Instructions

Final Report

COMS 6901 E sec 028

Spring 2018

Yuval Schaal

Jen-An Lien

May 8, 2018

Table of Contents

Abstract

Online games have become a huge part of today's' society. Many games use tutorials at the beginning stages of the games to aid the players in learning how to play the game. In the article First Five Minutes [1], Jarvinen discusses the importance of tutorials in games and how cold starts do not spike the user's interest to continue playing. He describes how tutorials can be used as metrics to measure the drop off rate, where players who have not completed the tutorials do not continue playing.

In this paper, we will describe our contributions to the Gameful Constructionism (Epic #4) aspect of SAGE (Socially Addictive Gameful Engineering) [2]. First of all, we will discuss computational thinking, the concept of an instruction in sage, and objectives we set to work on during the semester. Then we mention related works that pertain to tutorials and instructions other gameful educational systems have been using. We move on to discussing in details our accomplishments for this semester, followed by future work we see instructions can be used for. Then, we list out our conclusions and references.

1. Introduction

Since the early twentieth century, video games have become part of the daily lives of many individuals such as 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 semester, we have been working on a feature that will aid the instructors and students navigate the SAGE platform more easily. This feature is the instruction. The idea behind the instruction for SAGE has two main aspects. First, to allow instructors to create instructions for their games, to help students along the games the instructors created. Second, is to guide the instructors through creating different type of games (parsons/ cvg). Using these two aspects for the main idea of the instruction we went ahead and created specific objectives for implementing the instructions.

The objectives we came up with included the ability for an instructor to create an instruction while creating the game, enable the instructor to edit and update the instruction, allow a student user to be able to see the instruction when entering a game. Other goals included enabling a researcher user to create instructions for specific game types that will show up when an instructor type user goes to create an assignment for their students.

2. Related Work

Recent years, lots of app-development startup companies decided to choose young children as their targets [3]. These companies develop websites called digital playgrounds, which help children learn programming languages in a creative and funny way.

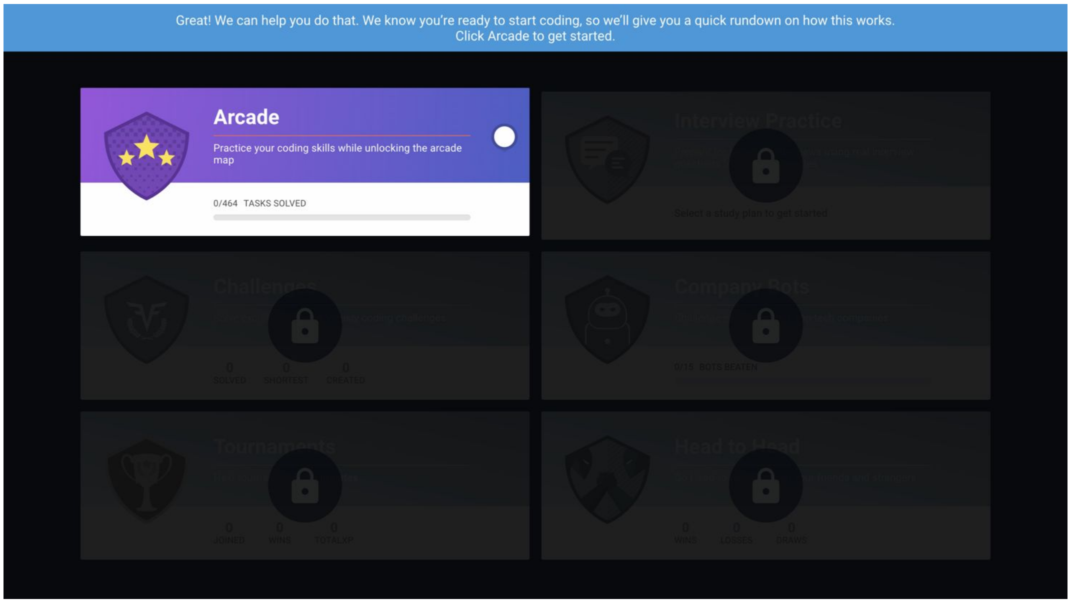
In addition, digital playgrounds tend to provide instructions for beginner, to help them get familiar with the system or games in very short time. In previous research[4], scientists found that entertainment and learning instruction can effective one's learning ability. They did research on people and gave them instruction to test if they can do the quiz better. As a result, people who were given instruction did a better job in the quiz. Using these concepts, digital playgrounds help people know deeper about what the game should be play by giving a clear instruction in their website.

