Plan of Attack

Name: Mokehon Park

UserID: m46park

Student Number: 20691007

Breakdown of the Project

December 3 – December 6:

I think I have not been doing well for this course, and since assignment 3, I have been considerably struggling with the related contents. Therefore, before starting the project (Straights), I am going to review some important topics, especially focusing on Classes, OOP, Inheritance, UML and Design Pattern. I believe spending several days for reviewing and correctly understanding the main concepts would be better than starting the project right away.

December 7 – December 8:

I am going to draw the initial UML diagram for the project. As implementing the project, I will fix UML diagram correctly, if needed.

After that, I am going to make all required header files (.h) based on the UML. I will declare the necessary classes and methods.

December 9 – December 13:

I am going to implement all required classes and methods in .cc files. To be honest, I expect that I will be struggling with the tasks, but I will do my best to complete them.

December 14 – December 16:

I am going to test whether my project works correctly. If something is wrong, I will try to fix it. I will also check that the program runs and my project compiles well.

After that, I am going to double-check the UML diagram and write the report.

Question & Answers

Question 1: What sort of class design or design pattern should you use to structure your game classes so that changing the interface or changing the game rules would have as little impact on the code as possible? Explain how your classes fir this framework.

Answer: In my opinion, I should you the Decorator design pattern to deal with the issues.

Question 2: If you want to allow computer players, in addition to human players, how might you structure your classes? Consider that different types of computer players might also have differing play strategies, and that strategies might change as the game progresses i.e. dynamically during the play of the game. How would that affect your structure?

Answer: I can make both Human class and Computer class inherited from the same base class (probably virtual class).

Question 3: If a human player wanted to stop playing, but the other players wished to continue, it would be reasonable to replace them with a computer player. How might you structure your classes to allow an easy transfer of the information associated with the human player to computer player?

Answer: I can make Computer class inherited from Human class so that the information of the human player can be easily transferred to computer player.