Hugh Han

Marc Feldman

Matthew Cowen-Green

Seunghwan Lee

**UI Sketch [**give a proposal on how you will use text-only input and output to get a nice looking game. To do this, present a detailed sample run of the game which shows several turns of the game play, demonstrating different use-case scenarios.]

Starting the Game

In the beginning of the game, the user will be prompted if he would like to start a new game or load an existing game. If he chooses to load an existing game, then he will be prompted to enter the file location and name of the existing game. This may throw and error, and in that case, he will be prompted again if would like to start a new game or load an existing game.

If the user chooses to start a new game, then he/she will be prompted to first enter the name of this game, and then enter the number of total players (including both humans and computers). He will then be prompted to enter the names of each player, and whether they are humans or computers.

Playing the Game

This game will be a pass-and-play type game. Player 1 starts the game. The visual representation of the game from the current player’s point of view will be printed to the screen. Player 1 will then have several options of moves he would like to make. He/she will first be prompted which of his cards (from his hand, discard pile, draw pile) that he would like to move into a different pile. He/she will then be prompted to which pile he/she would like to move the selected card. This can throw an error (such as trying to move something from the discard pile to you hand, or moving the 5 card to a build pile with 7 at the top). When the current player feels that he does not have any more moves, he can end his turn by moving a card from his hand to the discard pile, and the next player’s turn begins.

Ending the Game

Players will have the option to quit the game by pressing ‘q’. Whenever the game is quit the user will be prompted if he/she would like to save the game. When a player has used all of the cards in his/her stock pile, the game will automatically end and a winner will be declared.

**Class diagram** [include a UML class diagram showing your proposed classes. Show is-a relationships with UML inheritance arrows, and has-a relationships with UML associations. Include the data members and major public methods. (During the design process, remember to consider how you will manage unit testing of the various pieces.)]

**Class overview** [given your proposed classes and methods, describe how they will be used, in particular for the sample run you gave in the UI sketch. In other words, describe the information flow through objects and their methods to illustrate how the game play corresponds to your OO design.]

**Implementation Plan**

The most challenging part of this project will most likely be determining how to game will run, and how the structure should be designed. The roles described below will most likely change as the project progresses, and they will be just a rough idea of what each team member will be responsible for.

Coding will most likely be done in pairs, and the team will meet up approximately twice a week.

Team Members; Roles

*Hugh & Marc*

First, we will figure out how to implement Card.cpp, Pile.cpp and Hand.cpp. Afterwards, we will work on Human.cpp. After these are all finished, we will join up with Matthew & Seung Hwan to all work together on the rest of the game.

*Matthew & Seung Hwan*

First, we will figure out how to implement Player.cpp. Afterwards we will work on Computer.cpp. After these are all finished, we will join up with Hugh & Marc to all work together on the rest of the game.

|  |  |
| --- | --- |
| Dates | Events & Deadlines |
| 4/18/2015 | Discuss project; start Part A |
| 4/20/2015 | Finish Part A |
| 4/22/2015 | Finish implementing Card.cpp, Pile.cpp, Hand.cpp, and Player.cpp |
| 4/25/2015 | Finish implementing Human.cpp and Computer.cpp |
| 4/27/2015 | Discuss progress of Game.cpp and integrate code |
| 4/29/2015 | Finish Game.cpp with few bugs |
| 4/30/2015 | Finish testing Game.cpp; finish SkipBo.cpp |
| 5/01/2015 | Finish testing entire program as a whole |

(Note: integration will take much longer than you expect!)

Only one design document should be submitted on Blackboard per team. The document itself must be a *single* pdf document. It should be well written and nicely formatted. There is no length requirement for this document. However, a one page document is probably too short and a 15 page document would put us to sleep (unless it had lots of pictures). 4-8 pages with figures is appropriate.