2.1 CodeFight

CodeFight is one of the best examples in digital playground. It tries to help anyone who is interested in computer science to acquire CS knowledge in a very short time. Providing battle mode to compete with others helps users to get familiar with algorithm because it asks users to solve an algorithm problems within the given time. Users can choose different types of stages they want to accomplish, and stages become more and more challenging when users beat others and get more XPs in that stage.

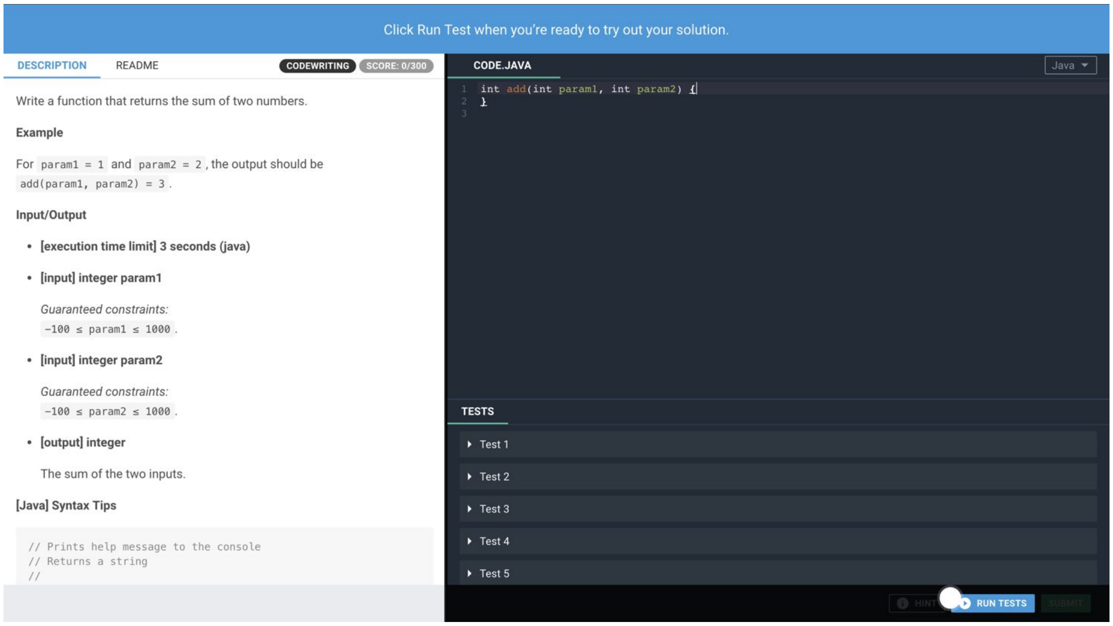
2.1.1 Instruction for Beginner

Traditionally, users look through the plain text instructions to know how to use website properly. CodeFight provides innovative way for users to get familiar with their system without looking through lots of plain text instruction. In CodeFIght website, it highlights some part of components they want to show to users in the page or use the pop-up dialog box to give the beginners clear guidance about the way to use the website. Normally, these hints are given only in the first time when players used the page. They assume that users should be get familiar with the page quickly. Screenshots below 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.

Figure1 : Highlighted instruction for beginner in CodeFight website [4]



Click Run Test when you're ready to try out your solution.

