Parson Programming Puzzles and Pilot Field Study
Final Report

December 19, 2019

Rebecca Cawkwell rgc2137

Emily Li el2895

Sandy Zhao bz2328

Juvaria Shahid js5160

Contents

0.1 Introduction

This report aims to summarize the work that has been made on the summative evaluation of the SAGE system. This has involved preparing all the materials and protocols necessary to complete the study and making contact with two potential host teachers.

Computational Thinking (CT) describes the problem-solving skill of thinking in the way a computer would solve a problem. Today, there are more computing jobs than any other type and not a sufficient number of graduates to fill those jobs. It is important to create educational programs that will teach the next generation of students CT in order to meet the needs of our economy. These educational programs need to be accessible, usable and inclusive to students of all backgrounds.

SAGE is a variation of Scratch that provides teachers a systematic way to build students' CT abilities. CT is broken down into concepts that describe various computational methodologies to solving a problem such as loops, conditionals, etc. Each of these CT concepts needs to have various Parson Puzzles designed in SAGE for students of different difficult level.

In order to know the effectiveness of this program, it is important to evaluate the effectiveness and usability of the program. Teachers can be evaluated to find gain insights into how to prepare study materials and the study protocol with students. Teachers have insights into interests of student, comprehension level and timing.

Student evaluations help to test the usability of the system, the effectiveness of Parson's Puzzles and of the possible positive use of distractors in puzzles.

0.2 Related Work

0.2.1 Parson Puzzle Design

How Do Students Solve Parsons Programming Problems? An Analysis of Interaction Traces (Helminen, Ihantola, Karavirta, Malmi)
This study primarily focused on understanding the process of how students work through and solve Parsons programming puzzles. The study was conducted on university students in Finland who were asked to solve five different programming puzzles. Interestingly enough, it was found from this study that the students used the automatic feedback function very sparingly while solving the assignments. The research also provided insight on common patterns that students used when solving problems, such as adding all the code fragments in a linear top-down order or first adding code fragments that defined a block or

control structure. Through this study, they were also able to observe common difficulties among the students and noticed that their solution paths often had loops. The study categorized this looping student behavior as backtracking, circular loops, separate sidetracks, concentrated sidetracks, and jumbled combinations. Though all the students were able to solve the problems, the researchers found that how they approached the problems varied greatly. As such, they believe that the problems should be assessing a student's solving strategy, which is more meaningful and valuable than merely checking their final solution.

Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking (Moreno-León, Robles, Román-González)
In this study, they proposed a web application named Dr.Scratch to facilitate teachers' evaluation of their students' CT proficiency and to provide effective and comprehensive feedback for students in order to motivate learning. Result has shows, through the incorporation of Dr.Scratch, student's overall CT scores were improved. Consequently, their coding skills are also improved. One of the ways that Dr.Scratch use to give effective feedback is by detecting insufficiency of logic. Thus, even students are able to complete the current task, they can still have not mastered the CT concept due to the use of repetitive codes. Dr.Scratch is able to detect that and further enhance the learning experience beyond correctly completing a task. Furthermore, Dr.Scratch, through analyzing the Scratch project, gives different amount of information to students at different levels since they have found too much information can be counterproductive for beginners.

Understanding Computational Thinking Before Programming: Developing Guidelines for the Design of Games to Learn Introductory Programming Through Game-Play (Bacon, Kazimoglu, Kiernan, MacKinnon)
This paper primary focuses on improving CT for first year undergraduates. It emphasises the difference be learning and gaming in GBL and the importance of designing challenges based on individual abilities. Bacon et al. believed that GBL should take into account students' prior knowledge, learning progress, preferences, etc to provide an individualized learning experience for all. They also made the distinction between learning CT concepts and CS concepts, in that mastering CT should enhance one's critical thinking and problem solving skills in any discipline. Moreover, they stressed the importance of receiving clear feedback to help beginner learn from their mistakes, such as learning early on the re-usability of logic and code. They claimed meaningful feedback that are different from compiler messages would also motivate students to reach their learning goals. Lastly, they

thought it's crucial to make all GBL gender and expertise neutral to encourage learning for all audiences.

