**INDEX**

[**EXPERIMENT 1: About me** 2](#_Toc499567365)

[**EXPERIMENT 2: SIGN UP FORM** 4](#_Toc499567366)

[**EXPERIMENT 3: RESUME** 7](#_Toc499567367)

[**EXPERIMENT 4: SMILE TRANSLATION** 11](#_Toc499567368)

[**EXPERIMTN 5: SMILE BIG AND SMALL** 13](#_Toc499567369)

[**EXPERIMENT 6: MOVE AN OBJECT** 15](#_Toc499567370)

[**EXPERIMENT 7: TWO OBJECT COLLISION** 16](#_Toc499567371)

[**EXPERIMENT 8: MOVING CAR** 18](#_Toc499567372)

[**EXPERIMENT 9: Line Drawing Algorithm** 20](#_Toc499567373)

[**EXPERIMENT 9: WEBGL HUT** 23](#_Toc499567374)

[**.JS File** 23](#_Toc499567375)

[**HTML Code** 32](#_Toc499567376)

# **EXPERIMENT 1: About me**

<!DOCTYPE html>

<html>

<head>

<title>My Details</title>

</head>

<body bgcolor="brown">

<h1 style="font-size: 38;text-decoration: underline;text-align: center;display: block;color: AQUA;">My Details</h1>

<fieldset>

<legend>Personal Details</legend>

First Name: Ashutosh Suman <br>

Last Name: Suman<br><br>

Email-Id: asutoshsuman24@gmail.com<br><br>

Contact No.: 7739043423<br><br>

BirthDate: 30/01/1998 ><br><br>

Address: Premnaga<br>

State: Uk<br>

City: Dehradun<br><br>

Father Name: Shivendra Kumar<br>

Mother Name: Punam Kumari<br><br>

School Name: D.A.V Public School<br><br>

College Name: UPES Dheradun<br><br>

</fieldset>

<fieldset>

<br>

<legend>Education Details</legend>

Intermediate: 90%<br>

High school: 86%<br>

Graduation: 8CGPA<br>

</fieldset><br><br>

<fieldset>

<legend>Skill Set</legend>

Skills: Right arm fast bowler<br>

Making logo<br>

Corel Draw<br>

c<br>

Da<br>

</fieldset>

<fieldset>

<legend>About My self</legend>

I am Gonna make<br>

The Rest of My Life<br>

The Best of My Life

</fieldset>

</form>

</body>

</html>

# **EXPERIMENT 2: SIGN UP FORM**

<!DOCTYPE html>

<html>

<head>

<title>Sign Up Form</title>

</head>

<body style="background-image: url(gradient-texture-cubes.jpg);">

<h1 style="font-size: 38;text-decoration: underline;text-align: center;display: block;color: AQUA;">Sign Up Form</h1>

<form action="name.php" method="post" target="\_blank" style="color: black;">

<fieldset>

<legend>Personal Details</legend>

First Name: <input type="text" name="firstname" placeholder="First Name" autofocus>

Middle Name: <input type="text" name="middlename" placeholder="Middle Name">

Last Name: <input type="text" name="lastname" placeholder="Last Name"><br><br>

Email-Id: <input type="email" name="email" size="50" placeholder="@gmail.com" ><br><br>

Contact No.: <input type="number" name="contact" maxlength="10" size="10"><br><br>

BirthDate: <input type="date" name="bdate" max="2005-12-31" min="1990-01-01" ><br><br>

Address: <input type="textarea" name="address" size="50">

State: <input type="text" name="state">

Pincode: <input type="number" name="pcode" maxlength="6" size="6">

City: <input type="text" name="city"><br><br>

Aadhar Card No.: <input type="number" name="aadharcardno." maxlength="16" size="16" ><br><br>

