Northeastern University

# Making Blackjack with PyGame

Xinrui Yi

## Introduction

This project presents a Python implementation of the classic card game Blackjack using the Pygame library, following the principles of Object-Oriented Programming (OOP).

The game features a graphical user interface with multiple pages, including a betting selection page, a game page where cards are drawn and displayed to the player, and a scoreboard page that shows the outcome of each round along with the corresponding monetary gain or loss.

By incorporating OOP principles and leveraging Pygame's graphical capabilities, this project provides an immersive and enjoyable gaming experience. It serves as an educational resource for understanding OOP concepts, game development, and building interactive applications with Pygame.

The project's modular structure allows for easy maintenance, extension, and customization, making it suitable for beginners and experienced developers alike.

## Objectives

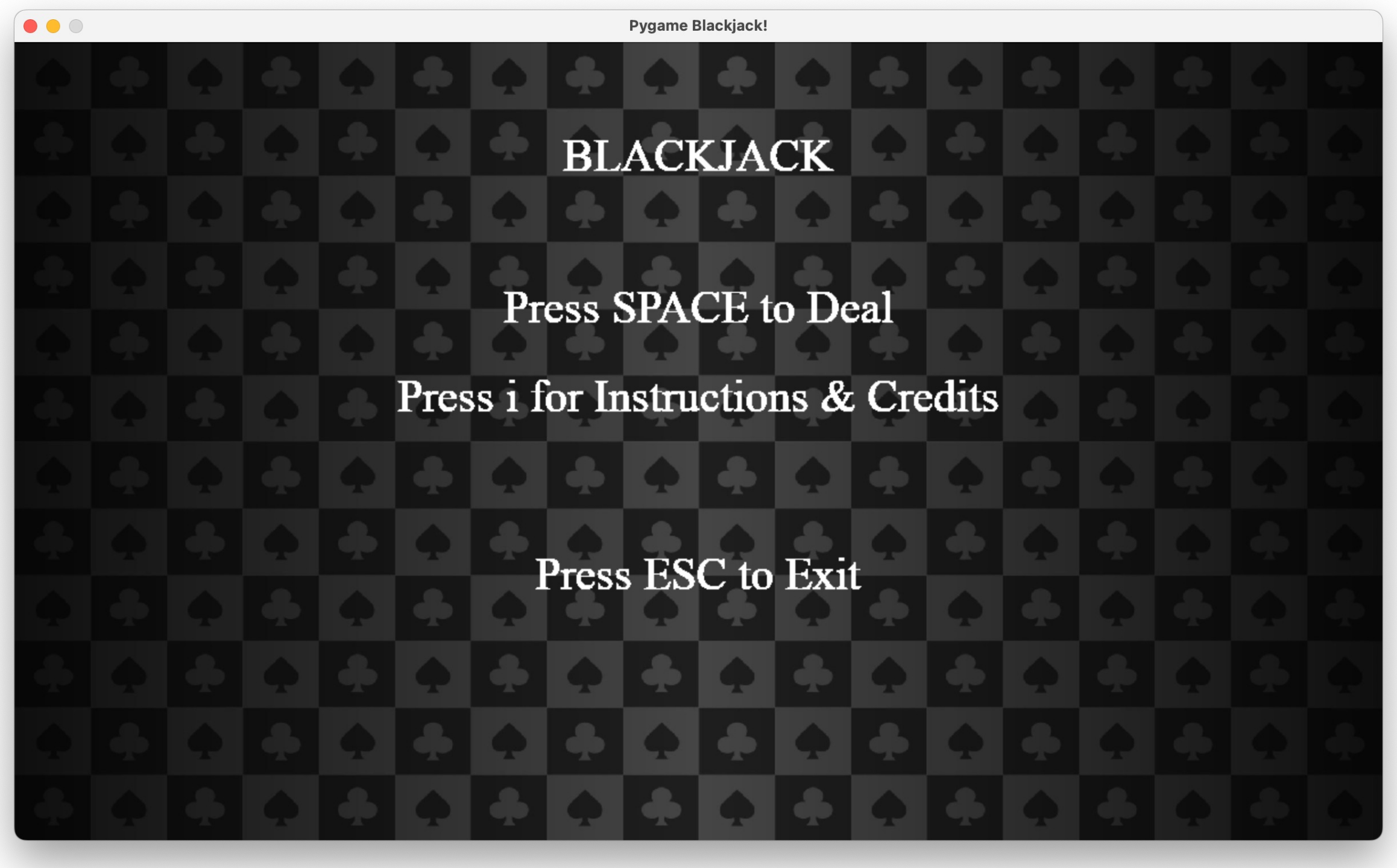
- ✓ Utilize Object-Oriented Programming (OOP) principles to create modular and reusable code
- ✓ Design and implement unit tests
- ✓ Allow players to place bets on each round, providing a realistic gambling experience.
- ✓ Display the drawn cards to the player during gameplay, creating an interactive and immersive game environment.
- ✓ Calculate and display the outcome of each round on the scoreboard
- ✓ Track and update the player's monetary gain or loss after each round, reflecting the betting results.

