MBARARA UNIVERSITY OF SCIENCE AND TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATICS

BARCHELORS OF SCIENCE IN COMPUTER SCIENCE

COMPUTER GRAPHICS

BY RICHARD NTWARI

2023/BCS/137/PS

AGABA DAVIS

```
<!DOCTYPE html>
<html>
<head>
  <title>Stick Man Scene</title>
  <style>
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
    }

    html, body {
      width: 100%;
      height: 100%;
      background-color: #f0f0f0;
  overflow: hidden;

  font-family: sans-serif;

}


body {

  display: flex;

  flex-direction: column;

  justify-content: center;

  align-items: center;

}


canvas {

  width: 100vw;

  height: 100vh;

  display: block;

  border: 2px solid #333;

  box-shadow: 0 0 10px rgba(0,0,0,0.3);

}


.controls {

  position: absolute;

  top: 0;

  padding: 10px;

  width 100%;

  flex-direction: column;

  order: -1;
```

```
            bottom: 20px;
            display: flex;
            gap: 5px;
            z-index: 10;
        }

        button {
            padding: 8px 16px;
            border: none;
            border-radius: 4px;
            background-color: #4CAF50;
            color: white;
            cursor: pointer;
            font-weight: bold;
        }

        button:hover {
            background-color: #45a049;
        }

    </style>
</head>
<body>
    <canvas id="myCanvas" width="800" height="500"></canvas>

    <script>
```

```
// Initialize canvas and context
const canvas = document.getElementById('myCanvas');
const ctx = canvas.getContext('2d');

// Scene variables
let isDay = true;
let stickManX = 400;
let stickManDirection = 1;
let isMoving = false;
let windIntensity = 0;
let sunMoonY = 80;
let sunMoonX = 700;
let cloudX = 600;
let cloudDirection = -1;

// Animation variables
let legAngle = 0;
let armAngle = 0;
let leavesAngle = 0;

// Draw the entire scene
function drawScene() {
  // Clear the canvas
  ctx.clearRect(0, 0, canvas.width, canvas.height);

  // Draw sky
```

```
    const skyGradient = ctx.createLinearGradient(0, 0, 0, canvas.height);
    if (isDay) {
      skyGradient.addColorStop(0, "#87CEEB");
      skyGradient.addColorStop(1, "#E0F7FF");
    } else {
      skyGradient.addColorStop(0, "#0C1445");
      skyGradient.addColorStop(1, "#2C3E7C");
    }
    ctx.fillStyle = skyGradient;
    ctx.fillRect(0, 0, canvas.width, canvas.height);

    //we Draw sun or moon
    if (isDay) {
      // Sun
      ctx.beginPath();
      ctx.arc(sunMoonX, sunMoonY, 40, 0, Math.PI * 2);
      const sunGradient = ctx.createRadialGradient(sunMoonX, sunMoonY, 0,
sunMoonX, sunMoonY, 40);
      sunGradient.addColorStop(0, "#FFFF00");
      sunGradient.addColorStop(0.7, "#FFA500");
      sunGradient.addColorStop(1, "#FF8C00");
      ctx.fillStyle = sunGradient;
      ctx.fill();

      // Sun rays
      ctx.save();
```

```
          ctx.translate(sunMoonX, sunMoonY);

          ctx.beginPath();

          for (let i = 0; i < 12; i++) {

            ctx.rotate(Math.PI / 6);

            ctx.moveTo(45, 0);

            ctx.lineTo(60, 0);

          }

          ctx.strokeStyle = "#FF8C00";

          ctx.lineWidth = 4;

          ctx.stroke();

          ctx.restore();


          // Draw clouds

          drawCloud(cloudX, 120);

          drawCloud(cloudX - 250, 90);

        } else {

          // Moon

          ctx.beginPath();

          ctx.arc(sunMoonX, sunMoonY, 35, 0, Math.PI * 2);

