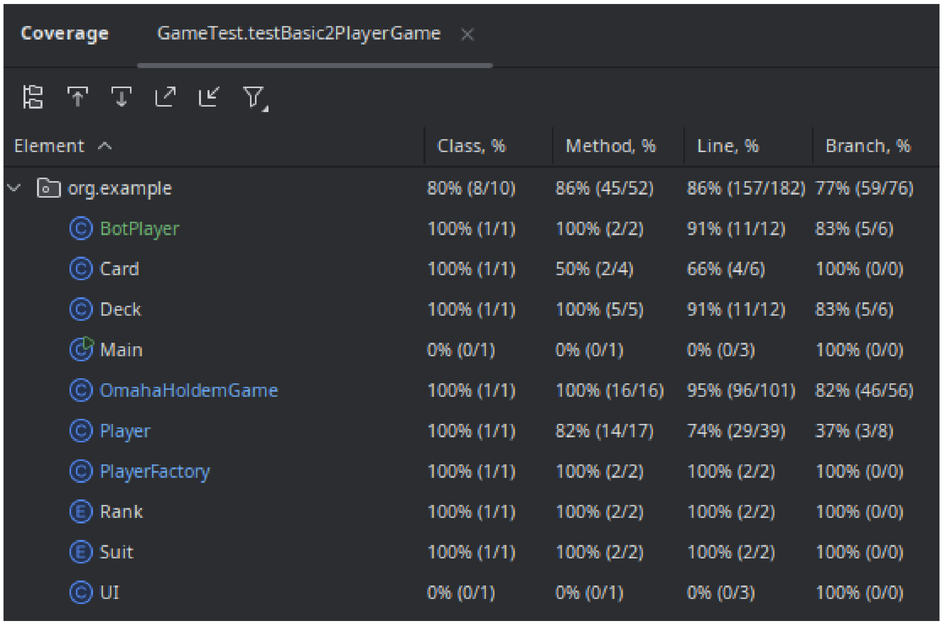
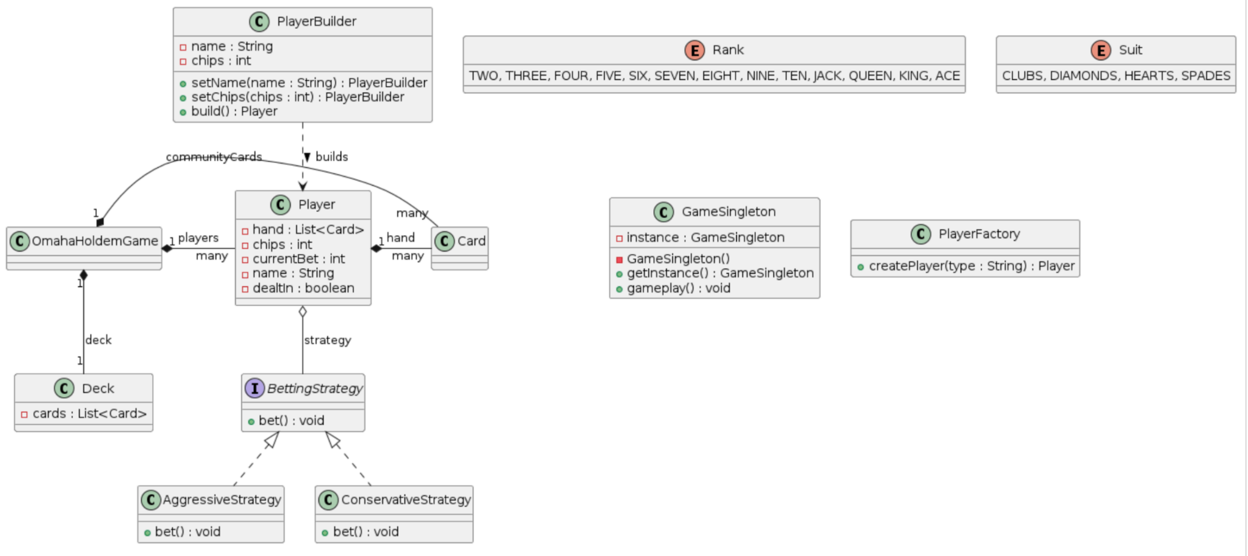
Homework 7: OOAD Semester Project Interim Report