## Methods & Materials

- ❖ Python
- ❖ GUI & UI: PyGame
- ❖ Vision Control: GitHub.
- ❖ Style: PEP 8

## Game Interface

Welcome Menu:


Pygame Blackjack!

BLACKJACK

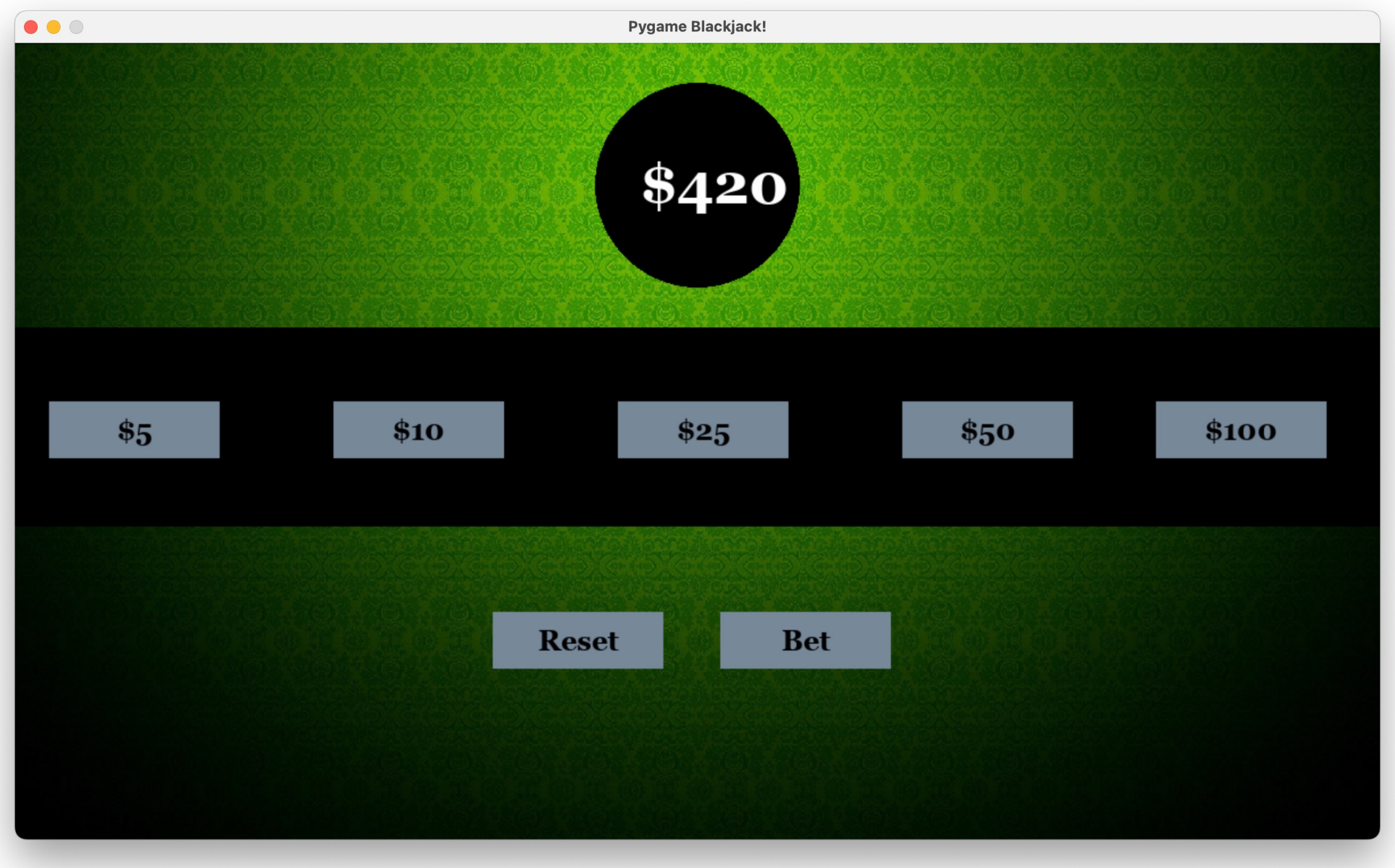
