

EXPERIMENT 2.3

AIM:

Interactive SVG Drawing Tool with Mouse Event Handlers

CODE:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>

  <title>SVG Drawing Tool</title>

  <style>

    body { font-family: Arial, sans-serif; text-align: center; }

    #toolbar { padding: 10px; background: #f5f5f5; border-bottom: 1px solid #ccc; }

    svg { border: 1px solid #ccc; margin-top: 10px; background: white; cursor: crosshair; }

  </style>

</head>

<body>

  <div id="toolbar">

    <select id="tool">

      <option value="pen">Pen</option>

      <option value="line">Line</option>

      <option value="rect">Rectangle</option>

      <option value="circle">Circle</option>

    </select>

    <input type="color" id="color" value="#000000" />

    <input type="number" id="strokeWidth" value="2" min="1" max="10" />

    <button id="clear">Clear</button>

  </div>
```

```
<svg id="drawing" width="800" height="500"></svg>
```

```
<script>
```

```
const svg = document.getElementById("drawing");  
const toolSelect = document.getElementById("tool");  
const colorPicker = document.getElementById("color");  
const strokeWidthInput = document.getElementById("strokeWidth");  
const clearBtn = document.getElementById("clear");
```

```
let drawing = false;
```

```
let currentElement = null;
```

```
let startX, startY;
```

```
svg.addEventListener("mousedown", (e) => {
```

```
    drawing = true;
```

```
    startX = e.offsetX;
```

```
    startY = e.offsetY;
```

```
    const tool = toolSelect.value;
```

```
    const stroke = colorPicker.value;
```

```
    const strokeWidth = strokeWidthInput.value;
```

```
    if (tool === "pen") {
```

```
        currentElement = document.createElementNS("http://www.w3.org/2000/svg", "path");
```

```
        currentElement.setAttribute("d", `M${startX},${startY}`);
```

```
    } else if (tool === "line") {
```

```
        currentElement = document.createElementNS("http://www.w3.org/2000/svg", "line");
```

```
        currentElement.setAttribute("x1", startX);
```

```
        currentElement.setAttribute("y1", startY);
```

```
        currentElement.setAttribute("x2", startX);
```

```
        currentElement.setAttribute("y2", startY);
```

```
    } else if (tool === "rect") {
```

```

currentElement = document.createElementNS("http://www.w3.org/2000/svg", "rect");
currentElement.setAttribute("x", startX);
currentElement.setAttribute("y", startY);
currentElement.setAttribute("width", 0);
currentElement.setAttribute("height", 0);
} else if (tool === "circle") {
currentElement = document.createElementNS("http://www.w3.org/2000/svg", "circle");
currentElement.setAttribute("cx", startX);
currentElement.setAttribute("cy", startY);
currentElement.setAttribute("r", 0);
}

currentElement.setAttribute("stroke", stroke);
currentElement.setAttribute("stroke-width", strokeWidth);
currentElement.setAttribute("fill", tool === "pen" || tool === "line" ? "none" : "rgba(0,0,0,0.1)");
svg.appendChild(currentElement);
});

svg.addEventListener("mousemove", (e) => {
if (!drawing || !currentElement) return;
const tool = toolSelect.value;
const x = e.offsetX, y = e.offsetY;

if (tool === "pen") {
let d = currentElement.getAttribute("d");
currentElement.setAttribute("d", d + ` L${x},${y}`);
} else if (tool === "line") {
currentElement.setAttribute("x2", x);
currentElement.setAttribute("y2", y);
} else if (tool === "rect") {
const width = x - startX;

```

```

const height = y - startY;
currentElement.setAttribute("width", Math.abs(width));
currentElement.setAttribute("height", Math.abs(height));
currentElement.setAttribute("x", width < 0 ? x : startX);
currentElement.setAttribute("y", height < 0 ? y : startY);
} else if (tool === "circle") {
const r = Math.sqrt(Math.pow(x - startX, 2) + Math.pow(y - startY, 2));
currentElement.setAttribute("r", r);
}
});

svg.addEventListener("mouseup", () => { drawing = false; currentElement = null; });
clearBtn.addEventListener("click", () => { svg.innerHTML = ""; });
</script>
</body>
</html>

```

OUTPUT:

