Testing Results

# Vehicle Module Unit Tests:

|  |  |  |  |
| --- | --- | --- | --- |
| Test Number | Test Description | Expected results | Actual Result |
| 1 | Hold the “W” key to check if the Tank will move forward | The vehicle should move forwards locally to where it is facing | Vehicle behaves as expected |
| 2 | Hold the “S” key to check if the Tank will move backwards | The vehicle should move backwards locally to where it is facing | Vehicle behaves as expected |
| 3 | Press the “1” key to test whether the Tank can correctly target an object | The vehicle’s turret should rotate and point towards the front left building | Vehicle behaves as expected |
| 4 | Press the “2” key to test whether the Tank can change targets | The vehicle’s turret should rotate and point towards the front right building | Vehicle behaves as expected |
| 5 | Hold the “W” key to determine whether the Tank can target an object whilst moving | The vehicle’s turret should stay pointing at the assigned target | Turret points at where the target was when it was initial locked as a target, and does not rotate to face it. |
| 6 | Hold the “Q” key to rotate the Tank anti-clockwise | The vehicle body should rotate anti-clockwise | Vehicle behaves as expected |
| 7 | Hold the “E” key to rotate the Tank clockwise | The vehicle body should rotate clockwise | Vehicle behaves as expected |
| 8 | Hold both the “E” key and the “W” key to test whether the vehicle can rotate and move at the same time | The vehicle should move forward locally to where it is facing, and so should move around in a large circle | Vehicle behaves as expected |

# Vehicle/Particle System Units Integration Tests:

|  |  |  |  |
| --- | --- | --- | --- |
| Test Number | Test Description | Expected results | Actual Results |
| 1 | Press the “1” key to assign a target to the tank | The Tank’s turret should rotate towards the building forward and to the right relative to the Tank | Vehicle behaves as expected |
| 2 | Once the target has been locked, wait for the tank to fire a shot at the target | The tank should fire a small pellet from its turret towards the target. Once the pellet reaches the target, it should create an explosion particle system, emitting from the target | The vehicle correctly fires the pellet and it behaves as expected |
| 3 | One the target has received an attack from the vehicle, it should emit smoke to indicate damage | A smoke system should emit from the target, and should increase in intensity over a short length of time | Smoke system behaves as expected |
| 4 | Press the “2” key to change the target | Like test 1, the tank turret should rotate to point at the target, which this time is back and to the right of the tank | Vehicle behaves as expected |
| 5 | Once the target has been locked, the tank should fire a shell at the new target | The outcome should be identical to the expected results of test 2, with a pellet from the turret and an explosion system upon impact | Both vehicle and pellet behaves identically to first attack, and behave as expected |
| 6 | Once the second target has been shot, a smoke system should emit from the target | Smoke particle systems should be updating on both targets without a significant hit to frame rate | Smoke system behaves as expected |
| 7 | Hold the “W” key to check if Tank movement is still supported | The vehicle should move forward locally to where it is facing | Vehicle movement as expected |
| 8 | Hold the “Q” key to ensure that the Tank rotation is still functioning | The vehicle should rotate anti-clockwise | Vehicle rotation behaves as expected |

# Testing Scripts

**Place Building Use Case**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case Number** | **Test Case** | **Expected Result** | **Actual Results** | **Pass/Fail** |
| 1 | User selects a building to place | A ‘ghost’ of building appears and moves with player’s mouse position | A ‘ghost’ of building appears and moves with player’s mouse position | Pass |
| 2 | The user has a building selected | A ‘ghost’ of building appears and moves with player’s mouse position | A ‘ghost’ of building appears and moves with player’s mouse position | Pass |
| 3 | Comparison between cost of building and minerals of the player | When player clicks, the game should branch down one of two paths; sufficient or insufficient funds | Branches down the correct path of sufficient funds | Pass |
| 4 | Player has sufficient funds | If player has sufficient funds, no error message is shown and game progresses to check surrounding area | No error message and surrounding area is checked next | Pass |
| 5 | Game checks surrounding area to ensure it is clear of other structures | If surrounding area is clear, ’ghost’ of building is shaded green | The ‘ghost’ of the building appears green. | Pass |
| 6 | Building is not overlapping edge of the grid | If the building is not overlapping, it should again appear green | Building ghost appears green | Pass |
| 7 | Game creates building at chosen location | Building no longer tracks the mouse, and the ghost appears white. Can no longer build in that location until building is deleted. | Building is fixed in place with a white ghost appearance. | Pass |
| 8 | Game deducts cost of the building from the player | The player should have less minerals relevant to the price of the building | The player has less minerals then before placing the building | Pass |
| 2A1 | Insufficient minerals with error message | An alert should display stating that the player does not have enough minerals | No error message is displayed however it does stop the player from purchasing the building | False |
| 4A1 | Surrounding area used with error message | An error message should display stating that the area is in use | No error message is displayed, but the ghost of building appears red | False |
| 5A1 | Building overlaps the edge with error message | An error message should display that the building is not in a valid location | Again the ghost turns red and the user cannot place it in that location, but no error message is displayed | False |

**Build Units Use Case**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Case Number** | **Test Case** | **Expected Result** | **Actual Results** | **Pass/Fail** |
| 1 | User selects building they want to purchase a unit from | Information for the current building, along with buttons for constructing units, are shown | The correct information about the building is displayed along with buttons relating to the units that particular building can purchase | Pass |
| 2 | Button for the unit the user wants to build is clicked | The game begins checks for the specific unit consistent with the button that is being clicked | When the worker unit button is clicked, for example, the game begins checks specific to the worker unit | Pass |
| 3 | Game compares cost of unit to the funds of the player | If the player has enough funds, the checks progress | The player did have enough funds and the game began checking against the production queue | Pass |
| 5 | Game checks to make sure unit production queue for that building is not full | If the production queue for the building is not full then the purchase of the unit will be allowed | The unit queue was not full and the purchase was allowed | Pass |
| 7 | Game deducts the cost of the unit from the player | The cost of the unit is deducted from the player | The cost of the unit is deducted from the player | Pass |
| 8 | Building begins production of the unit | The timer until the unit is finished building begins counting down | The timer until the unit is finished building begins counting down | Pass |
| 3A1 | An error message should display to show that the player does not have enough minerals for the unit purchase | The unit cannot be purchased and an error message is displayed | The unit cannot be purchased however no error message is displayed | False |
| 5A1 | An error message should display to state that the queue for this building is full | The unit cannot be purchased and an error message appears specifying that the unit queue is full | The unit cannot be purchased but there is no error messsage | False |