Deliverable 1 Testing Cases and Results

This testing was done using the format of gherkins to standardize the language across all manual test cases. The steps to the gherkins are below and also formatted clearly as a .feature file in the repo.

Test Number	Scenario	Result
1	Loading webXR through quest browser	Pass
When performing the steps described in the gherkin, The webXr technology performed as expected in the oculus rift by loading, and allowing VR		
2	Launch spike through firebase host	Pass
When attempting to launch the spike through a firebase host, and then connecting to the host through the url that firebase created, we could successfully see the content that we had created.		
3	Meta Quest controls interact with spike	Pass
When using the controls from the meta-quest, on the Firebase host, we could successfully use both the real and simulated controls. - There was reported some time that the controls would not release when releasing the buttons, could not reproduce but will need to keep an eye on this happening again		
4	Spike responsivity	Pass
To verify that the firebase and webxr technology were responsive with the application, we interacted with the model to view if there was any obvious latency, which there wasn't - Check to see if there are technologies that can accurately track performance of the application for future testing		
5	Quest can enter virtual reality mode	Pass
When loading a website that has webxr, to enter the virtual reality mode, the website will automatically determine if the device supports VR mode, and the Meta quest 2 could successfully enter this mode.		

Deliverable Gherkin test reproduction steps.

Feature: WebXr Spike

Spike to prove that WebXr can be run using the Meta Quest and confirm other assumptions.

Background:
Given That the user is on the Meta Quest 2

```
@spike @manualtesting @Test1
Scenario: Loading webXR through quest browser
   Given I have configured the meta quest
   And I have opened Quest Browser
   When I load a webXr based browser
   Then the meta quest should support the use of webXr

@spike @manualtesting @Test2
Scenario Outline: Launch spike through firebase host
   Given I have loaded Quest Browser
   When I load the <Firebase> url
   Then the meta quest should load the spike

   Examples:
      | Firebase                    |
      | https://oculus-3d-render.web.app/ |


@spike @manualtesting @Test3
Scenario Outline: Meta Quest controls interact with spike
   Given I have loaded the Quest Browser
   And I have loaded the firebase host
   When I use the <OculusControlls>
   Then the spike model should respond

   Examples:
      | OculusControlls |
      | "Left Trigger"  |
      | "Right Trigger" |
      | "Pointer"       |

@spike @manualtesting @Test4
Scenario: Spike responsivity
   Given I have loaded the Quest Browser
   And I have loaded the Firebase host
   When I interact with the model using quest controls
   And I adjust the viewport
   Then there should be no noticeable lag
   And the meta quest should handle the loading locally

@spike @manualtesting @Test5
Scenario: Quest can enter virtual reality mode
   Given I have loaded the Quest Browser
   And I have loaded the Firebase host
```

When I click the enter vr mode
Then the quest should enter the vr mode