Gamification Elements and Improving Real-Time Assessment in SAGE

Project Proposal

Yu-Chun (Julie) Chien | yc2937
Ray Tsai | ()
Iris Zhang | iz2140

COMS E6901, Section 14
Fall 2016

Table of Contents

1. Motivation and Goals

In 2002 MIT came out with their first iteration of Scratch. Since its release, Scratch has become popular among educators as a stepping stone towards teaching advanced computer programming concepts. It boasts 11 million unique visitors monthly, a testament to its ease of use and low entry barriers. As a result, educational researchers sought ways to improve for the platform as a tool for both students and teachers. Social Addictive Gameful Engineering (SAGE), first imagined by Bender in early 2015, sought to "tooling Scratch with a collaborative game-based learning system that catalyzes teachers and (grade 6-8) students to immerse in computational thinking." This project is a continuation of the implementation of SAGE, adding game design elements and reward mechanisms to motivate students' learning through gamification.

The project will focus on three driving factors: (1). Improve upon the Formative SAGE Assessment architecture through creating a HTTP API. (2) Scaffold Computational Thinking through real-time feedback within the Scratch Editor student view. (3) Motivate students through gamification, with a stretch goal of providing a visual competitive element through ranking.

2. Relevant Work

We reviewed relevant work in two stages. The first part involved assessing the project from a technical standpoint. To get started, each project member took a look at the existing Scratch tool available publicly for students to get a sense of the tool we aim to improve. Then we viewed existing Scratch projects (https://scratch.mit.edu/projects/92837313/) to understand how educators may possibly use Scratch as a teaching tool. This process helped guide our understanding of the motivation for the creation of SAGE, as well as form our first impressions of the technological possibilities. The paper "Formative SAGE Assessments" (Pava 2016) informed us of the state of the current SAGE project, outlining specifics of what technologies have been developed as well as suggestions for further expansions.

The later stage involved reading more theoretical papers about learning theory and computational thinking to inform our understanding of the foundations for SAGE. "Gamification in Education" (Lee & Hammer) described certain aspects of games that make them fun, in particular engaging players on a cognitive, emotional, and social level. Gamification bridges the gap between game elements with education by providing students with "opportunity to experiment with rules, emotions, and social roles." The paper later goes on to give various examples of how mundane school tasks can be reframed in a fun way. "Tooling Scratch" tied all this together by identifying "four learning principles which guide the formulation of design tactics fit for tooling Scratch with a collaborative game-based learning system that catalyzes teachers and students to immerse in computational thinking" (Bender). Finally, "Micro Adaptivity" provided

some insight into how to balance gaming and learning, while “Native Programmer Confusion Achievement” emphasized the critical need to correct confusion immediately after discovery. Both informed our design of Scratch enhancements which we hope addresses students’ learning needs as soon as possible without distracting them from the overall “game” experience.

3. Proposal

Our project will include two main components. One is additional gamification of the Scratch game-building process. The second part is a suite of enhancements to the real-time assessment server.

3.1 Game Design Elements in SAGE

We will add elements of gamification to the Scratch editor to make things fun and to encourage students to go in certain directions. The improvements will include real-time feedback for students as they build projects using Scratch. As students build projects, the Scratch editor will respond to student activity in a teacher-defined way.

Each project may be divided into levels, breaking the subject matter into distinct chunks. Each chunk will include assistance for students as they complete activities. This scaffolding will guide learners and provide a supportive learning environment for students. [1]

3.1.2 Reward Mechanisms

We will integrate reward mechanisms into the Scratch project creation process to gently encourage students to achieve teacher-defined milestones.

3.1.2.1 Points

A teacher may specify that a block be associated with a point value. For example, blocks that represent more complex programming concepts may be worth more points than blocks that represent simple concepts. In this scenario, teachers will encourage students to get more points by using blocks with greater point values.

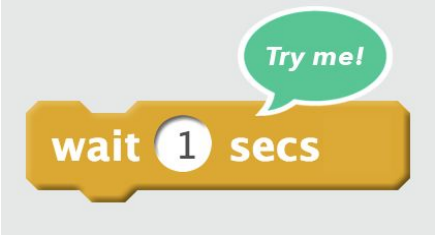
Alternatively, perhaps a teacher wants students to program concisely and elegantly, and she does so by encouraging students to complete the level with the *lowest* point value possible.