          const moonGradient = ctx.createRadialGradient(sunMoonX, sunMoonY, 0, sunMoonX, sunMoonY, 35);

          moonGradient.addColorStop(0, "#FFFFFF");

          moonGradient.addColorStop(0.7, "#F0F0F0");

          moonGradient.addColorStop(1, "#E0E0E0");

          ctx.fillStyle = moonGradient;

          ctx.fill();
          // Stars
          for (let i = 0; i < 100; i++) {
            const x = Math.random() * canvas.width;
            const y = Math.random() * canvas.height * 0.7;
            const radius = Math.random() * 1.5;

            ctx.beginPath();
            ctx.arc(x, y, radius, 0, Math.PI * 2);
            ctx.fillStyle = "white";
            ctx.fill();
          }
        }

        // Draw ground
        const groundGradient = ctx.createLinearGradient(0, canvas.height - 120, 0, canvas.height);
        if (isDay) {
          groundGradient.addColorStop(0, "#7CFC00");
          groundGradient.addColorStop(1, "#228B22");
        } else {
          groundGradient.addColorStop(0, "#1E3B05");
          groundGradient.addColorStop(1, "#0A1C00");
        }
        ctx.fillStyle = groundGradient;
        ctx.fillRect(0, canvas.height - 120, canvas.width, 120);
```

```
  // Draw compound fence
  drawFence();

  // Draw house
  drawHouse(250, canvas.height - 250, 200, 150);

  // Draw tree with shade
  drawTree(580, canvas.height - 120);

  // Draw car with tires properly positioned
  drawCar(canvas.width - 150, canvas.height - 65);

  // Draw improved stick man on the ground
  drawStickMan(stickManX, canvas.height - 60);
}

// Draw a cloud
function drawCloud(x, y) {
  ctx.fillStyle = "rgba(255, 255, 255, 0.8)";
  ctx.beginPath();
  ctx.arc(x, y, 25, 0, Math.PI * 2);
  ctx.arc(x + 25, y - 10, 25, 0, Math.PI * 2);
  ctx.arc(x + 50, y, 20, 0, Math.PI * 2);
  ctx.arc(x + 25, y + 10, 25, 0, Math.PI * 2);
  ctx.fill();
```

```
}

// Draw the fence
function drawFence() {
  ctx.fillStyle = isDay ? "#CD853F" : "#8B4513";
  ctx.strokeStyle = isDay ? "#A0522D" : "#5E2605";
  ctx.lineWidth = 2;

  // Draw horizontal beams
  ctx.fillRect(50, canvas.height - 150, 700, 10);
  ctx.fillRect(50, canvas.height - 190, 700, 10);

  // Draw vertical posts
  for (let i = 0; i <= 17; i++) {
    ctx.fillRect(50 + i * 40, canvas.height - 220, 10, 100);
  }

  // Draw gate
  ctx.fillStyle = isDay ? "#A0522D" : "#603311";
  ctx.fillRect(250, canvas.height - 220, 80, 100);

  // Gate details
  ctx.beginPath();
  ctx.moveTo(250, canvas.height - 220);
  ctx.lineTo(330, canvas.height - 170);
  ctx.moveTo(250, canvas.height - 170);
```

```
    ctx.lineTo(330, canvas.height - 220);
    ctx.stroke();

    // Gate handle
    ctx.fillStyle = "#FFD700";
    ctx.beginPath();
    ctx.arc(320, canvas.height - 190, 5, 0, Math.PI * 2);
    ctx.fill();
}

// Draw the house
function drawHouse(x, y, width, height) {
    // House body
    ctx.fillStyle = isDay ? "#F8F8FF" : "#BEBEBE";
    ctx.fillRect(x, y, width, height);

    // Roof
    ctx.fillStyle = isDay ? "#8B0000" : "#4B0000";
    ctx.beginPath();
    ctx.moveTo(x - 20, y);
    ctx.lineTo(x + width/2, y - 50);
    ctx.lineTo(x + width + 20, y);
    ctx.closePath();
    ctx.fill();

    // Door
```

```
ctx.fillStyle = isDay ? "#8B4513" : "#3B2613";
ctx.fillRect(x + width/2 - 20, y + height - 60, 40, 60);

// Door handle
ctx.fillStyle = "#FFD700";
ctx.beginPath();
ctx.arc(x + width/2 + 10, y + height - 30, 3, 0, Math.PI * 2);
ctx.fill();

// Windows
ctx.fillStyle = isDay ? "#87CEFA" : "#0F4C81";