0.2.2 Field Studies

Distractors in Parson Puzzles Decrease Learning Efficiency for Young Novice Programmers (Harms, Chen & Kelleher)
In this study, Harms, K. et al. studied the difference between learning PP with and without distractors. Overall they found that distractors did not increase their "transfer task performance" and decreased their learning efficiency. They studied three types of distractors: extra noise, familiar suboptimal paths and only one possible solution. They found that extra noise was easy to ignore. Suboptimal paths led to students not creating the correct solution and not understanding why their solution was not correct. Only one possible solution was the answer to this problem. However, in analysis they found that distractors created only extraneous cognitive load which is harmful to learning. They suggest that these results may be different for students who have prior programming experience and in different educational contexts. They also suggest the study of the impact of distractors on the motivation of students.

Enabling Independent Learning of Programming Concepts through Programming Completion Puzzles
This paper focuses on formative evaluation techniques. The group iteratively tested and developed a puzzle interface within a block based environment. It found that participants who learned using the puzzles did 26 % better on transfer tasks and 23 % in the learning phase.

Their formative evaluation was conducted on 23 users (age 10-15). Each session lasted 30 minutes.
The evaluation was broken down into two parts:

1. Designing the programming puzzles:
Through ten iterations of formative testing, 23 users (1-15 years age) were evaluated. Each session was 30 minutes.
Some lessons learned regarding the format and interface:

- Only show the user's work in the program's output
- Limit the editable dimensions of the puzzle
- Limit distractions and focus on the user's attention on the output
- Provide user with ambitious and incremental feedback towards the puzzle's solution

2. Creating the programming puzzle curriculum
The group focused on repeated execution, parallel execution, parallel nested within repeated execution, and repeated nested within parallel execution.
They evaluated 21 participants ages 10-15. Each session was 90 minutes.

0.3 Planning

Preparation

We will build upon the surveys and templates created by Ben and Jady this summer. We will design the field studies based on methodologies that gain the most amount of feedback from teachers. Additionally we will use our resources to connect with potential schools and teachers that we can talk with.

We could potentially work with teachers from the Upperline School of Code. Upperline School of Code employs around 20-30 teachers each summer to teach software engineering boot camps. Most of the teachers teach in schools in New York City. If we are able to secure a couple of teachers, we will have access to several classrooms. In the past, SAGE has connected with PS175, so these could be secondary locations to conduct our surveys for the pilot release.

Parson's Puzzle Design

We originally planned to design 10 puzzles focusing on loops. We planned to use our own experience as well as input we gain from teacher conversations. We also decided to use distractors to recreate a similar design of the Harms, Chen & Kelleher study. In these teacher conversations we hoped to gain key insights into the following:

1. How to design games to be gameful and interesting for students

2. What are typical challenges students face? This is important in order to choose meaningful distractors
3. How long these puzzles would take students?
4. Are the puzzles in the correct difficulty range for the study participants?

Using these teacher conversations, we will iterate through puzzle design in order to choose 3 puzzles and 1 familiarization task to be in the actual study.

Field Study Implementation

As we send out emails to teachers that could potentially work with us, we will provide two options: either they can choose to give us their teacher-side feedback or they can help set-up a study with their students.

If the teachers choose the first option, we will hold a 30 min to one hour session with them in which we walk through SAGE and sample puzzles. Before the walk-through, we will give a survey that gauges their experience teaching CT, Parson Puzzles, programming languages (specifically block-based languages such as Scratch). As we walk through SAGE and our current parson puzzles, we will gather their reactions and comments.

