112-2 Program Design (II)

Final Project

Midterm Deadline: 23:59, 2024/4/10
Final Deadline: 23:59, 2024/06/23

What do you need to do for the final project?

For your final project, you are tasked with the creative challenge of designing and implementing your own **game**, centered around a core theme: "**Randomness**." Randomness plays a pivotal role in the mechanics of numerous games, offering unexpected encounters, varying rewards, and a unique experience with each playthrough.

To deepen your understanding of how randomness can be incorporated into game design, we highly recommend watching an insightful video by Game Maker's Toolkit, which is accessible at [this link]. This resource, available with both English and Chinese subtitles, will provide you with valuable perspectives on leveraging randomness to enhance gameplay dynamics and player engagement. You are encouraged to think innovatively and employ the concept of randomness in a manner that enriches the gaming experience.

Find Your Teammate

- This is a **team project** that requires **5 or 6 members**. Please find your teammates and register your team at this link: https://forms.gle/TQQwKUCJmRvoMadt7
- **Please finish the registration before 2024/3/19**.
- In general, all team members will receive a similar score for the final project. However, we understand that collaboration is not always an easy task.
- Therefore, each team is **required** to use version control systems like **Github** to record the contributions of each team member.
    - Please ensure we know who you are by the Github account you used.
    - Each team will need to submit the final project using the link of the Github repo.
- Moreover, we will let each team member grade other team members' contributions, which will also be included in the grading of the final project.
- If there is any disagreement in a team, we will give different grades to each team member according to the records on the version control system, as requested.

Technical Requirements

- Although we focus on C in this course, we encourage you to explore and learn other programming languages and tools by yourselves.
- Therefore, you can choose to use the following programming languages to implement the game you designed: C, C++, Java, C#, or Python
    - These languages are selected based on the list of game engines [link].
    - Don’t worry if you only know how to use C, there are many helpful game engines or tools that support C programming: raylib, orx, Gunslinger, Mojoc
- Your game needs to meet the requirements described in the **Basic Part** to obtain essential points. All requirements in the Basic Part are designed to be achievable even if you only use the most basic way to write a C program.
    - However, since using C tends to require more effort than using other programming languages, we will give you extra points if you used C to achieve a similar effect with other projects that use different programming languages.
- There are some advanced items you can do to get more points in the **Advanced Part.**

Basic Part

Data Type and Data Structure

- Your game needs to have **at least three** basic data types (e.g., `int, float, char`)
- There should be at least one data with the **string** type in the system.
- There should be at least one kind of **customized data structure**. For example, in C, you can create a specific data structure using structure, union, or enumerations.

Operations

- **Traverse**: Display all items in the database in a specific format. For example, the game can display all the items the players possess.
- **Sort**: Sort all the items in ascending or descending order according to the given data. For example, the stages of the game are sorted by difficulty level.
- **Random:** Since the game's theme is randomness, your game needs to perform random operations.
- **File I/O**: Your game needs to perform the operations of reading and exporting the data to external files.

Advanced Part

- **Excitement and Engagement:** Evaluation of the game's ability to captivate and maintain player interest, with particular attention to its dynamics, challenges, and overall fun factor.
    - You are encouraged to upload the game to https://itch.io/ or any place where we can play it online.
- **Innovation in Theme Implementation**: Assessment of the game's originality and creativity in integrating the theme of randomness, focusing on unique gameplay elements and mechanics that stand out.
- **Code Quality and Organization**: Review of the game's code for clean structure, readability, and the inclusion of appropriate comments that facilitate understanding and future maintenance.
- **Completeness of README Documentation**: Examination of the README file in the game's GitHub repository for thoroughness and clarity for people to understand how to play the game. The Github README file needs to provide information on the work division of a team.

What do you need to submit and present?

Project Progress Report

- The purpose of the progress report is to help us to check if each team is on the right track.
- **The presentation will be in English. You need to present to the entire class.**
- The instructor will provide you with feedback to help you improve the final project
- The length of the presentation will be **5 minutes,** and you need to prepare the slides:
    - In the first slide, please introduce your group member with their name and student ID numbers.
    - For the rest slides, please briefly introduce what you plan to do
    - Each team needs to submit the **slides** on eCourse2 **before the midterm deadline**

Final Project Demo & Presentation

- Each team needs to submit the **GitHub repository URL** on eCourse2 before the **final deadline.**
    - The change on the GitHub repo after the deadline will be ignored during grading.
- We will run a lottery to determine the order of the presentation.
- **The presentation will be in English. You need to present to the entire class**
- The **length** of the presentation will be **10 minutes** depending on the number of teams. You need to prepare the **slides:**
    - In the first **slide**, please introduce your group member with their name and student ID numbers.
    - For the rest of slides, please introduce your game, the basic and the advanced parts.
    - Each team needs to submit the **slides** on eCourse2 **before the final deadline.**
- Lastly, you will need to do a **live demonstration** by executing your game and playing it to show the most significant and exciting features
    - You need to execute your code on your computer. Do not play a pre-recorded video.
    - Please rehearse your live demonstration. You need to show the basic and advanced parts and explain their functions at the same time, which is not an easy task.
    - If you don't know how to do a live demonstration. Welcome to this live demo video made by OpenAI!

Grading

- **Basic Part (35%)**
    - Data Type and Data Structure (15%)
    - Operations (20%)
- **Advanced Part (27%)**
- **Peer Evaluation: Grading from Other Teammates (8%)**
- **Presentation (30%)**
    - Midterm Progress Presentation (10%)
    - Final Presentation and Live Demonstration (20%)

The Policy of Late Submission

- If you failed to submit the slides or Github URL before the midterm or final deadlines, you could still submit it on eCourse **before the project progress report and final project demo & presentation**.
- However, you can only get a **70% score for that submission**.
- If you fail to do the project progress report or the final project demo and presentation, you will get a zero score for the final project.