澳門理工大學
Universidade Politécnica de Macau
Macao Polytechnic University

Faculty of Applied Sciences
Bachelor of Science in Computing

COMP490 Final Year Project
Project Proposal

Academic Year 2022/23

Parsons problem generator and solver

Project number:	19
Student ID:	P-19-0834-5
Student Name:	Jane Liu
Supervisor:	Philip Lei
Assessor:	Charles Lam
Submission Date:	September 8, 2022

Table of Contents

1 Project Description

In recent years, with the computer industry boom, an increasing number of relevant jobs in this area have been provided, and more professional personnel are needed. Consequently, a growing number of people prefer to major in science in computing at their universities to find decent jobs after graduating. However, for some of the learners of this program, the courses are not straightforward, especially some lessons about improving programming skills. One of the reasons for this tough study of programming is that the students need to practice programming by directly writing codes line by line, which causes a heavy cognitive load for students and makes it easy for students to feel frustrated and even lose interest in programming [1]. Therefore, the introductory programming courses introduce an advanced learning method Parsons problem. With the drag-and-drop programming style, this method allows the students to reorder the mixing up prepared blocks of this solution, which mitigates the thinking pressure for students. However, the existing Parsons problem work is limited to the introductory programming level. The previous work does not cover higher-level programming courses such as data structures and algorithms. Therefore, the expanded question scope of Parsons problems will be discussed and proposed in this project.

This project aims to design and build a website for computer usage to generate Parsons problems according to the original Python code, make students solve the generated Parsons problems, and provide feedback for the solutions. The major objectives of this project are as follows:

- Study the existing Parsons problems generators and solvers to get a general idea
- Design system architecture to have a blueprint of the project
- Design and develop a database for storing the account information including the types of users, the generated Parsons problems, the corresponding solution with distractors, the scope tag of problems, and the practice records of students.
- Design and develop a webpage for authorized users to input the original questions with the correct solution and necessary distractors and variants.
- Design and develop a function to generate Parsons problems according to the given questions, solutions, distractors, and variants.
- Design and develop a webpage

- Design and develop a webpage to show the generated Parsons problems and let students drag and drop the correct answers to the appropriate positions.
- Design and develop a function to evaluate the code generated by students and give them some feedback.
- (Optional) Design and develop a function to show the statistical analysis of the total number of completed practices and the mastery degree of students.
- (Optional) Design and develop a function to collect student practice progress and let teachers know the mastery degree of students.

In this project, the Parsons problem is used to assist students in learning programming. This approach allows students to drag and drop the ready coding blocks separated from the whole original codes instead of writing code directly. In this way, the logic is divided from the syntax [1], which reduces students' cognitive load and helps teachers to aim at the specific problems students have [1]. Besides, with timely feedback, students can better understand their codes and be encouraged to learn further. In addition, different scopes of the concepts covered by the problems can make this project not be limited to the introductory level of programming study.

There are several challenges in this project. Firstly, it needs to cut the original solution reasonably over all the scopes of the problems. Secondly, it needs to define an appropriate upper bound of the number of distractors to control the complexity of the problems. Thirdly, it needs to develop a 2D view for dragging and dropping the blocks. Lastly, it needs to know the results of students’ code. It may need to run the code on the webpage.

2 Summary of Related Work and Key References

There are several programming learning ways. Specifically, the most traditional one is code writing, which lets the students directly write codes line by line. Without a doubt, this way is most challenging for learners since this method mixes the syntax with the logic and makes it tough to identify their difficulties [1]. In this case, students need to make many attempts to solve the programming problems, and they may easily give up during this period. To ameliorate this method, a preparation action Coached Program Planning can be applied before writing code. This method guides students to analyze the problems and design the logical procedures with native-language style pseudocode for solving these problems [2, 3] so that the logical parts of codes can be partially separated from the syntax parts reducing the cognitive burden during coding to some extent. In addition, there is another way of studying programming, code tracing, which is a method for students to track the changes of variables by hand during the execution of codes [4]. Although this method can provide students with auxiliary help in improving code writing skills through the accumulation of experience of code reading, this method does not give any new ideas for programming practice when students need to write some codes. Thus, students still need to write the whole code alone when they have some programming exercises, which means they may still meet the problems mentioned in the previous code writing method. Besides, this method will also be quite cumbersome when the questions are complicated. As a consequence, this method cannot be applied to every programming question.

