**Appendix I**

**Test Plan**

**Acceptance Plan**

*This appendix also shows that all components of the system are tested, both via Acceptance and JUnit testing.*

| **ID** | **Use Case Name** | **Description of Test** | **Test Initialization** | **Test Inputs** | **Test Procedure** | **Expected Results** | **Passed?** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 01 | Use Case 1 | Testing Launch of Game/Main Menu is displayed | Launch the ArtemisLite Game | Game.java | Launch ‘Game.java’ in Eclipse | Welcome message and Main Menu are displayed | Y |
| 02 | Use Case 1 | Testing ‘Start New Game’ | Launch ‘Start New Game’ from Main Menu | Player input at Main Menu (int) | Input ‘2’ from Main Menu | Introductory dialogue is displayed. Player is asked to input number of Players | Y |
| 03 | Use Case 1 | Testing input for ‘ Enter Number of Players’ (Valid) | Enter number of Players, within the range 2-4 | Number of Players: 2 | Input number of Players (2) | Number of Players is accepted | Y |
| 04 | Use Case 1 | Testing input for “Enter Player Names’ | Enter names of Players when requested | Player input (String) | Enter names: ‘PlayerOne’, ‘PlayerTwo’ | Names entered are accepted and stored | Y |
| 05 | Use Case 1 (Alternative Flow) | Testing input for number of Players  (Invalid) | Enter number of Players, outside the range 2-4 | Number of Players: 5 | Input number of Players (5) | Error message appears, prompting the user to re-enter number between 2-4 | Y |
| 06 | Use Case 1 (Alternative Flow) | Testing input for “Enter Player Names’ (Invalid) | Enter names of Players when requested | Player input (String) | Enter names: ‘PlayerOne’, ‘PlayerOne | Error message appears, prompting the user to re-enter a new name | Y |
| 07 | Use Case 1 (Alternative Flow) | Testing ‘Display Rules’ | Launch ‘Display Rules’ from Main Menu | Player input at Main Menu (int) | Input (1) from Main Menu | Rules are displayed | Y |
| 08 | Use Case 1 (Alternative Flow) | Testing return from ‘Display Rules’ | Return from ‘Display Rules’ | Player input at Main Menu (Return Key) | Input (Enter) from ‘Display Rules’ | Returns to Main Menu | Y |
| 09 | Use Case 1 (Alternative Flow) | Testing ‘Restore Saved Game’ | ‘Restore Saved Game’ is selected from Main Menu | Player input (int) | Player input (3) at Main Menu | ‘Restoring game..‘ message displayed and saved game details are displayed. Game is restarted from save point | Y |
| 10 | Use Case 1 (Alternative Flow) | Testing ‘Enable Speech’ | ‘Enable Speech’ is selected from the Main Menu | Player input (int) | Player input (4) at Main Menu | Player is asked to enter ‘Y’ or ‘N' to enable speech. | Y |
| 11 | Use Case 1 (Alternative Flow) | Testing ‘Enable Speech’ | ‘Enable Speech’ is selected from the Main Menu | Player input (boolean) | Player input (Y) at ‘Enable Speech’ | The menu is displayed again and spoken, with option 4 being replaced with the option to disable speech | Y |
| 12 | Use Case 1 (Alternative Flow) | Testing ‘Enable Speech’ | ‘Enable Speech’ is selected from the Main Menu | Player input (boolean) | Player input (N) at ‘Enable Speech’ | Player is returned to Main Menu | Y |
| 13 | Use Case 2 | Testing ‘Dice’ function | Dice are rolled automatically as part of a Players’ move | Player | Players’ move is launched automatically / number of dice set automatically | Result of Player's roll (between 1-12) is displayed | Y |
| 14 | Use Case 2 | Testing ‘Move’ function | Player’s move is launched automatically moved a specified number of squares on the virtual game board. | Player | Player’s move is launched automatically on their turn. | A dialogue is displayed telling the current player what square they landed on and all relevant information about that square. | Y |
| 15 | Use Case 2 | Testing ‘End Turn’ | On move, Player selects from Option Menu | Player input number 1-6 (valid) | Player inputs (4), from the move Option Menu | Player’s move ends and Move is repeated for the next Player. | Y |
