Object Design Documentation

To refer to the UML Diagram of the entire game object structure, please open "Trianta Ena – UML Diagram.png"

This game architecture consists of 16 classes and 1 interface file. Each task or process of the game is in a separate segregated file, which makes the entire game architecture easy to understand.

Due to the Tic Tac Toe and Order and Chaos game architecture from Assignment 1, I was able to use the similar style of working here, due to its extendibility.

For example, MessageStorage class acts as a storage to display messages to the user. I was able to use the same class in this card game too (by just changing the methods and messages). This made it super easy to call and use the messages, as all the display messages are stored in their specific methods.

Another example is the PlayerAction interface in this card game structure. Due to the idea which I used in the Tic Tac Toe game by creating a PlayerAction class that handled everything related to players movement (either X or O), following the similar structure, the PlayerAction interface also does the same tasks here, like, what is the player's decision and choice - to hit or to stand, make a bet or cash out.

Lastly, it has a very concise main(), as it is of only one line (StartGame.java). Because of its low memory usage, it has an efficient time complexity, and the winner calculation process (even for many players) is super-quick!

The changes I made in this Trianta Game would also be useful for other similar card games say, Blackjack. Hence, extendibility has been highly valued in this assignment too!