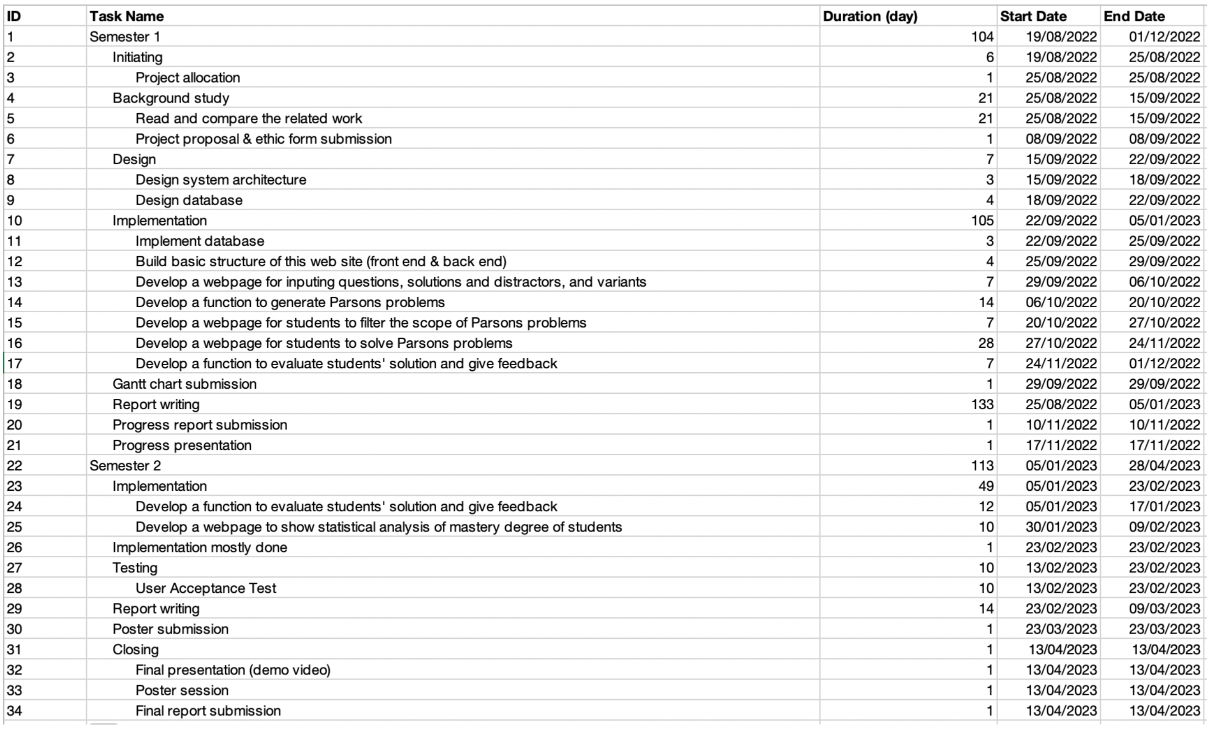
Unlike the previous two methods, the Parsons problems create a new way to learn programming courses – drag and drop prepared blocks instead of writing codes directly. Thanks to this new way, Parsons problems, at a certain point, look like a hybrid of code writing and code tracing. In other words, the Parsons problem not only keep some code reading and understanding parts like code tracing, which can assist students to build some basic ideas of some common solution structures of some questions and learn about some good coding habits like naming variables properly, but also reduce the need of writing syntax coming from code writing, which causes less cognitive load than the code writing. Because of the benefits of Parsons problems, this method has been used in loads of previous work about programming education. To be more specific, Parsons and Haden developed a web-based tool for generating Parsons problems for introductory programming in BASIC with jumbled distractors, pre-scaffolding, and absolute line-based feedback [1, 5]. Besides, Helminen et al.

developed a Parsons problems project called js-parsons for introductory programming in Python with no distractors, student-scaffolding, and relative line-based feedback [review, how to]. Lastly, Amruth developed Epplets for introductory programming in C+, C#, and Java with distractors and relative line-based feedback. In conclusion, although the previous work provides Parsons problem practice in several programming languages with different structures, these projects have one thing in common – they are only applied for introductory programming, which gives my project room for improvement.

Moving on to this project, from my perspective, Parsons problem still can demonstrate its superiority compared with the traditional code writing way in the middle level of programming learning. Admittedly, students taking intermediate-level programming courses should be able to write code instead of just rearranging the order of provided answer blocks. However, Parsons problem still can be one of the possible programming question types besides code writing since it can help teachers to know the grasping level of specific knowledge of individual students by setting related distractors. Furthermore, the game-style Parsons problem can improve students' engagement and motivation in learning programming, which is also vital in middle-level learning [5]. What’s more, Parsons problem can make students familiar with the logic of some algorithms or data structures when they first access the new knowledge. Lastly, Parsons problem can provide instant feedback for students, which is also essential for students.

3 Project Work Plan

This section illustrates the project work plan in Table 1.



