Blackjack Game

Final Report

Team Jack

By: Kyle Choate, Cooper McKenny, Leon Nguyen, Joseph Mock

Course: CPSC - 224 Software Development

I. Introduction and Project Description

Our project was a virtual implementation of the popular card game blackjack. Blackjack is a simple card game where players play against the dealer, hoping to draw cards that have values that add up to 21 or as close as possible where the winner is the person with cards that add up closest to 21 or 21 itself which is called a blackjack. If a player or dealer goes over the specified limit of 21, they bust and lose immediately. The project we put together surrounded these principles where it was split up into 3 main groups each consisting of more technical aspects. Those main groups were the cards used in the game of blackjack (which is a standard deck of 52 cards), the users / players of the game, and the game itself. Files within these groups were to give the game structure and interact with each other to provide a smooth experience when playing the game.

II. Team members - Bio and Project Roles

Cooper McKenny is a sophomore computer science major with interests in software development and computer efficiency. During this project he worked on debugging, handling variables and edge cases, game functionality, and other miscellaneous tasks required to keep the code clean and able to run smoothly. He has experience with C/C++, python, assembly, and java. Technical interests include both computer software and hardware, with a larger interest in networking and software development.

Kyle Choate is a student at Gonzaga University who majors in Computer Science with a minor in Catholic Studies. He enjoys designing and programming creative projects and seeing how different parts come together. During this project, he contributed to the overall structure in which Classes were connected to encapsulate attributes and methods. He has experience with C/C++, Python, Assembly, and Java. He enjoys math and abstract algorithms but also has an interest in software development and creative processes involved in projects.

Joseph Mock is a second year Computer Science student at Gonzaga University. He is highly interested in cyber security and artificial intelligence and plans to graduate in Spring 2027 with a concentration in cybersecurity. His skills include C / C++ and Java. During this project, he helped to build a framework for how the game is built creating small classes that will create a basis like cards, collections, and hands for the players.

Leon Nguyen is a student at Gonzaga University majoring in Mathematics and minoring in Software Development. Interests consist of statistical analysis, using quantitative methods, and mathematical concepts and seeing how these can be applied to data analytics in fields such as sports. Skills include experience with C/C++ and Java. During the project, he worked on building a class that handles betting, he also helped collaborate with others to help identify, debug, and test errors to ensure the game functions correctly.

III. Project Requirements

The requirements for this project were a game that included a UI where the user could input a number of players that they wish to play with. It was also required to have several smaller game components like a win lose screen with winner / loser information, settable player names or icons, and functionable game components if mimicking a real-life game. Specific requirements relating to the game of Blackjack itself is that players must be able to set a valid betting amount before starting their turn. Players then receive a hand of two cards where players should then be able to perform three main actions relating to their turn, hit where they continue to draw, stand if they feel comfortable with their current hand, and double where they double their bet and only draw once. The project should require an artificial intelligent bot which is the Dealer of the game, this is what the player's play against.

IV. Solution Approach

In solving the actual implementation of a blackjack game, the team started with creating a UML diagram and breaking down the aspects of Blackjack into small parts and classes. Each aspect of the game, whether it was a card, a deck of cards, the dealer, the player, were all their own classes and each had different ways of interacting with each other just like the real game of blackjack would be like.

V. Test Plan

When it comes to the game that we created, testing is fairly easy because blackjack is already a game that exists. Our project goal was to create a virtual blackjack that worked identically to the real game that can be played. For testing we asked questions like; does the deck object work like a real standard deck of 52 cards? Meaning is there no repeats of cards, and each card has its own suit and color and there should be 4 of each number each tied to a different suit and 2 black and 2 red. We also wanted to test if essential features to the game worked like the bet systems. Players should always win if the dealer busts and they don't and should also be paid out according to the bet that they placed. We also wanted to test that the project requirements were made to standard, so they could actually play with multiple players and each player could set their own name.

An effective way to play the game was to play the game, but with plenty of players. When this is done, there are a lot of very different situations that happen, allowing us to easily and quickly identify odd cases that defy our expectations.