If the teachers choose the second option, we will have several more meetings to finalize materials needed (various surveys, tests and puzzles) and schedule when the study will take place. Depending on the availability, teachers will host in one of their classes or ask 15-20 students who are willing to participate in the study. We will host an orientation session with the teachers. We will conduct our study over two days. This may be subject to change based on availability. On the first day, we will conduct a pre-test (to assess the student's computation skills), present a tutorial on SAGE and conduct a background survey. On the second day, we will teach the concept of loops with a short presentation and then provide the loop SAGE puzzles for the students to work through. After, we will ask the students to complete a post-test survey.

0.4 Implementation

0.4.1 Overview

The following provides an overview of the created documents and the systems we created them in. The notes provide a high-level overview of major changes and important information about the document

Document	System	Notes
Background Survey	Qualtrics	Remove identifying questions
Pre and Post Test	Qualtrics	Isomorphic, 10 total (7:sequences, 3:loops)
Cognitive Load Surveys	Qualtrics	Rewording for student comprehension
Teacher User Guide	GDocs	Detailed schedule of the 3 days
Student Instructions (3 versions)	GDocs	Must include attributes, Brief user guide
SAGE Tutorial	GSlides	Updated style, images and points
SAGE Study Puzzles	SAGE	“Your first recipes!”, 3 Quests with 4 Puzzles

0.4.2 Survey and Tests

Background Survey We used a previous Background Survey made by Jeff. During a conversation with our first study teacher, we were asked to remove any identifying pieces of information. We removed 3 questions about Ethnicity/Race, Native Language and Disability.

Pre and Post Test We created isomorphic pre and post tests. Each test has 10 questions. We created 7 questions that focused mostly on sequences and 3 on loops. The 7 that focused on sequences varied in difficulty. The question types involved: placing blocks in order, choosing the correct ordering of blocks and choosing the correct blocks. The 3 loop questions were made to look familiar to previous sequence questions. The final loop question was a short answer to test understanding of the motivation to use a loop.

Through teacher conversations we gained important feedback such as content of questions that students might not understand and the importance of not using variables. We learned that variables were a much harder topic. This was a challenge we had to work around to find ways of testing loops and understanding of sequences without using variables. When we originally created our loop tests, we also learned that one of the harder concepts for students is the boolean logic. Therefore when creating a pre and post test for loops it would be important to create questions that test an understanding of boolean logic and also an understanding of loops. This would help to distinguish what students understand.

Cognitive Loads We have two cognitive load tests in our study. One of them is taken after completion of the SAGE activity where they learn through the puzzles. The other