Status Summary:
Project Title: Omaha Hold'em
Team Members: Connor Doane, Chloe Cantalupo
Work done: Thus far, we've implemented the Player class for the human player, we haven't yet implemented the opposing automated players that you play against but it will be a subclass of the Player class that incorporates a BettingStrategy class to decide its actions, instead of taking in user input. Additionally, the Deck and Card classes have been implemented to be a complete deck, shuffled, and dealt out to each player as well as the community cards shared by the table. The OmahaHoldemGame class holds most of the logic of the gameplay, including handling the deck and the betting rounds held by the players. The amount of work was 50/50 between each of us, with each of us implementing multiple classes.
Changes/Issues: Initially we had implemented Card as a nested class within the Deck class, because we initially envisioned Cards as being intrinsic to the Deck itself. However, this introduced too much coupling between the two classes, especially when it came to having each player hold their own hand of cards and the community cards which are separate from the Deck after having been dealt. As a result, we had to change the class to be its own class, reducing its coupling.
Patterns: Thus far we've implemented the Factory pattern for the Player class, which easily allows us to add or remove new players from the game, and in the future will allow us to easily choose the strategy that Bot players will take during gameplay. Additionally, we've begun to implement the Builder strategy on the OmahaHoldemGame class that will allow us to build a more complex instance of that class that will configure the game, including initial chip count, amount of players, small blind, and more to set up many different types of games players may want to play.
Test Coverage:



Element	Class, %	Method, %	Line, %	Branch, %
org.example	80% (8/10)	86% (45/52)	86% (157/182)	77% (59/76)
BotPlayer	100% (1/1)	100% (2/2)	91% (11/12)	83% (5/6)
Card	100% (1/1)	50% (2/4)	66% (4/6)	100% (0/0)
Deck	100% (1/1)	100% (5/5)	91% (11/12)	83% (5/6)
Main	0% (0/1)	0% (0/1)	0% (0/3)	100% (0/0)
OmahaHoldemGame	100% (1/1)	100% (16/16)	95% (96/101)	82% (46/56)
Player	100% (1/1)	82% (14/17)	74% (29/39)	37% (3/8)
PlayerFactory	100% (1/1)	100% (2/2)	100% (2/2)	100% (0/0)
Rank	100% (1/1)	100% (2/2)	100% (2/2)	100% (0/0)
Suit	100% (1/1)	100% (2/2)	100% (2/2)	100% (0/0)
UI	0% (0/1)	0% (0/1)	0% (0/3)	100% (0/0)

UML Class Diagram:



BDD Scenarios:

1. Game Initialization

Feature: Initialize a new Omaha Hold'em game with specified parameters.

Scenario: Starting a new Omaha Hold'em game with specified player count and blinds

Given the system is ready to start a new game when I initialize a new Omaha Hold'em game with 4 players, each starting with 5000 chips and a small blind of 50. Then the game should start with 4 players each having 5000 chips and the small blind should be set at 50

2. Card Dealing

Feature: Deal cards to each player at the beginning of the game.

Scenario: Dealing initial hands to players

Given a new game of Omaha Hold'em has been initialized with 4 players when the cards are dealt at the start of the game, then each player should receive 4 cards

3. Betting Process

Feature: Conduct a betting round where players can bet or fold.

Scenario: Conducting a pre-flop betting round

Given each player has been dealt 4 cards in Omaha Hold'em when the pre-flop betting round is conducted
Then each player must either call the current bet, raise, or fold

4. Flop Dealing and Betting

Feature: Deal the flop and perform a betting round.

Scenario: Dealing the flop and conducting betting

Given the pre-flop betting round has completed without all players folding when the flop is dealt, then three community cards should be added to the table and a subsequent betting round should follow

5. Winner Determination

Feature: Determine the winner of the hand at the end of the game.

Scenario: Determining the hand winner

Given that all betting rounds have been completed in Omaha Hold'em when the showdown occurs, then the player with the best hand according to Omaha Hold'em rules wins the pot

Plan For Next Iteration:
For the next iteration, we need to implement the bot strategies to allow players to play against someone other than a human player. For this we will use the Strategy pattern to allow for multiple types of opponents, such as those who bluff a lot and those who play more passively, only betting when their hand is actually good. We also need to implement the hand grading process to determine the winner of each hand, to know who to award the Pot to after a hand of gameplay. Additionally, the UI is currently just a command line interface, and we’d like to create a full graphical interface to make the game more easily usable and visually appealing.