SAGE Midterm Report: Spring 2021

Lalitha Madduri, B.S. Candidate in Computer Science, 2021
Columbia University
Supervised by Jeffery Bender, Gail Kaiser

***Abstract*—The main purpose of this paper is to provide a review of work done until the midpoint of the semester with regards to progress with SAGE in the Spring 2021 semester. In the field studies area, the main objectives included iterating on the field study data obtained in previous semesters, pre-processing the data into an easy-to-parse format for data analysis, and setting up the parameters for future field studies. Progress on the SAGE platform itself included Parsons Puzzle authoring across conditions, instruction authoring, and clarification in the existing puzzle database. Bringing the currently authored puzzles to test was also a priority that is currently underway.**

I. Introduction

THE objective of SAGE (Social Addictive Gameful Engineering) is to provide teachers a platform to design puzzles emphasizing computational thinking concepts for students to complete self-motivated and rewarding manner. Computational thinking (CT) can be framed as a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use. Thus, SAGE places the programmer, rather than the computer, at the forefront in its learning environment. The user experience is a key component in achieving this goal, and assessing whether the student feels at the forefront must be done through thorough field studies involving both the students and instructors. Field Study 1 (FS1) concluded and was submitted for publication at the 26th annual conference on Innovation and Technology in Computer Science Education (ITiCSE), but was eventually not accepted for publication. However, valuable feedback was provided on the paper, and such feedback will be of use when moving forward with FS3. The goals for this semester that are currently underway are creating protocol content for looping CT concept for FS3, finalizing the condition set and any participant demographic constraints, and creating games within the platform across conditions with a gameful, thematic edge. Another key feature of SAGE improved upon this semester are the Parsons Programming Puzzles and associated instructions. We want the students to need to read the minimal amount of instructions in order to solve the puzzles, and shift the focus from instructions to clear directives. Thus, it was important to determine if any variation of instruction needed across conditions.

II. Completed Work

II.i. Finalization of Condition Set

For Field Study 3, the condition set and any participant demographic constraints were finalized in late February. These conditions will cover the looping Computational Thinking (CT) concept that will be assessed in FS3. There were 16 conditions in total. In terms of notation, ”PPP” refers to Parsons Programming Puzzles, ”d” marks distractors, ”o” marks objectives, ”ScratchIE” marks Parsons Puzzles without a palette, and ”CPP” includes the Constructionist Video Games. Below are the 16 conditions:

1) PPP 2) PPPd 3) ScratchIE 4) ScratchIEd 5) PPPo 6) PPPod 7) ScratchIEo 8) ScratchIEod 9) CPP 10) CPPd 11) CPPie 12) CPPied 13) PPP, PPPd, PPPo, PPPod, CPP, CPPd 14) ScratchIE, ScratchIEd, ScratchIEo, ScratchIEod, CPPie, CPPied. 15) PPP, PPPo, ScratchIE, ScratchIEo, CPP, CPPie 16) PPPd, PPPod, ScratchIEd, ScratchIEod, CPPd, CPPied

Additionally, there is a control condition that spans the computational thinking concept of ”sequences.” 6 puzzles need to be constructed for each condition, including one familiarization puzze, so 7 per condition in total, leading to a total of 112

puzzles that need to be authored for FS3.

We also need to determine if game instructions need to vary across conditions. We might consider configuring the CVG condition so that explicit step-by-step instruction is not necessary, if game objective feedback offers sufficient progress and completion guidance, and move-based PPP feedback is disabled. This configuration would optionally facilitate an alternative approach to looping instruction in which the participant is encouraged to construct a sequence first, then focus on the loop. In PPPs, it is less confusing when the participant is guided to build the puzzle from top to bottom, since when puzzles are solved that way, the result is a positive flow of correctness feedback and an increasing score. The adult instruction warm-up solutions provide us with the appropriate reference for this section.