| 16 | Use Case 2 (Alternative Flow) | Testing ‘Display All Systems’ | On move, Player selects from Option the Menu | Player input (int 1-6 (valid)) | Player inputs (1), from the Option Menu | All systems are displayed with the system name and the elements they contain. | Y |
| 17 | Use Case 2 (Alternative Flow) | Testing ‘Display your Resources & Properties Owned’ | On move, Player selects from Option the Menu | Player input (int 1-6 (valid)) | Player inputs (2), from the Option Menu | Resources & Properties Owned are displayed. Player is return to Option Menu. | Y |
| 18 | Use Case 2 (Alternative Flow) | Testing ‘Develop an Element’ (not yet owned) | On move, Player selects from Option Menu | Player input (int 1-6 (valid)) | Player inputs ‘3’, from the move Option Menu | ‘You need to own all elements in a system before you can develop!” message is displayed | Y |
| 19 | Use Case 2 (Alternative Flow) | Testing ‘Save Game’ | On move, Player selects from Option Menu | Player input number 1-6 (valid) | Player inputs ‘6’, from the move Option Menu | Game is saved, Player is told game has been saved and returned to Option Menu. | Y |
| 20 | Use Case 2 (Alternative Flow) | Testing ‘Display Game Map’ | On move, Player selects from Option Menu | Player input number 1-6 (valid) | Player inputs ‘3’, from the move Option Menu | Game Map is displayed | Y |
| 21 | Use Case 2 (Alternative Flow) | Testing ‘Quit Game’ | On move, Player selects from Option Menu | Player input number 1-6 (valid) | Player inputs ‘6’, from the move Option Menu | Game ends, final state of play is displayed. | Y |
| 22 | Use Case 2 (Alternative Flow) | Testing invalid input from Option Menu | On move, Player selects from Option Menu | Player input number 1-6 (Invalid) | Player inputs ‘9’, from the move Option Menu | Message is displayed ‘Problem with input’, and Player is returned to Menu | Y |
| 23 | Use Case 3 | Testing ‘Pass Go’ function | Player has been moved along entirety of board, which causes them to ‘Pass Go’ | Player’s turn | Players’ move has caused them to ‘pass go’. | Player is informed that they have ‘passed go’ . Resources are updated. | Y |
| 24 | Use Case 4 | Testing ‘Purchase Element’ function (Player decides to purchase) | Player is offered if they would like to purchase - chooses to purchase | Player input (boolean) | Player inputs ‘Y’ when asked if they wish to purchase | The appropriate resources are removed from the player,. Player’s name is assigned to appropriate Element | Y |
| 25 | Use Case 4 (Alternative Flow) | Testing ‘Purchase Element’ function (Player decides not to purchase) | Player is offered if they would like to purchase - chooses not to purchase | Player input (boolean) | Player inputs ‘N’ when asked if they wish to purchase | ‘Sorry you don’t want to purchase’ message is displayed, The square is not purchased by Player and their resources are not changed. Player is asked if they want to offer the square to another player. | Y |
| 26 | Use Case 4 (Alternative Flow) | Testing offering ‘Purchase Element’ to another Player (Player decides to offer) | Player has landed on the square, and has chosen not to buy. Offers to Player2 | Player input (boolean / int) | Player inputs ‘Y’ when asked if they want to offer to another Player. Player inputs number from list of available Players (1) | Sub-Menu of available Player’s to offer Element to is displayed. | Y |
| 27 | Use Case 4 (Alternative Flow) | Testing offering ‘Purchase Element’ to another Player (Player decides not to offer) | Player has landed on the square, and has chosen not to buy. Does not offer to another Player | Player input (boolean) | Player inputs ‘N’ when asked if they want to offer to another Player | Player is returned to Main Option Menu | Y |
| 28 | Use Case 4 (Alternative Flow) | Testing Player accepts Purchase Offer | Player offers the Element to Player2 | Player input (boolean) | Player being offered enters ‘Y’ | Player under offers resources are updated and shown to screen | Y |
