Gameful Constructionism
Project Proposal
COMS 6901 E sec 028
Spring 2018

Yuval Schaal
Jen-An Lien

February 2, 2018

Table of Contents

Abstract

In this paper, we will describe our proposal to add objective ui/ux and constructionist game creation abilities to SAGE (Socially Addictive Gameful Engineering). First of all, we will discuss related works that pertain to the intelligent hinting system and constructionist games. The proposal will describe the features the students will work on in this project such as objective UI/UX, objective points display, constructionist game creation, and constructionist game edit.

1. Introduction

Since the early twentieth century, video games have become part of the daily lives of many students. These games have captivated their mind and captured their attention in a way that was not done beforehand. Educators, alongside researchers, saw this as an opportunity to combine these addictive games and take a more educational approach to them. Nowadays they are known as educational games. Taking this a couple steps further, we come to constructionist games. These type of games involve free ranging constructionism, which is a large part of the programming language and online community, Scratch.

This proposal is concerning this semester's upcoming gameful constructionism work. We will be working on features such as the Objective UI/UX, which involves a timer that will count down when a students begin a game. The timer will be set by the instructor based on how long they think students should have to complete the game successfully. The Objective UI/UX feature objective that we are also planning to work on this semester is the intelligent hinting system, that will aid students through different games. Another part that will be worked on is the students’ scores while working on various games. Another feature that we are planning to work on is the constructionist game creation and edit. This feature will allow instructors to create constructionist games for students, as well as edit the game as needed when they need to change a game for various reasons.

2. Related Work

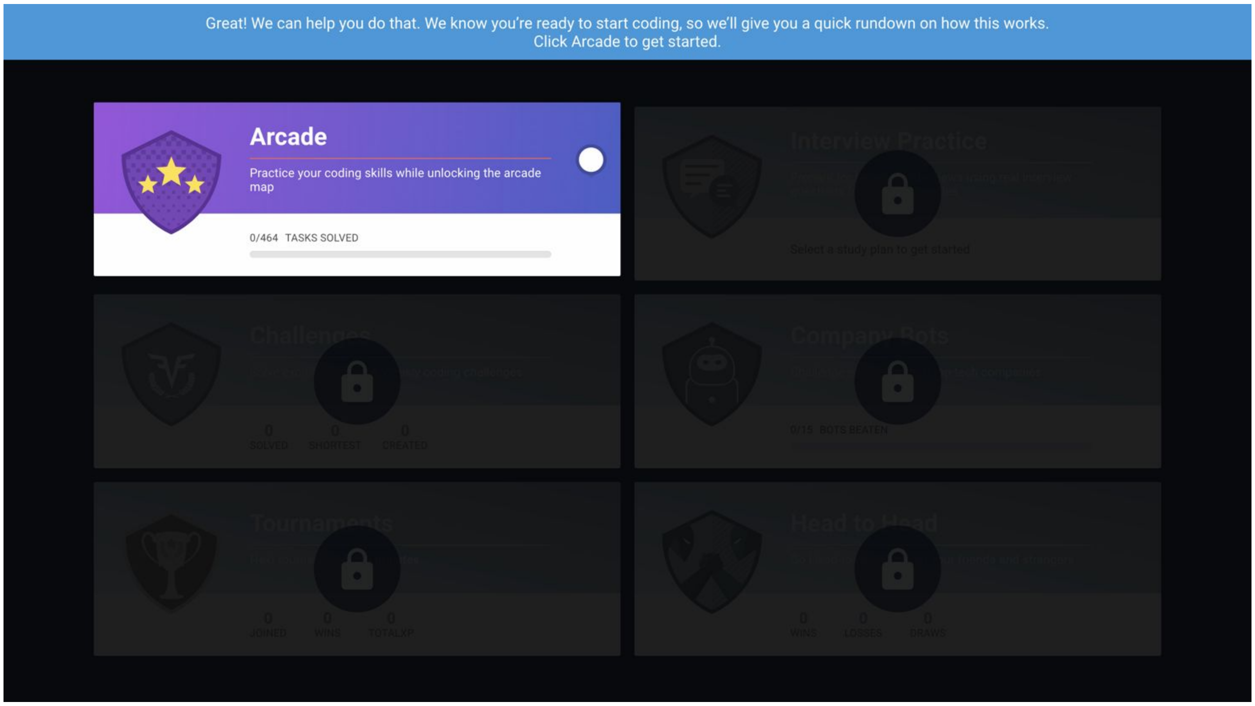
2.1 CodeFight

Recent years, lots of app-development startup companies decided to choose the young children as their targets. These companies develop the websites called digital playgrounds, which help children learn programming languages in a creative and funny way. For example, CodeCombat is a RPG game that asks players to solve the some problems before getting some money that could be used in buying some weapons or accessories. Most of these websites provide a friendly user interface to let players pick them up easily.

