**Use Case Brief Textual Descriptions**

**Brief Descriptions**

Use Case Name:

**Start Game**

Actors:

User

Use Case Description:

After the user has launched the game, they are greeted by the main menu. The main menu itself has buttons representing the loading of the game, starting a new game, editing game options, or quitting the game entirely.  
Upon clicking the ‘Start New Game’ button, the base grid for both the Earth are and Mars area is loaded. In the centre of the each grid is the respective faction’s Command Centre, and randomly dotted around the map are resources which can be gathered. The User Interface is loaded at this point, as well as the initialising of the AI.

Use Case Name:

**Place Building**

Actors:

User

Use Case Description:

The game is in play, and the user has selected a building from the User Interface which they want to construct. The user will click on a tile that they want the building to be placed. Upon doing this, various methods are called to ensure that the location the user wants to place this building is valid. If it is valid, the building will be placed. Otherwise, a message will be shown on the bottom left of the screen in red font.

Use Case Name:

**Quit To Main Menu**

Actors:

User

Use Case Description:

Upon the game being paused, the user is faced with various options; ‘Resume’, ‘Save’ or ‘Quit’. When ‘Quit’ is selected, the user is taken back to the main menu. Everything in the main menu state is created and displayed, whilst those in the playing state are unloaded and destroyed.

**In-Depth Textual Description**

Use Case Name:

**Place Building**

Actors:

User

Triggers:

User clicks whilst holding a building to place.

Preconditions:   
User is holding an object of type structure, and cursor is within the game grid.

Post-conditions:

The surrounding area is checked to ensure it is a valid location.

Player’s mineral amount is checked to ensure enough is owned to pay for building.

If the location is valid and enough minerals are owned, then the building is placed.

Flags are raised in the grid locations the building encompasses.

Cost of building deducted from player.

Successful Flow:  
1. The user hovers mouse over grid square in which they want the building to be placed.

2. The game compares the cost of the building to how many minerals the player currently has.

3. The game accepts that the player has sufficient funds.

4. The game checks the surrounding area which the building requires in order for the structure to be built.

5. The game passes the check – surrounding area is clear of other structures and is not overlapping the edge of the game grid.

6. The game creates the building at the chosen location.

7. The game deducts the cost of the building from the player’s funds.

8. The user continues to play.

Unsuccessful Flow:

2A1: The player does not have enough minerals to pay for the cost of constructing the building. An error message is returned at the bottom left of the screen letting the player be aware of the issue.

4A1: The area surrounding the building trying to be placed is being used by other structures, making it an invalid placement. An error message is returned to the player, displayed at the bottom left hand side of the screen.

4A2: The building overlaps the edge of the map, thus an invalid building placement. Again, an error would be displayed at the bottom left of the screen highlighting the issue.