Stevens Institute of Technology
CS 537 – Interactive Computer Graphics
Dr. Chloe LeGendre

Assignment 1

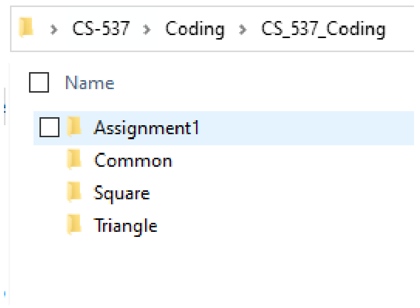
Date posted to Canvas: Feb 14, 2021
Due date: Feb 28, 2021; 11:59 PM

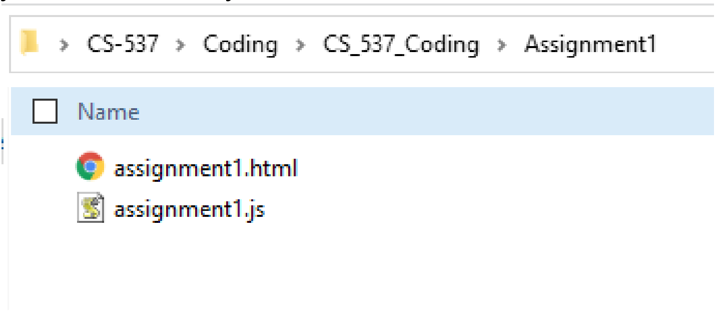
Overview

This assignment is meant to serve as the “hello world” of WebGL programs. You’ll learn about how to define geometry and colors, send data to the GPU, and adjust vertex and shader parameters to change the appearance of an object to be rendered to the screen. You’ll also get to learn about the interplay between the .html and .js files that we’ll use in this course to build WebGL programs. The content of this assignment will be covered predominantly in the readings and notes of Week 02 and Week 03. As such, it will be due at the end of Week 04. **This is an individual assignment and you should not work in a group or share code with your peers.** Discussion with peers is fine, but it should be higher-level concepts rather than code snippets.

Starter Code

As with all assignments in this course, you will be given “starter code” that compiles and runs in a browser. This code requires the “Common” folder that you were asked to download as a part of the exercises for Week 02. You should copy the provided two starter code files “assignment1.html” and “assignment1.js” into one folder labeled Assignment1. You should place this Assignment1 folder in the same directory as the “Common,” “Square,” and “Triangle” folders. So, your final directory structure should look like this:





The starter code will render a green or red triangle to the canvas. You have been provided with a set of menu buttons in the HTML and JavaScript files that will allow you to switch between different rendering modes, which you will be asked to complete for this assignment.

How to turn in your Assignment:

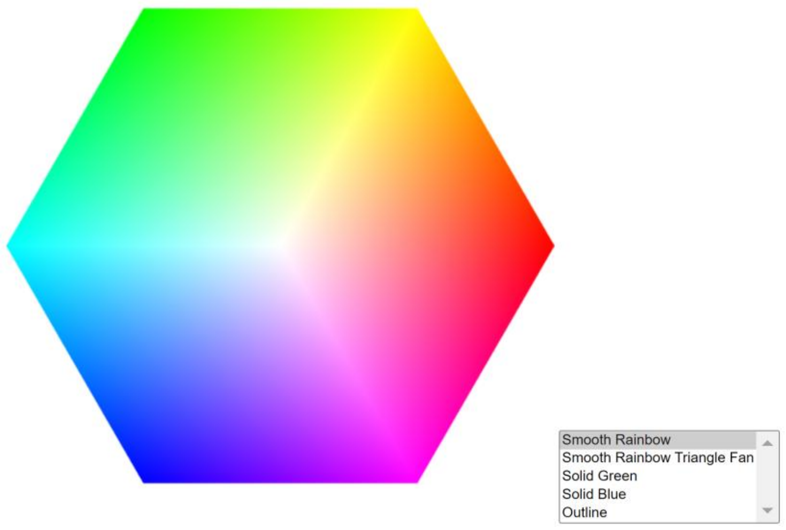
Upload a compressed .zip file to Canvas named as follows: Assignment1_yourusername.zip. The zip archive should contain your Assignment1 folder. This naming system helps us make sure the right assignment is paired with the right individual.

Goals

Your goal for this assignment is to adjust the JavaScript file and the vertex shader in the HTML file to generate the following appearances for each of the rendering modes of the drop-down menu. When you click on the dropdown menu item, the appearances should match the following renderings.

