CMPT 276 Project - Phase 1

v1 (05-15)

Project overview

You are asked to create a software implementation of a board game. A board game has multiple players, a win condition, and movable components on a board.

- Examples of board games: Monopoly, Risk, Twilight Struggle.
- Non-examples of board games: Sudoku (single player), Charades (no board), Ouija (no win condition).

Other examples can be found on board game websites online. You may choose an existing board game or design your own. If you choose an existing game, you are not required to faithfully implement it – you can pick and choose which features and game mechanics to include or create new ones.

Game requirements

In the end, your project shall have the following features:

- Interactable game elements, such as pieces, cards, and boards.
- An achievable win condition.
- An AI that makes decisions based on game factors.
- Save and load.
- A game UI that is not pure text.
- Achievements and game scoring.

More specifics for each phase, such as a demonstration of refactoring, test cases, and design patterns will be released later.

There are many features your project may have. To help you design your software, here are some questions you might ask:

- How should the user learn how to play the game?
    - Tooltips? Interactable tutorial? Help files?
- How should the UI enable and support player actions?
- How should the code architecture be designed to allow for the game's logic?
- What should the AI do to provide a challenge?
    - Different difficulty levels? Special challenge scenarios?

Phase 1

For Phase 1, you need to:

- Set up a GitHub account at sfu.github.ca. This is SFU's GitHub server.
    - One person should create the project for your group.
    - Add all group members, the professor and both TAs to your project. You can only add someone who has signed in once.

    - You can greatly ease your development by using SSH keys.
    - Remember to set up config to show your name and e-mail address.
- Demonstrate some understanding of git's features.
    - Create at least one feature branch and merge it back into the main branch.
    - Each group member should make at least one commit.
    - (It is not necessary for the feature branch to be a real feature, or the commits to contain anything substantial.)
- Submit a requirements document by putting it in your git repository (top folder, main branch).
    - The document should be a PDF file and it should be called requirements.pdf.
    - Follow the instructions listed below for your requirements document.
- You are strongly recommended to have one or multiple group meetings before the deadline to plan out your schedule, divide the workload, and break the ice. No submission is required here.

Phase 1 – Requirements documentation

The requirements document is about the game you will create. You may freely choose how to present, organize, and format the document, but it needs to include the following:

- An introduction/preface of what game you are implementing and high-level design choices you have made.
- Several user stories describing proposed features.
- Functional and non-functional requirements (clearly mark them as such). Use tables and structured requirements to present them.
- To show system usage and architecture, there must be at least one UML use-case diagram and at least one UML class diagram.

Grading

- 30%: GitHub setup
- 40%: Requirements documentation (hard factors)
    - Completion of the above
- 30%: Requirements documentation (soft factors)
    - Quality of writing, thoughtfulness of planning, clarity of presentation, quality of UML diagrams, etc.