Father Name: <input type="text" name="fathername" size="30" >

Mother Name: <input type="text" name="mothername" size="30"><br><br>

School Name: <input type="text" name="schoolname" size="80"><br><br>

College Name: <input type="text" name="collegename" size="80"><br><br>

</fieldset>

<fieldset>

<br>

<legend>Percentage</legend>

Intermediate: <input type="number" name="intermediate" size="2">

High school: <input type="number" name="highschool" size="2">

Graduation: <input type="number" name="graduation" size="2">

</fieldset><br><br>

Gender: <br>

<input type="radio" name="gender" value="male" checked>Male<br>

<input type="radio" name="gender" value="female">Female<br>

<input type="radio" name="gender" value="transgender">Transgender<br><br>

Username: <input type="text" name="usrname" size="25" >

Password: <input type="password" name="password" size="30" placeholder="letter a-z, A-Z, 0-9 & Special Characters" ><br><br>

<fieldset>

<legend>Program Selction</legend>

Program: <select name="program">

<option value="disabled">Select</option>

<option value="B.Tech">B.Tech</option>

<option value="B.Sc">B.Sc</option>

<option value="B.B.A.">B.B.A.</option>

<option value="B.A.">B.A.</option>

</select>

Branch: <select name="branch">

<option value="disabled">Select</option>

<option value="Aerospace Engineering">Aerospace Engineering</option>

<option value="Applied Petroleum Engineering">Applied Petroleum Engineering</option>

<option value="Computer Science Engineering">Computer Science Engineering</option>

<option value="Civil Engineering">Mechanica Engineering</option>

<option value="Fire And Safety Engineering">Fire And Safety Engineering</option>

<option value="Mechanical Engineering">Mechanical Engineering</option>

<option value="Mechatronics Engineering">Fire And Safety Engineering</option>

</select>

</fieldset>

<br><br>

Upload Photo: <input type="file" name="img" size="6" multiple>

<br><br><br><br><input type="submit" value="Submit">

<input type="reset" value="Reset" onclick="alert('reset completed!!!!!!')">

</form>

</body>

</html>

# **EXPERIMENT 3: RESUME**

<html>

<head>

<link type="text/css" rel="stylesheet" href="resume.css"/>

<title></title>

</head>

<body>

<div id="header">

<p id="name" >Ashutosh Suman <a href="ashutoshsuman24@gmail.com" target="\_blank"><p id="email">ashutoshsuman24@gmail.com</p></a></p>

</div>

<div class="left">

</div>

<div class="right">

<div id="headshot" >

<img src="headshot.jpg" alt="Ashutosh Suman" />

</div>

<center><h3>A young, determined hard and smart working person. I believe in task based roles and complete ownership of work.</h3></center>

<h3>Personal Information</h3>

<p>

<ul>

<li>

A young, determined hard and smart working person. I believe in task based roles and complete ownership of work.

<li>Currently going through B.tech CSE Graphics & Gaming Course at UPES dehradun</li>

<li>Learnt basic Python, JavaScript, HTML, CSS and graphics tools.</li>

<li>An independent, dedicated, efficient person. These attributes are proved through the series of courses I have taken or I am taking as of now independently through online platforms.</li>

<li>Good Communication Skills, Presentation Skills, attitude towards leadership, authorisation and delegation, conflict resolution and negotiation and a very good team worker.</li>

<li><span id="course-name">Languages Known:</span>English, Hindi, Maithili</li>

<li>

<span id="course-name">Hobbies:</span>I love reading Graphics and IT related books / magazines, playing Chess and outdoor games, listening music, surfing Internet, self-learning through e-courses.</li> </ul>

</ul>

</p>

<h3>Technical Skills</h3>

<p>

<ul>

<li>

<span id="course-name">Operating Systems:</span> Windows 2000, Windows XP, Windows Vista, Windows 7, Windows 8, Windows 10, Macintosh Computers (OS X), Linux (Ubuntu)</li>