VI. Project Implementation Description

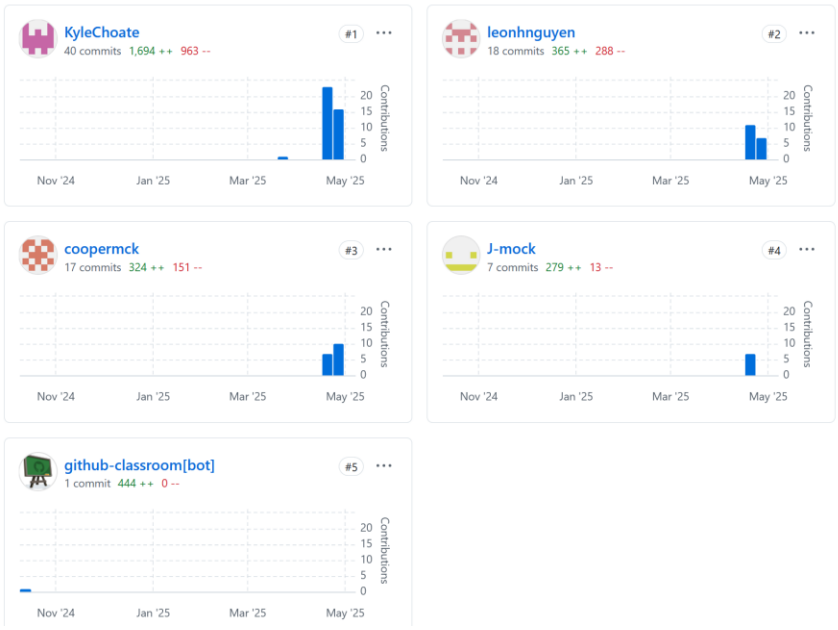
The team implemented several lasses all designed to mimic a different portion of the game of blackjack. The main functions were round and turn where round iterates through and gives turns to each player of the game where they get to make decisions on how they would like to play blackjack. Players are able to make a bet which is handled by their own class and if they win, they will be paid 2x that amount that they bet. If they lose, they will lose the amount that they bet. We implemented this by creating bet and player classes that used basic getters and setters when being pushed through the game. We then implemented the player classes to accept a list of 2 cards that will act as their hand when playing the game. If they choose to hit, they will then be given another card into their list and the deck will also lose a card based on the list that the deck has. With these implementations and many small technical checks and functions the team was able to create a functioning virtual game of blackjack.

View:

```
Hello Team Game
Welcome to Blackjack!

How many players would yoou like to play with?
 > 2

Player #1, what is your name?
 > Joe

Player #2, what is your name?
 >
Your name is now: Phoenix
Joe:  0 points
Phoenix:  0 points

-------------------------

We have entered round #0!

Joe, please place a bet
A: 5
B: 10
C: 100
D: 500

Your balance: 1000
```