II.ii. Parsons Puzzle Authoring Across Conditions Parsons puzzles are a relatively new type of practice problem aimed at helping novice programmers [1]. Each puzzle is made of a plain English description of a program's requirements, accompanied by its implementation. This "reference solution" is then scrambled, usually by isolating each line of code into a "fragment" and shuffling them. This paper examines the effectiveness of Parsons Programming Puzzles in teaching CT concepts through independent learning.
Originally in the semester, there were issues with the environment as Flash Player is no longer supported on browsers, and there needed to be an additional hotfix that enabled Flash to run retroactively on the browser. Furthermore, there were obstacles preventing the successful loading of Parsons Puzzles from previous field studies due to an issue on the SAGE platform that was also eventually resolved, allowing for puzzles to move from being authored on documents outside of the platform to being authored on the platform themselves.

The goals of puzzle authoring include:
1. Author puzzle programs with motivating scenarios
2. Author puzzle programs with memorable segments
3. Provide a challenge without being tricky
4. Leave the users with a positive impression

As such, for the looping content, the theme of "Training for a 5K" came into creation. Prior to the 5K training concept, a "Bank account" theme was developed that involved using loops to see how interest, compound interest, saving, and spending came into play. This was mainly because of SAGE's focus on adult learners, but this concept was abandoned due to multiple computational thinking concepts coming into play (variables and looping), leading to potential confusion within users.

The 5K Training concept begins with a warm-up that asks the user to make the sprite complete 3 jumping jacks, providing instructions on how to complete a jumping jack. A new concept, costume changes, is introduced in this looping concept within the "Looks" pane, visible within the user's PPP when he or she starts. This concept intends to make the play more gameful for users, as there is more variation with animation on the screen than prior puzzles which simply utilized the "Say" block in the "Looks" pane. The following puzzles iterate on this concept with differing varieties of loops and complexities. For example, another puzzles utilizes mainly "Motion" pane blocks and a repeat until that involves the sprite reaching a finish line. Further puzzles include making a sprite run around a track, and then running around a track while doing a jumping jack/squat sequence at each corner.
Functionality for game objectives has been largely implemented since the beginning of the semester, which is at the forefront of authoring as well. Game objectives and distractors are new for FS3, and in previous field studies, the only conditions tested were feedback inclusion and palette inclusion.

II.iii. Preliminary User Centered Design Sessions

Before planning more intentional user-centered design sessions for FS3, I plan to bring the current looping content to three adult learners who I've identified as potential test users for SAGE. The class and roster creation functionality will be implemented and the learners will be given separate logins to assess the platform and give feedback

on the looping content across conditions. Learners will begin with PPP (with a palette provided) and also be assessed on the use of distractors and objectives within the looping content, and any difficulties with either the platform, curriculum, or conditions will be noted. These preliminary user centered design sessions will act as guiding forces for more involved user-centered design sessions once looping content across conditions is finished being developed. Furthermore, an emphasis on casual interview to obtain journey maps and pain points during the students' use of SAGE will be noted.

III. Future Work

Much work lies ahead for the rest of the semester. As of late february, CPP, CPPd, CPPie, CPPied are deployed, and the functionality for objectives has advanced significantly enough to implement objectives across conditions within the looping content for FS3. Finishing the 112 puzzles within SAGE for the looping content is the next step. Other goals include crafting a pretest and posttest focused on CT in addition to programming. The qualtrics fs1 pretest posttest and fs2 pretest posttest are available, but are arguably programming-oriented. Some ideas for a more CT-focus pretest and posttest include:

1. No blocks and only text
2. Screenshots of stage with sprite and question related to number of blocks required to reach designated spot
3. Probe understanding of value (select reason for using loop out of incorrect reasons)
4. Identify puzzle in which looping would be useful (two types, 1 text, 1 picture)

Furthermore, we need to identify additional and/or modified survey content and create questions specific to our study for final survey similar to those created for fs2 final survey. Finally, the creation study guide in Qualtrics remains a goal (prolific entry and exit point) similar to the fs2 guide, but with one block per step.

References

[1] Bender, J. (2018). "Social addictive gameful engineering (sage): An intelligent game- based learning and assessment system that infuses computational thinking in grade 6-8 curricula."

[2] K. J. Harms, N. Rowlett and C. Kelleher, "Enabling independent learning of programming concepts through programming completion puzzles," 2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Atlanta, GA, 2015, pp. 271-279.

[3] K. J. Harms, E. Balzuweit, J. Chen and C. Kelleher, "Learning programming from tutorials and code puzzles: Children's perceptions of value," 2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Cambridge, 2016, pp. 59-67.