<li>

<span id="course-name">Application Software:</span> Office 2007-2013; Excel 2003/2007 for Financial Modelling spreadsheets, Photoshop, Corel Draw, Adobe Cloud, Blender, GIMP.</li>

<li>

<span id="course-name">Programming Skills:</span>HTML, CSS, C, C++, JavaScript, learning Python and .net</li></ul>

</p>

<h3>Educational Qualifications</h3>

<table>

<tr id="heading">

<td>Level/Qualification</td>

<td>Board/Universitiy</td>

<td>Percentage / Grades</td>

<td>Year</td>

</tr>

<tr>

<td>S.S.C</td>

<td>C.B.S.E India</td>

<td>93.1%</td>

<td>2013</td>

</tr>

<tr>

<td>H.S.C (Science Stream)</td>

<td>C.B.S.E India</td>

<td>86%</td>

<td>2015</td>

</tr>

<tr>

<td>B.tech (CSE Graphics & Gaming)</td>

<td>UPES (University Of Petroleum and Energy Studies), Dehradun, India</td>

<td>8.2 CGPA</td>

<td>2017</td>

</tr>

</table>

<h3>Favourit Subject</h3>

<p>

<ul>

<li>Mathematics</li>

<li>Computrt Graphics</li>

<li>C and C++ </li></ul>

</p>

<h3>Other Information</h3>

<p>

<ul>

<li>

<span id="course-name">Expected Salary:</span>As per company standards</li>

<li>

<span id="course-name">Area of Interest:</span>Software Development, Programming, Start-ups, Coding, App Development, Technical Support, Equity Research, Business Analysis</li>

<li>

<span id="course-name">Joining Date:</span>Immediate</li></ul>

</p>

<h3>Declaration</h3>

<p>

I hereby declare that the details furnished above are true and correct to the best of my knowledge and belief.</p>

<p> Ashutosh Suman.</p>

</div>

<div id="footer"></div>

</body>

</html>

# **EXPERIMENT 4: SMILE TRANSLATION**

<html>

<head>

<style>

.vert{

position: absolute;

top : 50%;

left : 74%;

transform: rotate(90deg);

width : 400px;

}

.hor{

position: absolute;

top : 94%;

left : 20%;

width : 600px;

}

</style>

</head>

<body onload="initFunc()">

<canvas id="mycanvas" width="1200" height="600" style="border:2px solid #d3d3d3;">

<script>

function initFunc()

{

var canvas = document.getElementById("mycanvas");

var cdx=canvas.getContext("2d");

cdx.clearRect(0, 0, canvas.width, canvas.height);

var X = document.getElementById("h").value;

var Y = document.getElementById("v").value;

X=X-250;

Y=Y-50;

var x = 400+X;

var y = 200+Y;

var radius = 150;

var startAngle = 0 \* Math.PI;

var endAngle = 2 \* Math.PI;

var counterClockwise = false;

cdx.beginPath();

cdx.arc(x, y, radius, startAngle, endAngle, counterClockwise);

cdx.lineWidth = 25;

cdx.strokeStyle = 'red';

cdx.stroke();

cdx.moveTo(325+X,175+Y);

cdx.arc(300+X, 175+Y,25, startAngle, endAngle, counterClockwise);

cdx.moveTo(525+X,175+Y);

//cdx.beginPath();

cdx.arc(500+X, 175+Y,25, startAngle, endAngle, counterClockwise);

cdx.moveTo(460+X,280+Y);

cdx.arc(400+X, 200+Y,100, 0.25\*Math.PI, 0.75\*Math.PI, counterClockwise);

cdx.lineWidth = 10; // line color

cdx.strokeStyle = 'pink';

cdx.stroke();

}

</script>

</canvas>

<input type = "range" id = "v" min = "0" max = "300" value = "20" class = "vert" onchange = "initFunc()">

<input type = "range" id = "h" min = "0" max = "900" value = "20" class="hor" onchange = "initFunc()">

