|  |
| --- |
|  |
| Testers:  Greaseball  Testees:  ………………… |
| **Test Case** | **Description** | **Expected Outcome** | **Mark** |
| Dice Rolling | Test rolling within range | Roll the dice multiple times and ensure the sum falls within 2 to 12. |  |
|  | Verify doubles identification | Roll doubles repeatedly and confirm they are correctly identified. |  |
|  | Confirm correct movement | Roll various combinations and validate that players move the appropriate number of spaces. |  |
|  | Test for randomness | Perform statistical analysis to ensure dice rolls exhibit fair randomness. |  |
|  | Edge cases testing | Specifically test for snake eyes, boxcars, and other edge cases. |  |
| Board Functionality | Test board display | Display the board and verify it matches the expected layout. |  |
|  | Check player location update | Move players to different spaces and ensure their positions are accurately updated. |  |
|  | Wraparound behaviour | Move players past the end of the board and confirm they wrap around correctly. |  |
|  | Dynamic board size | Test the game with boards of varying sizes to ensure compatibility. |  |
| Player Actions | Piece selection test | Allow players to choose pieces and confirm their choices are correctly stored. |  |
|  | Name storage and announcement | Ensure player names are stored and announced accurately during turns. |  |
|  | Money transactions | Add and deduct money from players and validate balance changes. |  |
|  | Win/loss identification | Simulate different scenarios to verify correct identification of winners and losers. |  |
|  | Interaction with game elements | Test player interactions with animals, cards, and other game components. |  |
| Animal Interaction | Purchase functionality | Allow players to buy animals and confirm ownership transfer. |  |
|  | Landing charges | Land on spaces with owned animals and validate correct charges. |  |
|  | Upgrade functionality | Upgrade animals and verify level increase at the correct cost. |  |
|  | Visit charges | Confirm players are charged the correct amount to visit animals based on their level. |  |
|  | Maximum level handling | Ensure the game handles animals reaching the maximum level gracefully. |  |
| Card Drawing | Random selection | Draw cards from the deck and ensure each card has an equal chance of being drawn. |  |
|  | Scenario and adjustment accuracy | Check that drawn cards present the correct scenario and monetary adjustment. |  |
|  | Money adjustment validation | Validate that players gain or lose the correct amount of money based on drawn cards. |  |
|  | Diversity in deck | Ensure the card deck contains a diverse range of scenarios and monetary adjustments. |  |
| Cross-Functionality | Integration of components | Verify smooth interaction between all game components without errors. |  |
|  | Handling of player actions | Test player prompts and action handling based on the current game state. |  |
|  | Error handling | Ensure appropriate error messages are displayed for invalid actions or scenarios. |  |
| Scalability and Maintainability | OOP adherence | Evaluate codebase for adherence to object-oriented principles. |  |
|  | Readability assessment | Check code readability and clarity for maintainability. |  |
|  | Flexibility of codebase | Introduce changes and verify ease of implementation without significant rework. |  |
| User Experience | Feedback gathering | Collect player feedback to assess user experience. |  |
|  | Improvement implementation | Implement necessary improvements based on collected feedback. |  |
| Regression Testing | No new issues introduction | Verify previously tested features continue to function correctly after modifications. |  |
|  | Post-modification functionality | Ensure all features work as expected after changes or bug fixes. |  |