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

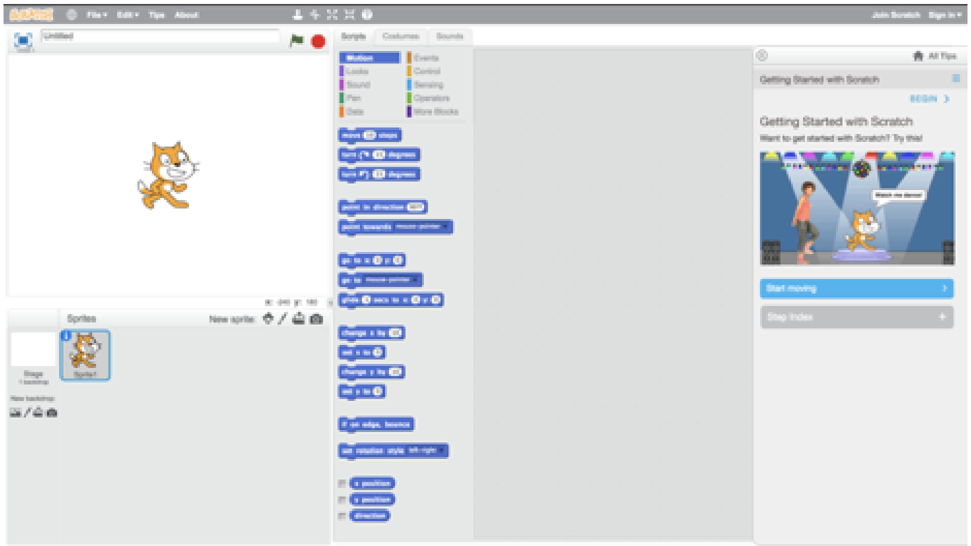
```
// Prints help message to the console
// Returns a string
//
```

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

Figure 2: Instruction for use 'how to run tests' in CodeFight website [4]

2.2 Scratch

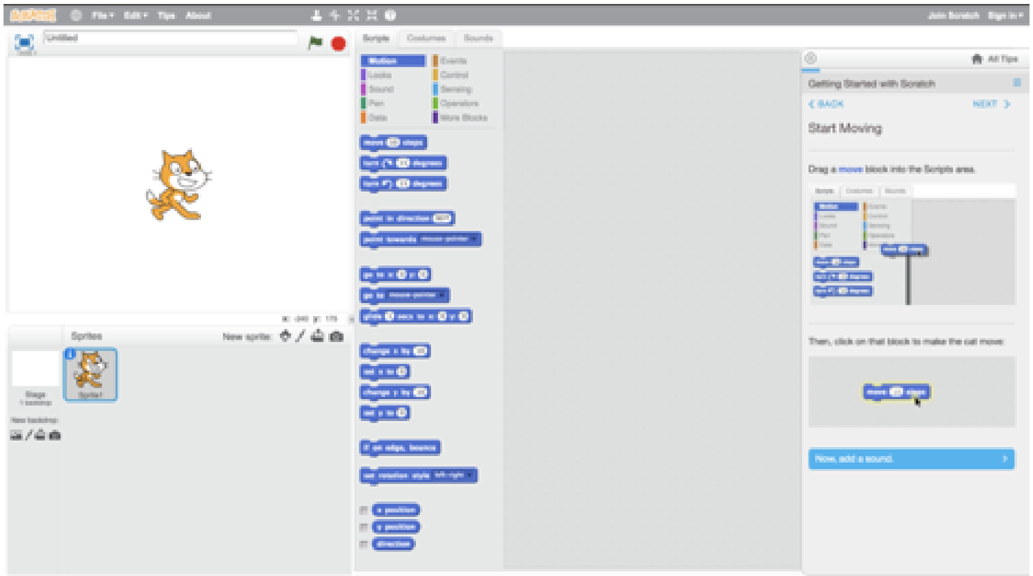
Scratch is a website that help children learn coding concepts by creating games, stories or animation without using any text-based programming language. It introduced blocks that stand for different operations in programming and children create their own project by assembling these blocks.



Getting Started with Scratch

Want to get started with Scratch? Try this!

Figure 3: The main page of Scratch [5]



Start Moving

Drag a move block into the Scripts area.

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

Figure 4: Instruction for building a project in Scratch [5]

2.2.1 Instruction for Beginner

Unlike CodeFight, it tries to blacken the unnecessary part and highlight the part it wants to show to the players. Scratch provides many animate instructions to help players to get familiar with how to build a easy Scratch application step by step. The biggest difference between them is while CodeFight only guides 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 Scratch project step by step, and players realize that they actually can finish a fancy within few steps.

3. Accomplishments

This semester, we focused on building a system that help instructors to easily build an instruction for students to follow while they go through a game. Also a way for researchers to build an instruction for instruction to follow while creating the games and assignments. We used AngularJS Material [6] to ensure the consistency of the visualization for the frontend pages, and help others who will contribute this project by giving them a template to follow. We also implemented the loading service on the instructor, student, and research side. This avoids the errors or unexpected results that asynchronous function may cause, discussed later in section 3.6 in more detail.

