# 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 and sage-scratch. 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 and sage-scratch that prevent basic usability of SAGE. In sage-frontend, 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.

The issues we've been dealing with in sage-scratch relate to a student's user experience, an instructor's user experience, as well as basic functionality and usability of Parsons Puzzles.

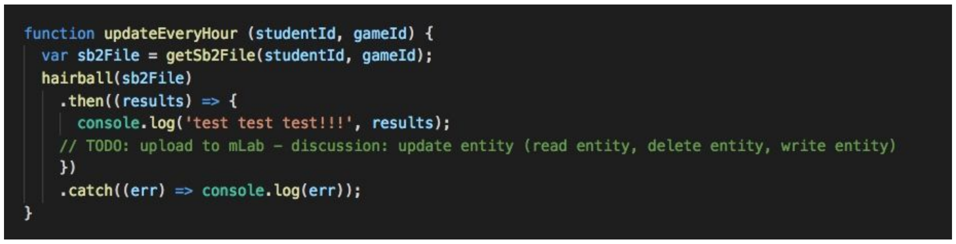
**Pausing parsons puzzle live scoring:** Currently the way we score parsons puzzles breaks down when tested with multiple unexpected scenarios. The scoring logic was designed with the assumption that students would work on their solutions linearly(top down) and correct their mistakes right away as prompted by live feedback from the scoring logic. However, this is not always the case. Ihantola and Karavitra (Two-Dimensional Parson's Puzzles, 2011) mention that even programming experts don't write programs linearly (i.e line by line). Even though it's easy for a student to insert a block between two other blocks in the current implementation of Parsons Puzzles, our live score feedback does not handle this situation. To remedy this, we've implemented this feature that essentially pauses the scoring logic and allows the student to build their solution without their score changing or their solution executing at every step. Once the student is ready, they can play the scoring logic again to see how close they are to the solution.

**Playing correct solution:** In addition to showing them an execution of their solution, we now show them an execution of the correct solution to make it easier for them to compare and adjust their solutions. The button for this has yet to customized. In addition, since the executions are very fast, having a watermark of the sprite in the initial location of the sprite would help students and provide a frame of reference for the final location of the sprite according to the student and instructor solutions.

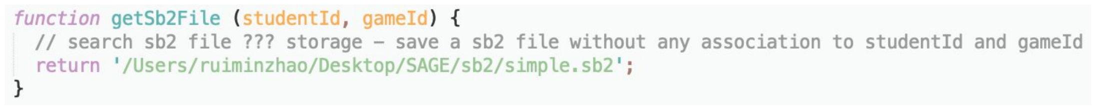
**Limitations and Assumptions**

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.

Pausing scoring:
- Since the students will no longer have immediate feedback to correct their actions, pausing scoring my increase their cognitive load.
- The scoring logic is designed so that, for example, a student with a completely wrong solution and a student with a solution 2 swapped blocks away from the correct solution, could get the same score. If these scores are directly used for any type of data analysis, they could lead to false results.

Bugs:
- There are bugs, like loading time for the game taking too long in certain instances, that have been hard to solve and might take more time to solve.

**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.

Improve parsons scoring:

- The goal would be to find an efficient algorithm that can differentiate between a completely wrong solution and a solution 2 swapped blocks away from the correct solution and still provide live feedback. We can perhaps use heuristics like the sum of manhattan distance of all the blocks from their correct positions or dot products to get a score on solutions while providing the feedback messages determined through the logic implemented currently.

Subgoal labeling and providing structure in parsons puzzles:

- Ericson, Margulieux and Rick in their paper “Solving Parsons Problems Versus Fixing and Writing Code” the advantages of subgoal labels to decrease cognitive load and help incorporate new information into existing knowledge schemas/ mental models.
- Providing what could be described as simple skeleton code to provide beginning structure could help in the same way.