COMPUTER MULTIMEDIA & ANIMATION

LAB MANUAL : PART-B

4TH SEM – BCA

Part -B:

1) Write a HTML/5 program to draw rectangle, line, polygon, polyline using SVG.

2) Write a HTML/5 program to draw linear and radial gradient ellipse using SVG.

3) Write a HTML/5 program to draw a star using SVG.

4) Write a HTML/5 program to draw line, circle, rectangle, gradient, text using canvas.

5) Write a HTML/5 program to demonstrate translation, rotation, scaling, and transform using canvas.

6) Write a HTML/5 program to demonstrate Bezier Curves and Quadratic Curves.

7) Write a HTML/5 program to create canvas and add a red square onto the game area with up/down/left/right controller buttons.

8) Write a HTML/5 canvas program to add random size obstacles with red square controller button.

1) Write a HTML/5 program to draw rectangle, line, polygon, polyline using SVG.

```
<!DOCTYPE html>
<html>
  <center>
   <body>
      <h2>HTML5 SVG Rectangle</h2>
      <svg width="400" height = "200">
         <rect x="50" y="20" width = "300" height = "100" fill = "red" />
      </svg>

      <h2>HTML5 SVG Line</h2>
      <svg width="200" height = "200">
         <line x1 = "0" y1 = "0" x2 = "200" y2 = "200" stroke="blue" stroke-width="5" />
      </svg>

      <h2>HTML5 SVG Polygon</h2>
      <svg width="300" height = "210">
         <polygon points="200,10 250,190 160,210" fill = "yellow" stroke="purple" stroke-width="2" />
      </svg>

      <h2>HTML5 SVG Polyline</h2>
     <svg width="200" height = "200">
      <polyline points="0,40 40,40 40,80 80,80 80,120 120,120 120,160"  fill = "white" stroke="green" stroke-
     width="5" />
      </svg>
   </body>
 </center>
</html>
```

2) Write a HTML/5 program to draw linear and radial gradient ellipse using SVG.

```
<!DOCTYPE html>
<html>
 <body>
  <center>
   <h2>HTML5 SVG LinearGradient Ellipse</h2>
    <svg height = "200" width="200" >
     <defs>
     <linearGradient id="gradient1" x1="0%" y1="0%" x2="100%" y2="0%">
     <stop offset="0%"  style="stop-color:yellow;stop-opacity:1" />
     <stop offset="100%" style="stop-color:red;stop-opacity:1" />
     </linearGradient>
     </defs>
     <ellipse cx = "100" cy = "50" rx = "100" ry = "50" fill="url(#gradient1)" />
    </svg>

   <h2>HTML5 SVG RadialGradient Ellipse</h2>
    <svg height = "200" width="200">
      <defs>
         <radialGradient id="gradient2" cx = "50%" cy = "50%" r = "50%" fx = "50%" fy = "50%">
         <stop offset = "0%" style = "stop-color:white; stop-opacity:0"/>
         <stop offset = "100%" style = "stop-color:blue; stop-opacity:1"/>
         </radialGradient>
      </defs>
      <ellipse cx = "100" cy = "50" rx = "100" ry = "50" fill="url(#gradient2)" />
    </svg>
   </center>
  </body>
</html>
```

3) Write a HTML/5 program to draw a star using SVG.

```
<!DOCTYPE html>

<html>

<body>
   <center>
    <h2>HTML5 SVG Star</h2>
      <svg width="200" height = "200" >
      <polygon points = "100,10 40,180 190,60 10,60 160,180" fill = "yellow"/>
    </svg>
   </center>
  </body>
 </html>
```

4) Write a HTML/5 program to draw line, circle, rectangle, gradient, text using canvas.

```
<!DOCTYPE html>
<html>
<body>
  <center>
   <h1> Canvas Line </h1>
   <canvas id="Canvasline" width="200" height="100" style="border:1px solid #d3d3d3;">
    </canvas>
   <h1> Canvas Circle </h1>
   <canvas id="Canvascircle" width="200" height="100" style="border:1px solid #d3d3d3;">
   </canvas>
   <h1> Canvas Rectangle </h1>
   <canvas id="CanvasRectangle" width="200" height="100" style="border:1px solid #d3d3d3;">
   </canvas>
   <h1> Canvas Text </h1>
   <canvas id="Canvastext" width="300" height="100" style="border:1px solid #d3d3d3;">
   </canvas>
   <h1> Canvas Gradient </h1>
   <canvas id="Canvasgradient" width="200" height="100" style="border:1px solid #d3d3d3;">
   </canvas>
<script>
// Canvas Line
var c = document.getElementById("Canvasline");
var ctx = c.getContext("2d");
ctx.moveTo(0,0);
ctx.lineTo(200,100);
ctx.stroke();
// Canvas Circle
var c = document.getElementById("Canvascircle");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95,50,40,0,2*Math.PI);
ctx.stroke();
// Canvas Rectangle
var c5 = document.getElementById("CanvasRectangle");
var ctx = c5.getContext("2d");
ctx.strokeRect(25, 10, 150, 80);
// Canvas Text
var c = document.getElementById("Canvastext");
var ctx = c.getContext("2d");
ctx.font = "30px Arial";
ctx.fillStyle="#00F";
ctx.fillText("GFGC Boy's College",10,50);
// Canvas Gradient
var c = document.getElementById("Canvasgradient");
var ctx = c.getContext("2d");
// Create gradient
var grd = ctx.createLinearGradient(0,0,200,0);
grd.addColorStop(0,"yellow");
grd.addColorStop(1,"red");
// Fill with gradient
ctx.fillStyle = grd;
ctx.fillRect(25,10,150,80);
</script>
</center>
</body>
</html>
```