CodeFight is one of the best examples in digital playground. It helps beginner to acquire CS knowledge very quickly. And provide the battle mode to compete with others in solving algorithm problems within the given time. It provides progressive stages for people to learn what they want to learn from scratch. By finishing more and more stages, users get XP and know their rank in their country or university.

2.1.1 Hints for Beginner

Traditionally, users look through the plain text instructions to know how to use website properly. CodeFight highlights some part of components in the page or use the pop-up dialog box to give the beginners clear guidance about the way to use the website. Normally, the hints are given only in the first time when players used the page. They assume that users should be get familiar with all the functions quickly. Screenshots give an example about how they guides the beginner. In addition to the highlight, CodeFight also provides simple text instruction to help users understand the reason why there is a highlight as well as help users really know what to do in next steps.

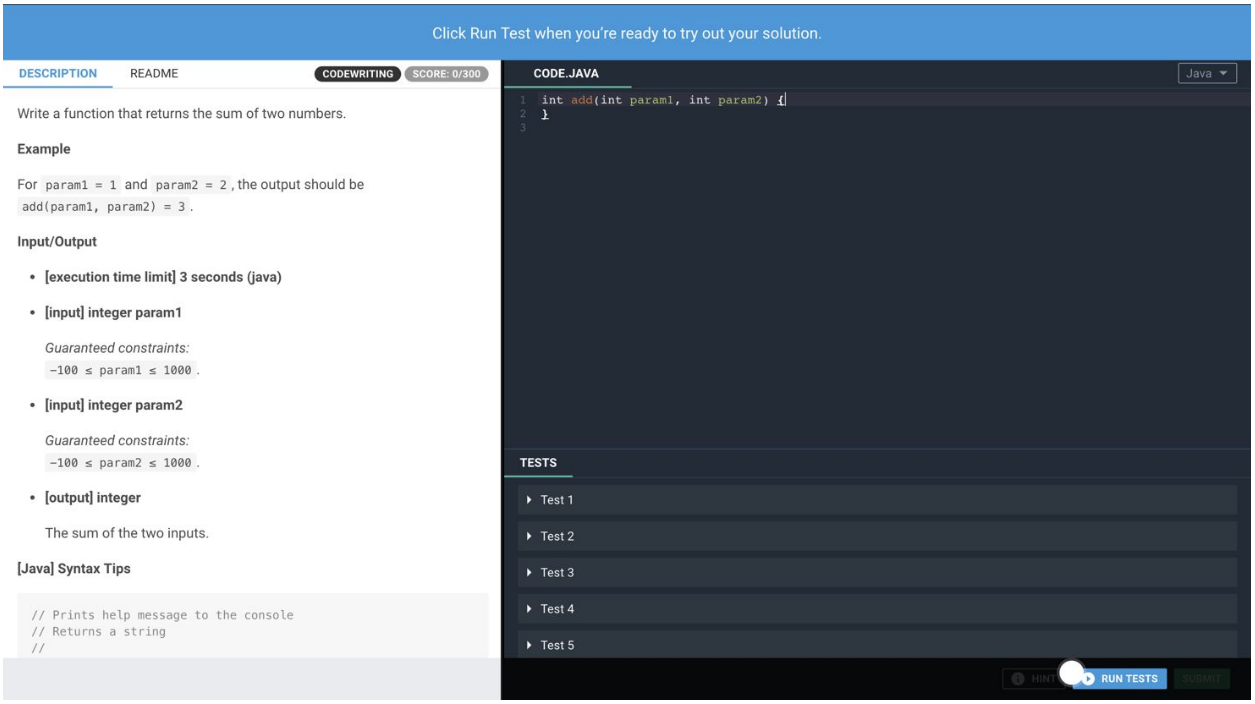


Great! We can help you do that. We know you're ready to start coding, so we'll give you a quick rundown on how this works.
Click Arcade to get started.

Arcade

Practice your coding skills while unlocking the arcade map

0/464 TASKS SOLVED



Write a function that returns the sum of two numbers.

Example

For `param1 = 1` and `param2 = 2`, the output should be `add(param1, param2) = 3`.