| 29 | Use Case 4 (Alternative Flow) | Testing Player accepts Purchase Offer (Cannot afford purchase) | Player offers the Element to Player2 (insufficient resources) | Player2r input (boolean) | Player being offered enters ‘Y’ | The player does not have sufficient resources to purchase. Message is displayed, and no resources change | Y |
| 30 | Use Case 4 (Alternative Flow) | Testing Player2 declines Purchase Offer | Player offers the Element to Player2 | Player2 input (boolean) | Player2 enters ‘N’ | Player is asked if they would like to make offer to another Player | Y |
| 31 | Use Case 4 (Alternative Flow) | Testing offering Element purchase to Player2 (Invalid) | Player has input ‘Y’ when asked if they want to offer to another Player | Player2 input (int) | Player inputs number from list of available Players (3) | Massage presented “Invalid Option selected, try again…” and returned to sub-Menu | Y |
| 32 | Use Case 5 | Testing Develop Element | Sub-Menu of Elements available for development is displayed - Player selects one | Player input (int 1-6 (valid)) | Player inputs (3), from the sub-Menu | The appropriate resources are removed from Player. A dialogue detailing the development taking place is displayed. The development menu is displayed | Y |
| 33 | Use Case 5 | Testing Develop Element (exiting) | Sub-Menu of Elements available for development is displayed - Player selects one | Player input (int 1-6 (valid)) | Player inputs (6), from the sub-Menu | ‘Exiting development Menu’ is displayed. Players resources and ‘Elements’ are updated. Player is returned to Option Menu | Y |
| 34 | Use Case 5 (Alternative Flow) | Testing Develop Element (fully developed) | Sub-Menu of Elements available for development is displayed - Player selects one | Player input (int 1-6 (valid)) | Player inputs (3), from the sub-Menu | ‘You've developed it!‘ message and description displayed, Element is removed from the menu of developable elements. | Y |
| 35 | Use Case 5 (Alternative Flow) | Testing Develop Element (no developments to make) | Player selects from Option Menu | Player input (int 1-6 (valid)) | Player inputs (3), from the move Option Menu | ‘You don't have any developments to make!’ message is displayed | Y |
| 36 | Use Case 5 (Alternative Flow) | Testing Develop Element (insufficient resources) | Player selects from Option Menu | Player input (int 1-6 (valid)) | Player inputs (3), from the move Option Menu | ‘You don't have enough resources to develop’ message is displayed. No development are made, and no resources change. | Y |
| 37 | Use Case 5  (Alternative Flow) | Testing Develop Element (invalid) | Sub-Menu of Elements available for development is displayed - Player selects one | Player input number( int !1-6 (invalid)) | Player inputs (10), from the sub-Menu | “Invalid input try again” is displayed, and Player is returned to Menu | Y |
| 38 | Use Case 6 | Testing ‘Pay Rent’’ (charges rent) | Player lands on square which is already owned by Player2 | Player2 input (boolean) | Player2 answers (Y) to the question ‘Do you want to charge (Player) rent?’ | ‘Rent’ is deducted from Player’s resources,The equivalent resources are added to Player2. Updated resources are displayed. | Y |
| 39 | Use Case 6 (Alternative Flow) | Testing ‘Pay Rent’ (declines to charge rent) | Player lands on square which is already owned by Player2 | Player2 input (boolean) | Player2 answers (N) to the question ‘Do you want to charge (Player) rent?’ | No rent is deducted from Player’s resources, updated resources are displayed. | Y |
| 40 | Use Case 6 (Alternative Flow) | Testing ‘Pay Rent’’ (charges rent causing bankruptcy) | Player lands on square which is already owned by Player2 | Player2 input (boolean) | Player2 answers (Y) to the question ‘Do you want to charge (Player) rent?’ | ‘Rent’ is deducted from Player’s resources,The equivalent resources are added to Player2. Player now has negative resources and ‘Game Over’ is triggered | 30 |