ctx.fillRect(x + 30, y + 30, 40, 40);
ctx.fillRect(x + width - 70, y + 30, 40, 40);

// Window frames
ctx.strokeStyle = isDay ? "#FFFFFF" : "#AAAAAA";
ctx.lineWidth = 3;
ctx.beginPath();
ctx.moveTo(x + 30, y + 50);
ctx.lineTo(x + 70, y + 50);
ctx.moveTo(x + 50, y + 30);
ctx.lineTo(x + 50, y + 70);
ctx.moveTo(x + width - 70, y + 50);
ctx.lineTo(x + width - 30, y + 50);
ctx.moveTo(x + width - 50, y + 30);
ctx.lineTo(x + width - 50, y + 70);
```

```
  ctx.stroke();
}

// Draw the car with proper tires
function drawCar(x, y) {
  // Car body
  ctx.fillStyle = "#FF0000";
  ctx.beginPath();
  ctx.moveTo(x - 60, y - 20);
  ctx.lineTo(x + 60, y - 20);
  ctx.lineTo(x + 60, y - 40);
  ctx.lineTo(x + 30, y - 60);
  ctx.lineTo(x - 30, y - 60);
  ctx.lineTo(x - 60, y - 40);
  ctx.closePath();
  ctx.fill();

  // Car body details - stripes or design
  ctx.strokeStyle = "#FFFFFF";
  ctx.lineWidth = 2;
  ctx.beginPath();
  ctx.moveTo(x - 50, y - 30);
  ctx.lineTo(x + 50, y - 30);
  ctx.stroke();

  // Windows
```

```
ctx.fillStyle = "#87CEFA";

// Front window
ctx.beginPath();
ctx.moveTo(x - 10, y - 57);
ctx.lineTo(x + 28, y - 57);
ctx.lineTo(x + 28, y - 40);
ctx.lineTo(x - 10, y - 40);
ctx.closePath();
ctx.fill();

// Back window
ctx.beginPath();
ctx.moveTo(x - 28, y - 57);
ctx.lineTo(x - 28, y - 40);
ctx.lineTo(x - 50, y - 40);
ctx.lineTo(x - 28, y - 57);
ctx.closePath();
ctx.fill();

// Tires with detailed styling
// Rear tire
drawTire(x - 30, y);

// Front tire
drawTire(x + 30, y);
```

```
  // Headlights
  if (!isDay) {
    // Headlight beams
    ctx.fillStyle = "rgba(255, 255, 150, 0.3)";
    ctx.beginPath();
    ctx.moveTo(x + 60, y - 30);
    ctx.lineTo(x + 120, y - 60);
    ctx.lineTo(x + 120, y);
    ctx.closePath();
    ctx.fill();
  }

  ctx.fillStyle = isDay ? "#FFFF00" : "#FFFFAA";
  ctx.beginPath();
  ctx.arc(x + 58, y - 30, 5, 0, Math.PI * 2);
  ctx.fill();

  // Taillights
  ctx.fillStyle = "#FF0000";
  ctx.beginPath();
  ctx.arc(x - 58, y - 30, 5, 0, Math.PI * 2);
  ctx.fill();
}

// Draw a detailed tire
```

```
function drawTire(x, y) {
    // Tire outer circle
    ctx.fillStyle = "#000000";
    ctx.beginPath();
    ctx.arc(x, y, 15, 0, Math.PI * 2);
    ctx.fill();

    // Tire rim
    ctx.fillStyle = "#CCCCCC";
    ctx.beginPath();
    ctx.arc(x, y, 7, 0, Math.PI * 2);
    ctx.fill();

    // Tire hub cap
    ctx.fillStyle = "#FFFFFF";
    ctx.beginPath();
    ctx.arc(x, y, 3, 0, Math.PI * 2);
    ctx.fill();

    // Tire details
    ctx.strokeStyle = "#333333";
    ctx.lineWidth = 1;
    ctx.beginPath();
    for (let i = 0; i < 6; i++) {
        const angle = i * Math.PI / 3;
        ctx.moveTo(x, y);
        ctx.lineTo(x + Math.cos(angle) * 7, y + Math.sin(angle) * 7);
    }
    ctx.stroke();
}

// Draw the redesigned tree with needle-like leaves and branches for shade (non-moving)
function drawTree(x, y) {
    // Tree trunk
    const trunkGradient = ctx.createLinearGradient(x - 15, 0, x + 15, 0);
    trunkGradient.addColorStop(0, "#8B4513");
    trunkGradient.addColorStop(0.5, "#A0522D");
    trunkGradient.addColorStop(1, "#8B4513");

    ctx.fillStyle = trunkGradient;
    ctx.beginPath();
    ctx.moveTo(x - 10, y);
    ctx.lineTo(x - 15, y - 120);
    ctx.lineTo(x + 15, y - 120);
    ctx.lineTo(x + 10, y);
    ctx.closePath();
    ctx.fill();

    // Draw shade on ground
    if (isDay) {
        ctx.fillStyle = "rgba(0,0,0,0.2)";
```

```
       ctx.beginPath();