Points will be displayed on each block. Blocks not explicitly assigned a point value will have a point value of 0.

Points displayed on blocks

3.1.2.2 Indicators

Indicators on blocks, displayed as callouts with text, can be used to suggest to the student to use the block. Indicators can be specified by the teacher to pop up at specific times. For example, when a student is stuck on a certain level, an indicator may pop up to hint to the student which block to use. These can be used with or in lieu of points.



Indicator on a block

3.1.2.3 Trophies

Trophies may be awarded after every level for using a certain number of points, using blocks that demonstrate important concepts, or achieving other milestones.

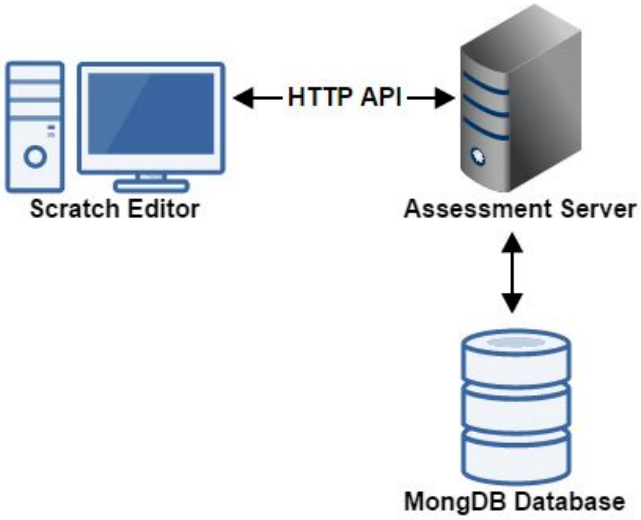
3.1.2.4 Leaderboard

A leaderboard that shows students with top scores will be updated as students complete levels. The motivation for building such a tool stems from studies that show that competition is a gamification element that adds a social and emotional element to learning. Although this is a reach goal for us, we have put thought into designing this feature to mitigate risks of negative

student outcome [2]. The leaderboards can anonymize personal information, leading to students only knowing how they individually achieved relative to the class.

3.3 Architecture

We will create a HTTP API for the SAGE assessment server which allows Scratch modules to communicate with the assessment server in real-time. In other words, the Scratch modules will be able to send and receive information from the assessment server while the module is running. The Scratch editor will also require new communication blocks in order to utilize the assessment server's API.



Completing the assessment server API will allow us to create infrastructure for a logging system on the assessment server. We will need to analyze the type of data to be recorded and enhance the MongoDB database to allow for storage of the logging data. Once the infrastructure is in place, the Scratch editor blocks will be updated to make use of the new logging functions.

4. Project Plan

4.1 Game Design Elements Milestones

Date	**Milestone**
Oct 7, 2016	Development environment setup
Oct 14, 2016	Design of game elements

Oct 28, 2016	Implementation of real-time Scratch editor feedback
Nov 11, 2016	Implementation of point/trophy system Communication with backend architecture
Nov 25, 2016	Implementation of leaderboards
Dec 9, 2016	Improvements

4.2 Architecture Milestones

Date	**Milestone**
Oct 7, 2016	Development environment setup
Oct 14, 2016	Design of API endpoints
Oct 28, 2016	Implementation of API endpoints
Nov 11, 2016	Implementation of Scratch editor communication blocks
Nov 25, 2016	Analysis of data to be logged
Dec 16, 2016	Implementation of database and logging

4.2 Roles

Back-End Architect

Ray will work on designing the infrastructure to support the SAGE Assessment Server.

Game Designers and Developers

Yu-Chun (Julie) Chien and Iris Zhang will work together on gamification of the Scratch Editor.

5. Works Cited

[1] https://www.knewton.com/resources/blog/ed-tech-101/scaffolding-instruction/

[2]
https://theses.lib.vt.edu/theses/available/etd-06302012-162750/unrestricted/Swab_AG_D_2012.pdf