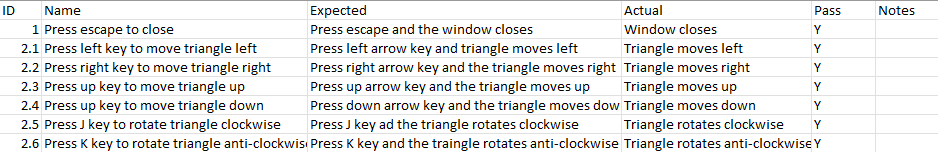
|  |  |  |  |
| --- | --- | --- | --- |
| **IMAT3904 Report Template**  Please write in the boxes below. Expand the boxes as you need to, however this report should not exceed 2 pages (not including test case data) | | | |
| Name: | Saul Batts | P Number: | P2585183 |
| Github Username: | saul1508 | Github Repo URI: | https://github.com/IMAT3904/cw23-saul1508 |
| **Please summarise the functionality of your game engine (bullet points are fine):** | | | |
| * Console logger for debugging and information and file logger. * General timer and chrono timer for windows. * Randomiser to generate floats or integers from uniform and normal distributions. * Event handler for window close, window focus, window lose focus, window resize, window move, key pressed, key typed, key released, mouse button press, mouse button release, mouse moved, and mouse scrolled events. * Window and graphics context to create a window to render to * Cubes and pyramids can be drawn to the window and updated with different shaders and textures. * Triangle can be moved and rotated through user input. | | | |
| **What testing have you performed and what testing strategy was used? (test cases on next page please)** | | | |
| * Google tests on handling for each event * Test cases for input | | | |
| **Did you change any code as a result of use a profiler or a GPU debugger, if so how?** | | | |
| * I have used Nsight to fix a lot of errors with OpenGL that aren’t obvious through visual basic such as checking if uniforms are being sent to the GPU correctly. | | | |
| **How have you approached your time management for this piece of work?** | | | |
| * Using Trello to break down tasks to be done, making sure to keep to the time limit. | | | |
| **What have your learned from whilst building your game engine what would you do differently next time?** | | | |
| * Be even more consistent with time management as it would have allowed me to get more done and add a 2D and 3D renderer * I would make sure to use a GPU debugger a lot more such as Nsight or RenderDoc as this would have made a lot of issues much easier to solve and I could also use it to change code to make OpenGL code more efficient * I would have used more comments in the source (cpp) files, especially the application.cpp file when adding the raw OpenGL and shader code before abstracting as it got very confusing, and I made many mistakes that could have been avoided. * I would make sure to stick to a particular formatting and naming convention throughout the code to make everything easier to understand * Commit more regularly as there were a few times that I had to undo big changes that I had forgot to commit before doing so was a lot harder to fix than just reverting to an earlier commit | | | |

Appendix:

**TEST CASE DATA**

****