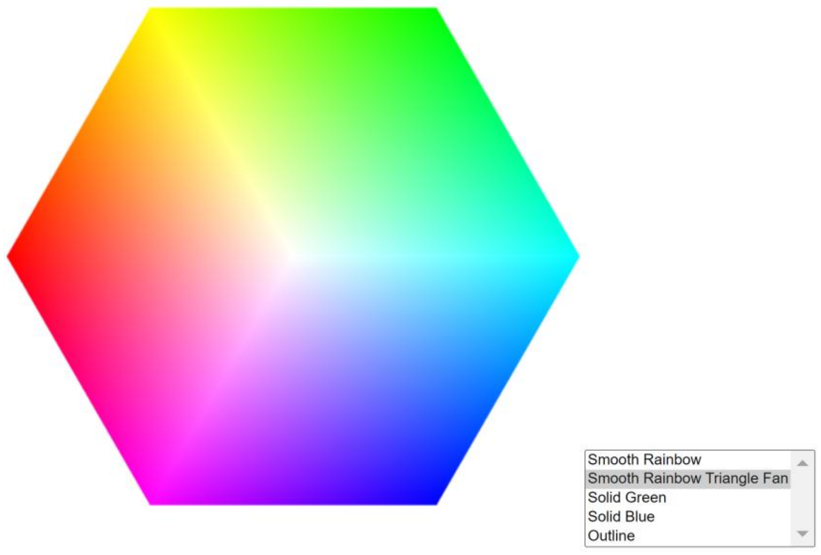
1. Render mode "Smooth Rainbow"

Goal: render a flat *hexagon* with differently colored vertices, using gl.TRIANGLES. Hint: the fragment shader will interpolate the vertex colors to produce a rainbow if the vertex colors are correctly specified.

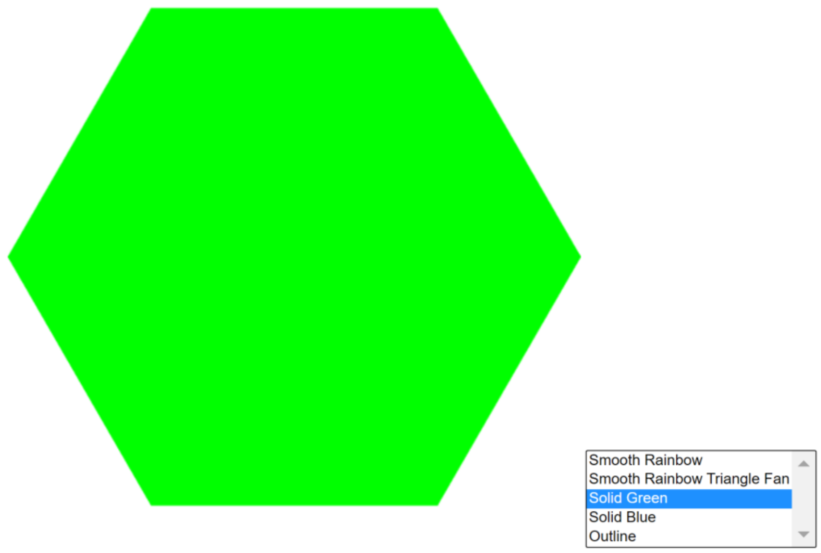


2. Render mode "Smooth Rainbow Triangle Fan"

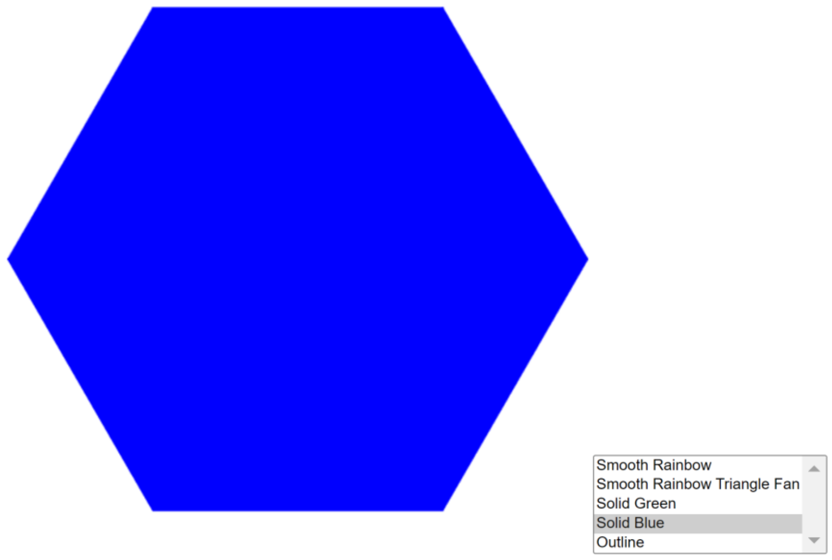
Goal: Render a hexagon with differently colored vertices, with different colors assigned to each vertex than the first mode. For this second mode, you must use gl.TRIANGLE_FAN instead of gl.TRIANGLES. This is to get you familiar with the different representations that can be rendered and the underlying data required for the shaders to render them.



