**Use Case Brief Textual Descriptions**

**Brief Descriptions**

Use Case Name:

**Build Units**

Actors:

User

Use Case Description:

Whilst the game in is play, the user is able to utilise the buildings they have created in order to construct units which will be used to launch attacks on the AI player.

When the play clicks on a constructed building, they will be shown a list of units that can be built from that particular building. From this, if the building is free for production, they will be able to either click the image of the unit or press a hotkey designated to that unit to build it. When a unit is selected, the building will be in production for the duration of the unit’s production time.

Use Case Name:

**Defend**

Actors:

User

Use Case Description:

Whilst the game is in play, the user will be at constant battle from the AI player. If the player loses a space battle, then the AI player will land units on earth and will try to destroy the player’s command centre in order to win the game. The player will have to build and command units in order to try and defend his/her base from destruction.

Use Case Name:

**Move Units**

Actors:

User

Use Case Description:

Throughout the game, the player will have the ability to manoeuvre his/her units around both earth and if there are able to launch a successful invasion, Mars. The player will have different unit types to choose from, and as such will have multiple movement types.

Throughout all the types of units though is the player’s ability to move them to where they are needed. The player will be able to click on units to select them, and right-clicking on a position on the map with a unit selected will instruct the unit to move to that position. If the position is being used by a building or another unit, the moving unit will attempt to get as close as possible to that position.

**In-Depth Textual Description**

Use Case Name:

**Build Units**

Actors:

User

Triggers:

User either clicks on unit image or presses hotkey with building selected to build unit.

Preconditions:   
User has selected a building which can produce a unit

Post-conditions:

The building’s current production is checked to ensure that it is not already in production

Players mineral amount is checked to ensure that the unit can be bought.

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

Cost of unit deducted from player.

Successful Flow:  
1. The user selects the building they want to produce from

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

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

4. The game checks the building is not already in production of a unit

5. The game passes the check – building is currently not producing a unit

6. The building begins production of the unit.

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 producing the unit. An error message is returned at the bottom left of the screen letting the player be aware of the issue.

4A1: The building is already producing a unit, and as such cannot produce two units at the same time. An error message is returned to the player, informing them of the issue.