

CS 246, Mokheon Park, 20691007

Overview:

I implemented the project (straights) with seven classes, which are Card, Deck, Table, Player, HumanPlayer, ComputerPlayer, and GamePlay. Hence, my project contains seven .h files, seven .cc files, main.cc and Makefile.

Design:

I did not use a specific design pattern such as decorator design pattern or observer design pattern, but I did my best to maximize cohesion and minimize coupling by using necessary classes, and their member variables and methods.

Resilience to Change:

I implemented the program with seven necessary classes, so I would deal with better a various change.

Answers to Questions:

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 use Observer design pattern.

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 Player class and Computer Player class inherited from the same base class (Player class) which is an abstract 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: First, I should make both Human Player class and Computer Player class inherited from the same base class (Player class) which is an abstract class. Then, I can make a pointer (type is Player*) which points a human player points a computer player.

Extra Credit Features:

I couldn't do this...

Final Questions:

Answer of Q1:

To be honest, I have been struggling with this course this term, so I was scared with the final project. However, I think I implemented the project better than I expected... At least, it compiles! Also, it is my first time to implement this kind of large programs. It was obviously difficult, and extremely time consuming, but I believe that it was a good experience for me.

Answer of Q2:

If I had a chance to start over, I would use a specific design pattern. After completing the project, I realized that my project could be much better and cleaner if I used an appropriate design pattern. Also, I would use more inheritance so that I can maximize cohesion and minimize coupling for my project.