|  |  |  |
| --- | --- | --- |
| **Flow of Events for the *Initialise Game* use-case** | | |
| **Objectives** | **To allow players to set up and begin a game.** |
| **Actor(s)** | Player. |
| **Precondition** | The programme has been launched. |
| **Main Flow** | 1. The menu is displayed in the console when the game is launched. 2. The player enters the ‘key’ for start game. 3. The player is prompted to enter the number of players. 4. The player enters a whole number 2-4. 5. The player is prompted to enter a player’s name. 6. The player enters a name. 7. Steps 5 and 6 are repeated for the number of players enter in step 4. |
| **Alternative Flow** | **At 2.**  The player enters the ‘key’ for How to Play.  The information is displayed.  The player is asked “Do you want to return to the menu? Yes/No”.  The player types ‘Yes’ and the menu is displayed again.  The player enters ‘No’ the information remains displayed with the question repeated until the player enters ‘Yes’ or a valid ‘key’ into the console.  **At 2.**  If anything, other than ‘YES’ or ‘NO’ (not case sensitive) is typed into the console an appropriate error message is displayed and the question is repeated until an accepted character is entered.  **At 4.**  If anything, other than ‘2, 3, or 4’ is typed into the console an appropriate error message is displayed and the question is repeated until an accepted character is entered.  **At 6.**  The player enters a name outside the accepted parameters.  An appropriate error message is displayed, and the player is prompted to enter their name again.  **At 6.**  The player enters name already entered by a previous player.  An appropriate error message is displayed, and the player is prompted to enter their name again. |
| **Post-condition** | The game is started. |

|  |  |  |
| --- | --- | --- |
| **Flow of Events for the *Display Player Information* use-case** | | |
| **Objectives** | **To display the current players level or resources and elements.** |
| **Actor(s)** | Player |
| **Precondition** | It is the players turn. |
| **Main Flow** | 1. A dialogue is displayed showing the resources and elements the current player owns. |
| **Alternative Flow** | None. |
| **Post-condition** | None. |

|  |  |  |
| --- | --- | --- |
| **Flow of Events for the *Move* use-case** | | |
| **Objectives** | **To move the player from their current position on the game board by the number randomly generated.** |
| **Actor(s)** | Player |
| **Precondition** | It is the players turn. |
| **Main Flow** | 1. Virtual dice are rolled, and a number randomly generated. 2. The player is told the total value. (For example, “[Player name] has rolled 10.” 3. Using the randomly generated value the current player is moved a specified number of squares on the virtual game board. 4. The players position on the board is temporarily saved. 5. A dialogue is displayed telling the current player what square they landed on and all relevant information about that square. |
| **Alternative Flow** | **After 5.**  Depending on what square the player passes or lands on the Pass Go, Purchase Element, Pay Rent, Develop Element or Trade uses cases may be performed. |
| **Post-condition** | Player’s move ends and Move is repeated for the next Player. |

|  |  |  |
| --- | --- | --- |
| **Flow of Events for the *Pass Go* use-case** | | |
| **Objectives** | **The current player has passed ‘GO’ and receives additional resources.** |
| **Actor(s)** | Player |
| **Precondition** | It’s the players turn.  The player has rolled the dice.  The player has passed or landed on the ‘GO’ element. |
| **Main Flow** | 1. A dialogue is displayed telling the current player they have passed go and received resources. 2. The appropriate resources are added to the current player. |
| **Alternative Flow** | None. |
| **Post-condition** | The current players resources are updated. |

|  |  |  |
| --- | --- | --- |
| **Flow of Events for the *Purchase Element* use-case** | | |
| **Objectives** | **To purchase an Element.** |
| **Actor(s)** | Player |
| **Precondition** | It’s the players turn.  The player lands on an element not owned by another player. |
| **Main Flow** | 1. The current player is asked “[Player name] Do you want to purchase ‘Element X’? Yes/No”. 2. The player types Y into the console. 3. The appropriate resources are removed from the player. 4. The players’ name is assigned to ‘Element X’. |
| **Alternative Flow** | **At 2.**  The current player types ‘No’ into the console, choosing not to purchase ‘Element X’.  The square is not purchased by the current player and their resources are not changed.  The square is then offered to each player in turn following the main flow.  If every player types ‘No’, no one is assigned to ‘Element X’ and no resources change.  **At 2**.  If anything, other than ‘YES’ or ‘NO’ (not case sensitive) is typed into the console an appropriate error message is displayed and the question is repeated until an accepted character is entered.  **At 3.**  The player does not have sufficient resources to purchase ‘Element X’. An appropriate message is displayed, no one is assigned to ‘Element X’ and no resources change. |
| **Post-condition** | The players resources are updated and ‘Elements’ are updated. |

|  |  |  |
| --- | --- | --- |
| **Flow of Events for the *Develop Element* use-case** | | |
| **Objectives** | **To develop an Element within a System the current player owns.** |
| **Actor(s)** | Player |
| **Precondition** | It’s the players turn.  The player owns all elements within a System.  The element is not fully developed. |
| **Main Flow** | 1. The current player is asked “[Player name] Do you want to develop ‘Element X’? Enter Yes/No”. 2. The current player types ‘Yes’ into the console. 3. The appropriate resources are removed from the current player. 4. The development level of ‘Element X’ is increased along with the cost of rent. 5. A dialogue detailing the development taking place is displayed. |
| **Alternative Flow** | **At 2.**  The current player types ‘No’ into the console, choosing not to develop ‘Element X’.  The current player resources are not changed.  The development level of the current players’ elements does not change.  **At 2**.  If anything, other than ‘YES’ or ‘NO’ (not case sensitive) is typed into the console an appropriate error message is displayed and the question is repeated until an accepted character is entered.  **At 3.**  The player does not have sufficient resources to develop ‘Element X’. An appropriate message is displayed, ‘Element X’ is not developed, and no resources change. |
| **Post-condition** | The current players resources and ‘Elements’ are updated. |