</body>

</html>

# **EXPERIMTN 5: SMILE BIG AND SMALL**

<html>

<head>

</head>

<body onload="animate()">

<canvas id="mycanvas" width="1000" height="500">

<script>

var z=0;

var q=0;

function animate()

{

setInterval (smile,100);

}

function smile()

{

var canvas = document.getElementById("mycanvas");

var cdx=canvas.getContext("2d");

cdx.clearRect(0,0,canvas.width,canvas.height);

if (z>90 && q>6)

{

z=0;

q=0;

}

else

{

z=z+10;

q=q+2;

}

cdx.beginPath();

cdx.arc(400,200,100+z,0,(Math.PI)\*2,false);

cdx.moveTo(380+q,225);

cdx.arc(355,225,25+q,0,(Math.PI)\*2,false);

cdx.moveTo(480+q,225);

cdx.arc(455,225,25+q,0,(Math.PI)\*2,false);

cdx.stroke();

//cdx.moveTo(330+z,190+z);

cdx.beginPath();

cdx.arc(410, 250,100+z, 1.2\*Math.PI, 1.75\*Math.PI,false);

cdx.stroke();

}

</script>

</canvas>

</body>

</html>

# **EXPERIMENT 6: MOVE AN OBJECT**

<!DOCTYPE HTML>

<html>

<head>

<style>

body {

margin: 0px;

padding: 0px;

}

</style>

</head>

<body background="rod.jpg" style=";background-repeat: no-repeat;">

<canvas id="myCanvas" width="1100" height="500"></canvas>

<script>

window.addEventListener('load', function () {

var

img = new Image,

ctx = document.getElementById('myCanvas').getContext('2d');

img.src = 'b.png';

img.addEventListener('load', function () {

var interval = setInterval(function() {

var x = 10, y = 10;

return function () {

ctx.clearRect(0, 0, ctx.canvas.width, ctx.canvas.height);

ctx.drawImage(img, x, y);

x += 5;

if (x > ctx.canvas.width) {

x = 0;

}

};

}(), 1000/40);

}, false);

}, false);

</script>

</body>

</html>

# **EXPERIMENT 7: TWO OBJECT COLLISION**

<html>

<head>

</head>

<body onload="animate()">

<canvas id="myCanvas" width="1320" height="800">

<script>

var z=0;

var q=0;

function animate()

{

setInterval (truck,100);

}

function truck()

{

var canvas = document.getElementById("myCanvas");

var cdx = canvas.getContext("2d");

cdx.clearRect(0,0,canvas.width,canvas.height);

if(z>1200)

{

z=0;

}

else

{

z=z+10;

q = Math.floor((Math.random() \* 1250) + 1);

}

var x = 155+z;

var x1 = 1250;

var y = 250;

var y1 = 50+q;

var score=0;

var object1;

var object2;

cdx.beginPath();

cdx.moveTo(1250,50+q);

object2=cdx.arc(x1, y1,50, 0\*Math.PI, 2\*Math.PI, false);

cdx.lineWidth = 5; // line color

cdx.strokeStyle = 'black';

cdx.stroke();

cdx.beginPath();

cdx.moveTo(130+z,250);

object1=cdx.arc(x, 250,25, 0\*Math.PI, 2\*Math.PI, false);

cdx.lineWidth = 5; // line color

cdx.strokeStyle = 'black';

cdx.stroke();

if (object1.x < object2.x1 && object1.x > object2.x1 &&

object1.y < object2.y1 && object1.y > object2.y1)

{

score = score+5;

}

cdx.font = "30px Arial";

cdx.fillText(score,100,500);

}

</script>

</body>

</html>

# **EXPERIMENT 8: MOVING CAR**

<html>

<head>

</head>

<body onload="animate()">

<canvas id="myCanvas" width="1420" height="800">

<script>

var z=0;

function animate()

{

setInterval (truck,100);

}

function truck()