Press SPACE to Deal

Press i for Instructions & Credits

Press ESC to Exit

## Betting Selection

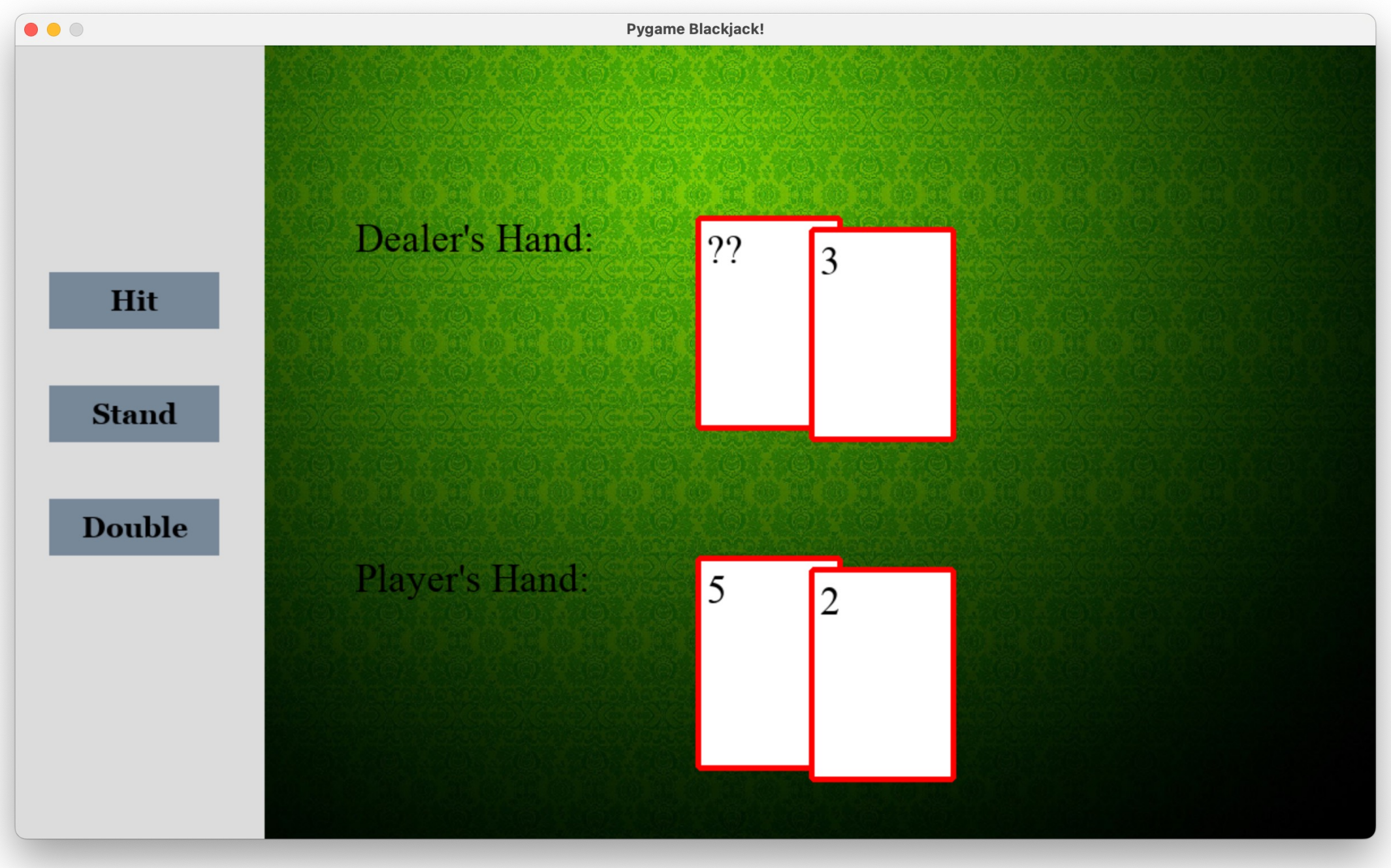
- ❑ Choose different amount to bet or reset


Pygame Blackjack!

