Software Development Report

CS 3300-002 Spring 2025

Jack Dodge, West Manison, George Morales, Sebastian Sanchez

1. Project Title: MLG John Doe Portfolio Website  
2. Test Plan for the final Blackjack Game test  
3. Test Objectives: Unit, Integration, or System test  
4. Test Approach:

1. First test: Play the game as expected, make sure that winning and losing conditions work properly. Checking to ensure that both the player and dealer can properly draw, stand, and bust.
2. Second test: Make sure that busts happen at the correct values, so for example no hands with 22 can win. Aces are the most important thing to check, making sure that they properly change from 11 to 1. Doing things like hardcoding hand values to simulate drawing 4 aces.
3. Third test functionality: Attempt to press other buttons on the keyboard to get an alternative effect. The only buttons that should work are the H button to hit, the S button to stand, and the R button to reset. Any other button that influences the game should be recorded as a bug.
4. Fourth test performance, stress, dependability: Mash the hit button to continuously hit, the expected result is that you will bust every time and the dealer will win. While simultaneously trying to restart the game before you are allowed to. This will make sure that the game runs smoothly through its cycle of resetting. Trying to get the program to mess up by rapidly going through the loop.
5. Usability Testing: Get a random person to try the program, if they can understand how it works by looking at it then it’ll be fine.

5. Manual or Automated Test: Because of the simple nature of the programs all the tests were done manually.   
6. Test tools: NONE  
7. Test environment: George’s MacBook – macOS, running a python program.  
8. Test criteria: This is a comprehensive test for all aspects of the Blackjack game. The simplistic nature of the game allows for a relatively fast testing phase even for a comprehensive review.

9. Test schedule Planned for April 17, 2025  
10. Test team:

Georges Cat – Ran tests, recorded outputs, and fixed bugs

George – Moral support

Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| Player Hand | User Input | Dealer Hand | Expected Output |
| Player Draw  5 & 5 = 10  Draws K  5 + 5 + K = 20 | Hit  Stand | Dealer draw  2 & 3 = 5  Draws 9  2 + 3 + 9 = 14  Draws 8  2 + 3 + 9 + 8 = 22  Dealer Busts | Player Wins!! |
| Player Draw  Q & J = 20 | Stand | Dealer Draws  9 + 6 = 15  Draws 6  9 + 6 + 6 = 21  Dealer stands | Dealer Wins!! |
| Player Draw  8 + 7 = 15  Draws J  8 + 7 + J = 25  Player Busts | Hit | Dealer Draws  5 + 3 = 8 | Dealer Wins!! |
| Player Draw  A(11) + A(1) = 12  Draws A  A(11) + A(1) + A(1) = 13  Draws A  A(11) + A(1) + A(1) + A(1) = 14  Draws K  A(1) + A(1) + A(1) + A(1) + K = 14  Draws 8  A(1) + A(1) + A(1) + A(1) + K + 8 = 22  Player Busts | Hit  Hit  Hit  Hit |  | Dealer Wins!! |
| Player Draws  K + A = 21 | Stand | Dealer Draws  7 + 9 = 16  Draws A  7 + 9 + A(1) = 17  Dealer Stands | Player Wins |
| Player Draws  A(11) + K = 21  Draws 2  A(11) + K + 2 = 23  Player Busts | Q  I  P  A  L  Z  :  “  !  Hit | Dealer Draws  A(11) + 8 = 19 | Dealer Wins!! |
| Player Draws  K + A(11) | Stand | Dealer Draws  A(11) + J = 21 | Tie Game! |
| Player Draws  Q + 9 = 19 | Stand | Dealer Draws  A(11) + 6 = 17  Dealer Stands | Player Wins!! |

Testing Reset Button

|  |  |  |  |
| --- | --- | --- | --- |
| Player Draw  Q & J = 20  Player Draw  6 + 4 = 10  Draw 8  6 + 4 + 8 = 18 | Hit  Stand  Restart  Hit  Stand | Dealer draw  2 & 3 = 5  Draws 9  2 + 3 + 9 = 14  Draws 8  2 + 3 + 9 + 8 = 22  Dealer Busts  Dealer Draws  2 + 7 = 9  Draws Q  2 + 7 + Q = 19  Dealer stands | Player Wins!!  Dealer Wins!! |
| Player Draws  4 + 9 = 13  Player Draws  Q + Q = 20 | Stand  Hit  Hit  Hit  Stand  Stand  0  P  /  Restart  Restart  Restart  Stand | Dealer Draws  K + 6 = 16  Draws 8  K + 6 + 8 = 24  Dealer Busts  Dealer Draws  3 + 9 = 12  Draws 8  3 + 9 + 8 = 20 | Player Wins!!  Tie Game!! |