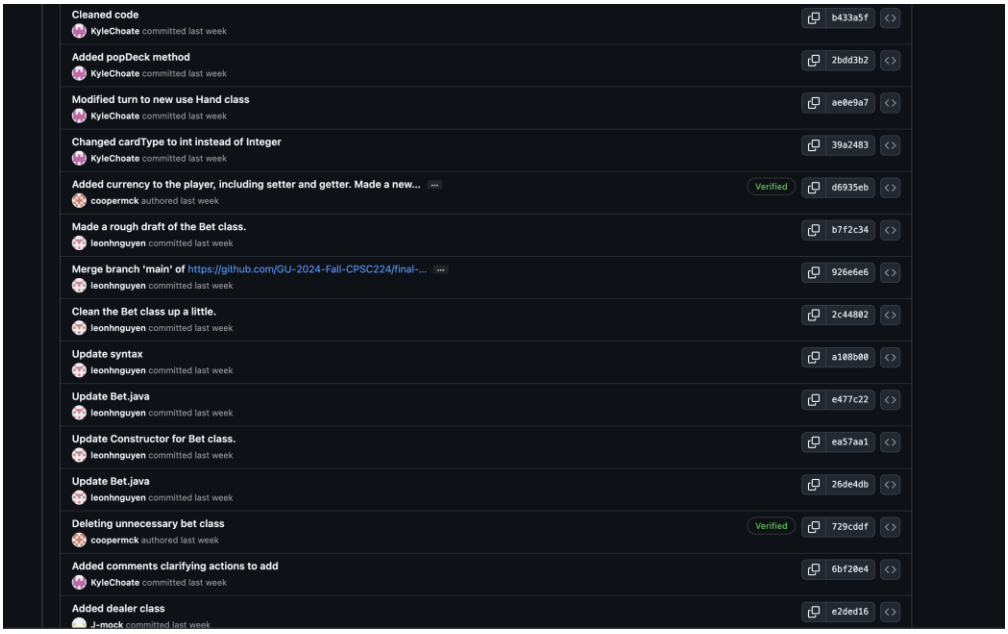
VI(a). Project Statistics

- 88 total commits
- 20 files changed
- 1300 total lines of code written across the team.



- One software test
- One branch (main)

Portion of Git Repo Tree:



https://github.com/GU-2024-Fall-CPSC224/final-game-project-teamjack

VII. Team Collaboration and Tools Used

The team engaged in direct collaboration with each other in-person. The primary case of asynchronous work was very small portions of code that were mainly just building the structure of different classes. We did this to avoid accidentally causing changes to files others were working on. Group coding was used a lot to develop and implement large portions of the game that were much more important and required more testing. Each member of the group kept in touch through texting and frequently checking the Git Hub repository for new commits. The team had little struggles with keeping up with the Git Hub as it was a goal to make pushes and pulls frequently where only a little code was changed each time.

VIII. Future Work

Advancing this project would mainly be adding a couple more features that are sometimes used in blackjack and creating a better GUI for the game itself with Java Swing. Playing the game on the console works fine, but adding a framework to have a better visual of the game of blackjack would be a good addition to the well-functioning game that the team has created. Also, in some games of blackjack players are able to split the two cards that they are given and play two hands at once, and sometimes they are able to take insurance on certain hands. It would be good to add options for optional game mechanics.

IX. Glossary

Bet – Placing an amount of currency to wager against the dealer in a game of blackjack

Class – java file that holds an object created

Hit - A decision in the game of blackjack where the user chooses to be dealt another card form the deck in hope of getting closer or to the desired value of 21 for their hand.

