems like the SGDA has increased in-class feedback, and this, in turn, has led to students attempting to improve their project based on verbal feedback received. The above quote about how more time would allow for increased SGDA comments, provides insight into why many students might prefer to instead give feedback verbally in-class. Since students come to class anyways, playing a fellow classmate's game and giving comments in-class does not take away from the time the student spends outside of class on the projects. Finally, it should be noted that this in-class interaction mirrors the interactions that actually happen in middle school computer clubs, and these interactions are the very ones we want to promote regardless of the feedback mechanism used.

4.4 Downloading and Viewing Project Code

In Figure 2 we see that in addition to running a classmate's game for a given assignment, students have the option of downloading the project as a .zip file. This allows students to look at other students' code to figure out how they accomplished different interactions. When downloading code, the student must login. The reasons the SGDA requires student login for downloading is to discourage outright cheating (identically copying someone else's code) and to track influence among students. Students are encouraged to investigate other students' code and even take parts of their classmate's code as long as they make it their own. When asked if they had ever downloaded a classmate's project code, 57.89% replied yes (11/19, 1 abstaining). Furthermore, of those who downloaded code, 75% replied that the code was helpful (6/8, 3 abstaining). The general consensus among those who were helped by downloading was they were trying to figure out something specific, and by downloading the code, it put them on the right track to accomplishing this. The following is a quote that is typical of what most students who found downloading helpful said:

"Yes, I downloaded code a few times. Usually they had a cool feature in their game and I wanted to see how they did it. Yes it did help, it showed me how the feature was implemented."

During the course of the semester, many students expressed reluctance to download code even though it was encouraged from the first day of class. When asked if they felt downloading a classmate's code for a given assignment before turning in the assignment was wrong, 25% (5/20) answered 'yes'. Furthermore, 15% (3/20) felt that it was a gray area. Many of these people thought that it was all right after the assignment was due or later in the semester when students submitted individual projects (as opposed to the first four weeks wherein everyone submitted similar games like 'Frogger' or 'Sokoban' etc.)

From the questionnaire answers it looks like the SGDA along with the Educational Game Design Class is effective in motivating students to download code allowing ideas to easily flow from one student to the next. It also seems like students want more guidelines as to when downloading and using someone else's code should be allowed and when it should be prohibited. A possible solution to this would be to first explain the middle school computer club motivation for allowing students to download a classmate's code. Furthermore, as was suggested with the star ratings, maybe if we could link the downloads with comments, such that students could openly provide feedback on the downloaded code, it might highlight the middle school computer club context. Overall, student-downloading seems to be occurring using the SGDA, and students who download often attempt to use the information garnered to improve their project.

5. CONCLUSIONS

The results of the questionnaire indicate that the Scalable Game Design Arcade is effective in implementing the Flow of Inspiration Principles in the Educational Game Design class. Students use the SGDA to view other students' projects. To varying degrees students also use one of three feedback mechanisms enabled by the SGDA to appraise classmates' projects. A sizeable portion of students download and view classmates' code. Evidence strongly suggests that students are attempting to increase the quality of their work after viewing classmates' projects, receiving feedback on their own projects, and downloading classmates' code. Informally, this is further verified by our prior experiences with the Educational Game Design Class; the current SGDA class implementation coincides with a marked increase in the quality of submitted projects. The SGDA seems to enable and motivate the emergent middle school computer club peer-to-peer interactions in the Educational Game Design Class.

Future SGDA iterations will focus on the shortcomings highlighted by the questionnaire. Creating a class with phased gamelet development such that students have more time to comment on fellow students' code might allow for increased numbers of students to use the SGDA online feedback mechanism more often. Allowing students to explain their critical star ratings could enable more honest evaluations of classmates' projects. Finally, making the download policy more explicit might allow students to be less tentative downloading and using classmates' projects for their own purposes.