Input/Output

- **[execution time limit] 3 seconds (java)**
- **[input] integer param1**

  Guaranteed constraints:
  `-100 ≤ param1 ≤ 1000`.
- **[input] integer param2**

  Guaranteed constraints:
  `-100 ≤ param2 ≤ 1000`.
- **[output] integer**

  The sum of the two inputs.

[Java] Syntax Tips

```
// Prints help message to the console
// Returns a string
//
```

```java
int add(int param1, int param2) {
}
```

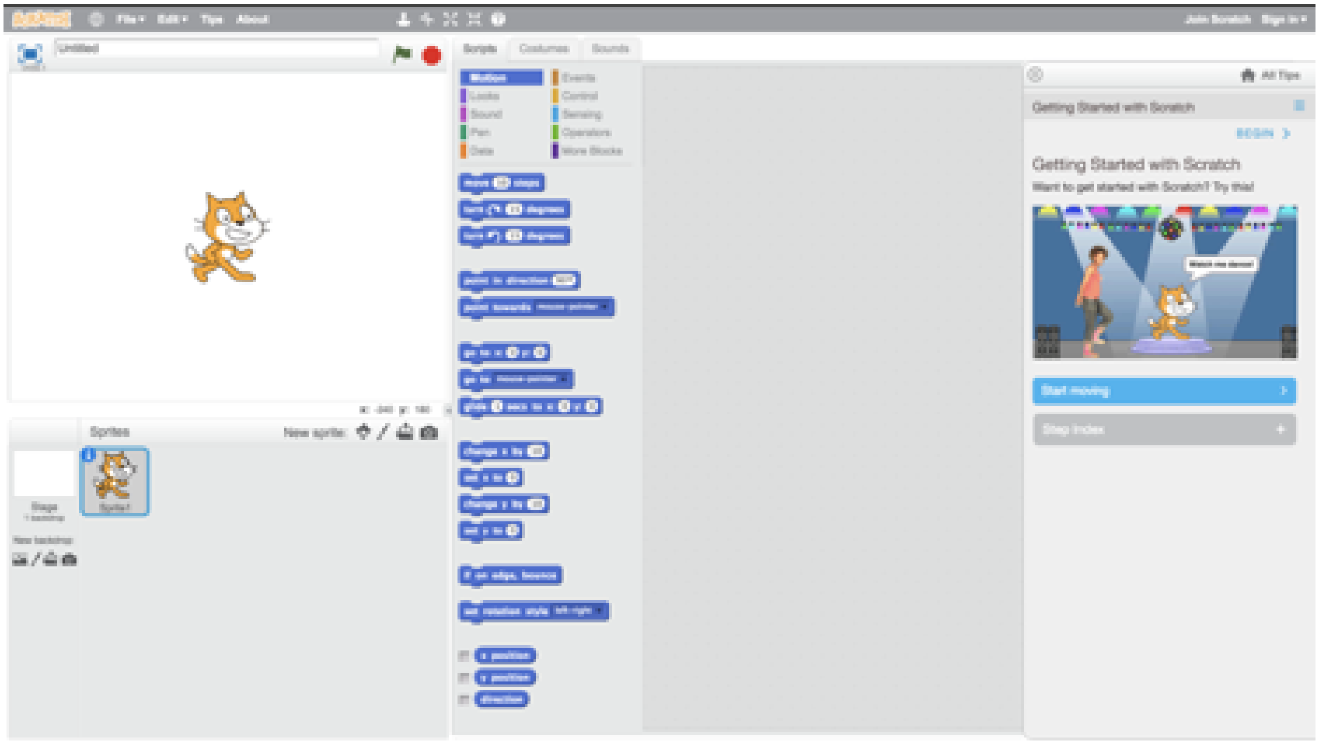
- Test 1
- Test 2
- Test 3
- Test 4
- Test 5

2.1.2 Score System

Users gain XP when completing one game in CodeFight, the amount of XP they get depends on rules set by the rulers and how well they did on their game. For example, if the game asked one algorithm problem, solvers come out the algorithm that could solve in most case but not some edge cases, they only get partial XP. Score calculation is defined by the problems' rule, which could be set by the administrator. In addition, XP could be money. Users buy new stage using their XP. The amount of XP also defines the hardness of stages and prevent the beginner to get access in hard stage when they do not have ability to complete hard one.