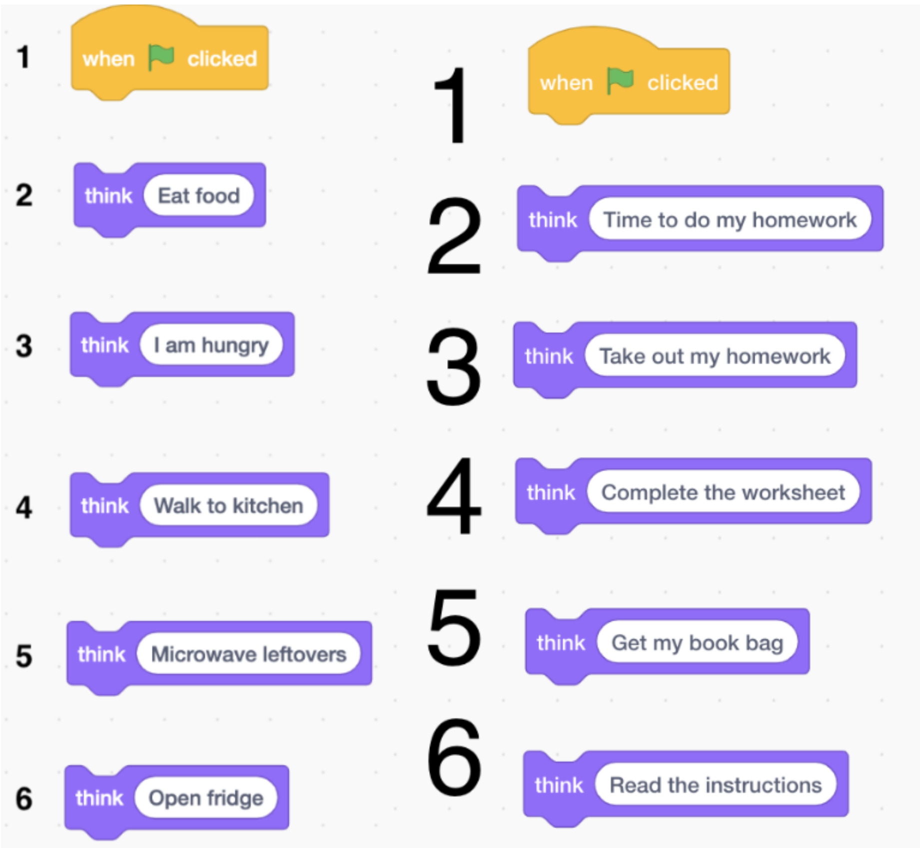

Figure 1: This is an example of two isomorphic questions. Both ask students to choose the correct ordering of the blocks.

is the cognitive load after taking the post-test. This is to help distinguish if there was a decrease in cognitive load for those who had distractors.

One piece of feedback we received from a teacher is that the wording of the cognitive load survey was above the comprehension of many of the students. As a result we went through each of the statements and reworded for better understanding. We then confirmed these new questions with the teacher.

Intrinsic Motivation Lastly, students at the end of the study take an intrinsic motivation survey to help us understand their experience of both the study and of SAGE. We based it off of standardized IMI. We chose the following subcategories: Interest/Enjoyment, Perceived Competence, Effort/Importance, Pressure/Tension, Value/Usefulness and Relatedness.

Qulatrics We transferred the surveys from Google Forms to Qualtrics. This was done to allow for the timing of questions. This new survey is uploaded as a Google Doc into the shared drive. Each "block" represents one question that is timed. Qualtrics provides

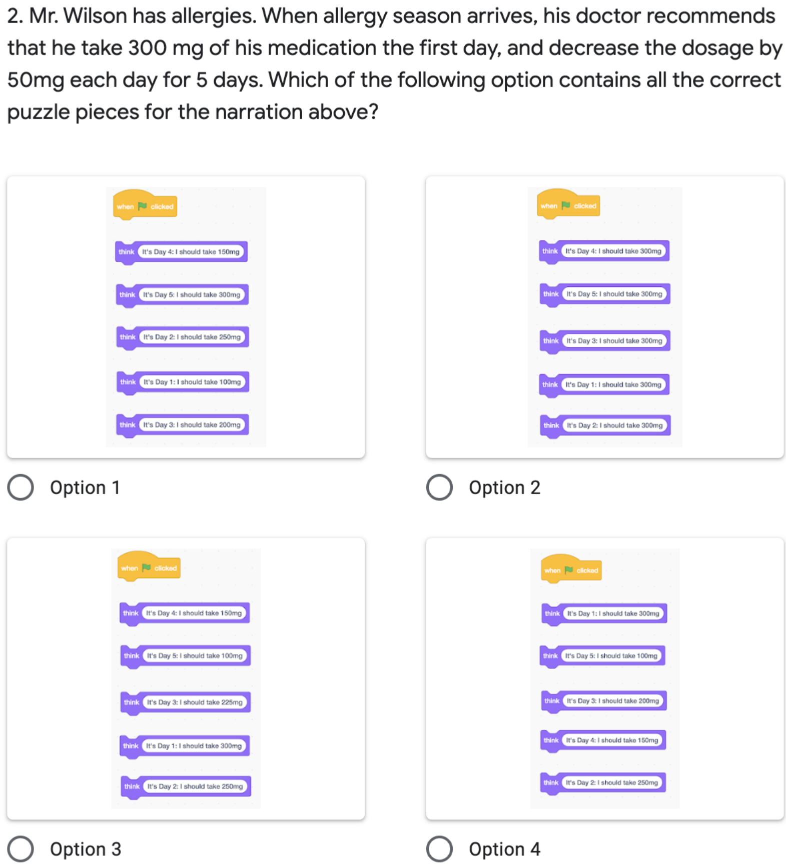

Figure 2: This is an example of another type of question in the pre and post test. This one is question two of the pre-test. This is how the question looks in Google Forms

