Assignment 2

Arve Nygård

1. Explain why the Scene 1,2,3 and 4 are displayed wrong after the user enables scene 5

OpenGL is sort of a state machine. When scene 5 is enabled, the program enables the vertex and fragment shaders. This results in all coordinates being mapped from `model space` to `screen space`. The teapot, for instance, has size 20, resulting in coordinates way outside of screenspace.

I fixed this by calling `glUseProgram(0)` in scene 1, 2, 3 and 4.

2. Report how you changed `RenderScene5`.

I modified the raw vertex data to move one of the points to `-1, 1, 0`. This could also have been done in the vertex shader, using the variable `gl_VertexID`. To change the color, i edited the fragment shader.

3. Report on how you created RenderScene6

Instead of using `glDrawArrays()`, i used `glDrawArraysInstanced()`. This lets you efficiently render the same mesh several times. I also created a new shader which was loaded with Scene 6, and made use of the built-in variable `gl_InstanceID`. The shader uses instanceId to offset each vertex' position by a small amount.

```
#version 330 core

layout(location = 0) in vec3 vertexPosition_modelspace;
void main(){
        gl_Position.xyz = vertexPosition_modelspace * 0.3 + gl_InstanceID * 0.2;
        gl_position.w = 1.0;
}
```