{

var canvas = document.getElementById("myCanvas");

var cdx = canvas.getContext("2d");

cdx.clearRect(0,0,canvas.width,canvas.height);

if(z>1300)

{

z=0;

}

else

{

z=z+10;

}

var x = 100+z;

var y = 50;

var width = 450;

var height = 250;

cdx.beginPath();

cdx.fillStyle = "#8ED6FF";

cdx.fillRect(x,y,width,height);

cdx.rect(x,y,width,height);

cdx.lineWidth = 15;

cdx.strokeStyle = 'red';

cdx.stroke();

//cdx.beginPath();

cdx.moveTo(565+z,250);

cdx.fillStyle = "#9f6a25";

cdx.fillRect(560+z,155,150,150);

cdx.rect(560+z,155,150,150);

cdx.moveTo(125+z,320);

cdx.arc(155+z, 335,30, 1.15\*Math.PI, 1.9\*Math.PI, false);

cdx.moveTo(480+z,320);

cdx.arc(500+z, 335,30, 1.15\*Math.PI, 1.9\*Math.PI, false);

cdx.moveTo(130+z,350);

cdx.arc(155+z, 350,25, 0\*Math.PI, 2\*Math.PI, false);

cdx.moveTo(475+z,350);

cdx.arc(500+z, 350,25, 0\*Math.PI, 2\*Math.PI, false);

cdx.lineWidth = 5; // line color

cdx.strokeStyle = 'pink';

cdx.stroke();

}

</script>

<input id="slider1" type="range" min="0" max="2000" step="10" />

<input id="slider2" type="range" min="0" max="2000" step="10" />

</body>

</html>

# **EXPERIMENT 9: Line Drawing Algorithm**

<!doctype html>

<head></head>

<body>

<canvas id="canvas" width="800" height="500" style="border: 1px solid black"></canvas>

<script>

var canvas = document.getElementById("canvas");

var context = canvas.getContext("2d");

canvas.addEventListener("mousedown", handleMouseDown);

canvas.addEventListener("mousemove", handleMouseMove);

canvas.addEventListener("mouseup", handleMouseUp);

var p0 = {x: 0, y:0};

var p1 = {x: 0, y:0};

var isdown = 0;

display();

function handleMouseUp(event)

{

isdown = 0;

p1.x = event.clientX - canvas.offsetLeft;

p1.y = event.clientY - canvas.offsetTop;

display();

}

function handleMouseDown(event)

{

isdown = 1;

p0.x = event.clientX - canvas.offsetLeft;

p0.y = event.clientY - canvas.offsetTop;

display();

}

function handleMouseMove(event)

{

if (isdown == 0)

return;

p1.x = event.clientX - canvas.offsetLeft;

p1.y = event.clientY - canvas.offsetTop;

display();

}

function midpoint(p0, p1)

{

var dy = p1.y - p0.y;

var dx = p1.x - p0.x;

var incE = -2\*dy;

var incNE = 2\*dx + incE;

var d = dx - 2\*dy;

var x, y;

y = p0.y;

for (x = p0.x; x <= p1.x; ++x)

{

context.fillRect(x, y, 1, 1);

if (d >= 0)

d += incE;

else

{

d += incNE;

++y;

}

}

}

function display()

{

context.clearRect(0, 0, canvas.width, canvas.height);

var dx = p1.x - p0.x;

var dy = p1.y - p0.y;

if (dx > 0 && dy > 0 && dx >= dy)

midpoint(p0, p1);

context.fillStyle = "#00FF00";

context.strokeStyle = "#FF0000";

context.beginPath();

context.moveTo(p0.x, p0.y);

context.lineTo(p1.x, p1.y);

context.stroke();

}

</script>

</body>

</html>

# **EXPERIMENT 9: WEBGL HUT**

## **.JS File**

var InitDemo = function () {

/\*======= Creating a canvas =========\*/

var canvas = document.getElementById('my\_Canvas');

var gl = canvas.getContext('experimental-webgl');

