Faculty of Computing, Engineering and Science

Assessment Cover Sheet and Feedback Form 2021-22

Module Code: CS3S664	Module Title: Real-time Rendering Techniques	Module Team: Carl Jones, Marius Miknis
Assessment Title and Tasks: Full Screen Anti-Aliasing		Assessment No. 1
Date Set: 27-Sep-2021 21:00	Submission Date: 18-Feb-2022 23:55	Return Date: 16-Mar-2022 23:55

IT IS YOUR RESPONSIBILITY TO KEEP RECORDS OF ALL WORK SUBMITTED

Marking and Assessment
This assignment will be marked out of 100% This assignment contributes to 40% of the total module marks.
Learning Outcomes to be assessed (as specified in the validated module descriptor https://icis.southwales.ac.uk/): 1) To be able to analyse and critically evaluate techniques used to render 3D scenes in real-time 2) To design, implement and evaluate GPU shaders in order to render effects in real-time
Provisional mark only: subject to change and / or confirmation by the Assessment Board

Assessment Task

Background Information:

Full-Screen Anti-Aliasing (FSAA) is a technique used to eliminate aliasing artefacts such as jagged edges or texture artefacts during animation for example. Anti-aliasing is popular in modern video games because of the improved visual quality it produces. There are numerous variations of FSAA that game engines and video cards support. Two popular approaches are Super Sample Anti-Aliasing (SSAA) and Multi Sample Anti-Aliasing (MSAA).

With **Super Sample Anti-Aliasing** , a scene is rendered to an off-screen texture at a higher resolution than the screen resolution. When the scene is rendered to the actual screen, each on-screen pixel has a number of associated 'sub-pixels' in the off-screen texture. These are combined to produce a final colour for the on-screen pixel. For example, a 2x super-sampled scene rendered into an 800x600 on-screen window will actually need to be rendered into a 1600x1200 off-screen texture. A 4x super-sampled scene would need a 3200 x 2400 off-screen texture.

Multi Sample Anti-Aliasing can be considered a special case of super-sampling. However, the scene does not have to be rendered into a high-resolution off-screen texture. Instead of sampling multiple off-screen pixels (each with their own lighting calculations) we still take multiple samples at different locations within a single pixel (at sub-pixel coordinates). Each sample shares the same texture and lighting calculations but can have different depth values at each sub-pixel location, so the overhead of having to calculate separate texture and lighting information at each sub-pixel is not present. Multi Sample Anti-Aliasing is popular with modern game engines and both OpenGL and DirectX provide built in support for this technique.

DirectX and OpenGL implement multi-sampling as standard features; *this assignment will look at setting up the context to enable multi-sampling and implementing super-sampling using shaders. This will give you an understanding of how each technique works, the cost in terms of speed in using each technique and how the results of each technique differ. This assignment will build upon the vertex shader, fragment shader and texture access tutorials covered on the course.*

The following steps outline the tasks required for this assignment:

1. Setup a simple 3D scene using a number of different objects and light sources. The objects must be setup using VBOs and VAOs and texture mapping must be applied to at least one object.

2. Render the scene using MSAA. Setup the OpenGL rendering context to use Multi-Sample Anti-Aliasing. Render your scene using different numbers of samples and document the results.

3. Implement Super-Sample Anti-Aliasing (SSAA). This requires you to implement the following steps:

    a. Setup an FBO and texture (render target) - the resolution of the texture should be at least 2x the resolution of the on-screen window.

    b. In the first pass, render your scene to the high resolution texture.

c. In the second pass, render a single quad to the screen which is to be textured with the high-resolution texture created in the first pass. The quad must be screen-aligned, so you need to determine the most appropriate coordinates and projection mode to make sure the quad fills the screen.

d. As part of the second pass, implement a pixel shader that samples multiple texels from the high-resolution texture, averages the values and outputs a final pixel colour to the screen. The sample pattern and number of points you sample will affect the results. Experiment with different resolutions and sampling patterns and document your results.

4. Write a report that discusses the results obtained by rendering your scene with no anti-aliasing, MSAA and Super-Sample Anti-Aliasing. For each technique, note the changes in visual quality / aliasing artefacts and discuss the impact on performance each technique has.

5. You will also be required to explain your design and implementation in a short 5-10 minute code demo which will take place in the tutorial sessions after the assignment has been submitted.

The code demo is mandatory. The above sections will also be marked according to how well you demonstrate your understanding of them in the code demo.

The report should be word processed and no longer than 2000 words. You may use the tutorial or lecture demo code as a starting point for your implementation.

Deliverables:

- A report that includes the details of the results obtained, 2/3 screenshots of your scene and your **edited*** code as an appendix at the end of the report.

    - *You do not need to include code of the files that you did not change.

- A 5-10 minute code demo discussing your implementation, the results obtained and the problems you faced in implementing the assignment.

Marking Scheme:

	Fail	Narrow Fail	3rd Class / Pass	Lower 2nd Class / Pass	Upper 2nd Class / Merit	1st Class / Distinction
Quality of the final 3D scene 10%	□ The overall quality of the final 3D scene is very poor □ No setup or implementation of lighting	□ The overall quality of the final 3D scene is poor □ Poor setup and implementation of lighting	□ Only a basic 3D scene has been implemented □ Setup and implementation of lighting is satisfactory	□ A 3D scene is presented that is of a good quality □ Setup and implementation of lighting is good	□ A 3D scene is presented that is of a very good quality □ Setup and implementation of lighting is very good	□ A 3D scene is presented that is of an excellent quality □ Setup and implementation of lighting is excellent
Multi-Sample Anti-Aliasing implementation 10%	□ No working MSAA implementation is given	□ Poor Multi-Sample Anti-Aliasing implementation	□ Satisfactory Multi-Sample Anti-Aliasing implementation	□ Good Multi-Sample Anti-Aliasing implementation	□ Very good Multi-Sample Anti-Aliasing implementation	□ Excellent Multi-Sample Anti-Aliasing implementation
Super-Sample Anti-Aliasing implementation 30%	□ No working SSAA implementation is given	□ A limited or incorrect SSAA implementation is given	□ A basic SSAA implementation is given, with a working screen-aligned quad	□ A good SSAA implementation is given, with a working screen-aligned quad	□ A very good SSAA implementation is given, with an efficiently calculated screen-aligned quad	□ An excellent SSAA implementation is given, with an efficiently calculated screen-aligned quad
Report 30%	□ No report was provided □ no meaningful discussion of the results provided □ No real testing of MSAA & SSAA	□ A basic report is given with no meaningful results or discussion □ Limited discussion of the results provided □ Limited or incorrect testing of MSAA & SSAA given	□ A basic report is given showing the qualitative (visual) and quantitative (timing) results □ Only a basic discussion of the results is also given □ Only a limited number of resolutions and sampling patterns have been tested for MSAA & SSAA	□ A good report is given showing the qualitative (visual) and quantitative (timing) results □ A good discussion of the results is also given □ A number of resolutions and sampling patterns tested for MSAA & SSAA	□ A detailed report is given showing the qualitative (visual) and quantitative (timing) results □ A very good discussion of the results is also given □ A number good of resolutions and sampling patterns tested for MSAA & SSAA	□ A very detailed report is given showing clearly the qualitative (visual) and quantitative (timing) results □ A detailed discussion of the results is also given □ Good variety of resolutions and sampling patterns tested for MSAA & SSAA
Demonstration 20%	□ No Demonstration	□ Does not understand the code	□ Has a simple knowledge of the code	□ Has a reasonable knowledge of the code	□ Has a good understanding of the code	□ Has an excellent understanding of the code