Group High Rollers Blackjack

Final Report

High Rollers
Harrison, Cooper, Asher

Course: CPSC 224 - Software Development

I. Introduction and Project Description

The purpose of this project description is to give insight into the development and engineering process from our group members into creating a fully working Blackjack java GUI representation. This document will go into the technical details about how the game is designed and operates. It will act as a summary of this project's progress and details about our efforts towards designing an entire game.

II. Team Members - Bios and Project Roles

Harrison Sheldon is a Bachelor of Science Computer Science Student interested in game development and software development. His prior projects include class projects such as a GUI Farkle and other terminal based games. Harrison’s skills include C++, Java, Python, and UI. For this project his responsibilities included fully creating the GUI representation of Blackjack

along with integrating other members backend logic and adding more logic to make a fully working GUI java game.

Cooper Stepanian is a Bachler of Arts Computer Science student interested in designing graphics in photoshop and game development. His prior projects include other computer science class projects and performing leetcode practice problems. Cooper's skills include Photoshop, IntelliJ, C++, Java, and python. For this project Cooper was responsible for creating photoshopped screens for our UI, backend game development, and helping to implement new feature along with Harrison. Additionally, Cooper made significant progress on our design documents and presentation.

Asher Weitz is a Bachelor of Arts Computer Science student interested java and python programming. His prior projects include class projects as such from 122 and 222 classes involving C++ and python. Asher's skills include C++, Java, python and other collaboration skills. For this project Asher was responsible for adding the main betting feature and backend code involving the blackjack game classes like card and deck. Asher also made significant progress on project documents and the UML class diagram.

III. Project Requirements

This section will provide a description of the major features that must be implemented for an effective and working blackjack game. It will include descriptions of the project game design and how the major features plan is to be implemented. This includes the user interface design and how the game visually will be presented and the software design.

For the initial splash/ intro screen of the game we want to let the user know what game they are playing, letting them know who made it, two sets of cards for UI/UX, and below that a start button for beginning the game. For our requirement of having settings to choose from, we want them to be able to choose between betting or playing till they win or lose. For the second screen, which is the play screen, we want to display the player and dealer cards in the middle of the screen. In the north side we prompt the user to hit or stand. On the south end of the screen, we have on the left side, the players wins/losses or their bankroll amount and on the right side we an area to show whether they won the round. For the final screen we will show that the player lost if they ran out of money from betting or if they just lost from selecting the win/loss setting. If they won, then we will show them that they won. If they chose to quit in the betting mode, then we will show them the end screen with their betting amount left.

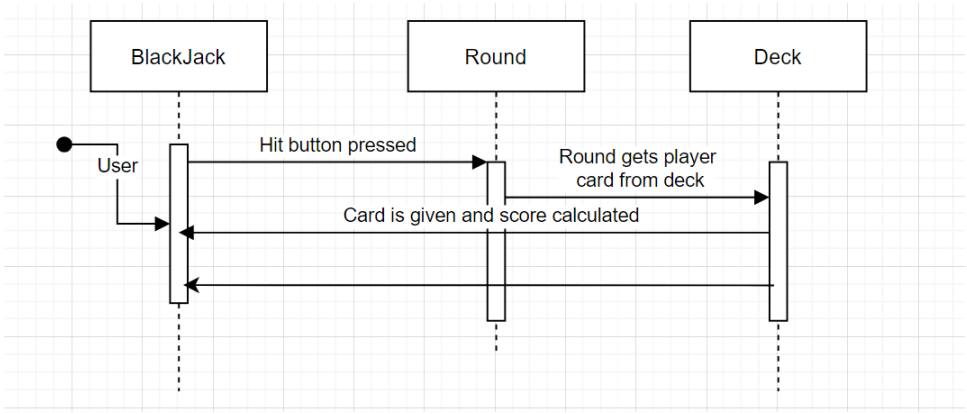
For our structure of our Blackjack game, we plan on having 8 classes. We plan on having a player and dealer class which has a hand object, and getters and setters for their name and hand. Classes for card and deck which deals cards to the player and dealer. A hand class that has an ArrayList of Cards and uses methods to add cards to the hand and to score the players and dealers card totals. A Blackjack class and a round class that oversees creating the objects and running the game, and finally a betting class that is in charge of adding and subtracting a player's Bankroll (the total amount of money that the player has to bet) amount.

