Kathmandu University

School of Engineering

Department of Computer Science and Engineering (DoCSE)

Dhulikhel, Kavre

Computer Graphics - Lab 1

Submitted to:

Mr. Dhiraj Shrestha

Department of Computer Science and Engineering

Submitted By:
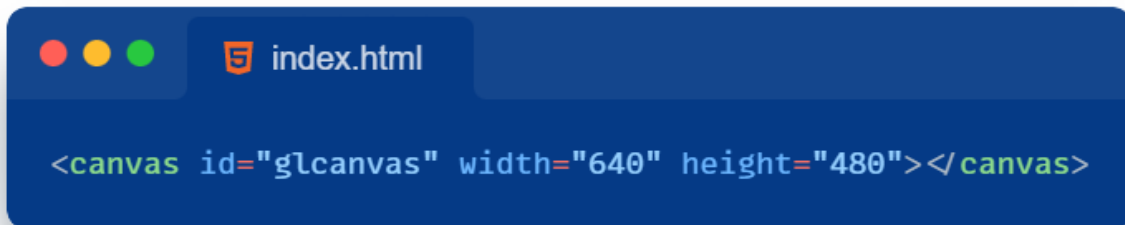
Kamal Pd Neupane (37)

Date of Submission: 2023/03/29

Mention the name of Programming language and Graphics Library you are using this semester for performing your Computer Graphics Lab and Project.

- I have used Javascript as the programming language and WebGL as the graphical library.
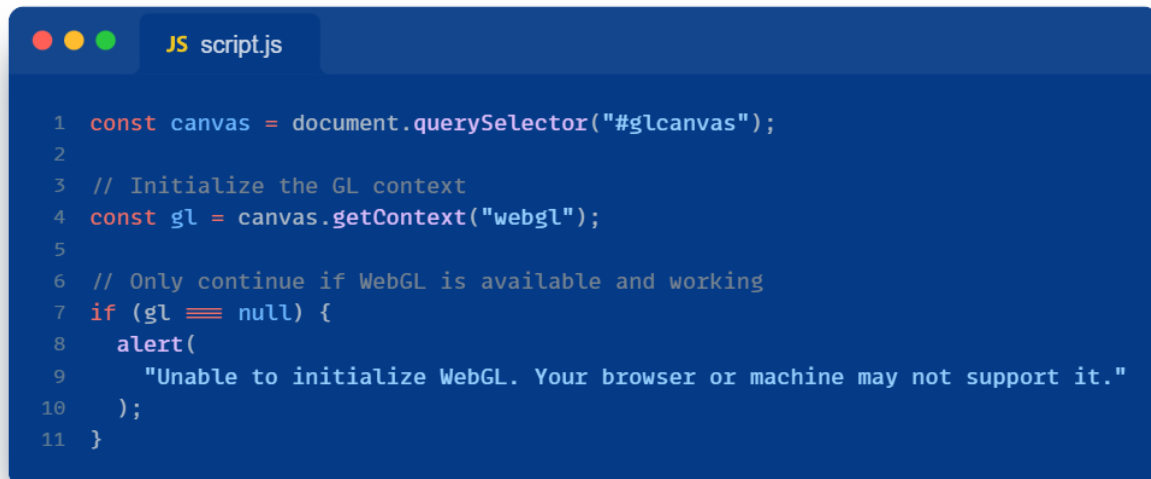
Write the code snippets for setting graphics environment in your chosen graphics library and display the resolution of your display system through functions/classes provided by your graphics library

Setting the graphical environment:

```html
index.html

<canvas id="glcanvas" width="640" height="480"></canvas>
```

```javascript
JS script.js

1  const canvas = document.querySelector("#glcanvas");
2
3  // Initialize the GL context
4  const gl = canvas.getContext("webgl");
5
6  // Only continue if WebGL is available and working
7  if (gl === null) {
8    alert(
9      "Unable to initialize WebGL. Your browser or machine may not support it."
10   );
11 }
```

Display the resolution of the system:

```js
document.querySelector(".resolution").innerHTML =
  "Your screen resolution is: " +
  (window.screen.width * window.devicePixelRatio).toFixed() +
  "x" +
  (window.screen.height * window.devicePixelRatio).toFixed();
```

Output:

Source Code : Click Here
Your screen resolution is: 1920×1080

Get Familiar with the coordinate system and Draw a flag of Nepal using the chosen Graphics geometrical functions/ classes provided by the your chosen graphics library and also color the flag accordingly.