        ctx.ellipse(x, y + 5, 100, 30, 0, 0, Math.PI * 2);
        ctx.fill();
    }

    // Tree branches and needle-like leaves
    const branchColor = isDay ? "#8B4513" : "#3B2506";
    const leafColor = isDay ? "#006400" : "#003200";

    // Main branches - now with no movement
    drawStaticBranch(x, y - 120, 60, -Math.PI/4, 5, branchColor, leafColor);
    drawStaticBranch(x, y - 120, 70, -Math.PI/6, 5, branchColor, leafColor);
    drawStaticBranch(x, y - 120, 50, -Math.PI/2.5, 5, branchColor, leafColor);
    drawStaticBranch(x, y - 120, 65, -Math.PI/1.7, 5, branchColor, leafColor);
    drawStaticBranch(x, y - 90, 55, -Math.PI/3.5, 4, branchColor, leafColor);
    drawStaticBranch(x, y - 90, 60, -Math.PI/1.5, 4, branchColor, leafColor);
    drawStaticBranch(x, y - 60, 40, -Math.PI/4, 3, branchColor, leafColor);
    drawStaticBranch(x, y - 60, 45, -Math.PI/1.3, 3, branchColor, leafColor);
}

// Function to draw a static branch with needle-like leaves (no movement)
function drawStaticBranch(startX, startY, length, angle, width, branchColor, leafColor) {
    ctx.save();
    ctx.translate(startX, startY);
    ctx.rotate(angle);
```

```
// Draw the branch
ctx.strokeStyle = branchColor;
ctx.lineWidth = width;
ctx.lineCap = "round";
ctx.beginPath();
ctx.moveTo(0, 0);
ctx.lineTo(length, 0);
ctx.stroke();

// Draw needle-like leaves along the branch
ctx.strokeStyle = leafColor;
ctx.lineWidth = 1;

const leafCount = Math.floor(length / 5);

for (let i = 10; i < length; i += 5) {
  // Upper leaves
  const leafAngle = Math.PI/2;
  const leafLength = 8 + Math.random() * 7;

  ctx.save();
  ctx.translate(i, 0);

  // Draw multiple needle-like leaves in a cluster
  for (let j = 0; j < 5; j++) {
    const spreadAngle = leafAngle + (j - 2) * Math.PI/16;
```

```
                    ctx.beginPath();
                    ctx.moveTo(0, 0);
                    ctx.lineTo(leafLength * Math.cos(spreadAngle), leafLength * Math.sin(spreadAngle));
                    ctx.stroke();
                }

                // Lower leaves
                for (let j = 0; j < 5; j++) {
                    const spreadAngle = -leafAngle + (j - 2) * Math.PI/16;
                    ctx.beginPath();
                    ctx.moveTo(0, 0);
                    ctx.lineTo(leafLength * Math.cos(spreadAngle), leafLength * Math.sin(spreadAngle));
                    ctx.stroke();
                }

                ctx.restore();
            }

            // Recursively draw smaller branches
            if (width > 1) {
                // Draw 2-3 sub-branches
                const branchCount = 2 + Math.floor(Math.random() * 2);
                for (let i = 0; i < branchCount; i++) {
                    const subLength = length * (0.5 + Math.random() * 0.3);
                    const subAngle = (Math.random() - 0.5) * Math.PI/2;
```

```
      const subStartPoint = length * (0.6 + Math.random() * 0.3);

      ctx.save();
      ctx.translate(subStartPoint, 0);
      drawStaticBranch(0, 0, subLength, subAngle, width - 1, branchColor, leafColor);
      ctx.restore();
    }
  }

  ctx.restore();
}

// Draw the improved stick man on the ground
function drawStickMan(x, y) {
  ctx.lineWidth = 3;
  ctx.lineCap = "round";
  ctx.lineJoin = "round";

  // Body color
  const bodyColor = "#000000";
  const skinTone = "#FFD700";  // Gold color for better visibility

  // Draw the shoes
  ctx.fillStyle = "#0000CC";  // Blue shoes
  ctx.beginPath();