/\*======= Defining and storing the geometry ======\*/

/\* STEP01: \*/

// Create an empty buffer object

var vertex\_buffer = gl.createBuffer();

/\* STEP02: \*/

// Bind appropriate array buffer to it

gl.bindBuffer(gl.ARRAY\_BUFFER, vertex\_buffer);

/\* STEP03: vertices here is user defined \*/

// Pass the vertex data to the buffer

//gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

/\* STEP04: \*/

// Unbind the buffer

gl.bindBuffer(gl.ARRAY\_BUFFER, null);

/\*=================== Shaders ====================\*/

/\* STEP05: coordinates here is linkage between JS and shader \*/

// Vertex shader source code

var vertCode =

'attribute vec3 coordinates;' +

'void main(void) {' +

' gl\_Position = vec4(coordinates, 1.0);' +

'}';

/\* STEP06: \*/

// Create a vertex shader object

var vertShader = gl.createShader(gl.VERTEX\_SHADER);

/\* STEP07: \*/

// Attach vertex shader source code

gl.shaderSource(vertShader, vertCode);

/\* STEP08: \*/

// Compile the vertex shader

gl.compileShader(vertShader);

/\* STEP09: \*/

// Fragment shader source code

var fragCode =

'void main(void) {' +

'gl\_FragColor = vec4(1.0, 1.0, 0.0, 0.1);' +

'}';

/\* STEP10: \*/

// Create fragment shader object

var fragShader = gl.createShader(gl.FRAGMENT\_SHADER);

/\* STEP11: \*/

// Attach fragment shader source code

gl.shaderSource(fragShader, fragCode);

/\* STEP12: \*/

// Compile the fragmentt shader

gl.compileShader(fragShader);

// Create a shader program object to store

/\* STEP13: \*/

// the combined shader program

var shaderProgram = gl.createProgram();

/\* STEP14: \*/

// Attach a vertex shader

gl.attachShader(shaderProgram, vertShader);

/\* STEP15: \*/

// Attach a fragment shader

gl.attachShader(shaderProgram, fragShader);

/\* STEP16: \*/

// Link both the programs

gl.linkProgram(shaderProgram);

/\* STEP17: \*/

// Use the combined shader program object

gl.useProgram(shaderProgram);

/\*======= Associating shaders to buffer objects ======\*/

/\* STEP18: \*/

// Bind vertex buffer object

gl.bindBuffer(gl.ARRAY\_BUFFER, vertex\_buffer);

/\* STEP19: coor, is user defined, coordinates is again linkage and its name should be same everywhere \*/

// Get the attribute location

var coord = gl.getAttribLocation(shaderProgram, "coordinates");

/\* STEP20: coor, is user defined \*/

// Point an attribute to the currently bound VBO

gl.vertexAttribPointer(coord, 3, gl.FLOAT, false, 0, 0);

/\* STEP21: coor, is user defined \*/

// Enable the attribute

gl.enableVertexAttribArray(coord);

/\*============ Drawing the HUT =============\*/

/\* STEP01: \*/

// Clear the canvas

gl.clearColor(0, 0, 0.3, 0.9);

/\* STEP01: \*/

// Enable the depth test

gl.enable(gl.DEPTH\_TEST);

/\* STEP01: \*/

// Clear the color and depth buffer

gl.clear(gl.COLOR\_BUFFER\_BIT | gl.DEPTH\_BUFFER\_BIT);

/\* STEP01: \*/

// Set the view port

gl.viewport(0,0,canvas.width,canvas.height);

/\* STEP01: \*/

//ground

/\*var vertices = [

-1,-0.25,0,

-1,-1,0,

1,-1,0,

1,-0.25,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.TRIANGLE\_FAN, 0, 4);\*/

