|  |  |  |  |
| --- | --- | --- | --- |
| **IMAT3904 Report Template**  Please write in the boxes below. Expand the boxes as you need to, however this report should not exceed 2 pages. | | | |
| Name: | Radoslaw Zajdel | P Number: | P17215040 |
| Github Username: | Radixos | Github Repo URI: | https://github.com/IMAT3904/cw-Radixos.git |
| **Please summarise the functionality of your game engine (bullet points are fine):** | | | |
| The game engine has almost all classes presented on the lectures implemented. It has a dispatch loader for shaders, it has layers, wide range of OpenGl classes for rendering, buffers, arrays, materials and including render commands, events with an event dispatcher, a timer, a 2D camera with a camera controller and resource management. Altogether, the implemented code allows an easy addition of objects and functionalities. | | | |
| **What testing have you performed and what testing strategy was used?** | | | |
| There was an in-depth testing performed on implemented events, including resizing the window, closing it, measuring the position of focus of the mouse on the window, measuring the position of the mouse and mouse buttons presses and releases. Testing events gave me an opportunity to test relevant dispatcher functionality. More tests would have been done if time management of the project were done more efficient.  There is a blackbox testing done and it is in the bottom of this document. | | | |
| **How have you approached your time management for this piece of work?** | | | |
| My time management approach was not as it should be for this type of work. For first couple weeks not enough time was spent on implementing given code and classed. Classes provided on lectures and videos were only retyped and mostly that was all that was done with the code. However, after around a month, more and more time was spent on the project. Unfortunately, very late I realised how helpful other students and lab tutors can be and that is a great opportunity lost. In the middle of the term there can be an increased commits number seen on github, with more and more code implemented. More hard-working time with this assessment was within last month before deadline. There were big chunks of code implemented every couple days and it significantly moved the work forward. The most efficiently spent time was in the last week before deadline, where many things like cameras, layers, resource manager and text was implemented. | | | |
| **What have your learned from whilst building your game engine?** | | | |
| Advanced programming including GLSL functionality, OpenGL functions, how to use and implement GLFW, including input code, to process user input, lambdas, casting, understood pointers, layers and rendering, rendering pipeline, how to use renderer and render command, what an API is and how to code in an API manner so the code can work on multiple platforms, how to use timer and chrono functions in cpp code so events actions and animations can be performed easily and in an understandable way, how to perform events and dispatch them so it is easy to implement more, what are callbacks and how should they be used, what macros are, how useful they can be and how much more better looking code can be when using them, what vertex array, vertex buffers and index buffer are and how to implement them with OpenGL, how the data is send to GPU with the use of OpenGL, how textures are being used and how they are implemented with the OpenGL, the usability of resource manager, how to implement cameras, start simple scenes, how to output text on the screen and how to efficiently manage the code. | | | |
| **If you were to undertake this piece of work again what would you do differently?** | | | |
| If given a chance to start the work on this project again, I would prepare a detailed plan on what needs to be done and strictly stick to it. I would also start spending more time on the project every week. A good idea would be to do more online research and try to reach to lecturers and lab tutors more often to find feedback and hear tips for development. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Test performed** | **Expected result** | **Received result** | **Problem description and potential fix** |
| 1 | Close event with ESC | Window closes | Window closed | - |
| 2 | Resize event with dragging | The window changes its size and a message is shown | The window changed size and message was shown | - |
| 3 | Mouse move over the window event | Whenever a mouse is moved over the window a message is shown | Whenever a mouse was moved over the window a message with coordinates was shown. | - |
| 4 | Mouse button press and release events | Whenever a mouse button is pressed a message is shown | The message was not shown | The problem was diagnosed with the GLFWCodes.h file, where not all codes were implemented and some of them were implemented incorrectly. Reworking the file helped solve the problem. |
| 5 | Data dispatching for shaders | Dispatched shaders with the use of uploadData function | The data was not dispatched at all | The function needed to be reworked as it was not written correctly. Additional online research was necessary to be done to understand how the dispatching works. |
| 6 | Text showing on the screen | Text is shown on the screen | There was no text on the screen | There was a problem with setDataElement function in OpenGLMaterial.cpp file, it was fixed. |
| 7 | Loading a texture for text displaying | The texture will be “DMU engine”. | The texture was different than “DMU engine”. | A string filepath needed to be fixed. |