C++ 3D Assignment 1 (2.5 pts) - Quads & Shaders

Learning goal 1: Applying basic knowledge of the OpenGL *rendering* pipeline (*see lecture 1*).

Getting started

- Download the library bundles;
- Download the lecture slides and examples;
- Get the lecture samples working in your IDE as described in the lecture.

Assignment

Turn the given triangle from the last example in lecture 1 into a quad with a checker pattern.

Grading criteria

- To **pass** this course, each assignment needs to be **finished at the 'mediocre' level at least**, and all assignments together need to result in a minimum grade of **at least 5.5 points** (e.g. 1 'mediocre' + 3 'averages').
- To **pass** a level for a specific rubric you also need to pass all the '**lower**' levels in that rubric ***and be able to explain all your implementation changes.***

Mediocre (1 pts):

You changed the given triangle into a quad using a **single** glDrawArray or glDrawElements call in TRIANGLE_MODE.
You've implemented a checkerboard pattern that moves along with your quad using a (vertex/fragment) shader.
The number of columns and rows can be set through uniforms.

Average (1.5 pts):

Your pattern rotates and/or scales around the centre of your quad with a speed that can be controlled using the keyboard.
A fake lighting effect based on the mouse-to-fragment distance has been implemented.

Good (2 pts):

You have two animating planes on screen at the same time, one with the regular checkerboard pattern and one with a radial checkerboard pattern.
More interesting effects are also allowed but discuss this with your lab teacher beforehand.
More than 2 parameters are controlled through uniforms.

Excellent (2.5 pts):

Your rectangular and radial checkerboard quads are planes made up of at least 10 * 10 vertices with triangles created using an index buffer and glDrawElements.
At least one of your planes animates in a wave pattern in both x & y direction.

Cookie bonus: you've implemented some animating plasma 😊