3.1 Instruction Creation

For instructors or researchers, they can create an instruction when creating a new game. When users are creating game, they would be asked to specify the name of game, the name of instruction and the type of game type. Instruction can be multiple steps, so instructor can choose to add steps or add specific steps (add steps between steps) in their instruction. For each step, it allows a heading, description, and image.

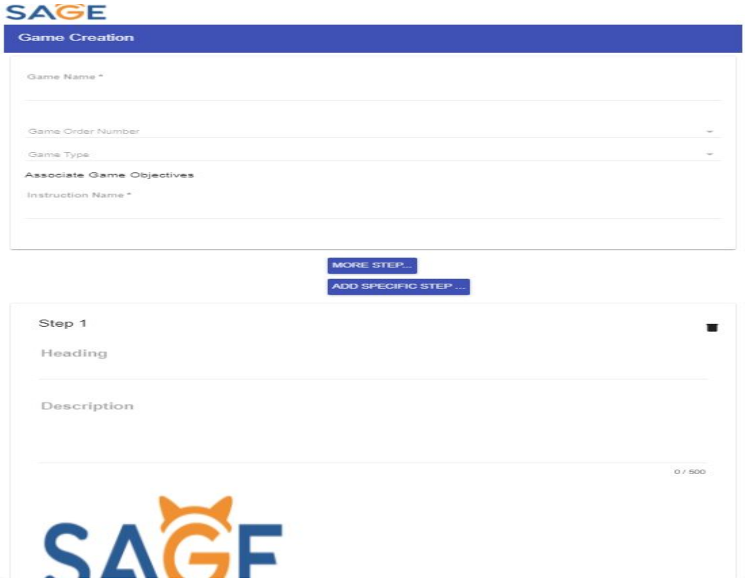
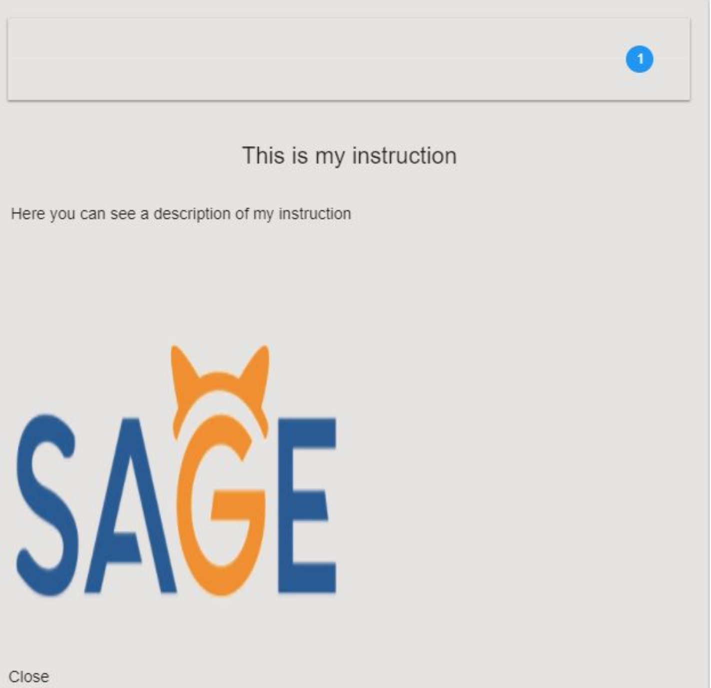


Figure 5: Instruction Creating View

3.2 Preview Instruction

When creating an instruction, many steps may be included. But instructors or researchers cannot imagine what instructions would be look like after their creation. Thus, we designed preview instruction for instructors and researchers. When adding any headings, texts or images, they easily see the instruction by clicking the ‘preview instruction’ button, and before submitting their instruction, they make sure the instruction is what they want to present to students or instructors.


