UOW
MALAYSIA

PART OF THE UNIVERSITY
OF WOLLONGONG AUSTRALIA
GLOBAL NETWORK

Bachelor of Game Development (Hons)

Game Engine Architecture and Design

XBGT3124N

Prepared by Mohamad Faris Zakwan
Semester September 2024

SCHOOL OF
COMPUTING
& CREATIVE
MEDIA

ASSIGNMENT 1

Course Title	: Game Engine Architecture & Design
Course Code	: XBGT3124
Course Lecturer	: Mohamad Faris Zakwan
	: faris.z@uow.edu.my

BRIEF

> ⚠ Important ⚠
>
> This course's assignments are continuous in nature.
>
> Your Assignment 1 work will serve as the base for Assignment 2, and your Assignment 2 will serve as the base for Assignment 3.
>
> Plan your work wisely. Taking shortcuts affects your future.

Create a core game engine framework (see Requirements).

Include a showcase demonstrating the features (see Showcase).

ASSESSMENT CRITERIA

Coursework marks allocated for this assignment is **20%**, of which:

• Application Framework	: 7%
• Input System	: 8%
• Showcase	: 5%

No submission or non-working submission is assessed as 0% of the allocated marks.

DUE DATE: 20 OCTOBER 2024 11:59:59PM

REQUIREMENTS

Core Game Engine Framework

- Programming language: C++
- Mandatory libraries: GLFW, glad, glm
- Programming Principles: Separation of Responsibility, Abstraction, Encapsulation
- **Game engine framework and its components MUST be in the same namespace.**
- **You MUST change the namespace from `MyGameEngine` to something else.**

Engine Features

Game developers should be able to (via code):

- Easily change the frame update limit.
- Easily close the application.
- Easily query the current application screen size and change the screen size.
  - Use `glfwSetFramebufferSizeCallback` instead of `glfwSetWindowSizeCallback`
- Easily query input (keyboard, mouse, and cursor) states.

Input System Features

STATES	
Mouse Buttons	Left, Right, Middle.
Keyboard Keys	• Standard Alphanumeric (A~Z, 0~9) • (Left & Right) Control, Alt, Shift • Space, Escape, Enter • Function keys (F1~F12)
FUNCTIONS	
`bool GetMouseButton(...)`	Returns true on frames where user is holding the button.
`bool GetMouseButtonDown(...)`	Returns true **ONLY** on the frame where user starts pressing the button.
`bool GetMouseButtonUp(...)`	Returns true **ONLY** on the frame where user releases the button.
`glm::vec2 GetMousePosition(...)`	Returns the cursor position relative to the top-left of the window.
`bool GetKey(...)`	Returns true on frames where user is holding the key.
`bool GetKeyDown(...)`	Returns true **ONLY** on the frame where user starts pressing the key.
`bool GetKeyUp(...)`	Returns true **ONLY** on the frame where user releases the key.

Showcase

Your showcase **MUST** demonstrate:

1. Alphanumeric number keys 1, 2, 3, 4, and 5 to set frame limit to 10, 20, 30, 60, and unlimited, respectively.
2. Function keys F1, F2, F3 to change window size to 200x150, 400x300, and 800x600 respectively.
3. Key state showcase:
   - Escape (On Pressing): Quit application.
   - Space (Just Pressed): Print the screen size to the console.
   - A (Just Pressed): Do something visually.
   - D (Just Released): Do something visually.
4. Mouse button state showcase:
   - Right button (Just Pressed): Print that the button just pressed and the cursor position to the console.
   - Right button (Just Released): Print that the button just released and the cursor position to the console.
   - Left button (On Pressing): Do something visually continuously.
5. Window event showcase:
   - On window close button pressed: print to console.
   - On window framebuffer resized: print the new size to console.