ID	Task Name	Duration (day)	Start Date	End Date
1	Semester 1	104	19/08/2022	01/12/2022
2	Initiating	6	19/08/2022	25/08/2022
3	Project allocation	1	25/08/2022	25/08/2022
4	Background study	21	25/08/2022	15/09/2022
5	Read and compare the related work	21	25/08/2022	15/09/2022
6	Project proposal & ethic form submission	1	08/09/2022	08/09/2022
7	Design	7	15/09/2022	22/09/2022
8	Design system architecture	3	15/09/2022	18/09/2022
9	Design database	4	18/09/2022	22/09/2022
10	Implementation	105	22/09/2022	05/01/2023
11	Implement database	3	22/09/2022	25/09/2022
12	Build basic structure of this web site (front end & back end)	4	25/09/2022	29/09/2022
13	Develop a webpage for inputing questions, solutions and distractors, and variants	7	29/09/2022	06/10/2022
14	Develop a function to generate Parsons problems	14	06/10/2022	20/10/2022
15	Develop a webpage for students to filter the scope of Parsons problems	7	20/10/2022	27/10/2022
16	Develop a webpage for students to solve Parsons problems	28	27/10/2022	24/11/2022
17	Develop a function to evaluate students' solution and give feedback	7	24/11/2022	01/12/2022
18	Gantt chart submission	1	29/09/2022	29/09/2022
19	Report writing	133	25/08/2022	05/01/2023
20	Progress report submission	1	10/11/2022	10/11/2022
21	Progress presentation	1	17/11/2022	17/11/2022
22	Semester 2	113	05/01/2023	28/04/2023
23	Implementation	49	05/01/2023	23/02/2023
24	Develop a function to evaluate students' solution and give feedback	12	05/01/2023	17/01/2023
25	Develop a webpage to show statistical analysis of mastery degree of students	10	30/01/2023	09/02/2023
26	Implementation mostly done	1	23/02/2023	23/02/2023
27	Testing	10	13/02/2023	23/02/2023
28	User Acceptance Test	10	13/02/2023	23/02/2023
29	Report writing	14	23/02/2023	09/03/2023
30	Poster submission	1	23/03/2023	23/03/2023
31	Closing	1	13/04/2023	13/04/2023
32	Final presentation (demo video)	1	13/04/2023	13/04/2023
33	Poster session	1	13/04/2023	13/04/2023
34	Final report submission	1	13/04/2023	13/04/2023

Table 1: Table of the project work plan

4 Risk Assessment

This section will describe three risks encountered in the project, illustrate the severity priorities of these three risks, and show the effectiveness of the related solution by comparing the impact and possibility before and after.

Risk 1: The computer used for implementation may break down

Since the whole project work, including codes and documents, is stored on the computer, it will be a disaster when computer errors occur. All work will disappear in a moment, and the previous effort will go down the drain, which will seriously affect the project's progress.

Solution: Take advantage of online backups such as GitHub or Google Drive to ensure the progress will not be affected when the computer acts up.

Risk 2: The system does not support different devices

Although this web-based project is designed to be used on computers majorly, some users may try to open it on their phones or tablets. Since the changes in screen sizes between these three kinds of devices, some interfaces of web pages may look too small or too strange, which may bring users an awful experience and reduce their motivation for further use of this website. In that case, it does not seem conducive to the future growth of this project.

Solution: Design a responsive website to fit the size of different devices satisfying users' needs on various devices.

Risk 3: Users reject this website because of the inferior user interface

If the user interface is unfriendly, the user may misunderstand some instructions given by this website and repeat some unexpected wrong actions again and again, which wastes a lot of users' time and may cause users to be impatient and give up easily. And this situation is not beneficial for the long-term development of this project.

Solution: Apply common design principles like Eight Golden Roles to the interfaces. In addition, simulate a user to access some related projects and get some ideas from their design.

To sum up the serious level of these previously mentioned risks, a priority risk table has been shown in Table 2.

Priority	Risk identifier and Description
1	Risk 1: The computer used for implementation may break down
2	Risk 3: Users reject this website because of the inferior user interface
3	Risk 2: The system does not support different devices

Notes: Priority 1 is the highest risk

Table 2: Table of priority risk

As shown in Table 2, there are three related risks. Specifically, the first risk has the highest priority since it may redo the whole work, which may significantly delay the development of this project. Subsequently, the third risk has lower priority than the previous risk since the project can still perform when affected by this risk. However, this risk may cause the users to feel confused and fretful, so they may not want to use the website again, which is not good for the popularization of the website. Lastly, the second risk has the lowest priority since this website will be major used in computers, and only a few people using other platforms are affected by this risk.

The probability impact matrixes have been shown in the following two figures to have a direct show between before and after using solutions.

