**Project: Three Shell Game**

**Date:**

**Tester: Paul Leanca**

**Testing type: Black Box Testing**

1. **Objective**

To verify the functionality of the Python code for Three Shell Game without inspecting or alter the code. The focus is on input – output behaviour, game rules and user interaction.

1. **Test Environment**

Operating System : Windows

Python Version : 3.12.2

IDE : Visual Studio Code

1. **Test Scenarios**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case ID** | **Scenarios description** | **Input / Action** | **Expected Output** | **Result** | **Status** |
| TC\_01 | Game Launch | Run the script | Game window opens with background, buttons, and shells visible | As expected | Pass |
| TC\_02 | Pop-up Image Display | Start the game | Pop-up image appears for 2 seconds | As expected | Pass |
| TC\_03 | Bet Increase | Click '+£10' button | Bet increases by £10 | As expected | Pass |
| TC\_04 | Bet Decrease | Click '-£10' button | Bet decreases by £10 | As expected | Pass |
| TC\_05 | Bet Limit Handling | Decrease bet below £10 | Bet stays at minimum £10 | As expected | Pass |
| TC\_06 | Bet Limit Handling | Increase bet above available credit | Bet stays within credit limit | As expected | Pass |
| TC\_07 | Shell Selection (Win) | Select the shell with the ball | 'You Won!' message and credit increase | As expected | Pass |
| TC\_08 | Shell Selection (Lose) | Select the wrong shell | 'You Lost!' message and credit decrease | As expected | Pass |
| TC\_09 | Play Again Button | Click 'Play Again' | Game resets with new ball position | As expected | Pass |
| TC\_10 | Quit Button | Click 'Quit' | Game window closes | As expected | Pass |
| TC\_11 | Credit Depletion Handling | Continue losing until credit is 0 | Game exits automatically | As expected | Pass |
| TC\_12 | GUI Element Visibility | Inspect all GUI elements | All buttons and texts are visible and readable | As expected | Pass |

1. **Defect Tracking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Defect ID | Test case | Description of issue | Severity | Status |
| D\_01 | UI | Winning image not completely overlaps the previous image | Low | Fail  Not rectified |
| D\_02 | Bet functions | The game doesn't enforce a minimum bet properly | High | Fail  Not rectified |
|  |  |  |  |  |

1. **Conclusion**

Normal Flow Test

* 1. User places bets and guesses shells correctly and incorrectly.
  2. Verify credit updates correctly and result messages appear as expected.

Edge Case Handling

* 1. Attempt to increase the bet beyond the available credit.
  2. Attempt to decrease the bet below the minimum bet of £10.

Boundary Testing

* 1. Winning or losing with exactly £10 left.
  2. Game handling when credit reaches zero.

GUI and Usability Testing

* 1. Inspect UI visibility, button responsiveness, and visual feedback (e.g., victory animation).

Performance Testing

* 1. Check game responsiveness when rapidly switching between shells and adjusting bets.

The Three Shell Game successfully meets functional requirements with appropriate handling of user interactions and credit management. The game is stable and visually engaging with no critical bugs detected.

**Recommendations**

* **Visual Feedback:** Victory animation works smoothly.
* **Bug Handling:** No crashes or unexpected behaviour observed.
* **Potential Improvement:** Add a restart option when credit reaches zero instead of exiting the game.