IV. Solution Approach

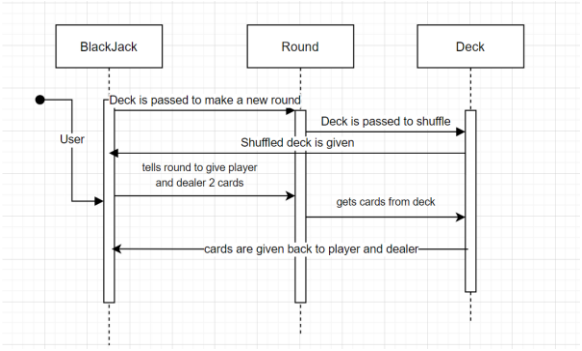
Lift materials from your presentation and any other materials about your solution. This should demonstrate to the reader what your project included, at least according to the current design. Include your UML and sequence diagrams here.

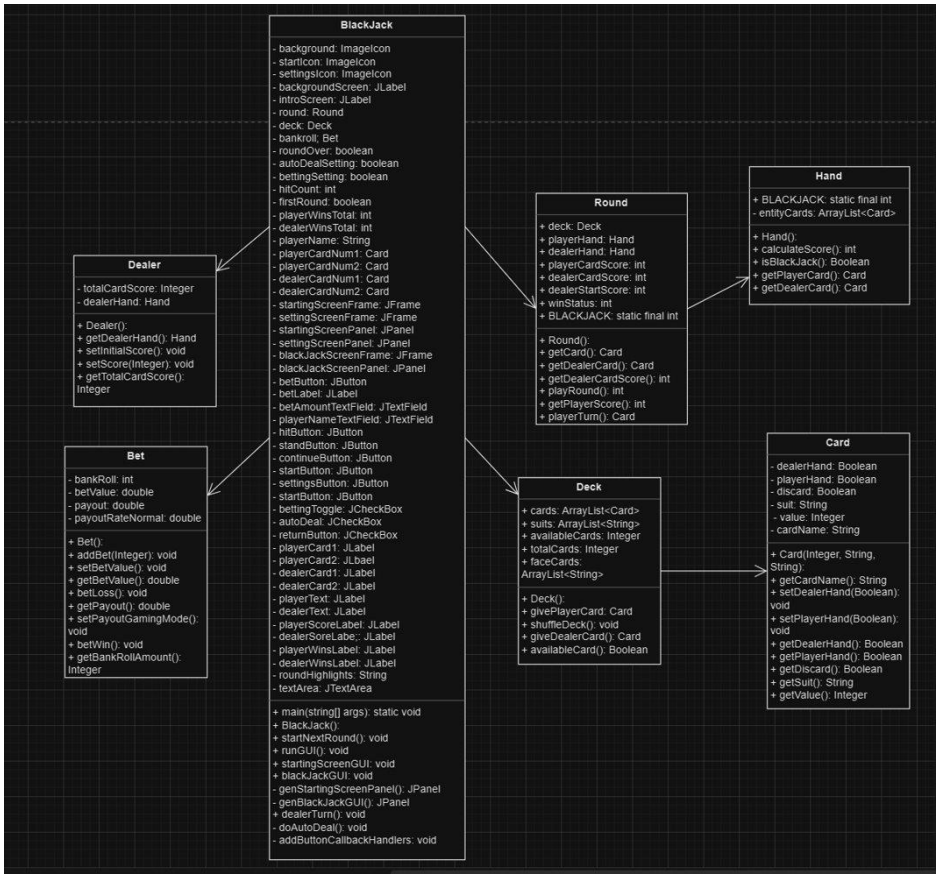
Some of the main game features include tracking a deck to ensure accurate odds of winning , working GUI , working settings , and accurate player and dealer logic . The Game and UI features include card layouts , terminal printing info, and game settings. The cards are arranged on the screen on the left side with the new cards showing up to the right. The games settings include an auto bet mode, 4 to 2 beat the house payout, setting a player's name, and a bet mode.

When a player hits:



When a round starts:





V. Test Plan

Bring in any of your test plans and results here. These should demonstrate how you validated your solution to determine whether it works and if it satisfies the requirements of the project. Look over the plan to ensure your requirements all have at least some kind of unit, functional, or user checklist test that addresses them.

