```
var vertices = [

        //-----------K----------------------
        vec2(-.6, 0),
        vec2(-.55, 0),
        vec2(-.2, -.6),

        vec2(-.2, -.6),
        vec2(-.25, -.6),
        vec2(-.6, 0),

        vec2(-.6, 0),
        vec2(-.55, 0),
        vec2(-.2, .6),

        vec2(-.25, .6),
        vec2(-.6, 0),
        vec2(-.2, .6),

        vec2(-.6, -.6),
        vec2(-.55, -.6),
        vec2(-.55, .6),

        vec2(-.6, -.6),
        vec2(-.55, .6),
        vec2(-.6, .6),
        //-------------A----------------
        vec2(.1, -.6),
        vec2(.15, -.6),
        vec2(.5, .6),

        vec2(.5, .6),
        vec2(.45, .6),
        vec2(.1, -.6),

        vec2(.5, .6),
        vec2(.45, .6),
        vec2(.75, -0.6),

        vec2(.75, -0.6),
        vec2(.8, -0.6),
        vec2(.5, .6),

        vec2(.23, -.23),
        vec2(.23, -.18),
        vec2(.67, -.23),

        vec2(.67, -.23),
        vec2(.23, -.18),
        vec2(.67, -.18)
```

DRAWING LETTERS:

I use many triangles for drawing my name's and surname's first letter. Save it to vertices array and send to bufferData.

```
var bufferId = gl.createBuffer();
gl.bindBuffer( gl.ARRAY_BUFFER, bufferId );
gl.bufferData( gl.ARRAY_BUFFER,flatten(vertices), gl.STATIC_DRAW );
```

COLORING LETTERS:

```
<select id="mymenu" size="3">
  <option value="0">Red</option>
  <option value="1">Green</option>
  <option value="2">Blue</option>
</select>
```

I create a option menu which has red,green and blue options in html.

```
}

</script>
<script id="fragment-shader" type="x-shader/x-fragment">
   precision mediump float;
   uniform vec4 color;

   void main() {

      gl_FragColor = color;

     }

</script>
```

Write a function which creates color vector and send it to void main() in html.

```
var positionLocation = gl.getAttribLocation (program, "a_position");
gl.vertexAttribPointer( positionLocation, 4, gl.FLOAT, false, 0, 0 );
gl.enableVertexAttribArray( positionLocation );

colorLoc = gl.getUniformLocation(program, "color");


var m = document.getElementById("mymenu");
m.addEventListener("click", function() {
  {
    var x = document.getElementById("mymenu").value;

   if(x==0){
     gl.uniform4f (colorLoc,1.0,0.0,0.0,1.0);//red
  }
  if(x==1){
     gl.uniform4f (colorLoc,0.0, 1.0, 0.0, 1.0);//green
   }
  if(x==2){
     gl.uniform4f (colorLoc, 0.0, 0.0, 1.0, 1.0);//blue
   }

 }});
```

ROTATION LETTERS:

I created buttons which aims to Start Rotation,Stop Rotation and Change Rotation Direction.

```
<button id="StartRotationButton">Start Rotation</button>
<button id="StopRotationButton">Stop Rotation</button>
<button id = "DirectionButton">
 Change Rotation Direction
</button>
```

delay for speed,isDirClockwise for rotation direction,isRunning for stop and start.

```
var gl;
var theta;
var thetaLoc;
var isDirClockwise = false;
var delay = 50;
var isRunning=true;
var color;

function buttonPressedFunc(){
  isDirClockwise=!isDirClockwise;
}
function startPressedFunc(){
  isRunning=true;
}
function stopPressedFunc(){
  isRunning=false;
}


  var program = initShaders(gl, "vertex-shader", "fragment-shader")
  gl.useProgram( program );

  var myButton = document.getElementById("DirectionButton");
  myButton.addEventListener("click", buttonPressedFunc);

  var startButton = document.getElementById("StartRotationButton");
  startButton.addEventListener("click",startPressedFunc);

  var stopButton = document.getElementById("StopRotationButton");
  stopButton.addEventListener("click",stopPressedFunc);
```

This code sends theta and vPosition to void main() in html.

```
var vPosition = gl.getAttribLocation( program, "vPosition" );
gl.vertexAttribPointer( vPosition, 2, gl.FLOAT, false, 0, 0 );
gl.enableVertexAttribArray( vPosition );

thetaLoc = gl.getUniformLocation(program, "theta");

theta = 0;
gl.uniform1f(thetaLoc, theta);
```

This code generates rotation.

```
<script id="vertex-shader" type="x-shader/x-vertex">
   attribute vec4 vPosition;
   uniform float theta;


   void main() {
  gl_Position.x = cos(theta) * vPosition.x - sin(theta) * vPosition.y;
  gl_Position.y = sin(theta) * vPosition.x + cos(theta) * vPosition.y;
  gl_Position.z = 0.0;
  gl_Position.w = 1.0;


}

</script>
```

Delay determines rotation speed,isRunning determines start or stop program,isDirClockwise determines rotation way.

```
68   function render(){
69     setTimeout(function() {
70       // Clear the color buffer with specified clear color
71       gl.clear(gl.COLOR_BUFFER_BIT);
72       if(isRunning){
73         theta += (isDirClockwise ? -0.1 : 0.1);}
74       gl.uniform1f(thetaLoc, theta);
75       gl.drawArrays(gl.TRIANGLES, 0, 36);
76       render();
77       }, delay);
```

ROTATION SPEED:

I created slider which aims to slow down or speed up rotation.

```
</script>
<p>Speed Slider (+ -)</p>
<div class="slidecontainer">
<input type="range" min="0" max="4" value="2" class="slider" id="myRange">
</div>
```

Decrease or increase delay for chancing the speed.

```
var slider = document.getElementById("myRange");
slider.addEventListener("click", function(){
  var x = document.getElementById("myRange").value;

  if(x==4){
      delay = 150;
  }
  if(x==3){
      delay = 100;
  }
  if(x==2){
    delay = 50;
  }
  if(x==1){
    delay = 25;
  }
  if(x==0){
    delay = 10;
  }
})
```

```
document.addEventListener('keydown', (event) => {

  var code = event.code;
  if(code==="NumpadAdd"&&delay>=0){
    delay-=10;
  }
  if(code==="NumpadSubtract"&&delay<=100){
    delay+=10;
  }
  // Alert the key name and key code on keydown
  console.log(delay);
}, false);
```

Speed chancing with keyboard.