

SAGE
Parson's Feedback Improvement
on Student's Side

COMS 6901 – Fall 2019

Linnan Li (ll3235)

Penghe Zhang (pz2244)

Content

1. Introduction.....	2
1.1 Parson's HUD styling	2
1.2 Parson's Per-Block Feedback.....	2
2. Related Works.....	3
2.1 Computational Thinking	3
2.2 Parson's Programming Puzzles	4
3. Proposal.....	4
3.1 Parson's HUD Styling	4
3.2 Parson's Per-Block Feedback.....	5
3.3 Bug Fix.....	6
4. Timeline.....	7
5. Future Works.....	8
6. References	8

1. Introduction

Parson's programming puzzles are great formative assessments that ask students to arrange blocks of scrambled code, allowing them to focus on the purpose and flow of the code while not worrying about syntax.^[1] Instructors can use this way to help students have a better understanding on the thought of coding easily, instead of letting students stuck in those hard and complex syntax and format of a coding language. Currently, our Social Addictive Gameful Engineering project has already embedded such a Parson's programming puzzles inside the system. However, there are still lots of efforts we need to do to improve the user experience both for students and instructors. In our project, we will most work on the HUD styling and feedback per block on student's side. Also, we will help fix some user interface bugs during the project.

1.1 Parson's HUD styling

When a student is solving problems in his/her scratch interface, he/she will find his/her points for each step and proficiency levels showing on the top panel of the scratch interface (shown as figure 1).

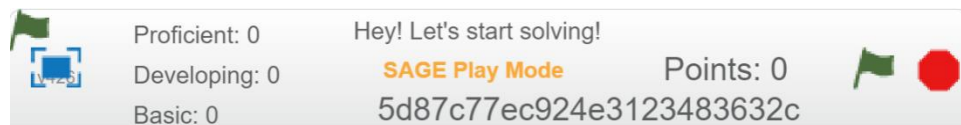


Figure 1. Current Top Panel of Scratch UI

We can see they are just some points and it is hard to see connections between these points. Thus, we need to modify this part to make it more readable, like using a graph. It would probably be better to render these via sage-frontend beneath the intelligent tutor avatar via an attractive charting tool.

1.2 Parson's Per-Block Feedback

Think in a student perspective, whenever you move, replace or add a block in your puzzle, it is always an important thing to know whether current action is a right or wrong one. Thus, we need to add such a feedback module to let students know whether they are in the correct track.

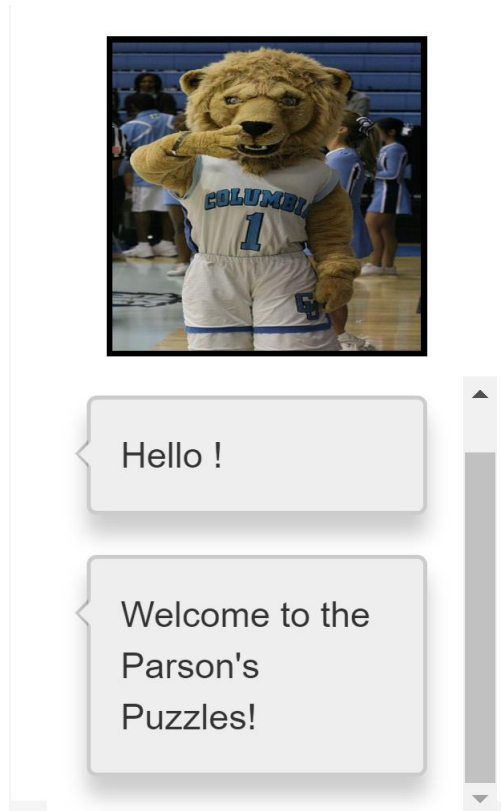


Figure 2. Current Feedback Module

2. Related Works

2.1 Computational Thinking

Computational thinking is a new concept in the area of computer science and educational research. It involves solving problems, designing systems and understanding human behavior by using fundamental concepts in computer science. ^[2] Even though the name seems like a study restricted to computer scientists, it is a fundamental skill for everyone actually.

Generally speaking, computational thinking is representing problems in our daily lives as computer science concepts or models, which makes them easier to handle. However, computational thinking does not equal to programming. It requires thinking at higher levels of abstractions and thinking as human beings. Therefore, computational thinking is aiming

at delivering ideas rather than artifacts.^[3]

In SAGE, we are building an educational game platform to teach students from 6 to 8 grade the skills of computational thinking drawing on Parson's Programming Puzzles and other approaches.

2.2 Parson's Programming Puzzles

Parson's programming puzzles is a new teaching approach for computer science and programming concepts. It is usually in a drag-and-drop style and each block of puzzles involving a subroutine of program.^[4]

Parson's programming puzzles are designed to have more improve the engagement of student because it uses a puzzle-like pattern. It also has some other properties such as permitting common errors, setting distractors and modeling good structured code. Another appealing strength of the puzzles is that it can provide immediate feedback. In the practical puzzles, students can click check button and get feedback.