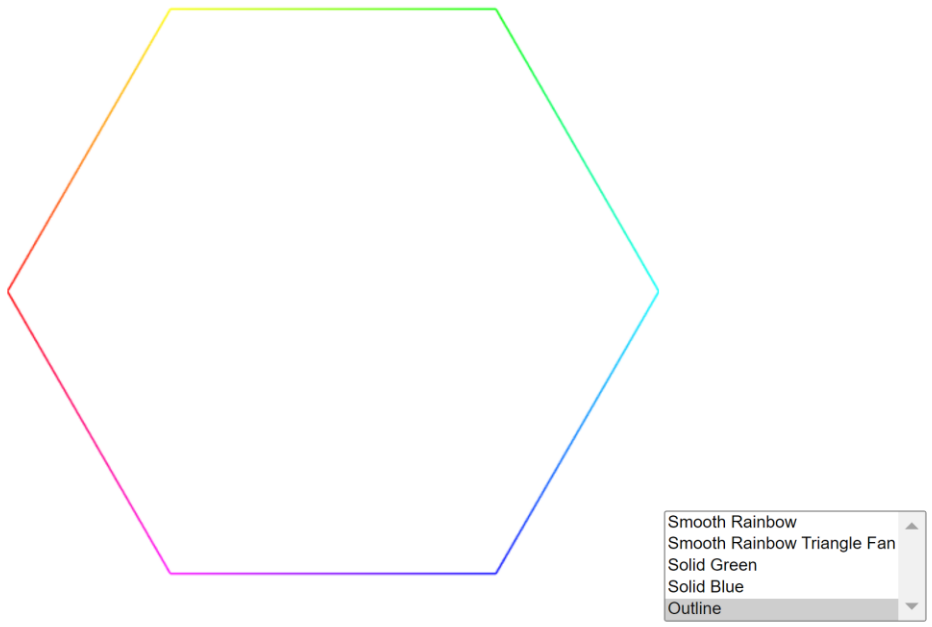
3. Render mode “Solid Green”
Goal: Render a solid green hexagon.



4. Render mode “Solid Blue”
Goal: Render a solid blue hexagon.



5. Render mode “Outline”
Goal: Render the outline of the hexagon only. For full credit, the interior portion of the hexagon must not contain any lines and should look like the below image. The vertices should be rainbow colored again.



How-To Guide:

There are several functions that we suggest you edit, and these are all marked with //TODO comments in the JavaScript file. You also may need to edit the vertex shader in the .html file. However, you should note that there are several ways to achieve these visual results. For instance, color values for single or solid-colored objects can be passed to the GPU in an array, set as uniforms, or set to a fixed value in the shader. Any of these approaches is fine for this assignment, so long as you match the above images with each render mode. The implementation is left up to you. The one exception is that you must use gl.TRIANGLES for render mode 1 and gl.TRIANGLE_FAN for render mode 2. This means that you do not necessarily need to follow the //TODO comments; they are merely included as a suggested place to begin.

Grading:

Smooth Rainbow Mode	**28**
Defines new vertices to render a hexagon (instead of triangle in starter code)	6
Defines vertex colors to achieve rainbow coloring for Smooth Rainbow Mode	6
Uses gl.TRIANGLES rendering mode	6
Rendering mode connected to appropriate dropdown menu	3
Visually correct appearance	7
Smooth Rainbow Triangle Fan Mode	**28**
Defines new vertices to render a hexagon in Triangle Fan render mode	6
Defines vertex colors to achieve rainbow coloring in Triangle Fan render mode	6
Uses gl.TRIANGLE_FAN rendering mode	6
Rendering mode connected to appropriate dropdown menu	3
Visually correct appearance	7
Solid Modes (Green/Blue)	**16**
Defines new vertex colors or adjusts shader to achieve solid appearance (half credit if only green or only blue is working)	6
Rendering modes connected to appropriate dropdown menu (half credit if only green or only blue is working)	3
Visually correct appearance (half credit if only green or only blue is working)	7
Outline Mode	**28**
Defines new vertices to render the empty hexagon instead of filled	6
Defines vertex colors to achieve rainbow coloring for Outline mode	6
Renders outline appearance instead of filled	6
Rendering mode connected to appropriate dropdown menu	3
Visually correct appearance	7
TOTAL	**100**

Partial credit may be given if some parts of code are completed but the visual result is incorrect, at the graders' discretion.