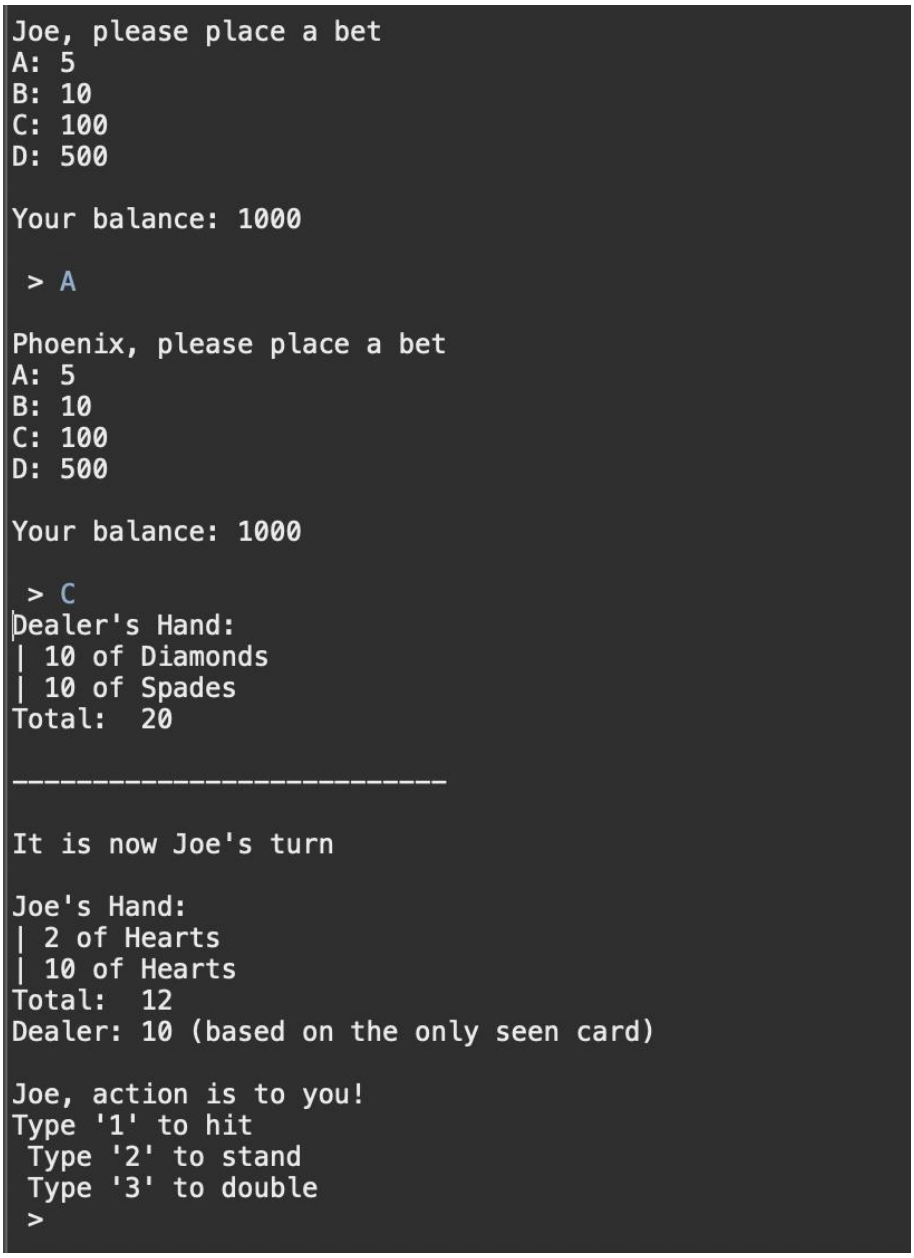
Stand – A decision in the game of blackjack where the user chooses NOT to be dealt nother card from the deck in hope of having a better chance of winning against the dealer.

X. References

Bicycle Cards. "How to Play Blackjack." Accessed May 1, 2025. https://bicyclecards.com/how-to-play/blackjack.

XI. Appendices

Appendix A: Game View



In this text example, we can see user prompts are on lines with “ > “ text. Hands are listed in a neat manner with text-based cards and score displays.

```
It is now Joe's turn

Joe's Hand:
| 2 of Hearts
| 10 of Hearts
Total:  12
Dealer: 10 (based on the only seen card)

Joe, action is to you!
Type '1' to hit
 Type '2' to stand
 Type '3' to double
 > 1
Hit!
3 of Clubs

Joe's Hand:
| 2 of Hearts
| 10 of Hearts
| 3 of Clubs
Total:  15
Dealer: 10 (based on the only seen card)

Joe, action is to you!
Type '1' to hit
 Type '2' to stand
 > 1
Hit!
10 of Clubs
Bust!

Joe ended their turn with 0 points

Joe:  0 points
Phoenix:  0 points

--------------------------
```

```
Phoenix, action is to you!
Type '1' to hit
 Type '2' to stand
 Type '3' to double
 > 2
Stand! Ending action

Phoenix ended their turn with 17 points

Joe:  0 points
Phoenix:  17 points

The dealer's full hull hand shall now be shown
Dealer's Hand:
| 10 of Diamonds
| 10 of Spades
Total:  20

Dealer's Score: 20
Now we will see who won their bets...

You lost... :(

Winner Winner!

--------------------------

We have entered round #1!

Joe, please place a bet
A: 5
B: 10
C: 100
D: 500

Your balance: 995
```