| 41 | Use Case 6 (Alternative Flow) | Testing ‘Pay Rent’’(Invalid) | Player lands on square which is already owned by Player2 | Player2 input (boolean) | Player 2 answers ‘X’ to the question ‘Do you want to charge (Player) rent?’ | Message displayed ‘Not valid input. Please enter Y / N’ | Y |
| 42 | Use Case 7 | Testing ’Trade an Element’ | Player selects from Option Menu | Player input number 1-6 (valid) | Player inputs (4), from the move Option Menu | A trade menu is displayed with all elements available for the current player to trade with a corresponding number | Y |
| 43 | Use Case 7 | Testing ’Trade an Element’ ( selecting an Element to trade) | Player enters the number for the element to trade. | Player input number 1-6 (valid) | Player inputs (1) | Element to trade is selected. Menu showing the other players with a corresponding number is shown for the current player to choose who to trade with | Y |
| 44 | Use Case 7 | Testing ’Trade an Element’ (offering a trade) | Player selects from list of players to trade with | Player input number 1-6 (valid) | Player inputs (1) | ‘Selected Player (Player2) is asked ‘would you like to buy ‘Element’?’ | Y |
| 45 | Use Case 7 | Testing ’Trade an Element’ (Player2 accepts trade) | Player2 is decides to accept trade | Player2 input (boolean) | Player2 inputs (Y) when asked if they would like to make trade | Player is told they have sold Element to Player2. The appropriate resources are removed from the Player2 and added to Player. Player is returned to Option Menu | Y |
| 46 | Use Case 7 (Alternative Flow) | Testing ’Trade an Element’ ( Player2 declines trade) | Player2 is decides to decline to trade | Player2 input (boolean) | Player2 inputs ‘N’ when asked if they would like to make trade | Player is told ‘(Player2) doesn't want to trade, hard luck!’. Player is returned to Option Menu. Resources are not changed | Y |
| 47 | Use Case 7 (Alternative Flow) | Testing ’Trade an Element’ (no elements to trade) | Player selects from Option Menu | Player input number 1-6 (valid) | Player inputs (3), from the move Option Menu | ‘You have no elements to trade’ message is displayed. Player is returned to Option Menu | Y |
| 48 | Use Case 7 (Alternative Flow) | Testing ’Trade an Element’ ( selecting an Element to trade(invalid)) | Player selects from available Elements to Trade sub-Menu | Player input number( int !1-6 (invalid)) | Player inputs (10), from the sub-Menu | ‘Incorrect selection’ message is displayed. Player is returned to sub-Menu | Y |
| 49 | Use Case 7 (Alternative Flow) | Testing ’Trade an Element’ ( offering a trade(invalid)) | Player selects from list of players to trade with | Player input number (int !1-6 (invalid)) | Player inputs (10), from the sub-Menu | ‘Incorrect selection’ message is displayed. Player is returned to sub-Menu | Y |
| 50 | Use Case 7 (Alternative Flow) | Testing ’Trade an Element’ ( Player2 enters invalid input) | Player2 enters invalid input | Player2 input (invalid) | Player2 inputs (X) when asked if they would like to make trade | ‘Incorrect selection’ message is displayed. Player2 is asked if they would like to make trade | Y |
| 51 | Use Case 8 | Testing ‘Random Event’ function | Random Events are triggered at the start of a Player's turn. | Will appear at random | Will appear at random | ‘\*\*\*EVENT\*\*’ will appear on screen. Appropriate randomly generated resources are added/removed from Player’s balance | Y |
| 52 | Use Case 9 | Testing end of play ‘Launch Sequence’ | Player develops all Element Systems | Player fully develops final Element System | Player inputs ‘1’, from the sub-Menu | Final epilogue and ‘Launch Sequence’ is displayed. Final detaIls are displayed. A dialogue is displayed detailing the game state is displayed | Y |
| 53 | Use Case 9 | Testing ‘Game Over’ | Player has a negative amount of resources. | Player has a negative amount of resources. | Player has a negative amount of resources. | Game Over message is displayed. A dialogue is displayed detailing the game state is displayed | Y |

**JUnit Test Cases**

*Screengrabs of Junit tests*









