Damon Leaf
Carmen Yon
CSCI 4448

Project 6

Status Summary

Damon Leaf and Carmen Michael Yon's Blackjack Simulator.

The first two weeks were reserved for learning and getting a strong handle on JavaFX. We wanted to take time to focus because neither of us has had previous experience with JavaFX and only one of us had made a GUI before. Our current status has four fleshed-out wireframes for the main functions of our game (Creation, Customization, Gameplay, and Save States). Additionally, we have successfully made visual playing card creation in the GUI. By this I mean we have created FX animations to deal cards to the player, NPC, and dealer card slots (Done by Damon).

Currently, we are designing the skeleton of our 'Model View Controller' and starting to meld the actual logic and sense behind all of our graphics. We haven't completely implemented our patterns yet because we struggled much more than we initially thought we would with FX. We had a lot of trouble getting JavaFX to work on both our machines. However, Carmen is now making great progress on our MVC and has started building the structure for our model Dealer Model that will communicate with our GamePageController. After more deliberation, we decided to not use the Observer pattern to observe game states and will be handling it through the command pattern in our Dealer class.

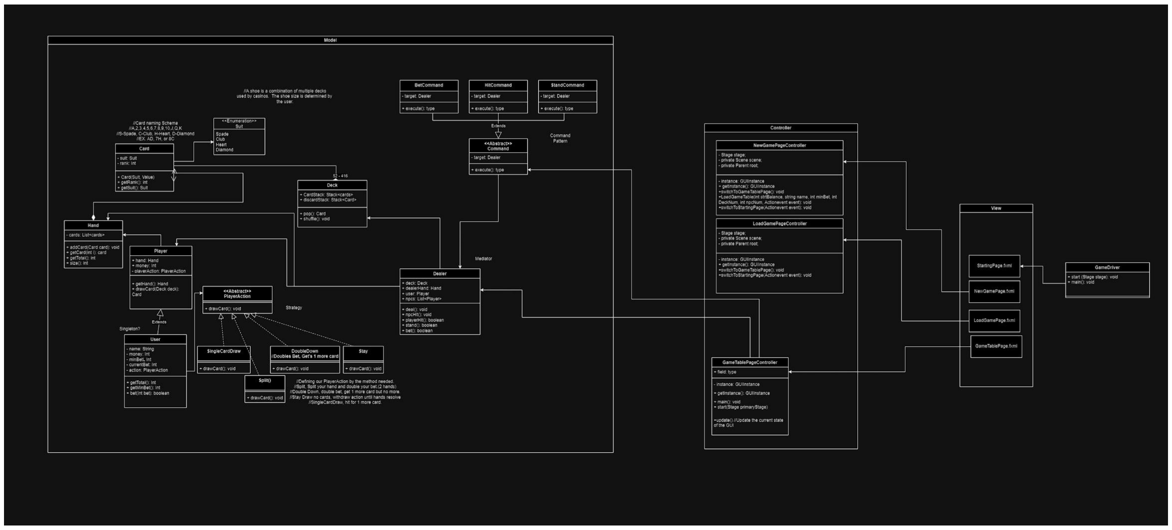
Carmen's work these past two weeks included beginning the implementation of the "model" section of our MVC, which is the simulated game of blackjack. This mostly included creating classes such as Player, User, Deck, Dealer, etc. The simulation is implemented using the mediator pattern, with the Dealer class representing the mediator. While functionality is not yet working, with several methods implemented, class structure is mostly complete.

Additionally, Carmen began implementing a command interface between the Controller and the Dealer, rather than the originally planned observer pattern. We decided against this because there was only one class that needed to observe the state of the model, rendering the observer pattern redundant for this specific application. We also have to deal with event driven IO, and having another event management system would be cumbersome. This is mostly implemented, with much room for extension as the project continues.

Class Diagram (10 Points):

- Class diagram of currently implemented classes. Highlight patterns in use

Link:
https://drive.google.com/file/d/1N1pT8hbXMsE1Tw4eTkc7YWFwBcRpG8mx/view?usp=drive_link



Plan for Next Iteration:

We have three main things left to complete: finishing the blackjack game logic and if time allows, we want to be able to create save states for our game. Now that we have most of the graphic work done, completing the blackjack logic shouldn't be too difficult. We first need to implement the MVC and sync up our GUI animations with the logic for our blackjack gameplay. Once we have basic blackjack running for a single player, we will code logic for the NPC players following the basic blackjack strategy chart. If we still have enough time left we will design a way to save a player game state as a byte stream in JSON, allowing players to load the previous game they were playing. By the due date, we would like our game to run with NPCs smoothly and have improved visual styling to theme the GUI to a real Blackjack table.

Demonstration (40 Points): DECEMBER 7TH 11 AM on Zoom
https://cuboulder.zoom.us/j/7117560160