|  |  |  |  |
| --- | --- | --- | --- |
| **IMAT3905 Report Template**  Please write in the boxes below. Expand the boxes as you need to, however this report should not exceed 2 pages without appendices. | | | |
| Name: | Jacob McIntosh | P Number: | P2431308 |
| Github Username: | PandaGotTaken | Github Repo URI:  Branch to be marked: | https://github.com/PandaGotTaken/ResitCW2020 |
| **Describe the functionality of your Key Controller Component:** | | | |
| The key controller allows the user to move any object with the component whilst the object’s key component is selected within the UI. The user can move the object in any of 4 directions relative to the direction it is facing and also rotate the object by holding the specific keys. | | | |
| **Describe the functionality of your AI Controller Component:** | | | |
| Ai controller was not implemented | | | |
| **Describe the changes you made to the UI:** | | | |
| No notable changes have been made to the UI apart from the additional shape being added to the editor (“capsule”) and also text and inputs in the render and keyboard component sections to guide the user on the functionality. | | | |
| **Describe any additional functionality you have added:** | | | |
| Added additional shape called capsule with the base shape that hold no ai or key controller with just the base components.  Added render component options to allow the user to tint the objects and change the mesh type used for each object. This uses radiobuttons to allow the user to select each meshtype and a colour selector to allow the user to choose the colour. | | | |

Test Cases and any other testing information:

Render component Tests

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Test | Method | Expected | Actual | Remedy (if applicable) |
| 1 | Changing all objects into a cuboid mesh type | Test to check that the cuboid radiobutton and relevant code appropriately change mesh | All object’ mesh type should change into a cuboid when the button is selected | As Expected, all objects change into the correct mesh |  |
| 2 | Changing all objects into a capsule mesh type | Test to check that the capsule radiobutton and relevant code appropriately change mesh | All object’ mesh type should change into a capsule when the button is selected | As Expected, all objects change into the correct mesh |  |
| 3 | Changing all objects into a sphere mesh type | Test to check that the sphere radiobutton and relevant code appropriately change mesh | All object’ mesh type should change into a sphere when the button is selected | As Expected, all objects change into the correct mesh |  |
| 4 | Adjusting the colour on all objects to check if the tint changes accordingly using the colour selector | In order to test that the colour is appropriately applying to the object the component belongs to | The object colour should change to the selected colour in the render component | The tint of the object changes appropriately |  |
| 5 | Using the RBG values to determine the colour for the object | This is to test if the RGB settings will find the correct colour to display (100, 50, 50 should equal and greyish red) | When the corresponding RGB numbers are entered the correct colour should display and tint the object | The correct colour is displayed on the colour selector which then tints the object |  |
| 6 | Using the HSV values to determine the colour for the object | The identical colour to the previous test (5) for HSV is 0, 128, 100 | The same colour as used in the RBG test should appear in the colour selector | The correct colour is displayed in the colour selector which then tints the object colour |  |
| 7 | Using the HexCode to select the colour for the colour selector | the same colour as test 5 can be replicated by the hexcode #633232 | The same colour as test 5 should appear in the colour selector | When #633232 is entered in the hexcode section the greyish red hue is generated |  |
| 8 | Entering value above the 255 limit for RGB colours | Increasing the R value to 255555 | The colour should limit at 255 and not increase further | The colour does increase past 255 and as a result the V (brightness) value increase resulting in an extremely sharp colour |  |

Keyboard component tests

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Test | Method | Expected | Actual | Remedy (if applicable) |
| 1 | Testing each direction to test if the cuboid moves correctly | Testing the Front direction by pressing up arrow | The cuboid should move forward in the direction its facing | The cuboid moves in the correct direction |  |
| 2 | ^ | Testing the Back direction by pressing down arrow | The cuboid should move away from the direction it is facing | The cuboid moves in the correct direction |  |
| 3 | ^ | Testing the Left direction by pressing left arrow | The cuboid should move towards the left of the direction its facing | The cuboid moves towards the opposite direction | The directions for Left and Right have to be swapped to move the cuboid appropriately |
| 4 | ^ | Testing the Right direction by pressing right arrow | The cuboid should move towards the right of the direction its facing | The cuboid moves towards the opposite direction | The directions for Left and Right have to be swapped to move the cuboid appropriately |
| 5 | Pressing opposite directions to check if the cuboid moves properly | Pressing both up and back at the same time | should move the cuboid in a forward direction | As expected |  |
| 6 | ^ | Pressing both left and right at the same time | should move the cuboid in a left direction | As expected |  |