CMP_SC 4610 - Computer Graphics

Ye Duan

Samuel Bishop

01/26/21

a. Objectives

Run the HTML files associated with the assignment and look at the HTML and JavaScript source code in order to understand how the web graphics are being drawn. Running these programs will allow students to become familiar with common computer graphics conventions in WebGL. Within this assignment students should learn how to render basic objects created in JavaScript, understand the concept of vertices, understand why composition of triangles is important in modern graphics, and understand how to use HTML code to format the JavaScript objects.

The List of Tasks to be Completed is Included Below:

Part A: Understanding Events (3 points)

Download and run the program Cad1 (index.html)

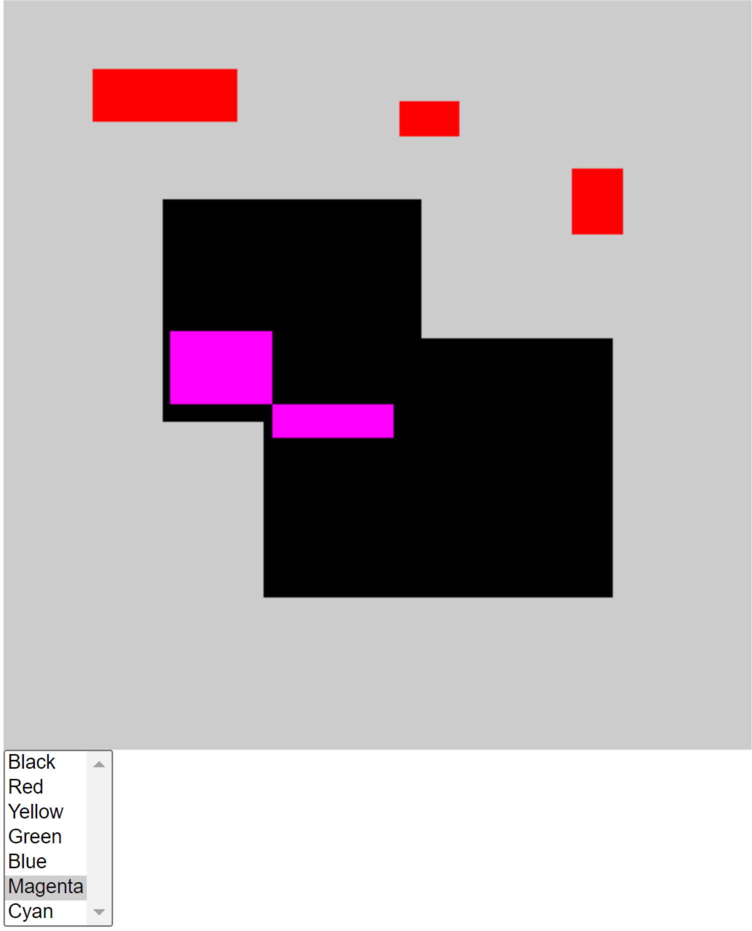

Figure 1. The graphical result of executing Cad1's index.html file

Download and run the program Cad2 (index.html)

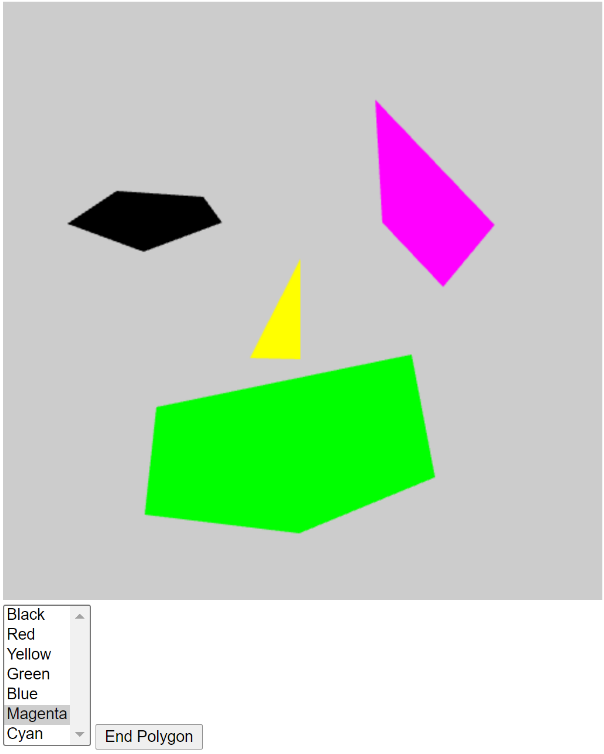

Figure 2. The graphical result of executing Cad2's index.html file

Part B: Implementing Mouse and Keyboard Movements (7 points)