- Source code Link: Click here

- Site hosted at: https://flag-of-nepal.vercel.app/

Following contain the screenshot of the source code.

```javascript
function createVertex(radius, xo, yo) {
  let temp = [];
  for (i = 0; i < 360; i += 15) {
    temp.push(...[xo, yo, 0]);
    x = radius * Math.cos((Math.PI / 180) * i) + xo;
    y = radius * Math.sin((Math.PI / 180) * i) + yo;
    temp.push(...[x, y, 0]);
    x = radius * Math.cos((Math.PI / 180) * (i + 15)) + xo;
    y = radius * Math.sin((Math.PI / 180) * (i + 15)) + yo;
    temp.push(...[x, y, 0]);
  }

  return temp;
}

function createHalfVertex(radius, xo, yo) {
  let temp = [];
  for (i = 175; i < 360; i += 15) {
    temp.push(...[xo, yo, 0]);
    x = radius * Math.cos((Math.PI / 180) * i) + xo;
    y = radius * Math.sin((Math.PI / 180) * i) + yo;
    temp.push(...[x, y, 0]);
    x = radius * Math.cos((Math.PI / 180) * (i + 15)) + xo;
    y = radius * Math.sin((Math.PI / 180) * (i + 15)) + yo;
    temp.push(...[x, y, 0]);
  }

  return temp;
}

function createSpikeData(radius, xo, yo) {
  let temp = [];
  midWidth = radius / 8;
  for (i = -15; i < 345; i += 30) {
    x = radius * Math.cos((Math.PI / 180) * i) + xo;
    y = radius * Math.sin((Math.PI / 180) * i) + yo;
    temp.push(...[x, y, 0]);
    x = radius * Math.cos((Math.PI / 180) * (i + 30)) + xo;
    y = radius * Math.sin((Math.PI / 180) * (i + 30)) + yo;
    temp.push(...[x, y, 0]);
    if (i <= 90) {
      x = radius * Math.cos((Math.PI / 180) * (i + 15)) + xo + midWidth;
      y = radius * Math.sin((Math.PI / 180) * (i + 15)) + yo + midWidth;
      temp.push(...[x, y, 0]);
    } else if (i <= 180) {
      x = radius * Math.cos((Math.PI / 180) * (i + 15)) + xo - midWidth;
      y = radius * Math.sin((Math.PI / 180) * (i + 15)) + yo + midWidth;
      temp.push(...[x, y, 0]);
    } else if (i <= 270) {
      x = radius * Math.cos((Math.PI / 180) * (i + 15)) + xo - midWidth;
      y = radius * Math.sin((Math.PI / 180) * (i + 15)) + yo - midWidth;
      temp.push(...[x, y, 0]);
    } else if (i <= 360) {
      x = radius * Math.cos((Math.PI / 180) * (i + 15)) + xo + midWidth;
      y = radius * Math.sin((Math.PI / 180) * (i + 15)) + yo - midWidth;
      temp.push(...[x, y, 0]);
    }
  }
  return temp;
}
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flag of Nepal</title>
    <link rel="stylesheet" href="Style/style.css">
</head>
<body>
    <div class="my-info">
        Kamal Neupane, CE, 37
    </div>
    <canvas id="glcanvas" width="640" height="480"></canvas>
    <img src="Image/nepal-flag.png" alt="flag" height="480" />
    <div style="margin-top: 50px;">Source Code :
        <a href="https://github.com/kamal0620040/COMP-342-Lab/tree/main/Lab%201/Flag%20of%20Nepal">
            Click Here
        </a>
    </div>
    <div class="resolution"></div>
    <script src="JS/utils.js"></script>
    <script src="JS/script.js"></script>
</body>
</html>
```
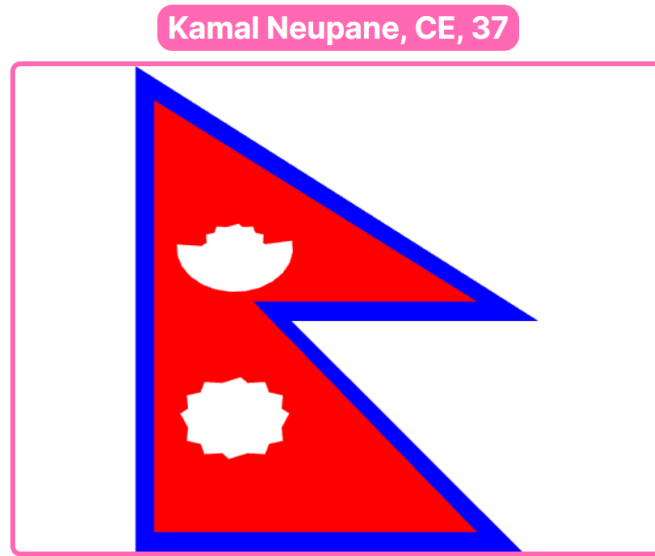
```css
@import url('https://fonts.googleapis.com/css2?family=
Inter:wght@400;500;600;700&display=swap');

body{
    font-family: 'Inter', sans-serif;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;

    height: 90vh;
}

body canvas{
    border: 5px solid hotpink;
    border-radius: 10px;

    /* opacity: 70%; */
    z-index: 10;
}

img{
    position: absolute;

    opacity: 0%;
}

.my-info{
    font-size: 30px;
    background-color: hotpink;
    color: white;
    border-radius: 15px;
    padding: 5px 10px;
    font-weight: bold;

    margin-bottom: 10px;
}

div a{
    /* text-decoration: none; */
    color: black;
}
```

Output:

Conclusion:

I was able to properly draw the flag of Nepal by using the geometrical functions and classes offered by webgl. By doing this lab, I was able to gain a deeper understanding of how webgl actually works.