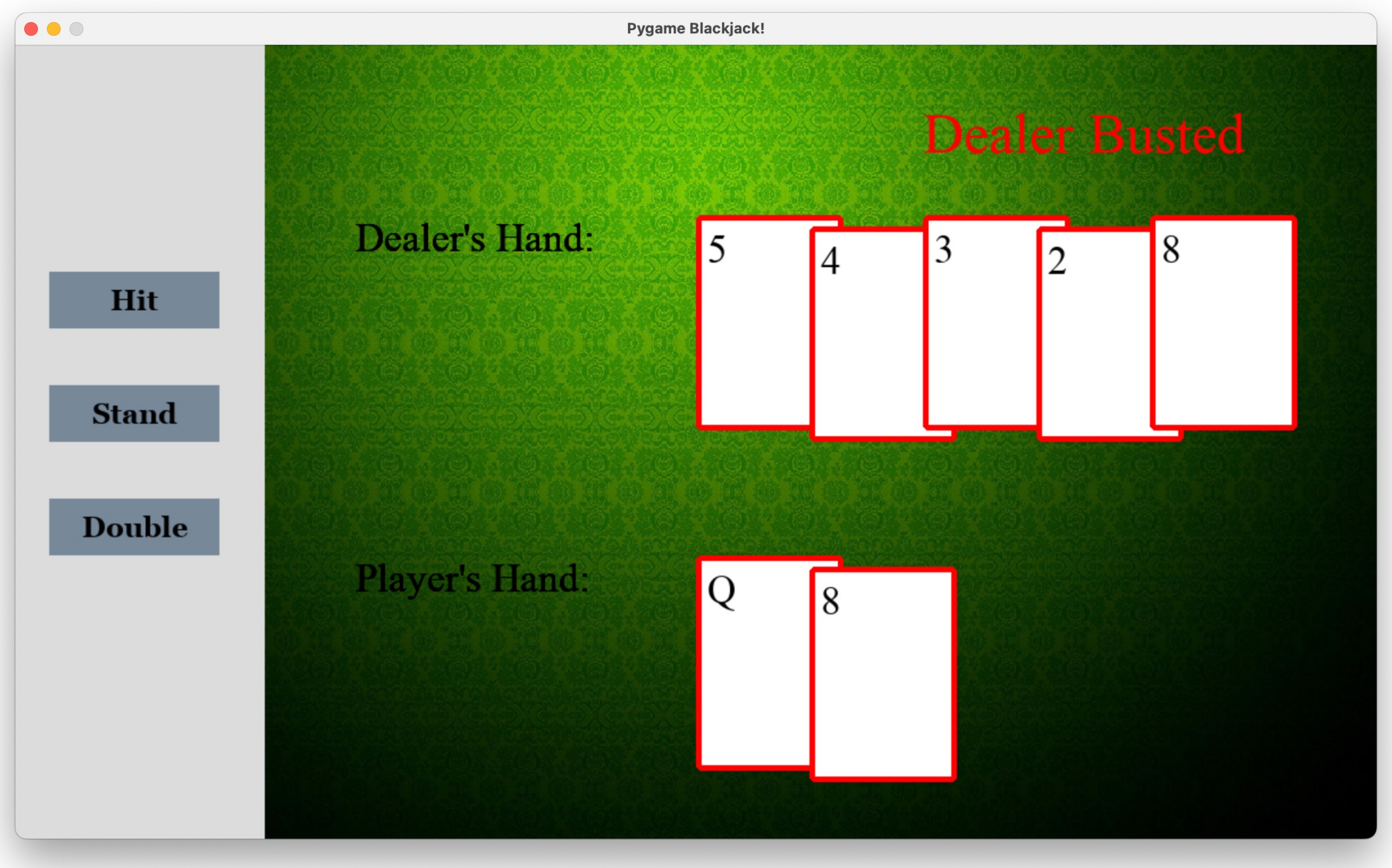
## Main Game

Multiple Player Commands

- Hit
- Stand
- Double


Pygame Blackjack!