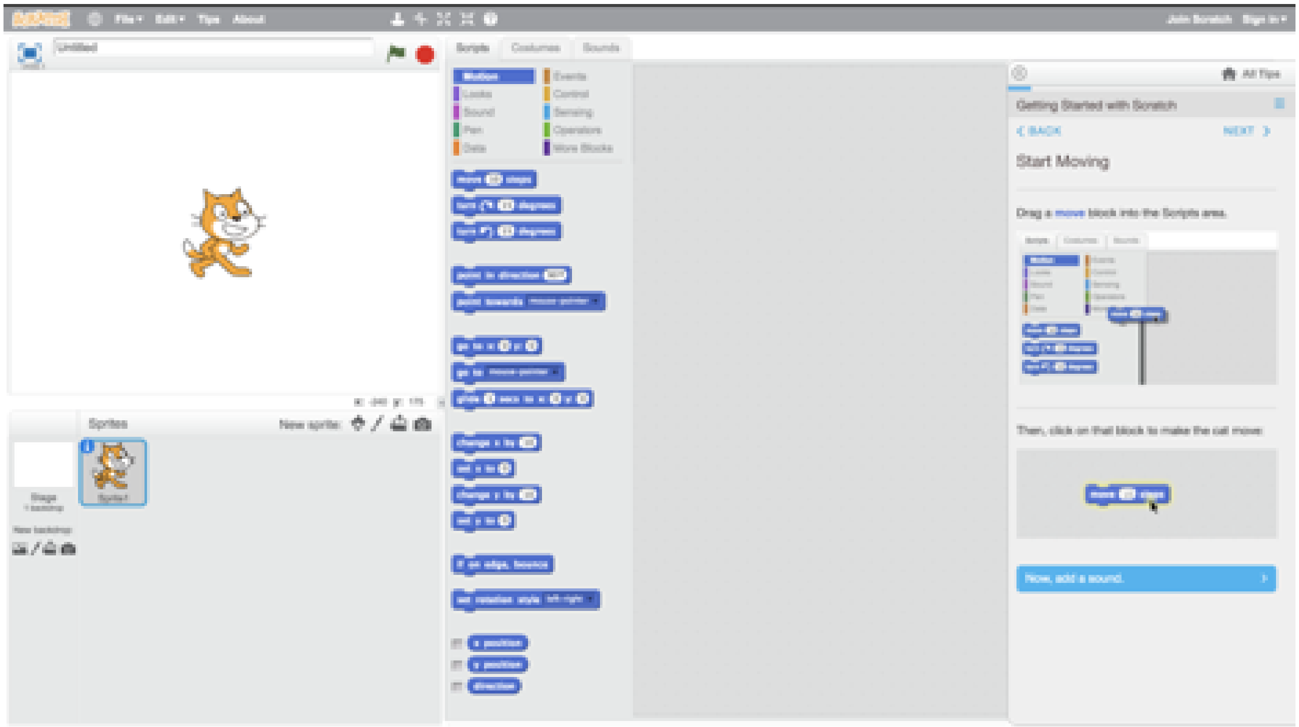
2.2 Scratch

Scratch is a website that help children learn coding concepts and create a project without using any text-based programming language. It used blocks which stand for different operations in programming. Children could create its own project by assembling the blocks.



Getting Started with Scratch

Want to get started with Scratch? Try this!



Getting Started with Scratch

Start Moving

Drag a move block into the Scripts area.

Then, click on that block to make the cat move.

Now, add a sound.

2.2.1 Hints for Beginner

Unlike CodeFight, it tries to blacken the unnecessary part and highlight the part it wants to show to the players, Scratch provide many animate instructions to help the player to get familiar with all the function step by step. The biggest difference between them is while CodeFight only guide players to know how to use the simple operation such as ‘how to compile’ or ‘how to run the test cases’, and let the users figure out the rest of problems by themselves, Scratch gives hints and helps players to finish the whole project step by step, and players realize that they actually could finish a fancy within few steps.

2.3 Constructionism

It is extremely important to understand that constructionism is an essential part and united part of Scratch. Supplying a sufficient degree of freedom that encourages exploration, as discussed in *Weintrop Et Al 2012.* It is also imperative to recognize and foresee constructionist games' resistance to grading, ranking, and classifying students as underachievers, since inherent in the idea of constructionism is that all students can engage in deep learning if the environment and tools are well designed. In our work this semester, we will work on the intelligent hiniting system to provide the students the best aid to being successful. We will also be working on the constructionist game creation to allow instructors to create constructionist games for their students as well as edit the games they have created.

