|  |  |  |  |
| --- | --- | --- | --- |
| **IMAT3904 Game Engine Report**  Please write in the boxes below. Expand the boxes as you need to, however this report should not exceed 2 pages (not including test case data) | | | |
| Name: | Daniel Ball | P Number: | 2405075 |
| Github Username: | AlSahHim | Github Repo URI: | <https://github.com/IMAT3904/cw21-AlSahHim.git> |
| **Please summarise the functionality of your game engine (bullet points are fine):** | | | |
| * 3D Rendering Space with ability to create 2D and 3D objects, with various tints, textures and lighting. * Render that manages renders such as Open GL / Direct X. * Basic Camera Controls to navigate Up, Down, Left, right. * Event tracking, to enable the tracking of key/mouse inputs as well as window changes. | | | |
| **What testing have you performed and what testing strategy was used? (test cases on next page please)** | | | |
| The testing performed was based on a proof of concept, this meant that for the implementation of each feature of the engine examples were made to check that this was functional and working correctly. | | | |
| **How have you approached your time management for this piece of work?** | | | |
| Time management when it came to this project was crucial as with a strict time frame, everything needed doing within a certain amount of time. This was planned out using the project management feature on github. This allowed me to split my work off into the relevant sections so that keeping track of everything was easy. The target when working on this project was to have my Open GL functions finished by the start of January. This target was met and enable me to continue the remaining piece of my work over the holidays and into the new year. | | | |
| **What have your learned from whilst building your game engine?** | | | |
| Building this Engine has taught me a tremendous amount in many areas. In particular it has shown me the implementation and effect of using Open GL, with the various steps to rendering 3D and 2D objects. It has also shown me how to make this code agnostic which enables you to encapsulate Open GL in generic classes so that if you wanted to use other renderers like Direct X or Vulcan then the code would stay the same. Lastly it demonstrated the necessity for debugging and how making things like event catchers can really help later down the line when there is a mistake in the code. | | | |
| **If you were to undertake this piece of work again what would you do differently?** | | | |
| If I was to undertake this project again, I would really want to dive more into personalizing my code more, with implementing the choice to have multiple ways to get a Camera, window etc. Making my engine as modular as possible. | | | |

Appendix:

**TEST CASE DATA**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Case** | **Plan** | **Expected** | **Actual** | **Comments** | **Pass/Fail** |
| **1** | Console should be able to Log various message types. | Logger can send messages of various types i.e: info, warning, error | Logger can send messages of various types i.e: info, warning, error | **null** |  |
| **2** | Timer should be able to track time since last check. | Timer should be able to track time since last check. | Timer can track time since last check. | **null** |  |
| **3** | Window should be created with specified properties. | When started a basic window with specified properties should appear. | When started a basic window with specified properties appears. | **Null** |  |
| **4** | Events for Window should be triggered on activation. (i.e., resize / move) | Events for Window should be triggered on activation. (i.e. resize / move) | Events for Window are triggered on activation. (i.e. resize / move) | **Null** |  |
| **5** | Input Poller should pick up on key/mouse input events | When run in if statement imput poller should track when a key is pressed. | When run in if statement imput poller tracks when a key is pressed. | **Null** |  |
| **6** | Open GL Error Logger should allow for easier debugging. | Implementation of Open GL Error Logger helps catch errors. | Implementation of Open GL Error Logger helps catch errors. | **Null** |  |
| **7** | Implementation of Open GL to Allow the rendering of basic 3D Shapes with textures and lighting. | Implementation of Open GL to Allow the rendering of basic 3D Shapes with textures and lighting. | Implementation of Open GL Allowed the rendering of basic 3D Shapes with textures and lighting. | **Null** |  |
| **8** | Make the Open GL code agnostic to allow the code to be universal to any renderer. | Open GL code functional with agnostic code. | Open GL code functional with agnostic code. | **Null** |  |
| **9** | Implementation of 2D Rendering. | Rending should allow visible 2D shapes in the window. | 2D Shapes render in window. | **Null** |  |
| **10** | Camera Controls (Basic) | Controlling your view with WASD. | User can move up, down, left & right using WASD keys. | **Null** |  |