|  |  |  |
| --- | --- | --- |
| **Flow of Events for the *Pay Rent* use-case** | | |
| **Objectives** | **To pay rent on an Element owned by another player.** |
| **Actor(s)** | Player |
| **Precondition** | It’s the players turn.  The player lands on an Element owned by another player. |
| **Main Flow** | 1. The player who owns ‘Element X’ is asked “[Player name] Do you want to charge rent for ‘Element X’? Enter Yes/No”. 2. The player types ‘Yes’ into the console. 3. The appropriate resources are removed from the current player. 4. The equivalent resources are added to the player who owns ‘Element X’. |
| **Alternative Flow** | **At 2.**  The player who owns ‘Element X’ types ‘No’ into the console, choosing not to charge rent.  The current player resources are not changed.  **At 2**.  If anything, other than ‘YES’ or ‘NO’ (not case sensitive) is typed into the console an appropriate error message is displayed and the question is repeated until an accepted character is entered.  **At 3.**  The current player does not have sufficient resources to pay rent for ‘Element X’. The current player has negative resources and Game Over use case is performed. |
| **Post-condition** | The current players resources are updated. The player who owns ‘Element X’ resources are updated. |

|  |  |  |
| --- | --- | --- |
| **Flow of Events for the *Trade Element* use-case** | | |
| **Objectives** | **To trade an Element the current player owns to another player.** |
| **Actor(s)** | Player |
| **Precondition** | It’s the players turn.  The player owns the element. |
| **Main Flow** | 1. The current player is asked “[Player name] Do you want to trade an element? Yes/No”. 2. The current player types ‘Yes’ into the console. 3. The elements and the value of them are displayed with an appropriate key assigned to each one. 4. The current player is asked “Which element do you want to trade?” 5. The player types the appropriate key into the console. 6. The current player is asked “Which player do you want to trade with?” 7. The current player types the player’s name into the console. 8. The selected player is asked “[Player name] Do you want to purchase ‘Element X’ for ‘X’ amount of resources? Yes/No”. 9. The selected player types ‘Yes’ into the console. 10. The appropriate resources are removed from the selected player and added to the current player. 11. The current players’ name is removed from ‘Element X’ and the selected players’ name is assigned. |
| **Alternative Flow** | **At 2.**  The current player types ‘No’ into the console, choosing not to trade ‘Element X’.  The current players’ resources and elements are not changed.  **At 2**.  If anything, other than ‘YES’ or ‘NO’ (not case sensitive) is typed into the console an appropriate error message is displayed and the question is repeated until an accepted character is entered.  **At 5**.  If anything, other than a valid key is typed into the console an appropriate error message is displayed and the question is repeated until a valid key is entered.  **At 7**.  If anything, other than a valid player name is typed into the console an appropriate error message is displayed and the question is repeated until a valid key is entered.  **At 9.**  The selected player types ‘No’ into the console, choosing not to purchase ‘Element X’.  The current players’ resources and elements are not changed.  The selected players’ resources and elements are not changed.  **At 10.**  The selected player does not have sufficient resources to purchase ‘Element X’. An appropriate message is displayed. |
| **Post-condition** | The current players resources and ‘Elements’ are updated.  The selected players resources and ‘Elements’ are updated. |

|  |  |  |
| --- | --- | --- |
| **Flow of Events for the *Random Event* use-case** | | |
| **Objectives** | **To generate a random event (positive or negative), which adds or removes resources appropriately for a randomly chosen player** |
| **Actor(s)** | Player |
| **Precondition** | All players have completed a turn. |
| **Main Flow** | 1. A dialogue is displayed to indicate an event will be generated 2. A dialogue is displayed detailing a positive event 3. A dialogue is displayed congratulating a randomly selected player 4. Appropriate randomly generated resources are added to the selected player’s balance 5. A dialogue is displayed indicating the end of the event |
| **Alternative Flow** | **At 2.**  A dialogue is displayed detailing a negative event.  A dialogue is displayed indicating a randomly selected player will be fined  Appropriate randomly generated resources are removed from the selected player’s balance  Main flow continues at step 5. |
| **Post-condition** | The selected player’s resources are updated |

|  |  |  |
| --- | --- | --- |
| **Flow of Events for the *Game Over* use-case** | | |
| **Objectives** | **To conclude the game.** |
| **Actor(s)** | Player |
| **Precondition** | None. |
| **Main Flow** | 1. All systems have been fully developed. 2. A dialogue is displayed congratulating players on a successful launch of Artemis. 3. An epilogue is displayed detailing the launch. 4. A dialogue is displayed detailing the game state. For example, the elements along with their development levels and owners, and the amount of resources each player had. |
| **Alternative Flow** | **At 1.**  The player enters the ‘key’ for quit game.  The player is asked “[Player name] Are you sure you want to quit? This will end it for all players. Enter Yes/No”.  The player types ‘Yes’.  The player quits the game.  Main flow continues at step 4.  The player enters ‘N’ into the console, choosing not to quit the game. The game continues.  **At 1.**  The current player has a negative amount of resources.  Main flow continues at step 4. |
| **Post-condition** | The game is over. |