Unit tests were used to address the functionality of the backend code to make sure that the logic for the game worked. This includes testing the deck, betting feature, hand, and cards to ensure functionality. To verify the main game's functionality, we enlisted the help of other individuals to play test as well as we tried to break the game. Through playtesting the different settings and features the game was able to be stamped with a testing approval.

VI. Project Implementation Description

Explain what parts/subsystems of your proposed architecture your team implemented. Mention the current progress at each part, i.e., how much of the proposed functionality you have completed.

Our biggest subsystem of our project was creating the GUI using Java Swing and implementing different windows that housed the buttons and panels for the start, settings, game, and end screen of our game. More parts of the architecture from our project include creating a Dealer and Card class that is used to keep track of what cards in the deck are assigned to the player and the dealer per round. A major subsystem for our project was the game flow that included an algorithm for the dealer and the Hand class that kept track of the cards in each hand and the ability to draw a card for the dealer/player hands. Having two different modes of scoring was another subsystem, one being the betting feature and the other being a win/loss mode.

I recommend to include plenty of images and pictures of the following where appropriate:

- any diagrams/figures that visualize various features of your prototype;
- the **screenshots of your user interfaces running**;
- pictures of your team testing and debugging the devices, programs, etc.

Note: A well-thought and clear diagram is better than long and descriptive text.

More screenshots of user interface in Appendix A

If your document starts to be very long due to screenshots and diagrams, please put at least some of them into an appendix to this document.

Also include a link to your git repo here.

GU-2023-Fall-CPSC224/final-game-project-high-rollers: final-game-project-high-rollers created by GitHub Classroom

VII. Future Work

The next logical steps for building the project would be to redesign some of the features to be much harder to break and less susceptible to bugs. Additionally, adding more animations such as when the dealer hits or flipping over the cards. The betting feature could also be improved by adding chips that can be bet instead of typing into a text box with the bet. The game is not necessarily close to a release as there are a lot of redesign and bug fixes to improve run efficiency and user playability.

VIII. Glossary

Define technical terms used in the document.

IX. References

Cite your references here if you've got any. At the very least you can cite this:

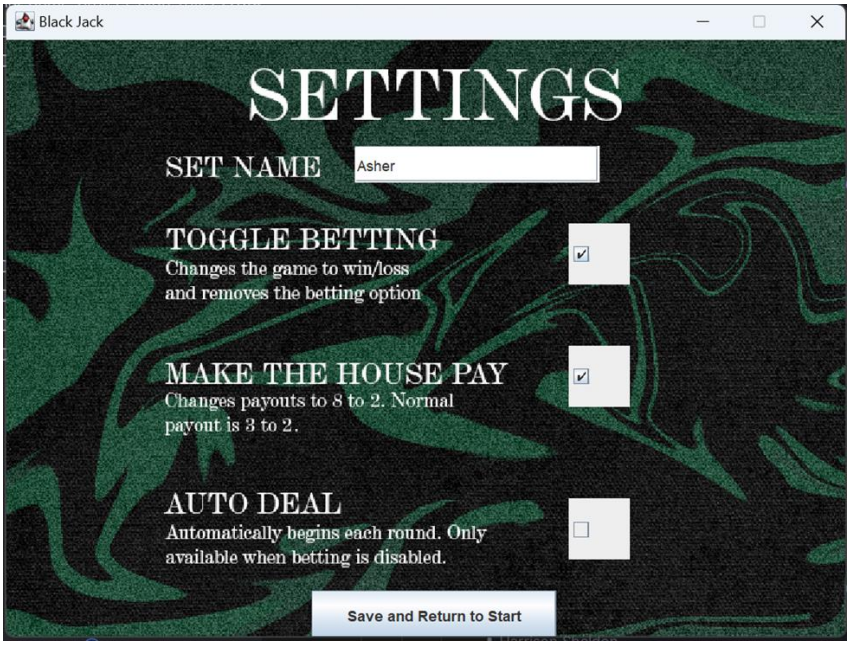
For the papers you cite, include the author(s), the title of the article, the journal name, journal volume number, date of publication and inclusive page numbers. Giving only the URL for the journal is not appropriate.

For the websites, give the title, author (if applicable) and the website URL.

Please use either Chicago or IEEE format for your citations

X. Appendices

As needed, copy over your appendices for the various sections. You can have as many appendices as required. Normally, they're numbered with letters:
Appendix A Screenshots:



Appendix B

…

Appendix *n*