Download the program DrawCube that will Render a 3D cube in WebGL. Implement the following:

1. Press any key to make the cube a random color

   To complete this part of assignment an uniform 4 vector variable was added within the html code (index.html) and linked to the JavaScript code (index.js) through WebGL's getUniformLocation() function and assigned to the colorLoc variable within the JS file.

   Next a keydown listener was added to the window object within the init function of the window. If the code attribute of the event object was equal to "space" the color was made white by assigning the colorLoc variable to zeros with the gl.uniform4f() function. Otherwise that same function was used to assign random values from 0-1 to those vector values using Math.random().

2. Press the spacebar key to return the cube back to white

   As mentioned above:

   “Next a keydown listener was added to the window object within the init function of the window. If the code attribute of the event object was equal to “space” the color was made white by assigning the colorLoc variable to zeros with the gl.uniform4f() function. Otherwise that same function was used to assign random values from 0-1 to those vector values using Math.random().”

3. Click on the canvas and change the color of the cube based on the click position
   a. The RGBA value of the cube will be (x, y, 1, 1) where
   b. x will go smoothly from 0 to 1 based on the horizontal click position, so it will be 0 if you click on the left, 0.5 in the middle, and 1 on the right.
   c. y will do the same, but vertically, so it will be 0 if you click on the bottom, 0.5 in the middle, and 1 on the top.

      When correctly implemented, the cube will be blue when you click in the bottom left, magenta when you click in the bottom right, white in the top right, and cyan in the top left.

In order to accomplish the above description a series of actions were taken. Firstly an event listener was added to the canvas that looks for mousedown events. Using the offsetX and offsetY attributes from the event object two values (red and green) were determined. Then the gl.uniform4f() function was used to assign the color of the cube to those red and green values along with a 1 value for blue and a 1 value for alpha.

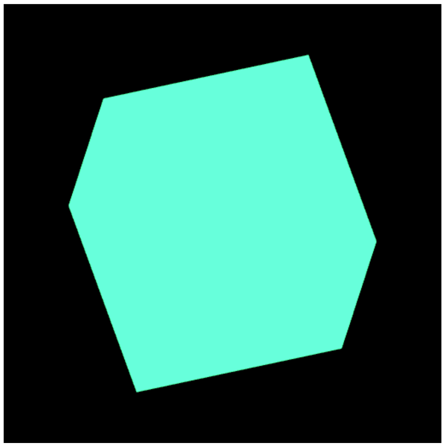

Figure 3. The graphical result of executing the DrawCube index.html file

b. Methods

Methods used to complete tasks are listed task-by-task below:

Number	Method(s)	Parameter(s)	Explanation
1	gl.getUniformLocation()	Program, variable name	Gets the address of the uniform variable declared in the html file.
2	obj.addEventListener()	Listener, function(event)	Allows for creation of event object upon certain listener action
3	gl.uniform4f()	Uniform variable, value1, value2, value3, value4	Assigns float values to a uniform vec4 variablef
4	Math.random()	N/A	Generates a random floating value from 0-1

c. Encountered Issues and Solutions

Few issues were encountered during the programming of this assignment. After Callum's explanation it was a just matter of implementing. However a few minor issues will be mentioned below:

1. I did not realize the uColor vec4 variable in the index.html file was supposed to be declared with the prefix uniform instead of out. This caused confusion for about 5 minutes before figuring it out.
2. Using the event object to grab information about the keys pressed and the offset locations was slightly confusing at first. But later became intuitive.
3. The concept of binding the memory location of the uColor variable in the index.html file to a variable in index.js was slightly confusing. But it was easy to figure out how to use this location variable in the gl.uniform4f() function.

d. Youtube Video Demo

https://youtu.be/k0fXMdnskbU

References

Angel, Edward, and Dave Shreiner. *Interactive Computer Graphics*. 8th ed., Pearson, 2020.