// POINTS, LINE\_STRIP, LINE\_LOOP, LINES,

// TRIANGLE\_STRIP,TRIANGLE\_FAN, TRIANGLES

//front3

var vertices= [

-0.5,0.35,0,

-0.5,0.2,0,

0.1,0.2,0,

0.1,0.35,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.LINE\_LOOP, 0, 4);

//front 2

var vertices= [

-0.9,-0.05,0,

-0.9,-0.5,0,

-0.46,-0.5,0,

-0.46,-0.15,0,

-0.635,0.175,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.LINE\_LOOP, 0, 5);

//roof2

var vertices= [

-0.2,0.2,0,

-0.65,0.2,0,

-0.46,-0.15,0,

-0.3,-0.15,0,

-0.3,0.05,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.TRIANGLE\_FAN, 0, 5);

//front1

var vertices= [

-0.3,0.05,0,

-0.3,-0.5,0,

0.19,-0.5,0,

0.19,0.1,0,

0.04,0.39,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.LINE\_LOOP, 0, 5);

//front

var vertices= [

0.19,0.1,0,

0.76,0,0,

0.76,-0.5,0,

0.19,-0.5,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.LINE\_LOOP, 0, 4);

// roof1

var vertices= [

0,0.4,0,

0.2,0,0,

0.8,0,0,

0.6,0.4,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.TRIANGLE\_FAN, 0, 4);

//roof2.1

var vertices= [

-0.65,0.2,0,

-0.95,-0.05,0,

-0.9,-0.05,0,

-0.635,0.175,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.TRIANGLE\_FAN, 0, 4);

//roof1.1

var vertices= [

0,0.4,0,

-0.35,0.05,0,

-0.3,0.05,0,

0.04,0.39,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.TRIANGLE\_FAN, 0, 4);

//front2.1

var vertices= [

-0.46,-0.5,0,

-0.3,-0.49,0,

-0.3,-0.15,0,

-0.46,-0.15,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.LINE\_LOOP, 0, 4);

//roof3

var vertices= [

0.0,0.4,0,

-0.47,0.4,0,

-0.55,0.35,0,

-0.05,0.35,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.LINE\_LOOP, 0, 4);

//roof4

var vertices= [

0.,0.8,0,

0.2,0.6,0,

0.5,0.6,0,

0.30,0.8,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.TRIANGLE\_FAN, 0, 4);

//roof4.1

var vertices= [

0.,0.8,0,

-0.25,0.6,0,

-0.20,0.6,0,

0.035,0.78,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.TRIANGLE\_FAN, 0, 4);

//front4

var vertices= [

-0.2,0.6,0,

-0.2,0.4,0,

0.2,0.4,0,

0.2,0.6,0,

0.035,0.78,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.LINE\_LOOP, 0, 5);

//front4.1

var vertices= [

0.48,0.6,0,

0.48,0.4,0,

0.2,0.4,0,

0.2,0.6,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.LINE\_LOOP, 0, 4);

//window1

var vertices= [

-0.05,0.65,0,

-0.05,0.55,0,

0.09,0.55,0,

0.09,0.65,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.LINE\_LOOP, 0, 4);

var vertices= [

0.32,0.45,0,

0.42,0.45,0,

0.42,0.55,0,

0.32,0.55,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.LINE\_LOOP, 0, 4);

var vertices= [

0.1,-0.05,0,

-0.1,-0.05,0,

-0.1,-0.45,0,

0.1,-0.45,0,

]

gl.bufferData(gl.ARRAY\_BUFFER, new Float32Array(vertices), gl.STATIC\_DRAW);

gl.drawArrays(gl.LINE\_LOOP, 0, 4);

}

## **HTML Code**

<html>

<head>

<title>WEBGL01 - 500046700 </title>

</head>

<body onload="InitDemo();">

<canvas id="my\_Canvas" width="1000" height="450">

Your browser does not support HTML5

</canvas>

<br/>

<script src ="1\_first.js"></script>

</body>

</html>