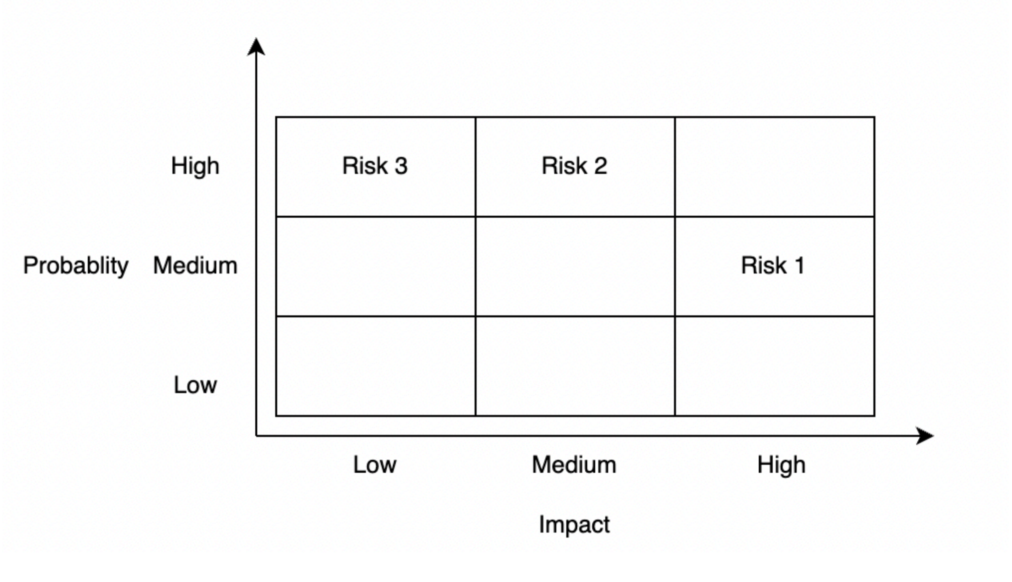


Figure 1 -Probability Impact Matrix – Initially

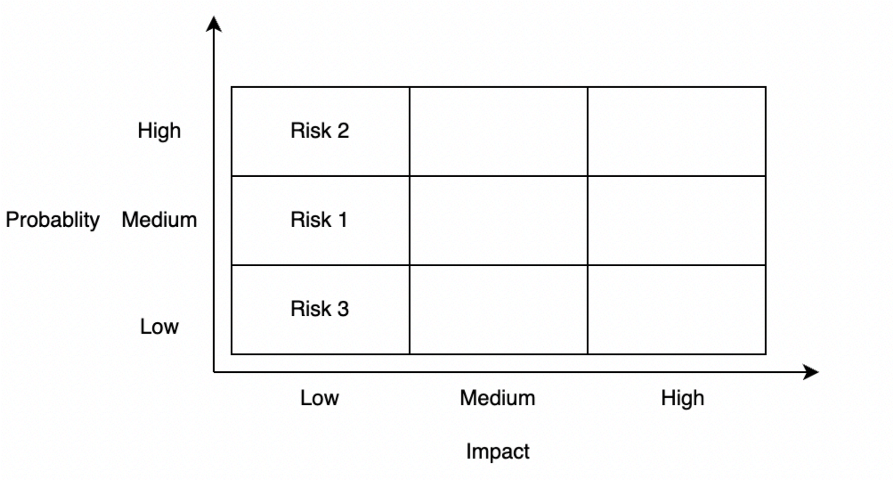


Figure 2 - Probability Impact Matrix - Solutions

References

[1] Yuemeng Du, Andrew Luxton-Reilly, and Paul Denny. 2020. A Review of Research on Parsons Problems. In Proceedings of the Twenty-Second Australasian Computing Education Conference (ACE'20), February 3–7, 2020, Melbourne, VIC, Australia. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373165.3373187

[2] Rita Garcia, Katrina Falkner, and Rebecca Vivian. 2018. Scaffolding the Design Process Using Parsons Problems. In Proceedings of the 18th Koli Calling International Conference on Computing Education Research (Koli Calling '18). ACM, New York, NY, USA, Article 26, 2 pages. https://doi.org/10.1145/3279720.3279746

[3] H. Chad Lane and Kurt VanLehn. 2003. Coached program planning: dialogue-based support for novice program design. SIGCSE Bull. 35, 1 (January 2003), 148–152. https://doi.org/10.1145/792548.611955

[4] Code Tracing. Retrieved September 7, 2022, from https://microcredentials.digitalpromise.org/explore/code-tracing

[5] Dale Parsons and Patricia Haden. 2006. Parson's Programming Puzzles: A Fun and Effective Learning Tool for First Programming Courses. In Proceedings of the 8th Australasian Conference on Computing Education - Volume 52 (ACE '06). Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 157–163. http://dl.acm.org.ezproxy.auckland.ac.nz/citation.cfm?id=1151869.1151890