1

This is my instruction

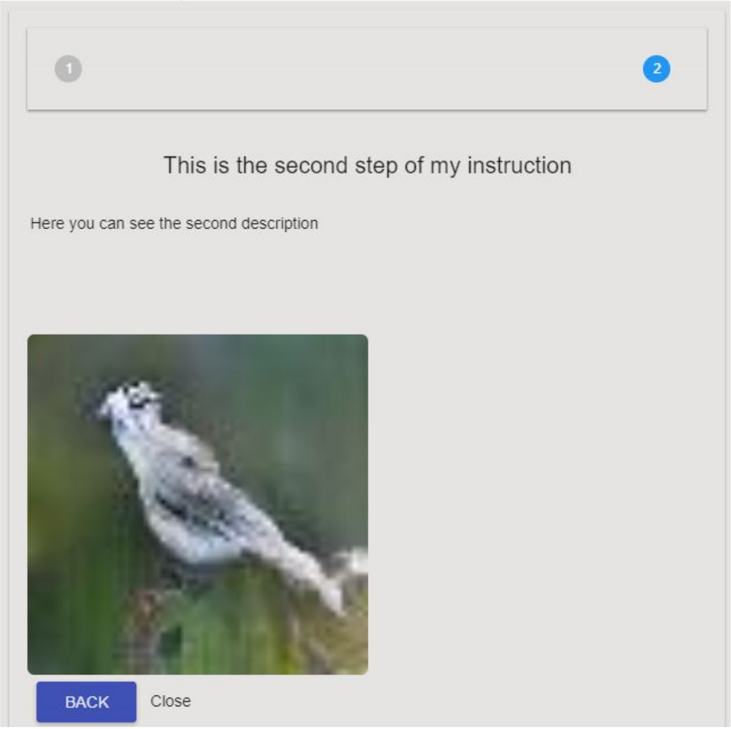
Here you can see a description of my instruction

Figure 6: Preview of a created Instruction

3.3 Adding Instruction Steps

Inspired by Scratch, we implemented instructions with steps which can be controlled by ‘continue’ and ‘back’ button. When users finish reading and understanding instruction in one step, they can click ‘continue’ button to see instruction in next step. When users want to check again about the previous instruction, they can click ‘back’ button to see the previous instruction.

Also, in order to save largest space for games in the web page, we created instruction that users can drag to any place in the web page, and closed the instruction if they do not want to see that instruction anymore. By clicking ‘instruction’ button in the web page, instruction will be shown again.



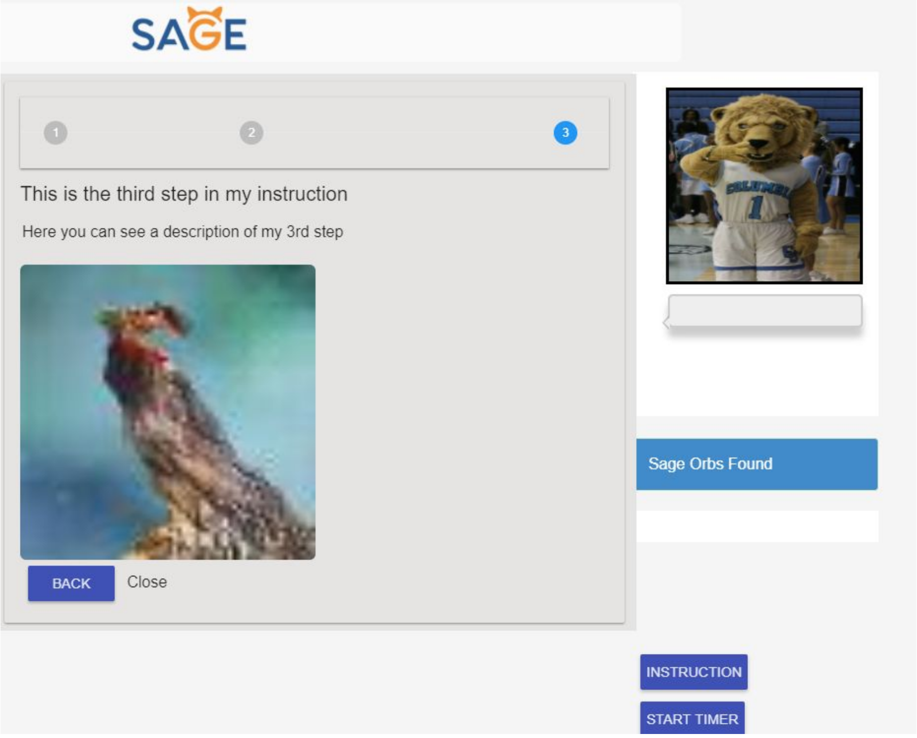
This is the second step of my instruction

Here you can see the second description

Figure 7: Preview of Instruction with Multiple Steps

3.4 Viewing Instruction

When instructors or students finish creating the game with instruction, students will see the instruction in the game and instructors will see the instruction provided by researcher. We stored every data about game and instruction in mongoDB, and designed many endpoints for any frontend developer to easily access.