5) Write a HTML/5 program to demonstrate translation, rotation, scaling, and transform using canvas.

```
<!DOCTYPE html>
<html>
<body>
<h2>HTML5 CANVAS</h2>
<canvas id="myCanvas" width="700" height="600" style="border:1px solid grey"></canvas>
<script>
const c = document.getElementById("myCanvas");
const ctx = c.getContext("2d");

//Canvas Rotate
ctx.strokeText("Canvas Rotate", 50, 10);
ctx.rotate(20 * Math.PI / 180);
ctx.fillStyle = "pink";
ctx.fillRect(50, 20, 200, 80);
ctx.setTransform(1, 0, 0, 1, 0, 0);

//Canvas Scale
ctx.translate(300, 10);
ctx.strokeText("Canvas Scale", 100, 0);
ctx.strokeRect(5, 5, 25, 15);
ctx.scale(2, 2);
ctx.strokeRect(5, 5, 25, 15);
ctx.scale(2, 2);
ctx.strokeRect(5, 5, 25, 15);
ctx.scale(2, 2);
ctx.strokeRect(5, 5, 25, 15);
ctx.setTransform(1, 0, 0, 1, 0, 0);

//Canvas Translate
ctx.strokeText("Canvas Translate", 400, 260);
ctx.fillStyle = "orange";
ctx.fillRect(400, 300, 200, 70);
ctx.fillStyle = "green";
ctx.translate(90, 90);
ctx.fillRect(400, 300, 200, 70);
ctx.setTransform(1, 0, 0, 1, 0, 0);

//Canvas Transform
ctx.strokeText("Canvas Transform", 50, 260);
ctx.translate(0, 220);
ctx.fillStyle = "yellow";
ctx.fillRect(10, 70, 200, 100)
ctx.transform(1, 0.5, -0.5, 1, 30, 10);
ctx.fillStyle = "red";
ctx.fillRect(10, 60, 200, 100);
ctx.transform(1, 0.5, -0.5, 1, 30, 10);
ctx.fillStyle = "blue";
ctx.fillRect(10, 50, 200, 100);
</script>
</body>
</html>
```

6) Write a HTML/5 program to demonstrate Bezier Curves and Quadratic Curves.

```
<!DOCTYPE html>
<html>
<body>
<h2>HTML5 CANVAS</h2>
<canvas id="myCanvas" width="500" height="500" style="border:1px solid grey"></canvas>
<script>
const c = document.getElementById("myCanvas");
const ctx = c.getContext("2d");

//  Adding Text to the Canvas
ctx.strokeText("Bezier Curve", 10, 20);
ctx.strokeText("Quadratic Curve", 10, 250);

// Canvas Bezier Curve
ctx.beginPath();
ctx.moveTo(100, 50);
ctx.bezierCurveTo(100, 200, 400, 200, 400, 50);
ctx.strokeStyle = "Orange";
ctx.lineWidth = 8;
ctx.stroke();

//Canvas Quadratic Curve
ctx.beginPath();
ctx.moveTo(100, 350);
ctx.quadraticCurveTo(100, 550, 400, 350);
ctx.strokeStyle = "green";
ctx.lineWidth = 8;
ctx.stroke();
</script>
</body>
</html>
```

7) Write a HTML/5 program to create canvas and add a red square onto the game area with up/down/left/right controller buttons.

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
<style>
canvas {
  border:1px solid #d3d3d3;
  background-color: #f1f1f1;
}
</style>
</head>
<body onload="startGame()">
<script>
var myGamePiece;
function startGame() {
  myGamePiece = new component(30, 30, "red", 10, 120);
  myGameArea.start();
}
var myGameArea = {
  canvas : document.createElement("canvas"),
  start : function() {
    this.canvas.width = 480;
    this.canvas.height = 270;
    this.context = this.canvas.getContext("2d");
    document.body.insertBefore(this.canvas, document.body.childNodes[0]);
    this.interval = setInterval(updateGameArea, 20);
  },
  clear : function() {
    this.context.clearRect(0, 0, this.canvas.width, this.canvas.height);
  }
}
```