This questionnaire was meant to analyze the cyberlearning infrastructure and its effects on the Educational Game Design Class. However, the questionnaire data not only indicates the extent to which the Scalable Game Design Arcade accomplishes the Flow of Inspiration Principles, but also, possibly gives insight into how the more successful SGDA characteristics are actually enabled by the physical classroom. For example, students preferred to give in-class verbal feedback of classmates' games; furthermore, in-class feedback led to many students altering their games based on this feedback. More investigation is necessary to see the reasons why in-class feedback was popular, but one possibility is that face-to-face interaction might be preferred over online feedback mechanisms. Or possibly, being in the vicinity of others lends itself to feedback that might have not otherwise been voiced. Some quotes from the questionnaire support the idea that when students were programming their games in-class often people would randomly stop by and give unsolicited feedback. Furthermore, a few students said they asked fellow classmates' in-class for game feedback presumably as a quick and easy way to see what was working and what needed improving. Though it might be possible to provide and receive this classmate interaction online, it seems more efficient to do it in class. Regardless of the reason, the point still remains that the questionnaire pointed to classroom interactions being a very important part of accomplishing the Flow of Inspiration Principles. Thus, the "Sage on the Stage" teaching approach might have little need for a physical classroom anymore; however, these initial findings look like a promising argument for the essential role of the physical classroom within teaching strategies that employ peer-to-peer student learning.

6. ACKNOWLEDGMENTS

We would like to thank all the students of the Educational Game Design Class at the University of Colorado Boulder. Our thanks to Centennial Middle School and the Boulder Valley School District. A special thanks to David Webb for helping with the evaluation methodology. This material is based in part upon work supported by the National Science Foundation under Grant Number No. DRL-0833612. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

7. REFERENCES

[1] Fischer, G., Rohde, M., Wulf, V. 2007: Community-based learning: The core competency of residential, research-based universities. International Journal on Computer-Supported Collaborative Learning (iJCSCL), Vol. 2, No. 1., pp. 9-40.

[2] Williams, L., Layman, L., Slaten, K. M., Berenson, S. B., and Seaman, C. 2007. On the Impact of a Collaborative Pedagogy on African American Millennial Students in Software Engineering. In Proceedings of the 29th international Conference on Software Engineering (May 20 - 26, 2007). International Conference on Software Engineering. IEEE Computer Society, Washington, DC, 677-687.

[3] Pollard, S. and Duvall, R. C. 2006. Everything I needed to know about teaching I learned in kindergarten: bringing elementary education techniques to undergraduate computer science classes. In Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education (Houston, Texas, USA, March 03 - 05, 2006). SIGCSE '06. ACM, New York, NY, 224-228.

[4] Hamer, J., Cutts, Q., Jackova, J., Luxton-Reilly, A., McCartney, R., Purchase, H., Riedesel, C., Saeli, M., Sanders, K., and Sheard, J. 2008. Contributing student pedagogy. SIGCSE Bull. 40, 4 (Nov. 2008), 194-212

[5] Repenning, A. and Ioannidou, A. 2004. Agent-based end-user development. Commun. ACM 47, 9 (Sep. 2004), 43-46.

[6] Repenning, A. and Ioannidou, A. 2008. Broadening participation through scalable game design. In Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education (Portland, OR, USA, March 12 - 15, 2008). SIGCSE '08. ACM, New York, NY, 305-309.

[7] Waite, W. M., Jackson, M. H., Diwan, A., and Leonardi, P. M. 2004. Student culture vs group work in computer science. SIGCSE Bull. 36, 1 (Mar. 2004), 12-16.

[8] McKinney,D., Dyck, J., Luber, E. 2009. ITunes University and the classroom: Can podcasts replace Professors? Computers and Education. Vol 52, 617-623.

[9] Chorost, M. (2008). Educating Learning Technology Designers: Guiding and Inspiring Creators of Innovative Educational Tools . New York: LEA. 357

[10] Christine, H. Abelson, L. Dirks, R. Johnson, K. R. Koedinger, M. C. Linn, C. A. Lynch, D. G. Oblinger, R. D. Pea, K. Salen, M. S. Smith, and A. Szalay, "Fostering learning in the networked world: The cyberlearning opportunity and challenge, a 21st century agenda for the National Science Foundation," NSF Report, June 2008.

[11] Head, C. C. and Wolfman, S. A. 2008. Poogle and the unknown-answer assignment: open-ended, sharable cs1 assignments. In Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education (Portland, OR, USA, March 12 - 15, 2008). SIGCSE '08. ACM, New York, NY, 133-137.

[12] Wing, J. M. 2006. Computational thinking. Communications of the ACM 49, 3 (Mar. 2006), 33-35.

[13] Papert, S. 1996. An Exploration in the Space of Mathematics Education. Journal of Computers for Mathematical Learning. Vol. 1, No. 1, pp. 95-123