3. Proposal

3.1 Objective UI/UX

In the previous version of SAGE, players see "Scratch editor", "Objective Feedback" and "Dialogue" when they enter the game. For beginners, the functions easily confuse them because they don't know what 'Scratch editor' is for, how to move the blocks to satisfy the objective set by the instructor. Therefore, the hints for beginners would be very convenient and would help them to get familiar with the game more quickly.

In my new version, when beginners enter the game page, they would be introduced to what dialogue is for, the objectives they are asked to finish and the basic operations of Scratch, by using the CSS in order to make all the unnecessary components black and highlight the parts I want to show them. Besides, in CodeFight, the pop-up dialog, which is displayed on the top of the page, guides players to the next step. In my version, a dialog box with an ‘OK’ button would appear near the element to make sure players actually see the instructions. Players could do nothing before clicking the ‘OK’ button.

In addition, in the current version of SAGE, the timer would start only after users click the button. I propose that the timer should start after the instructions that appear when players first play this game or start once they enter the pages. This helps the timer count the players’ behaviors. Also, in the current version, the timer is set for a fixed 10 minutes. But the harder the game is, the time players need would be more. This setting can be changed in the instruction page. When instructors designed a new game, and set the objective for the users, they could also set the time depend on the hardness of the game.

3.2 Objective Points Display

In the previous version of SAGE, points are only shown in the Scratch editor. When players move the wrong block or do some wrong actions, scores are taken off. I propose that the system should have another score system. Each objectives would have one score. When players finish an objective, they get a score. Besides, instructors can set the rule of how much points a player can get when they finish the objectives.

What’s more, CodeFight allows users to get partial XPs when players only finish the part of the goal. I propose that players get partial scores in the game but it is sometimes hard to define how players get scores when they finish ‘part of objectives’.

3.3 Constructionist Game Creation

Constructionist game creation in sage is one of the essential parts of allowing the instructors of sage to create games for their students. This feature will enable the instructors to create games for students through the instructor dashboard. As the instructor, they should be able to open scratch in CG design mode with the guidance of what to do to more easily navigate through the game creation.

3.4 Constructionist Game Edit

The constructionist game edit feature will allow teachers to edit previously created games in case they need to make any changes to either the game name or structure. In this case, the guidance of creating a new CG game will not be presented. The CG will be opened in design mode and when saved, the CG should be overwritten, where the affinity space will have a created a saved copy.

4. Timeline

Table 1 below outlines the various milestones through each sprint along with the end date of each sprint. Dates are also defined for midterm and final project deliverables.

Sprint Number	Milestone	Sprint End Date
Project Proposal	Project Proposal	February 2nd, 2018
0	Environment Setup	February 16th, 2018
1	Hinting (student side), open CG in design mode	March 2nd, 2018
2	Timer, hinting (instructor CG side)	March 30th, 2018
3	CG creation finalized	April 8th, 2018
4	Points, CG edit hinting off	April 13th, 2018
5	CG edit finalized	April 27th, 2018
6	Final Report & Presentation	May 11th, 2018

Table 1: Proposed Milestones Completion Dates

5. Conclusion

The SAGE platform provides a highly innovative and interactive way to teach computational thinking to young students. The entire process of interactive education is based on gamification and game based learning. Hence, it is imperative that the system is encompassed in a platform that is highly intuitive to use both for the students and teachers. Gameful constructionism will help students and teachers use the system more easily. The features proposed will help guide the instructors and students to navigate and use the platform in a way that will let them learn and quickly accomplish their goals, using hinting and guidance.

6. References

[1] J. Bender, “Tooling Scratch: Designing a Collaborative Game-Based Learning System to Infuse Computational Thinking within Grade 6-8 Curricula,” Columbia University, New York, NY, 2015.

[2] Dichev, Christo, et al. “From Gamification to Gameful Design and Gameful Experience in Learning.” Cybernetics and Information Technologies, De Gruyter Open, 5 May 2015

[3] Resnick, Mitchel, and John Maloney. “Scratch: Programming for All.” Communications of the ACM, Nov. 2009.

[4] Brennan, K., Balch, C., & Chung, M. (2014). Creative Computing. Harvard Graduate School of Education.

[5] Burman, Sanna. “A Gameful Change: How Gamification Can Be Used as a Communication Tool in Change Management.” DIVA, 31 July 2017

[6] Kumar, Deepak. “Digital Playgrounds for Early Computing Education.” Acm.org, Mar. 2014,