UOW MALAYSIA

PART OF THE UNIVERSITY
OF WOLLONGONG AUSTRALIA
GLOBAL NETWORK

Bachelor of Game Development (Hons)
3D Game Programming
XBGT2094N/XBGT2094
Prepared by Mohamad Faris Zakwan
Semester May 2024

SCHOOL OF COMPUTING & CREATIVE MEDIA

ASSIGNMENT

Course Title   : 3D Game Programming
Course Code   : XBGT2094N/XBGT2094
Course Lecturer   : Mohamad Faris Zakwan
                      : faris.z@uow.edu.my

BRIEF

1. Construct a 3D scene with several sources of lighting. (25%)
2. Then, using framebuffer objects:
   a. Add one renderer feature (10%)
   b. Add one themed post-processing effect (10%)
3. Provide documentation describing your renderer feature and post-processing implementation (5%)

NOTES

1. You will need to code in C++ and GLSL (vertex and fragment).
2. You will need to obtain your own 3D models that also meet the requirements of this assignment.
   a. It is OK to use 3D models from your peers specializing in Art, ensure that you include attributions in the documentation.

3D SCENE (25%)

Your scene MUST:
- Have a base.
  - E.g. a floor, a floating island, or a body of water.
- Have at least 6 unique objects.
  - **MUST fit a theme of your choosing.**
  - **The base is not counted.**
  - See Useful Resources section.
- **Have at least 1 object that is constantly moving.**

Objects in the scene MUST:
- Be illuminated by active sources of light.
- Utilize diffuse, specular and normal textures.
  - **1 object MUST also utilize emissive texture.**

Scene lighting requirements:
- 1 directional light
- 4 positional lights (Point and/or Spot)
  - **These lights MUST look like they come from a physical source.**
    - E.g. a candle, a lamp, alien spaceship.
- 1 positional light that is:
  - Automatically moving.
  - Cycling colour smoothly following rainbow pattern in exercise 2.
- All lights MUST be editable:
  - Enable/Disable light.
  - Alter light properties.

NOTES
1. Highlighted sentences are requirements that were frequently overlooked by students.
2. For normal mapping, the application already provides Tangent and Normal vectors. Use a relevant vector operation on them to obtain Bitangent.

RENDERER FEATURE (10%)

Renderer features provide enhancement to rendering result. **You are to implement one of the following:**

SHADOW
- Only one directional shadow is required (Point light can be complex).
- Provide means to change shadow strength and bias

HDR
- Requires understanding Gamma Correction.
  - TextureUtils have a function to load textures as sRGB.
- Provide means to change exposure value.

DEFERRED RENDERING
- Requires major rendering restructure.
- May make post-processing more complex to implement (or easier if you know exactly what to do).
- Provide means to view individual buffers before composition.
- Extra 5% marks for implementing this (10% > 15%)

NOTES
1. **Only implement ONE renderer feature.**
2. The linked articles implement renderer features using OpenGL Framebuffer API. **You are to use ColourDepthFBO and/or DepthOnlyFBO classes instead.**
3. To get maximum marks, you need to provide means to tweak the renderer feature.

THEMED POST-PROCESSING (10%)

Make ONE themed post-processing effect that changes the rendering result. **Provide controls to tweak the effect and to enable/disable it.**

Examples:
- Horror
- Post-apocalyptic
- Security camera
- Night vision
- CRT TV
- Black and White film

NOTES
1. A themed post-processing is composed using multiple individual effects applied together (such as vignette, sharpen)
2. This is not a numbers game! Too many effects do not necessarily mean the result will look good, and too little effects mean the result will look simplistic. Find a balance!

DOCUMENTATION (5%)

Explain, in your own words, the methodology used to implement the renderer feature and the themed post-processing effect.

- Explanations should be concise and straightforward.
- Show and explain the responsible code snippets.
- Make use of pictures/figures to help your explanation.
- Provide attributions to all tutorial/codes you learn from or make use of.
    - This is also where you provide 3D model attributions.

USEFUL RESOURCES

You can find models or textures from:
- OpenGameArt.org
    - Yughues | OpenGameArt.org
    - CC-Zero Models | OpenGameArt.org
- itch.io - Game Assets

If the model you want to use does not come with normal or specular images, you can use this online tool to generate them.
- For best results, use diffuse image that has grayscale applied as the input.

You can refer to any resources for guidance, such as LearnOpenGL.com.

SUBMISSION REQUIREMENTS

⚠ **Do not include the highlighted folders in your project/ folder!** ⚠

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*_XBGT2094_ASGN

    e.g., 0120123_XBGT2094_ASGN.zip

Your zip file MUST contain:
1. project/ folder **(REMEMBER TO DELETE THE HIGHLIGHTED FOLDERS!)**
2. deps/ folder.
3. Documentation **(in PDF)**

Each student is given access to a OneDrive folder for coursework submissions. Check your student email.
- 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 CRITERIA

Coursework marks allocated for this assignment is **50%**:

- 3D Scene   : 25%
  - Objects   : 10%
  - Lighting   : 10%
  - Code Quality   : 5%
- Renderer Feature   : 10%
- Themed Post Processing   : 10%
- Documentation   : 5%

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

DUE DATE:
<u>WEDNESDAY, 21 AUGUST 2024, 11:59:59PM</u>