This is the third step in my instruction

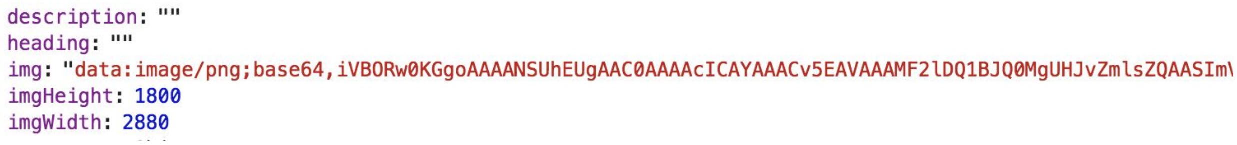
Here you can see a description of my 3rd step

Figure 8: Instruction Preview Student View

3.5 Image Encoding and Decoding

In the earlier version of SAGE, developers tended to store images in the server side. Whenever developers needed them, they loaded static file in the server side. But for instruction, many instructors or researcher would upload lots of instructions with many steps and images. And the way how developer stored images will become an issue.

Instead storing every image that users upload, we tried to convert the image into binary strings. However, another issue arises. The binary string was too long to send to server, especially for the picture with high dimension. Thus, we used base64 to solve this issue. To encode the original binary string to the format of base64 and sent that new string to the server side to store in the database. When anyone wants to get any image, they can get that string and convert to the original image.

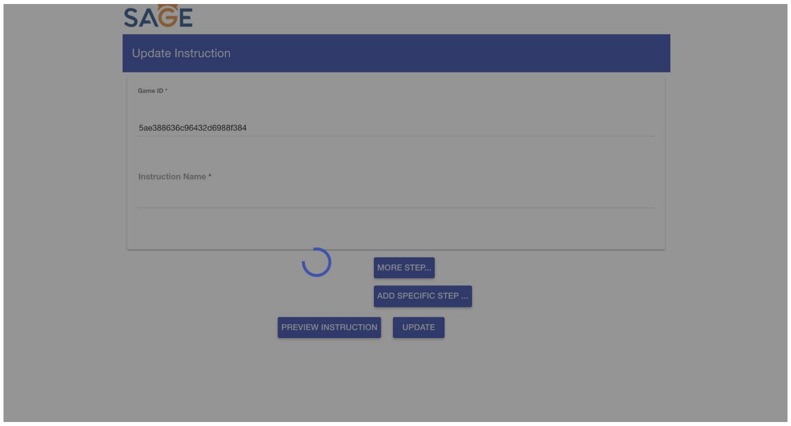


```
description: ""
heading: ""
img: "data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAC0AAAAcICAYAAACv5EAVAAAAMF2lDQ1BJQ0MgUHJvZmlsZQAASIm\
imgHeight: 1800
imgWidth: 2880
```

Figure 9:Encoding Code Snippet

3.6 Loading Services

When calling asynchronous functions such as POST, GET, the time that users get their needed data depend on the latency of server. The latency causes some problems such as clicking another button or any element in the web page that trigger some other functions while waiting for the finish of asynchronous functions, it sometimes leads to the unexpected results. Thus, we designed loading services that can be used everywhere in SAGE. Before starting asynchronous functions, the loading services starts to wait until the finish of asynchronous functions. Once they finish, we can call loading service to stop. This make sure asynchronous functions works well and avoids some unexpected mistakes.