```

```
    ctx.ellipse(x - 10 * stickManDirection + Math.sin(legAngle) * 3, y + 5, 12, 5, 0, 0, Math.PI * 2);
    ctx.fill();

    ctx.beginPath();
    ctx.ellipse(x + 10 * stickManDirection + Math.sin(legAngle + Math.PI) * 3, y + 5, 12, 5, 0, 0, Math.PI * 2);
    ctx.fill();

    // Draw the pants
    ctx.fillStyle = "#3333CC";  // Blue pants
    ctx.beginPath();
    ctx.moveTo(x - 12, y - 25);
    ctx.lineTo(x + 12, y - 25);
    ctx.lineTo(x + 10, y - 10);
    ctx.lineTo(x - 10, y - 10);
    ctx.closePath();
    ctx.fill();

    // Legs with improved design
    ctx.strokeStyle = "#3333CC";  // Blue pants
    ctx.lineWidth = 5;
    ctx.beginPath();
    ctx.moveTo(x - 5, y - 10);
    ctx.lineTo(x - 10 * stickManDirection, y + Math.sin(legAngle) * 5);
    ctx.moveTo(x + 5, y - 10);
    ctx.lineTo(x + 10 * stickManDirection, y + Math.sin(legAngle + Math.PI) * 5);
```

```
ctx.stroke();

// Draw the shirt
ctx.fillStyle = "#CC3333";  // Red shirt
ctx.beginPath();
ctx.moveTo(x - 12, y - 25);
ctx.lineTo(x + 12, y - 25);
ctx.lineTo(x + 15, y - 40);
ctx.lineTo(x - 15, y - 40);
ctx.closePath();
ctx.fill();

// Arms with improved design
ctx.strokeStyle = skinTone;
ctx.lineWidth = 3;
ctx.beginPath();
ctx.moveTo(x - 12, y - 35);
ctx.lineTo(x - 22 * stickManDirection, y - 20 + Math.sin(armAngle) * 5);
ctx.moveTo(x + 12, y - 35);
ctx.lineTo(x + 22 * stickManDirection, y - 20 + Math.sin(armAngle + Math.PI) * 5);
ctx.stroke();

// Hands
ctx.fillStyle = skinTone;
ctx.beginPath();
ctx.arc(x - 22 * stickManDirection, y - 20 + Math.sin(armAngle) * 5, 4, 0, Math.PI * 2);
```

```
    ctx.arc(x + 22 * stickManDirection, y - 20 + Math.sin(armAngle + Math.PI) * 5, 4, 0, Math.PI * 2);
    ctx.fill();

    // Head
    ctx.fillStyle = skinTone;
    ctx.beginPath();
    ctx.arc(x, y - 55, 15, 0, Math.PI * 2);
    ctx.fill();

    // Hair
    ctx.fillStyle = "#663300";  // Brown hair
    ctx.beginPath();
    ctx.arc(x, y - 65, 10, Math.PI, 2 * Math.PI);
    ctx.fill();

    // Face details
    ctx.fillStyle = "#000000";

    // Eyes
    ctx.beginPath();
    ctx.arc(x - 5, y - 55, 2, 0, Math.PI * 2);
    ctx.arc(x + 5, y - 55, 2, 0, Math.PI * 2);
    ctx.fill();

    // Smile
    ctx.strokeStyle = "#000000";
    ctx.lineWidth = 1;
    ctx.beginPath();
    ctx.arc(x, y - 50, 8, 0.2 * Math.PI, 0.8 * Math.PI);
    ctx.stroke();

    // Nose
    ctx.beginPath();
    ctx.moveTo(x, y - 55);
    ctx.lineTo(x, y - 50);
    ctx.lineTo(x + 2, y - 50);
    ctx.stroke();
}

// Function to move the stick man
function moveStickMan() {
    isMoving = !isMoving;
}

// Function to toggle day/night
function dayNightToggle() {
    isDay = !isDay;
}

// Function to make it a windy day - now has no effect on tree
function windyDay() {
    windIntensity = windIntensity > 0 ? 0 : 5;
}

// Animation loop
function animate() {
    // Update animation variables
    legAngle += 0.05;
    armAngle += 0.05;

    // Move clouds
    cloudX += cloudDirection * 0.5;
    if (cloudX < 0 || cloudX > canvas.width) {
        cloudDirection *= -1;
    }

    // Move stick man if moving
    if (isMoving) {
        stickManX += stickManDirection * 2;

        // Change direction if reaching boundaries
        if (stickManX > 750 || stickManX < 100) {
            stickManDirection *= -1;
        }
    }

    // Draw the scene
```

```
          drawScene();



      }

       // Start the animation
       animate();
    </script>
</body>
</html>
```