Name: Kevin Nguyen

Professor: Dr. Lehr

Object Oriented programming is a very important development architecture when developing games. Its helps in breaking the domain information down into smaller pieces of data that can easily be b solved singularly. This concept therefore makes it easy to do code-reusability which makes it easier to maintain the code base of any system. Object Oriented programming also help maintain the development constructs such as Don’t Repeat Yourself, program reliability and ease of maintenance. In Object oriented programming we treat problems that we want to solve as objects with features and develop methods which will interact with those features.

In this project I am creating a Card Game with two players. I chose this game because it embodies the concepts of Object-Oriented programming to the core and shows better how efficient Object-Oriented programming is when solving real world problems. Building the game was a bit challenging for me so I took about 7 days programming the game and its rules and wrote about 750 lines of code to for the complete game. The game has been developed with three classes Player class, Card class and Game class which combines all the other class and defines the rules of playing the game.

We have taken Object-Oriented approach when developing the system and all entities that interact with the system are treated as objects with methods and properties. The players logic is written under the player class, Card’s logic is under the card class and a Game class that contains the rule of the game, how to play it and how and when the game ends.

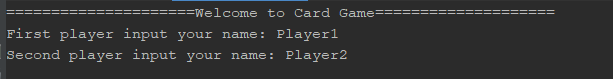
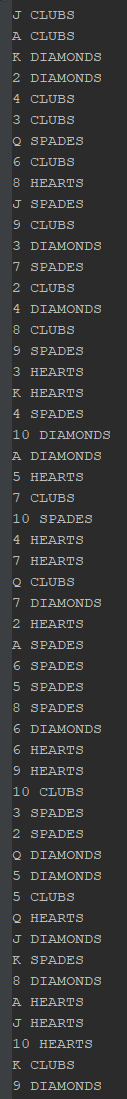
The version control tool that is used for the project is git. This tool has been chosen because of its ease of use and efficiency in managing different version of committed code during the phase of development.

**Rules of the game include the following**:

1. Players have a total of 4 cards at the beginning of the game
2. Before playing, the cards have to be shuffled in each iteration.
3. The game can only be played by two players at the same time.
4. Only eight cards can be dealt during a game iteration.
5. Players can only drop a card that matched the suit or the number/letter of the designated top card.
6. Players must draw a card from the shuffled deck if they do not have a matching card.
7. Player who matched all the cards in hand is the winner.

NOTE: Each player needs to [view] his/her deck before select the option to [drop] a card.

**Example Screenshot:**

1. Game Menu
2. A shuffled deck shown every card is mixed.
3. Text

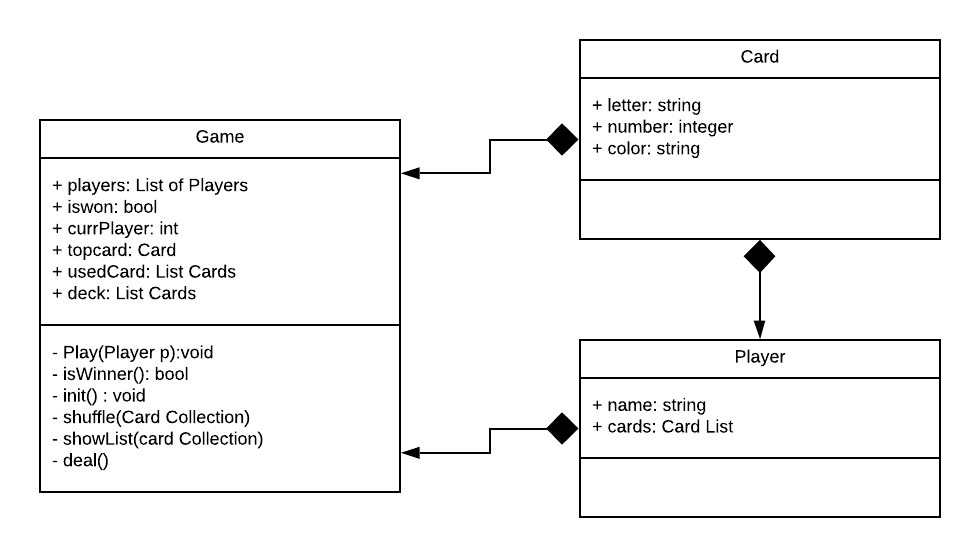
   Description automatically generatedOptions to play the game. Always [view] your card before [drop] or [pick] a card
4. Text

   Description automatically generatedIf you have a matching card, you can [drop] it. This case shows a matching suit (A of Hearts and 10 of Hearts matched with the top card Q of Hearts)
5. Text

   Description automatically generatedEnter the card number/letter to drop. Player who dropped all the cards in hand is the winner

The code base is written following OOP design padigram, the Player class contains the methods and properties of the players of the game, the Card class contains the methods and properties of a card and the finally the game class. The game class defines the rules of the games, how it is played and how to determine the winner. The Game class has the accessor and mutator methods of interacting with its variables. The main.cpp is the entry point to the application and it’s the file that has the functions that initializes the game and calls items in order to make sure that the rules of the game are properly followed.

**UML**

****