The activity **really enhanced my understanding of the concepts and definitions.**	The activity I did **helped me understand the concept**
The activity covered concepts and definitions **that I perceived as** very complex.	The concepts and definitions in the activity **were** very complex.
The **instructions and/or explanations during the activity** were very unclear.	The **activity's instructions** were very unclear.

Figure 3: These are three examples of changes made between the original Cognitive Load survey and ours. The column on the right shows the old and the left shows the new. The bolded sections highlight the words we changed

information about timing to: First Click, Last Click, Page Submit; and click count. This information will hopefully be useful in the analysis of student responses.

Puzzle Creation In creating our puzzles, we first created puzzles for loops. We focused mostly on the desired solution and learning outcome. We relied on teacher feedback to make them both gameful and for which distractors to be included.

As we shifted away from loops to sequences, we created various puzzles at many difficulties. We relied on teacher feedback to learn what was more challenging for students. For instance, using the broadcast blocks were more challenging for students.

We imported in pictures from Google Images in order to make the game look more professional.

Ultimately, we decided that 1 familiarization puzzles and 3 study puzzles would make the most sense and fit into the time frame. The first two (familiarization and puzzle 1) were sequences. Puzzle 2 helped to transition students into loops by creating a motivation for using a loop. The puzzle is long and repetitive. Puzzle 3 is the same as Puzzle 2 except that it requires the use of a loop. The quest is built around the plot that you are baking a cake because you are going to go visit a family friend.

The puzzles are saved in SAGE as the following:

1. Quest: "Your first recipes!" Mission: Pilot Class 10 (Parson Puzzle Group)

2. Quest: "Your first recipes!!" Mission: Pilot Class 20 (Parson Puzzle with Distractors Group)

3. Quest: "Your first recipes!!!" Mission: Pilot Class 30 ("Scratch" - No parson and no feedback - Group)

We learned that distractors for sequence35s was not understanding the x, y coordinate system. We changed "move" blocks to "go to x: go to y:" blocks in order to create significant distractors. We also learned that for loops students don't have problems choosing between different types of loop blocks (repeat until or repeat x). Instead, students struggle with understanding boolean logic, variables and the concept of inside and outside of a loop. It is therefore important to create nested loops and distinguish between blocks inside and outside of a loop.

0.4.3 Teacher Feedback and Student Study

We planned to conduct a student study from 12/16 to 12/18. This happened with the teacher Elissa Levy at Life Sciences Secondary School. This would be in a 9^{th} grade physics classroom who had never done any computer science before.

