02561 COMPUTER GRAPHICS
REPORT FOR FINAL PROJECT
INTERACTIVE PLANE MODEL

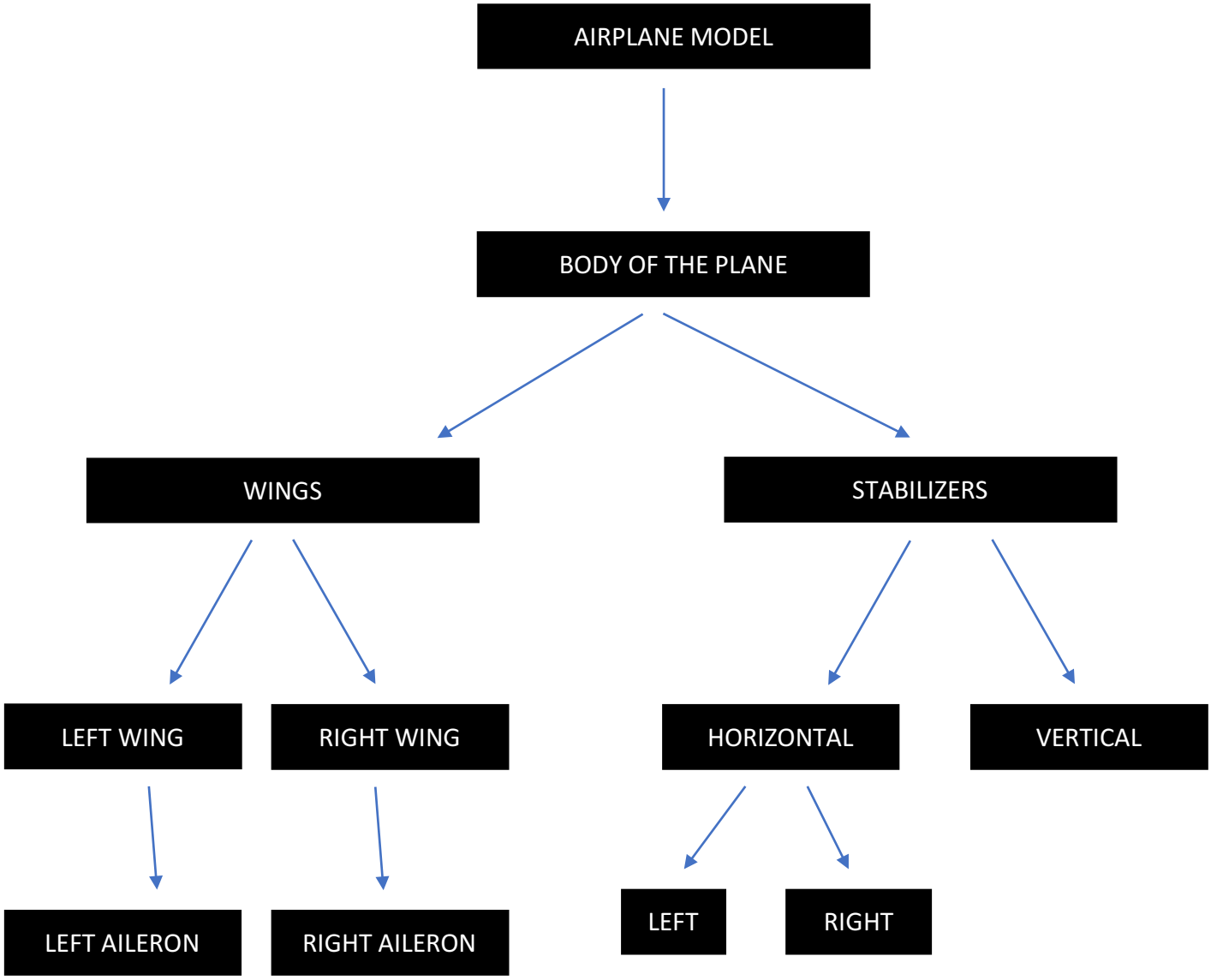
REPORT AND PROJECT PREPARED BY
S172674 HASAN EROGLU

The Main Objective

The main objective of project is coding interactive airplane model in WebGL.

Creating Interactive Airplane Model

Airplane is an hierarchical model. It has one body, two wings attached to this body, one aileron attached to each wing, two horizontal stabilizers with elevators and one vertical stabilizer with rudder.



I started to the project by initializing vertices for a box:

```
var x = 1;
var vertices = new Float32Array([
  -x, -x, -x,  x, -x, -x,  -x,  x, -x,  x,  x, -x, //front v0-v1-v2-v3
   x, -x, -x,  x, -x,  x,   x,  x,  x,  x,  x, -x, //right v1-v4-v5-v3
  -x, -x, -x, -x, -x,  x,  -x,  x,  x, -x,  x, -x, //left  v0-v6-v7-v2
  -x,  x, -x, -x,  x,  x,   x,  x,  x,  x,  x, -x, //top   v2-v7-v5-v3
  -x, -x,  x,  x, -x,  x,   x, -x, -x, -x, -x, -x, //bot   v6-v4-v1-v0
   x,  x,  x, -x,  x,  x,  -x, -x,  x,  x, -x,  x  //back  v5-v7-v6-v4
]);
```