Parson's programming puzzles is an important component of SAGE. As an instructor, a user can design puzzles and assign it to students. SAGE will provide feedback to students when they are solving the puzzle and report the score accordingly.

Currently, the score is only a plain number that calculated by SAGE, it cannot show more details about the student's performance. Besides, SAGE only provide an overall score for one quest. Adding per-block feedback will give students more instruction and enhance the engagement as well.

3. Proposal

3.1 Parson's HUD Styling

In the current look of a quest, the student's level of proficiency, development and score is shown on the top of the scratch and beneath instructor's avatar. However, the information is displayed as plain numbers, and there is no relationship between these data shown either.

There are a lot of useful tools for statistical display, for example, D3.js and chart.js. We propose to use one of these modules to display the data graphically, such as spider chart or line chart. We may reference the chart appeared in the Metric session to keep a similar UI of the whole system.

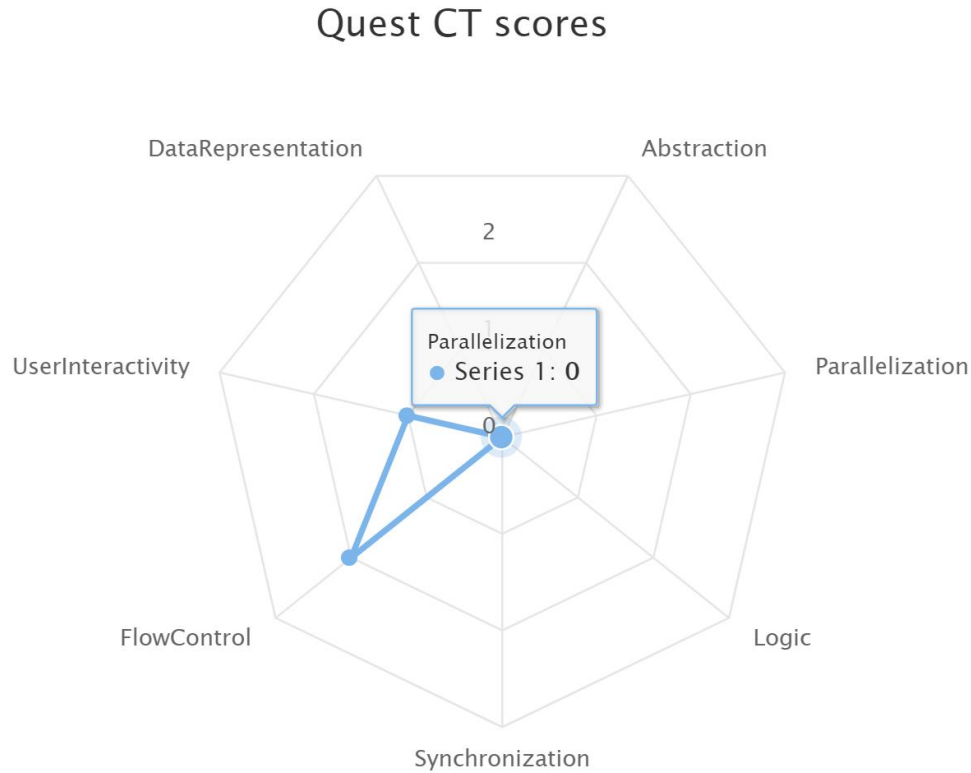


Figure 3. Chart in Metric Session

The workflow for this part is clear: gathering the score and proficiency level through sage-node, employing D3.js or Chart.js to draw graphs. We also propose that the per-block feedback can also provide data to be displayed in the charts.

After implementing the basic functionalities, we are going to refine the front-end look and style of this part. First of all, all the debug messages such as game IDs should be cleaned-up for representation. Besides, we are planning to rearrange every block for better user experience.

3.2 Parson's Per-Block Feedback

As a student, he/she wants to receive explicit feedback on the accuracy of placement of each block so that he/she can be continuously coached toward the correct solution.

Based on previous discussions, we may first have four types of feedback based on each block action, which are neutral, correct, incorrect and distractor. Basically, if they drop a block on to the scripts pane but not attach it to anything, they'd get a neutral feedback. If they attach it to some other blocks, we can tell them if it is correct/incorrect based on the block they attach it to or the blocks around it. if it's a distractor then we give them feedback specific to distractors.

Since the feedback module is mainly a suggestion module, we should make it an independent module instead of reusing the scoring system. And it should not be related to the final score. On the other hand, students may come out the solution which is totally different method with the standard, we may also think this scenario.

On the front-end side, the basic idea is to show the feedback as a bubble in the right panel of the web page, different types feedback may appear with different colors (we can use those pre-defined colors in Bootstrap). And to ensure the users can easily know the feedback's belonging, we should also include the action or the step number in the feedback module.