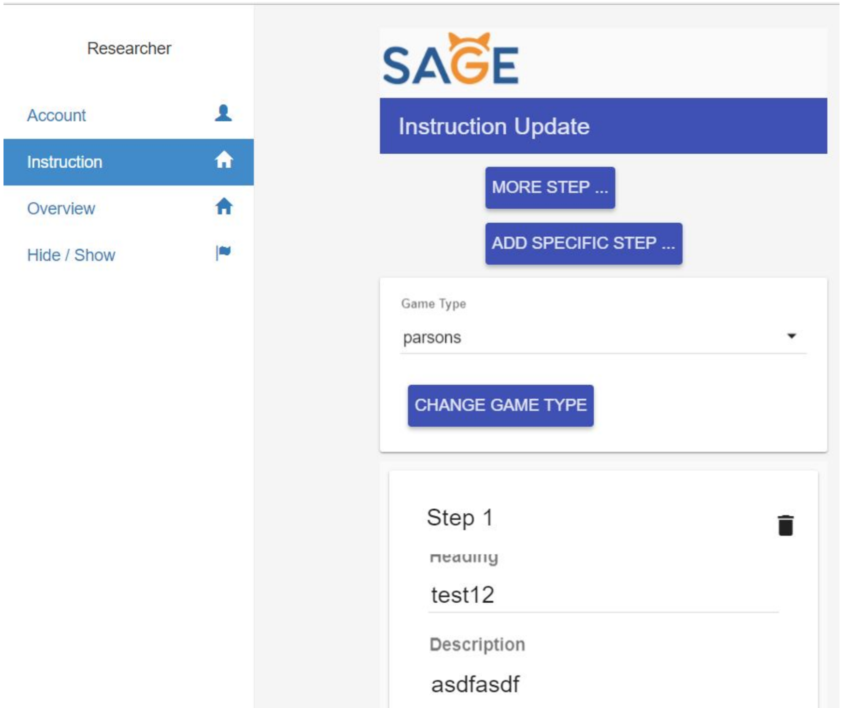
Game ID *
5ae388636c96432d6988f384
Instruction Name *

Figure 10: Loading Service Working

3.7 Adding Instructor Side Instruction

Instructions can also be created for instructors as a guide for creating an assignment. To do this, a researcher type user has to log in and go to the instruction section as seen in Figure 11. From there, the researcher can create an instruction for a specific game type (either

Parsons or CVG). They would choose the game type and edit that instruction for that game type. Then, the instruction would be seen by instructor type users when they begin creating an assessment,



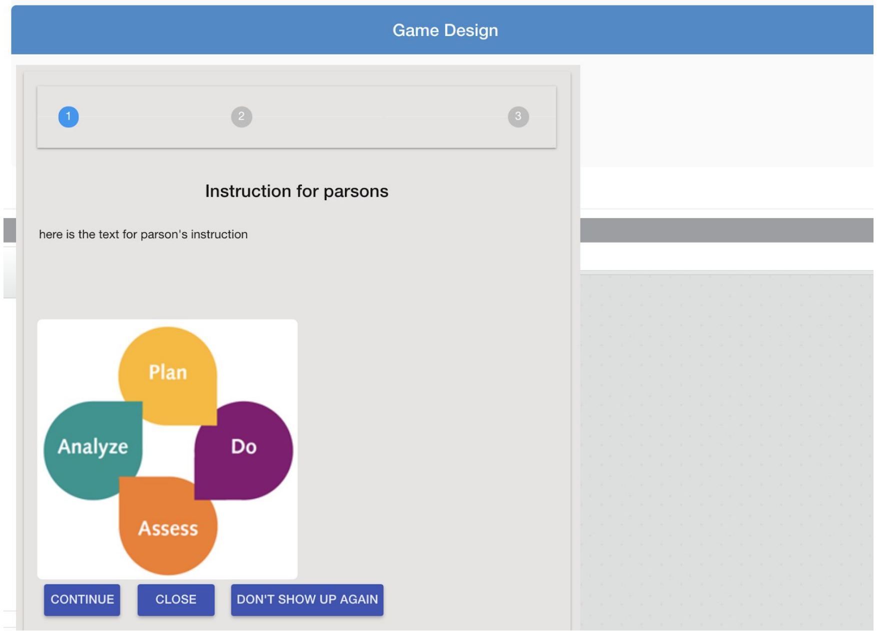
Game Type
parsons

Step 1
Heading
test12
Description
asdfasdf

Figure 11: Researcher view creating instruction

3.7.1 Parsons

When an instructor goes to design an assessment within a game, an instruction correlating to the game type appears, as shown in figure 12 for the parsons game type. The instructor can click through the instruction steps to view everything, or simply close the instruction to hide it from view. The instructor also has the option for ‘Don’t show up again’ which will disable the popping up of this instruction when the instructor user begins designing the game for a couple of days.



