Final Project (V1)

Please make an app that will have the following features:

FBX mesh loader see xGPU as an example 6 for reference.

- Use Assimp to load your meshes
- Primitive must have (Tangent, Bitangent, and Normal)
- Others; UV, Position also needed.
- Pass the L2C matrix as a push constant

Texture and lighting (Look at xGPU example 11, and 12)

- Texture map should be loading using a DDS loader
- Support the following textures maps:
    - Albedo/Diffuse texture (DXT1 compress)
    - Normal map should be an OpenGL compatible texture. (Green UP)
        - Should be BC5 compress (Also known as 3Dc) This is the standard.
    - Ambient Occlusion (DXT1 compress)
    - Glossiness is parametric value which represents a fading factor for the specular intensity (DXT1 compress)
    - Roughness is parametric value which represents the exponent of the specular (DXT1 compress)
- Remember that each texture may be linear or SRGB please check with the examples about this.

Please mirror the input from (xGPU example 11, 12)

- Rotating camera
- Space Locks the light position to where the camera was when you pressed it, and if you press space again will lock back to where the camera is.
- Pass the light position as a push constant.

The fragment shader should:

- Compute the diffuse intensity.
- Compute the specular intensity
- Compute the Ambient lighting
- Please look at xGPU example 12 for texture references.
- Please remember to use Linear and Gamma correctly

Push constants:

- Pass the EYE position in local space
- LC2
- Light direction in local space
- Light Color
- Ambient lighting
- See xGPU Example 12.

Extra Credit

- Can load different meshes
- Can have multiple lights

Delivery

- Always look at xGPU as a reference.
- Make sure the directory structure of your project is clean
- Make sure you have a build directory
    - Here make sure that you have a batch file to download any 3rd party libraries that you may need.
    - Here is the right place to put the files for the solution
- Make sure you have a src directory
    - Here is where the source files for your Vulkan Engine should go
    - Do not put the mesh viewer files (.cpp .h) mix with your Vulkan Engine
        - Either create a subdirectory for the meshviewer *OR* Put your example in a different root directory entirely.
- Have root Asset directory for all your textures, meshes, etc.
- At the root include the .exe already built, I should be able to just double click on it.

Please DO NOT SUBMIT:

- Temporary files like .obj, .pdb, etc

As always should be delivery as a zip file with your Distance learning Name.