Midterm Report

Eleanor Murguia, Zoë Gordin, Cherie Chu, Veronica T. Woldehanna

Abstract

In this report we will describe the progress made in Gameful Direct Instruction, Field Study Operalization, and Learning Metrics. The features and stories we have completed so far are: (482) Game Instruction Image Aspect Ratio, (481) Managing Quests in Missions, (478) Quest Edit | Description Size, (466) Create Mission - Save Functionality, (474) Left-nav Section Removal | Instructors, (483) Game Design Display & Instruction Behavior, (476) Class Mission Metrics, (468) Point Deduction On Drop, (467) Sign-In Disable. Features and stories in progress are: Game-objective Editor UI (455), Instruction Editing Display (484)

Architecture

The architecture for the student learning metrics consists of aspects from sage-frontend, sage-node, and sage-scratch. The current state of this architecture is that sage-node runs the Hairball analysis on an arbitrary sample sb2 file, sage-frontend displays the metrics that are drawn from the database, and sage-scratch manages the submission of the game and uploads the results to the server. The goal for the learning metrics architecture is for sage-frontend to send the contents of the sb2 file for a certain student and game ID to the database, and sage-node to get the contents of the sb2 file, run Hairball on the file, and then send the results of Hairball to the database.

The architecture for field study operalization consists primarily of sage-frontend. Sage-frontend currently operates with a Model-View-Controller pattern, with angular controllers and HTML views.

Implementation

Currently, hairball is only implemented to run on a dummy sample file once a game is submitted. In order to implement Hairball metrics on the student's actual submission, the sb2 information will need to be parsed from the HTTP response on the frontend, and the database will need to be implemented in an automatic way, rather than a manual way. This should be possible through adding a fields to student models that include their sb2 files, and recording this in the database whenever a student submits a game.

The goal of field study operalization is to fix bugs on sage-frontend that prevent basic usability of SAGE. We have focused primarily on bugs on the

instructor side of SAGE that prevented instructors from creating games, managing missions, and generally interacting with the app in a easy and intuitive way. Most of these bug fixes have involved modifying Angular controllers to prevent errors. For example, we modified the mission management page to prevent users from adding the same game twice to a mission.

Limitations and Assumptions
Learning Metrics:
- There is a disconnect between the metrics data on the backend (sage-node) and the frontend (sage-frontend).
  - The data that is generated from running Hairball is not stored in mLab.

```
function updateEveryHour (studentId, gameId) {
  var sb2File = getSb2File(studentId, gameId);
  hairball(sb2File)
    .then((results) => {
      console.log('test test test!!!', results);
      // TODO: upload to mLab — discussion: update entity (read entity, delete entity, write entity)
    })
    .catch((err) => console.log(err));
}
```

  - All the data currently visible from running sage-frontend is dummy data manually inserted into the database.
- Hairball is triggered upon running, but it does not run on the sb2 of the submitted game; it is currently hardcoded to run on a predetermined sample file.

```
function getSb2File (studentId, gameId) {
  // search sb2 file ??? storage — save a sb2 file without any association to studentId and gameId
  return '/Users/ruiminzhao/Desktop/SAGE/sb2/simple.sb2';
}
```

- The sb2 file created from the Scratch game is not saved, only logged to terminal; the file that is supposed to be created upon submission is not created correctly (this currently does not matter because there is no attempt to run hairball on the incorrectly created file)
  - In order to evaluate performance, a file must be created to run Hairball on it.

Future Work
Learning Metrics:
- The goal for the remainder of the semester is to bridge the gap between the backend and frontend.
  - First, we have to create a new sb2 file upon every game submission.
  - Next, we need to find a way to run Hairball on the new sb2 file after its creation.

- - Last, we need to send the data generated from Hairball into mLab, so that the frontend can display the correct metrics.

Field Study Operalization
- The goal for the remainder of the semester is to further enhance the user experience, on both the instructor and student sides of SAGE. For example, we plan to implement a change to the objective and game models, such that the human-readable name of both is stored and can be rendered for the user.