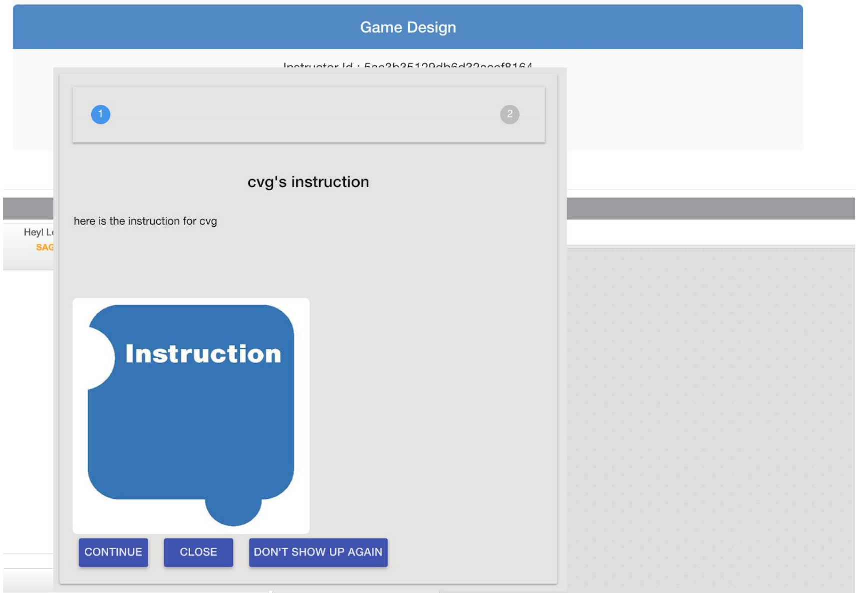
Instruction for parsons

here is the text for parson's instruction

Figure 12: Instructor viewing instruction created for parsons game

3.7.2 CVG

For a different game type, a different instruction will be displayed to the instructor user when they design an assessment. As can be shown in figure 13, the CVG content differs than the parsons type game instruction. This, as previously mentioned, can be managed by a researcher user in the instruction section.



cvg's instruction

here is the instruction for cvg

Figure 13: Instructor viewing instruction created for CVG game

4. Documentation

Documentation for all endpoints created for instructions as well as the instruction model, loading service, and mongodb image to binary string conversion can be found here: https://gudangdaya.atlassian.net/wiki/spaces/SAGE/pages/390758415/Instruction+Documentation

5. Future Work

Future work that can be done relating to SAGE instructions that will improve the SAGE platform as a whole include creating real instructions for the instructors, content such as explaining how to use the SAGE platform as well as differentiation between creating different types of games such as CVG or Parsons using instructions. This will allow for instructions to actually be used correctly in the system to explain usage of the platform.

Taking instructions created by researchers and going to real users of the SAGE platform will also be useful. This will allow future researchers to know what is best needed and wanted by SAGE platform users to be able to use the system more easily. The testing of the instructions with real users will be extremely beneficial to the system as a whole. Once the real instructions are created, this testing step can be done to experiment with the best ways to get users used to working and inside the SAGE platform.

6. Conclusion

The contributions we have made this semester will be extremely useful to SAGE and its future user base as a whole. The SAGE platform now contains instructions for students to be able to be guided through assignments created by their instructors. Researchers are now able to provide guidance for instructors to use and create different types of assignments on the SAGE system. Throughout the process of creating these items, a loading service has been created for use by future researchers as well. We also know how to store images now in MongoDB by encoding and decoding binary strings. All these items and endpoints are documented as shown in the preceding section (4. Documentation) so that future researchers can understand the contributions added and items implemented with ease.

References

[1] Jarvinen, Aki. “Introduction: Tutorials and the Freemium Business Model.” Gamasutra - First Five Minutes: How Tutorials Make or Break Your Social Game, www.gamasutra.com/view/feature/132715/first_five_minutes_how_tutorials_.php

[2] 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.

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

[4] Erhel, S., and E. Jamet. “Digital Game-Based Learning: Impact of Instructions and Feedback on Motivation and Learning Effectiveness.” Egyptian Journal of Medical Human Genetics, Elsevier, 13 Mar. 2013, www.sciencedirect.com/science/article/pii/S0360131513000699.

[5] https://codefights.com/

[6] https://scratch.mit.edu/projects/editor/?tip_bar=home

[7] https://material.angularjs.org/latest/