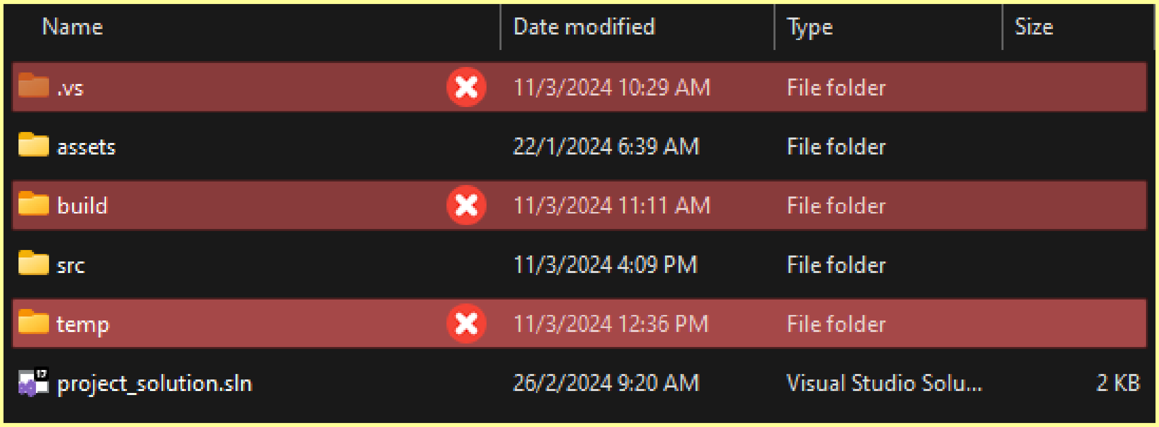
Notes

- Make a class named `ShowcaseScene` for the showcase, like how `TestScene` works in the tutorials.
- `ShowcaseScene` contains functions that are called in `Application::run()`:
  - `static void init()` - initialization.
  - `static void update()` - input related logic.
  - `static void draw()` - basic OpenGL rendering.
- `ShowcaseScene` IS NOT part of game engine components.

SUBMISSION REQUIREMENTS

⚠ Do not include the highlighted folders in your submission! ⚠

By default, .vs folder is hidden in Windows Explorer. Enable Hidden items to see it.



Name	Date modified	Type	Size
.vs ✖	11/3/2024 10:29 AM	File folder	
assets	22/1/2024 6:39 AM	File folder	
build ✖	11/3/2024 11:11 AM	File folder	
src	11/3/2024 4:09 PM	File folder	
temp ✖	11/3/2024 12:36 PM	File folder	
project_solution.sln	26/2/2024 9:20 AM	Visual Studio Solu...	2 KB

Delete the highlighted folders before zipping

Zip Filename Format: *StudentID*_XBGT3124N_A1
e.g., 0120123_XBGT3124N_A1.zip

Your zip file MUST contain:

1. project/ folder **(REMEMBER TO DELETE THE HIGHLIGHTED FOLDERS!)**
2. deps/ folder
3. Readme or documentation about your project.

Each student is given access to an individual OneDrive folder for assignment submissions.

- Only the student and lecturer-in-charge will be able to access the contents inside the folder.
- The same folder is used for all coursework submissions.

Penalty applies:

- **If you submit past the deadline: -20% of marks per day late.**
- **If you update your submission past the deadline without receiving permission from the lecturer beforehand.**

ASSESSMENT RUBRIC

CRITERIA	MARKS					
Framework	**-**	**-**	**-**	**2**	**1**	**0**
Core (2)				Application Framework exists following tutorials 1 and 2. Supports two (2) of the following window events: • Close • Screen Resize	Application Framework exists following tutorials 1 and 2. Supports one (1) of the following window events: • Close • Screen Resize	One of the following: • No submission. • Does not meet minimum requirements.
Ease of Access Feature (2)				Supports two (2) of the following via code: • Limit Frame Update • Window Resizing	Supports one (1) of the following via code: • Limit Frame Update • Window Resizing	
Abstraction (1)					GLFW and gl functions are hidden from game engine users. Note: this means it is impossible to use GLFW or gl functions in Application class.	
Namespacing (2)				All classes are namespaced correctly.	Some classes are not namespaced or are inside a wrong namespace.	

Continued next page…

Input	**-**	**-**	**-**	**2**	**1**	**0**
Key and Button Enumerations (2)				Contains all the stated key and mouse button enumeration.	Missing some key and/or mouse button enumeration.	One of the following: • No submission. Does not meet minimum requirements.
Keyboard Key States (2)				Handles key press, just press, and just released states.	One (1) of the following: • Handles key press state only. • Buggy implementation.	
Mouse Button States (2)				Handles mouse button press, just press, and just released states	One (1) of the following: • Handles key press state only. • Buggy implementation.	
Mouse Cursor (2)				Can obtain cursor X & Y position from one function call.	Can obtain cursor X & Y position from two separate function calls.	
Showcase	**5**	**4**	**3**	**2**	**1**	
Content (5)	Meets five (5) of the showcase requirements.	Meets four (4) of the showcase requirements.	Meets three (3) of the showcase requirements.	Meets two (2) of the showcase requirements.	Meets one (1) of the showcase requirements.	
PENALTIES					**Namespace**	**Late**
Infractions					Not changing namespace from MyGameEngine to something else (-1)	-20% assessed marks per day late. How days are counted: x = hours late Days = ceil(x / 24) e.g. 5 minutes late: x = 5/60 = 0.0833 Days = ceil(0.0833/24) = ceil(0.003472) = 1 day 27 hours late: x = 27 Days = ceil(27/24) = ceil(1.125) = 2 days