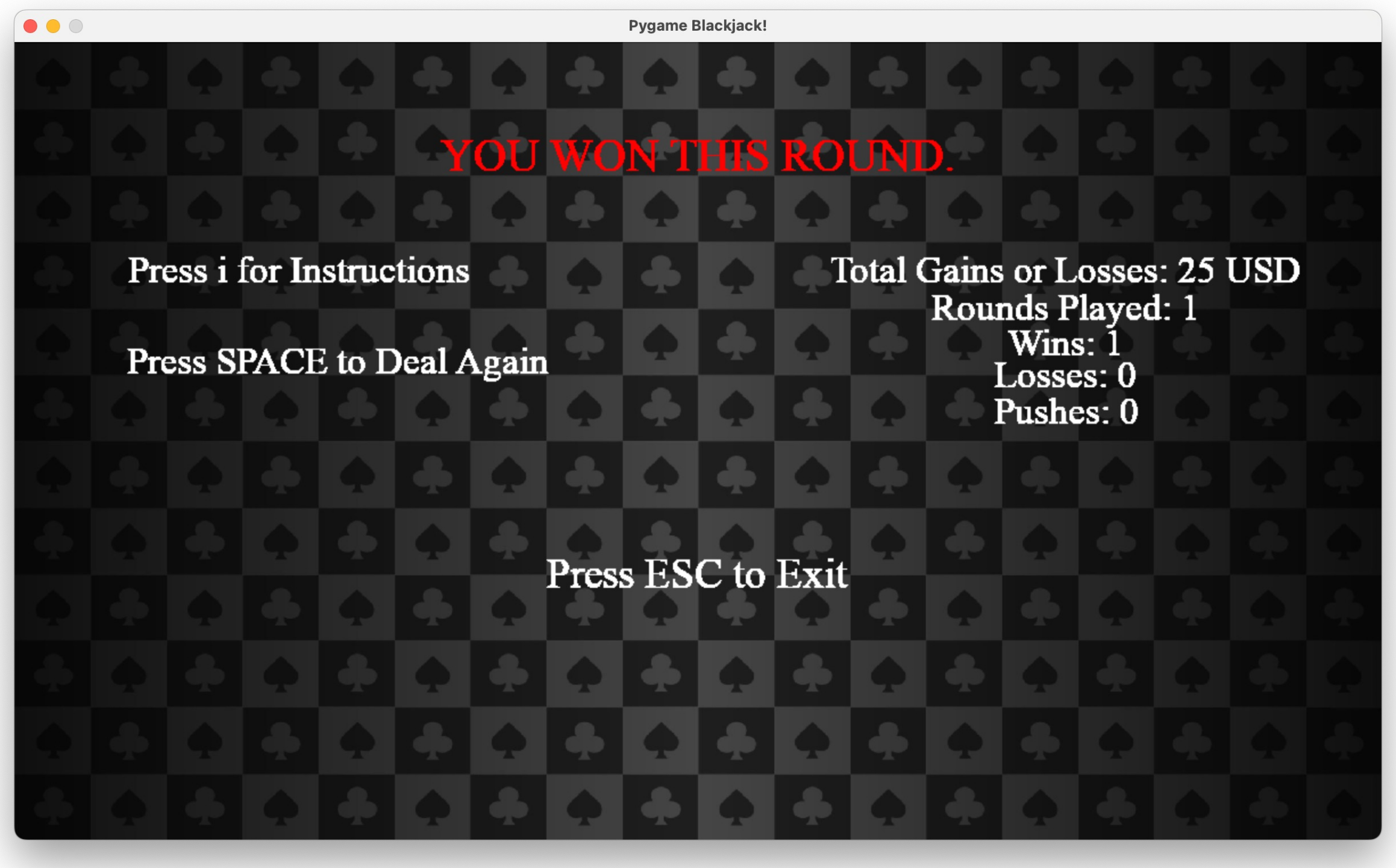
Instant in-game response


Pygame Blackjack!

Dealer Busted

## Scoreboard

- ❑ Game results display
- ❑ Players' overall gain or loss
- ❑ Win loss or push statistics


Pygame Blackjack!

YOU WON THIS ROUND.

Press i for Instructions

Press SPACE to Deal Again

Total Gains or Losses: 25 USD
Rounds Played: 1
Wins: 1
Losses: 0
Pushes: 0

Press ESC to Exit

## Conclusions

The project accomplished its objectives by developing a fully functional game with a graphical user interface, encompassing features such as a betting selection page, game page, and scoreboard page.

The integration of unit tests for the Hand and Deck classes ensured the reliability and accuracy of their operations, instilling confidence in their functionality. By utilizing the Pygame library, the game provided visually appealing graphics and intuitive user interactions, enhancing the overall gameplay experience.

While the project has achieved its primary goals, there are several potential avenues for future improvements and expansions. Some of these include:

- Adding additional game features: Introducing new elements such as split hands, insurance bets
- Incorporating multiplayer functionality: Implementing multiplayer capabilities would enable players to compete against each other, either locally or online, further increasing the game's engagement and appeal.

## References

https://medium.com/geekculture/understanding-oop-python-with-a-deck-of-cards-%EF%B8%8E%EF%B8%8F-99c31ea5acaa
https://www.makeuseof.com/start-menu-and-game-over-screen-with-pygame/
https://stackoverflow.com/questions/68761290/how-to-insert-url-link-inside-a-pygame
https://www.vecteezy.com/vector-art/3520134-dark-poker-background-of-spades-and-clubs
https://github.com/Mozes721/BlackJack

## Acknowledgements

Finished by:
Byunghyun (Ben) Ko
Deo Gracias (Gio) Ong
Xinrui (Richard) Yi

Advised by:
Dr. Mark Miller

Northeastern Khoury College of Computer and Information Sciences