And normals:

```
var normals = new Float32Array([
   0.0,  0.0,  1.0,   0.0,  0.0,  1.0,   0.0,  0.0,  1.0,   0.0,  0.0,  1.0, //front v0-v1-v2-v3
   1.0,  0.0,  0.0,   1.0,  0.0,  0.0,   1.0,  0.0,  0.0,   1.0,  0.0,  0.0, //right v1-v4-v5-v3
  -1.0,  0.0,  0.0,  -1.0,  0.0,  0.0,  -1.0,  0.0,  0.0,  -1.0,  0.0,  0.0, //left  v0-v6-v7-
   0.0,  1.0,  0.0,   0.0,  1.0,  0.0,   0.0,  1.0,  0.0,   0.0,  1.0,  0.0, //top   v2-v7-v5-v3
   0.0, -1.0,  0.0,   0.0, -1.0,  0.0,   0.0, -1.0,  0.0,   0.0, -1.0,  0.0, //bot   v6-v4-v1-v0
   0.0,  0.0, -1.0,   0.0,  0.0, -1.0,   0.0,  0.0, -1.0,   0.0,  0.0, -1.0, //back  v5-v7-v6-v4
]);
```

And indices to draw my box:

```
var indices = new Uint8Array([
    0, 1, 2,   1, 2, 3,     // front
    4, 5, 6,   4, 6, 7,     // right
    8, 9,10,   8,10,11,     // left
   12,13,14,  12,14,15,     // top
   16,17,18,  16,18,19,     // bot
   20,21,22,  20,22,23      // back
]);
```

Afterwards I initializied my shaders and started to create buffers for my box:

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

var vBuffer = gl.createBuffer();
gl.bindBuffer( gl.ARRAY_BUFFER, vBuffer );
gl.bufferData( gl.ARRAY_BUFFER, vertices, gl.STATIC_DRAW );

var vPosition = gl.getAttribLocation( program, "vPosition" );
gl.vertexAttribPointer( vPosition, 3, gl.FLOAT, false, 0, 0 );
gl.enableVertexAttribArray( vPosition );

var nBuffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, nBuffer);
gl.bufferData(gl.ARRAY_BUFFER, normals, gl.STATIC_DRAW);

var vNormal = gl.getAttribLocation( program, "vNormal" );
gl.vertexAttribPointer( vNormal, 3, gl.FLOAT, false, 0, 0 );
gl.enableVertexAttribArray( vNormal );

var indexBuffer = gl.createBuffer();
gl.bindBuffer(gl.ELEMENT_ARRAY_BUFFER, indexBuffer);
gl.bufferData(gl.ELEMENT_ARRAY_BUFFER, indices, gl.STATIC_DRAW);
```

I have three matrices to adjust my scene and these are viewMatrix , projectionMatrix and modelMatrix which is used for all kinds of transformations of my airplane model.

```
viewMatrixLoc = gl.getUniformLocation(program, "viewMatrix");
projectionMatrixLoc = gl.getUniformLocation(program, "projectionMatrix");
modelMatrixLoc = gl.getUniformLocation(program, "modelMatrix");
```

viewMatrix and projectionMatrix:

```
eye = vec3(-30.0, 30.0, 0.0);
view = lookAt(eye, at, up);
projection = perspective(fovy, aspect, near, far);
gl.uniformMatrix4fv(viewMatrixLoc, false, flatten(view));
gl.uniformMatrix4fv(projectionMatrixLoc, false, flatten(projection));
```

Usage of modelMatrix

Render function is for drawing whole airplane model. It starts with drawing body first(following hiererchical table).

```
//body
modelMatrix = translate(0.0,0.0,0.0);
modelMatrix = mult(modelMatrix, rotateX(rollAngle));
modelMatrix = mult(modelMatrix, rotateY(rotationAngle));
modelMatrix = mult(modelMatrix, rotateZ(diveAngle));
drawBox(10.0,1.0,1.0,modelMatrix);
```

Another function called drawBox is used for drawing boxes. Scaling is done in this function otherwise rotating boxes in a proper way is not possible.
drawBox function:

```
function drawBox(width, height, depth, m){
  pushMatrix(m);
  m = mult(m, scalem(width,height,depth));
  nMatrix = normalMatrix(m, false);
  gl.uniformMatrix4fv(viewMatrixLoc, false, flatten(view));
  gl.uniformMatrix4fv(projectionMatrixLoc, false, flatten(projection));
  gl.uniformMatrix4fv(modelMatrixLoc,false, flatten(m));
  gl.uniformMatrix4fv(normalMatrixLoc,false,flatten(nMatrix));

  gl.drawElements(gl.TRIANGLES,36,gl.UNSIGNED_BYTE,0);
  popMatrix(m);
}
```

And I have another two functions called pushMatrix and popMatrix. Since I am following hierarchical model I need to save my modelMatrix so I can use it again. For example, after drawing body of my plane I need to save my modelMatrix because I have two wings attached to the body.

Interaction with Plane

Left and right arrow keys used for rolling plane also ailerons react to this.
Up and down arrow keys used for elevating plane also elevators react to this.
A and D keys used for rotating plane side-to-side also rudder react to this.