UNIVERSITY OF LEEDS

School of Computing: Assessment brief

Module title	Foundations of Modelling and Rendering
Module code	COMP5812M
Assignment title	A2 – Advanced Techniques
Assignment type and description	Programming assignment.
Rationale	Applying advanced concepts described in the taught material, and extrapolating known concepts to novel proposed problems.
Word limit and guidance	8 Page limit for the report component.
Weighting	50%
Submission deadline	25/01/2024
Submission method	Gradescope
Feedback provision	Gradescope will detail the feedback for both the report and programming components.
Learning outcomes assessed	LO1, LO2, LO3, LO4, LO5, LO6, LO7, LO8, LO9, LO10
Module lead	Rafael Kuffner dos Anjos
Other Staff contact	

1. Assignment guidance

This is your second assignment, which is worth 50 points. You can complete this assignment after completing the first one, as it builds upon the codebase from the first assignment. It consists of a list of tasks that you can choose to tackle to gain points. With every task you implement, it is required you write a short description of your method and results in a report that will be assessed with your submission.

2. Assessment tasks:

Raytracing tasks

The following tasks should be implemented on top of your raytracer. It is recommended that they are addressed in this particular order, as there is a dependency from one to the next.

Refraction and Fresnel effect: 7 marks

Use the material properties to verify if the object is transparent (values are 0 or 1). If it is, calculate the refraction ray keeping track of the correct IOR and the appropriate shaded result when it hits an opaque object. Pay attention to total internal reflection. Also implement the Fresnel equations (e.g. Schlick approximation), which will transform part of the refracted light into reflection, and also part of the diffuse (considered internal refraction) into reflection. Combine results accordingly.

Montecarlo sampling for indirect lighting: 6 Marks

Replace the ambient component of your computation with a value calculated using monte-carlo sampling. For this, you will change the code to run as a path tracer when this checkbox is on, so a single indirect ray per loop should suffice. You should still use next event estimation (direct lighting rays), but being careful not to account for light sources twice. Moreover, pay attention to how the samples are generated so the result is unbiased. Add anti-aliasing to your path-tracer, with each primary ray being sent through a random position inside the pixel.

Area Lights: 6 Marks

Change the code that samples the light sources when doing monte-carlo rendering, along the whole surface of the light source as defined in the obj files (and their sizes in the Light class). This will allow you to obtain soft shadows. Implement a function that provides this sampled position and use it accordingly.

Caustics: 6 Marks

With refraction, monte-carlo, and area lights implemented, you should see caustics being formed when light goes through transparent objects. I'd like to see this effect in your raytracer, with next event estimation still working. For this, you should change the way NEE works, where rays should refract, and a collision with the surface of the area light should be confirmed.

Rasterisation Tasks

The following tasks should be implemented on top of your terrain renderer. There is no dependency in between them, so you can implement them in any order you want.

Dynamic LOD: 7 Marks

Instead of using a higher number of points (n_points) to create a higher quality mesh, you will implement using tessellation shaders to refine our mesh in GPU. Create a tessellation control shader which will set the tessellation level to a fraction of the resolution of the heightmap, or its total size (if you have the GPU for it), depending on how far they are from the camera. Then create a tessellation evaluation shader which should now process the newly generated vertices and transform them. You should also work on the C++ side to generate GL_PATCHES instead of GL_TRIANGLES in the LoadModel function, given that these are the primitives expected by the tessellation shaders.

Billboard vegetation: 6 marks

Add a detail pass where you render vegetation (bushes, trees, flowers) as billboards in a geometry shader. If the height of a vertex being processed (which one should you use?) is in the range where you would texture it with grass, it should now have a random chance of having a billboard with vegetation. This should be done as an additional render pass to the previous one.

Animated Water: 6 Marks

Add a new render pass where water is rendered in the lowest parts of your terrain. Water should be slightly transparent, and animated using sinusoidal functions to simulate the effect of light wind over its surface. Your water should also have a simple texture.

Skybox: 6 Marks

Implement a skybox with clouds on the topmost hemisphere. If you implement the water, the reflections should be seen in the surface of the water.

3. General guidance and study support

Consult the Minerva page to obtain the slides for this module, and references in the reading list. Your main source of support for this assignment is the laboratory sessions.

4. Assessment criteria and marking process

For each task you implement, you should write a short description of it (no more than 2 paragraphs) in the report. If implemented correctly, and clearly explained, you will be awarded the marks described in this document. Partial marks will be awarded according to how well it was implemented/described in the report.

5. Presentation and referencing

Report should be written in English, and respect the page limit. You will be penalized with 5 marks for each page you go over the page limit. Any research material used to complete these tasks should be correctly cited and referenced. Missing a reference can be considered a case of academic malpractice.

6. Submission requirements

:Please prepare a zip file for each one of the codebases (in a similar way to the way you have submitted A1), but making sure to add any additional files you have created or require for your rendering (e.g. textures for vegetation/water, shaders you created). The expected file structure is as follows

- Raytracing
    - RaytraceRenderWindow
        - …
- Rasterisation
    - Src
    - … extra files you need
- Report.pdf

Submit to Gradescope.

7. Academic misconduct and plagiarism

All of the code submitted will be verified using plagiarism detection software. You are not allowed to submit code that was not written by you. All the submitted code must be written from the start by you. You can use external references to understand a topic, not to copy paste a solution into your codebase.

8. Assessment/ marking criteria grid

Refraction and Fresnel effect	**7**
Montecarlo sampling for indirect lighting	**6**
Area Lights	**6**
Caustics	**6**
Dynamic LOD	**7**
Billboard vegetation	**6**
Animated Water	**6**
Skybox	**6**