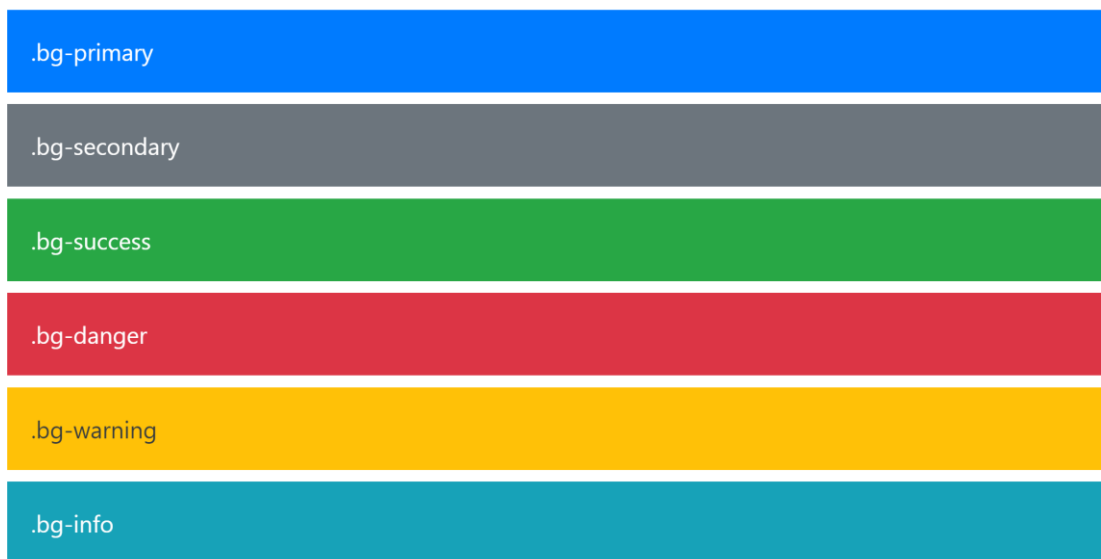


Figure 4. Pre-defined colors in Bootstrap ^[5]

3.3 Bug Fix

Besides the above two stories we need to do this semester. During this week's exploring in the current code of SAGE, we found some bugs and logic errors we need to fix.

The first one is the auth bug. When we tried to register a new user in the current SAGE system,

we created a new instructor role, but still when we are trying to login with this role, the page will still forward us to the student page instead of instructor page.

Meanwhile, we found there may be some critical flaws in the system, the url is something like this: <http://dev.cu-sage.org/student/#/metrics/5d869704ab891e3a0c0ff06e>. and when we modify the url to <http://dev.cu-sage.org/instructor/#/metrics/5d869704ab891e3a0c0ff06e>, the web page will be redirected to an instructor page. And in the browser local storage I also find that:

```
▼ {_id: "59f369cc748499467c32a414", email: "student1@sage.com", fullname: "Mike", role: "student",...}
  email: "student1@sage.com"
  fullname: "Mike"
  role: "student"
  __v: 0
  _id: "59f369cc748499467c32a414"
```

Figure 5. Local Storage

The user's role is student, but he can still visit instructor page with whole permission on all items for instructors, which is definitely something not right. We think this bug should also be fixed and we will try to fix it.

The second is the logical between instructors and students on Parson's programming is misleading. We are not sure whether we didn't understand the system or not, when we tried to design a puzzle in the teacher's side, then I published it to the students. But when we tried to solve the puzzle in the exactly same solution as we designed, we still get a 0 score. We are not sure why this happens, if this is a bug, we need to fix it; if this is only because we don't know the logical behind that, we should also write a wiki to tell people how to use this.

Also, there are some UI bugs need to fix, like in the Mission page, the cards in the Mission and the cards in the Quests are not in the same line, etc.

4. Timeline

Sprint 1: Play around with SAGE and read the source code.

Sprint 2: Implement HUD styling and improve the front-end look of the feedback block.

Sprint 3: Implement per-block feedback feature and connect it with overall feedback.

Sprint 4: improve test coverage and debug.

5. Future Works

The current scoring system and the feedback system is using traditional algorithm to calculate. In the future, we can apply machine learning approaches on this problem in order to get more accurate scores and feedback.

6. References

- [1] Parsons, D. and Haden, P. (2006). Parson's Programming Puzzles: A fun and effective learning tool for first programming courses.
- [2] Bender, J. (2015). Developing a collaborative Game-Based Learning System to infuse computational thinking within Grade 6-8 Curricula.
- [3] Jeannette M. Wing. 2006. Computational thinking. Commun. ACM 49, 3 (March 2006), 33-35. DOI: <https://doi.org/10.1145/1118178.1118215>
- [4] Mohan. (2017). Gameful Direct Instruction (Parson's Puzzle)
- [5] <https://getbootstrap.com/docs/4.3/utilities/colors/>