```
function component(width, height, color, x, y) {
    this.width = width;
    this.height = height;
    this.speedX = 0;
    this.speedY = 0;
    this.x = x;
    this.y = y;
    this.update = function() {
        ctx = myGameArea.context;
        ctx.fillStyle = color;
        ctx.fillRect(this.x, this.y, this.width, this.height);
    }
    this.newPos = function() {
        this.x += this.speedX;
        this.y += this.speedY;
    }
}
function updateGameArea() {
    myGameArea.clear();
    myGamePiece.newPos();
    myGamePiece.update();
}
function moveup() {
    myGamePiece.speedY -= 1;
}
function movedown() {
    myGamePiece.speedY += 1;
}
function moveleft() {
    myGamePiece.speedX -= 1;
}
function moveright() {
    myGamePiece.speedX += 1;
}
```

```
  </script>
  <div style="text-align:center;width:480px;">
  <button onclick="moveup()">UP</button><br><br>
  <button onclick="moveleft()">LEFT</button>
  <button onclick="moveright()">RIGHT</button><br><br>
  <button onclick="movedown()">DOWN</button>
</div>
</body>
</html>
```

8) Write a HTML/5 canvas program to add random size obstacles with red square controller button.

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
<style>
canvas {
   border:1px solid #d3d3d3;
   background-color: #f1f1f1;
}
</style>
</head>
<body onload="startGame()">
<script>

var myGamePiece;
var myObstacles = [];

function startGame() {
   myGamePiece = new component(30, 30, "red", 10, 120);
   myGameArea.start();
}

var myGameArea = {
   canvas : document.createElement("canvas"),
   start : function() {
      this.canvas.width = 480;
      this.canvas.height = 270;
      this.context = this.canvas.getContext("2d");
```

```
        document.body.insertBefore(this.canvas, document.body.childNodes[0]);
        this.frameNo = 0;
        this.interval = setInterval(updateGameArea, 20);
        },
    clear : function() {
        this.context.clearRect(0, 0, this.canvas.width, this.canvas.height);
    },
    stop : function() {
        clearInterval(this.interval);
    }
}

function component(width, height, color, x, y) {
    this.width = width;
    this.height = height;
    this.speedX = 0;
    this.speedY = 0;
    this.x = x;
    this.y = y;
    this.update = function() {
        ctx = myGameArea.context;
        ctx.fillStyle = color;
        ctx.fillRect(this.x, this.y, this.width, this.height);
    }
    this.newPos = function() {
        this.x += this.speedX;
        this.y += this.speedY;
    }
    this.crashWith = function(otherobj) {
        var myleft = this.x;
        var myright = this.x + (this.width);
        var mytop = this.y;
        var mybottom = this.y + (this.height);
        var otherleft = otherobj.x;
        var otherright = otherobj.x + (otherobj.width);
        var othertop = otherobj.y;
        var otherbottom = otherobj.y + (otherobj.height);
        var crash = true;
        if ((mybottom < othertop) || (mytop > otherbottom) || (myright < otherleft) || (myleft > otherright)) {
            crash = false;
        }
        return crash;
    }
}
```

```
function updateGameArea() {
   var x, height, gap, minHeight, maxHeight, minGap, maxGap;
   for (i = 0; i < myObstacles.length; i += 1) {
      if (myGamePiece.crashWith(myObstacles[i])) {
         myGameArea.stop();
         return;
      }
   }
   myGameArea.clear();
   myGameArea.frameNo += 1;
   if (myGameArea.frameNo == 1 || everyinterval(150)) {
      x = myGameArea.canvas.width;
      minHeight = 20;
      maxHeight = 200;
      height = Math.floor(Math.random()*(maxHeight-minHeight+1)+minHeight);
      minGap = 50;
      maxGap = 200;
      gap = Math.floor(Math.random()*(maxGap-minGap+1)+minGap);
      myObstacles.push(new component(10, height, "green", x, 0));
      myObstacles.push(new component(10, x - height - gap, "green", x, height + gap));
   }
   for (i = 0; i < myObstacles.length; i += 1) {
      myObstacles[i].x += -1;
      myObstacles[i].update();
   }
   myGamePiece.newPos();
   myGamePiece.update();
}

function everyinterval(n) {
   if ((myGameArea.frameNo / n) % 1 == 0) {return true;}
   return false;
}
function moveup() {
   myGamePiece.speedY = -1;
}
function movedown() {
   myGamePiece.speedY = 1;
}
function moveleft() {
   myGamePiece.speedX = -1;
}
function moveright() {
   myGamePiece.speedX = 1;
}
```

```
function clearmove() {
   myGamePiece.speedX = 0;
   myGamePiece.speedY = 0;
}
</script>
<div style="text-align:center;width:480px;">
  <button onmousedown="moveup()" onmouseup="clearmove()"
    ontouchstart="moveup()">UP</button><br><br>
  <button onmousedown="moveleft()" onmouseup="clearmove()"
    ontouchstart="moveleft()">LEFT</button>
  <button onmousedown="moveright()" onmouseup="clearmove()"
    ontouchstart="moveright()">RIGHT</button><br><br>
  <button onmousedown="movedown()" onmouseup="clearmove()"
    ontouchstart="movedown()">DOWN</button>
</div>
</body>
</html>
```