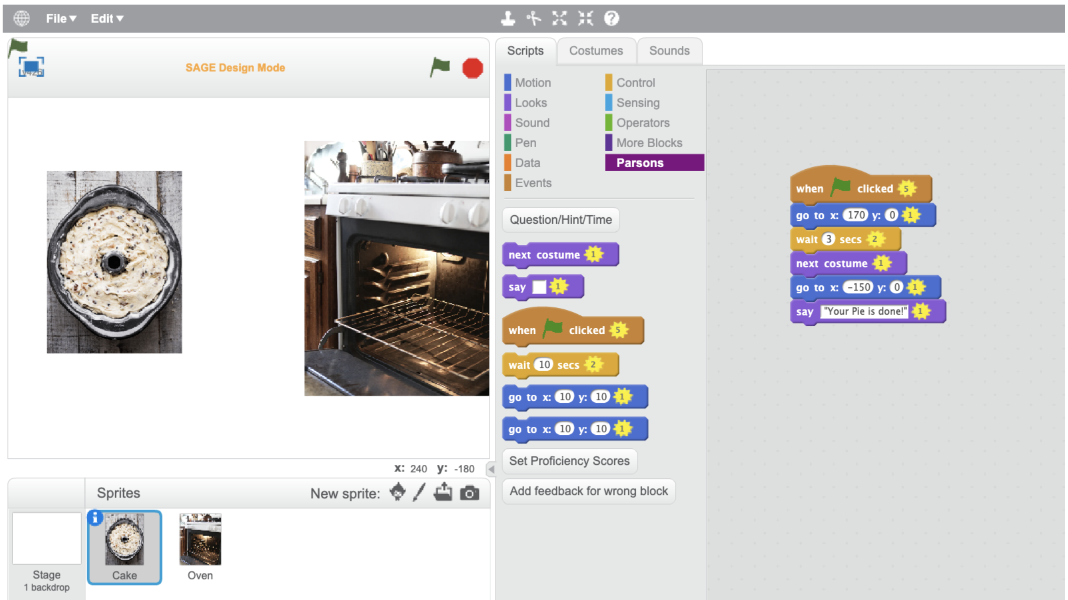

Figure 4: This is the final draft of the familiarization puzzle "Let's Bake a Cake."

Figure 5: These are included in the student instructions for the parson puzzle and parson puzzle with distractors groups. The group on the left is for the parson puzzle. The group on the right include distractors in the form of other x,y coordinates

The Plan On the first day we planned for Elissa to conduct her own Intro to Programming Lesson. Her plan was to have students create their own definitions of "computer," and discuss what a computer is and how people created programming languages to govern the behavior of computers. Input, processing, output. This would then be followed by checking that students could access the SAGE system and a SAGE tutorial. They would be given their credentials. The username they were given would be entered in each of their surveys/tests. Then they would complete the background survey and the pre-test.

On day two, we planned for a lesson on sequences that Elissa designed: "Have students create an algorithm to get Ms. Levy to walk across the room and write a student's name on the board. Talk about how computers are designed to take many instructions in a particular order, and that's what they'll be learning." They would be given their instruction sheet and then they would complete the familiarization puzzle and the three study puzzles. After this they would complete the cognitive load (after puzzle) survey.

On day three, students would complete the remaining tests/surveys: post test, post-test cognitive load and intrinsic motivation.

Reality On the first day of the study we learned that there were problems with using SAGE on the computers possibly due to Flash. We also found out that a lot of time was lost because students had to type in the URL of the survey.

On the second day, Juvaria and Jeff were able to be on site. They learned that there was a problem with the firewall and that prevented the study from taking place. Eventually, they were able to create a solution. The study has been postponed until next semester.

0.5 Limitations and Assumptions

As we had to wait for teachers to respond to our emails, we weren't able to gain feedback to implement in our study design and puzzle creation. Additionally, we only recently learned that for the study we could do a lesson, but that they would have no programming experience. This means that instead of doing loops, it made more sense to switch our focus to creating sequence puzzles that very slowly build to loops.

Since attributes aren't saved in the SAGE environment, it became difficult to create distractors for the students. We were able to work around this by creating a written instruction sheet that included pictures of the blocks with the desired attributes.

The actual study had to be postponed due to technical difficulties at the school location.

0.6 Future Work

At the beginning of next semester, the pilot study with Elissa Levy and her physics classroom can be completed. This initial study will likely provide more insights into protocols that need to be changed for future iterations. For instance, it would help if we could go into the school prior to the study being completed in order to check that

SAGE can run on their machines. Additionally, we should create some way of possible embedding links or sending out links so that students do not need to type out the URL.

It will then be essential to reconnect with the other teacher who agreed to do a study with us. Additionally, future work will include connecting with more teachers to perform the study.

0.7 Conclusion

This semester was mainly focused on the creation, feedback gaining and finalizing of study materials. This incldues the daily schedule of events, various surveys, pre/post tests, puzzles and puzzle instructions. Future work will be able to use these materials in actual studies. Our first trial run of the study showed that there may be technical difficulties that should tried to be tested and resolved prior to study day.