**Initialise Game Sequence Diagram**

Initialise Game is the first use case, starting once the game is launched. A welcome menu followed by a start up menu is displayed. The user is prompted to select one of the menu options.

* Option 1 will display the rules of the game and then automatically display the menu again and prompt the user for input.
* Option 2 is the main flow of the use case and when the user chooses it, they will be prompted to enter the number of players.

The user must enter a valid number between the MIN\_USERS and MAX\_USERS, for this game it is 2 to 4 players. Any number outside this range or a different character will display an appropriate message and prompt the user to input a number again.

When a valid number of players is accepted the players will be prompted to input a username. If the username is only whitespace, empty or another user has already used the name, an appropriate message is displayed, and the player prompted to enter a name again.

Once usernames have been entered for all players the usernames are used to create players and the players are passed into Game.

* Option 3 will restore a saved game.

Instead of asking for usernames and empty list of usernames is passed to Game and the saved players and board will resume from the last save point.

* Option 4 will enable speech.

After selecting option 4 the user will be prompted by both text and speech to enter Y or N if they want to enable speech. Entering Y will set speak to true and all text will be spoken as well as displayed to screen for the duration of the game. The menu options will be updated to display disable speech and if entered again will disable to speech functionality.

**Move Sequence Diagram**

Following Initialise Game, Move will repeat for each player until Game Over occurs. Two dice are rolled using diceRoller (Roller class) which returns the total of dice rolled. This number is used to move a player on the board. The move method calls the onPass() and onLand() methods which lead into the PassGo, Purchase Element or Pay Rent use cases depending on which square the player passes or lands on and the ownership of the square. Once the player has completed their purchase or paying rent if needed, a bankrupt check is completed and if the player has negative resources and appropriate message is displayed and Game Over occurs. A menu is then displayed, and the player is prompted to select an option.

* Option one displays all the elements grouped within their systems and their system name. The menu will then automatically be displayed again, and the player prompted for input.
* Option two displays the players resources and any elements they own. The menu will then automatically be displayed again, and the player prompted for input.
* Option three will display a development menu and leads into the Develop Element use case. The player will be returned to the menu and prompted for input.
* Option four will display a trade menu and leads into the Trade Element use case. The player will be returned to the menu and prompted for input.
* Option five is the main flow and ends a players turn. This will complete the Move use case for one player and switch to the next player, repeating until Game Over occurs.
* Option six allows the player to save the game. The players and board will be saved to allow the game to resume at a certain state of play if the player quits the game.
* Option seven allows the player to quit the game. This is one way the Game Over use case occurs.
* Option eight displays the board layout. The menu will then automatically be displayed again, and the player prompted for input.

**Purchase Element and Pay Rent Sequence Diagram**

Following Move, Purchase Element or Pay Rent occurs for each player once per move and is dependent on which square, they land on and its ownership.

The main flow for Purchase Element is when the square the player lands on is an Element and is unowned. The attemptPurchaseElement() method is called.

* The current player is offered the unowned element to purchase. If they enter ‘Y’ choosing to purchase the element and they have sufficient resources, they are set as the owner of the element, the element is added to their set of elements owned and the cost of the element is removed from the players resources. The Move menu is then displayed.
* If they enter ‘Y’ choosing to purchase the element and they do not have sufficient resources, an appropriate message is displayed, and the Move menu is displayed.
* If they enter ‘N’ choosing not to purchase the element they are then asked if they want to offer the element to another player to purchase.
  + If they enter ‘Y’ choosing to offer to another player, the other players are mapped to a number and displayed to the current player prompting them to enter a number corresponding to the player they wish to offer the element to. They are also given the option to cancel by selecting the number after the last player. If they cancel the Move menu is, then displayed. If they select a player the attemptPurchaseElement() method is called for the selected player and they are offered the unowned element to purchase following the main flow. Once the selected player makes their choice the Move menu is displayed for the current player.
  + If they enter ‘N’ choosing not to offer to another player, an appropriate message is displayed, and the Move menu is displayed.

If the player lands on an element they already own an appropriate message is displayed and the Move menu is, then displayed.

If the player lands on an element owned by another player, the Pay Rent use case occurs. The main flow for Pay Rent is when the square the player lands on is an Element and is owned by another player. The requestRent() method is called.

* The owner of the element is asked if they wish to charge the current player rent. If they enter ‘Y’ choosing to enforce the rent payment, the cost of rent is removed from the current players resources and added to the owners’ resources. The Move menu is then displayed.
  + If the current player does not have sufficient resources an appropriate message is displayed, and their resources will be negative. Within Move a bankrupt check is completed and the negative resources will cause Game Over.
* If they enter ‘N’ choosing not to enforce rent, an appropriate message is displayed, and no resources change. The Move menu is then displayed.