ENGG1340 / COMP2113

Course Project (2020-21)

Introduction

In this project, you will apply what you learned from the course (e.g., github usage, C/C++ programming, etc.) and **implement a text-based game of your own choice**. Your game should be a console application that can be accessed using the terminal or SSH. Note carefully the requirements for the project scope and submission detailed in this document.

Important note: We will take plagiarism seriously, so make sure that all submissions for your project are your own work or you will risk losing marks in full for your project.

Please also check the late submission and plagiarism policies from the course information on Moodle.

Milestones and Timeline

There are two submission deadlines that you should observe. Details for each milestone will be given in subsequent sections.

- **Stage 1: Proposal (due Oct 24, ~3 weeks from now)**
  - You will receive endorsement on your proposal from the teaching team within 2 week after submission, but you should continue to work on the project without idling.
- **Stage 2: Final Work (due Nov 21, ~7 weeks from now)**

Check what you need to submit for each milestone and note carefully the specific requirements for the project.

Grouping

You must work in a group of 2 members for the project. Please make sure that you have indicated your grouping on Moodle, as some submissions would be done on a group basis on Moodle. Grouping cannot be changed after proposal submission. You may make use of the Moodle forum to find a project partner.

Please indicate your grouping, as soon as possible before proposal submission, in this Moodle page (https://moodle.hku.hk/mod/choicegroup/view.php?id=1953611) by having your project partner joining the same group.

Requirements

- **Code Requirement.** Your implementation should encompass the following coding elements:
  1. Generation of random game sets or events enemy number is also random.
  2. Data structures for storing game status
  3. Dynamic memory management
  4. File input/output (e.g., for loading/saving game status)
  5. Program codes in multiple files
  6. Proper indentation and naming styles
  7. In-code documentation
- All projects should be committed to a private Github repo.
- Each student should contribute to at least 25% added(+) lines of the code of their project, gauged by GitHub's "Contributors Graph" of each project.
- Commit comment should not be empty and should be written sensibly.
- For each function, comments on "what it does", "what the inputs are" and "what the outputs are" are needed.
- You may use any of the C/C++ libraries.
- Your programs will be tested on the standard coding environment on the CS servers, so make sure that you work can be compiled and executed under the same environment.

Stage 1 Submission - Proposal

The purpose of the proposal is to help you define the project scope and consolidate your ideas before you start the implementation.

What to do for this stage:

- To identify a text-based game that you would like to implement
- To define what features that your proposed text-based game would incorporate.
- To set up a Github repo where your work will be hosted.

20201004

What to submit (for each group)

- Set up a private Github repo, which should include the two members and the teaching team account **engg1340comp2113**.
- Create a readme.md file there in the repo, which should contain:
  - A brief identification of the team members.

20201010

- A game description with basic game rules.
- A list of features that you **plan to implement**, vis-a-vis each of the items 1 to 5 listed under **coding requirements** above.

- Hand in a link to the repo to Moodle.

Reference:

Getting started with writing and formatting on GitHub
About READMEs

20201010~20201025: Code for the game.

Stage 2 Submission - Final Work

What to do for this milestone:

- To implement the text-based game according to the proposal
- To document the developed program

What to submit (for each group)

- In the same Github repo that you set up in Stage 1,
  - An updated readme.md with
    - A brief identification of the team members.
    - A game description with basic game rules.
    - A list of features that you **have implemented**, vis-a-vis each of the items 1 to 5 listed under **coding requirements** above.
    - Any non-standard C/C++ libraries, if any, that are used in your work, and what features in your game are supported by these libraries.
    - Compilation and execution instructions. Simply put, this serves like a manual to your program.
  - Your programs including **Makefile and source files (.h / .cpp / .c)**
  - Sample input/output files (whenever appropriate)
- Hand in a link to the repo to Moodle.
- A video (at most 3 minutes long) demonstrating a gameplay and the implemented features of your program.

20201025
~20201031

20201031
~20201121

What to submit (for each individual student)

- Submit on Moodle
  - a list of work that you have done for the project
  - a peer evaluation among your group members. Make comments on your project partner's work.

Weighting

Your work will be accessed in the following aspects:

Project total (10%)

- Problem statement (including setting & assumptions) (1%)
- Implementation (including the use of functions, dynamic memory management, file I/O, data manipulations) (5%)
- Program functionality and special features (including creative and fun elements) (2%)
- Documentation (including readme.md) and coding style (1%)
- Collaboration (1%)