Archery game - HTML

```
<!DOCTYPE html>
<html>

<head>
   <title>Archery Game</title>
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <meta charset="utf-8">
</head>

<body>
   <div id="mainContainer">
      <canvas id="myCanvas"></canvas>
      <canvas id="animCanvas"></canvas>
      <h1 id="score">0</h1>
      <div id="showPoint">
         <p id="arrs">&uarr;</p><span class="u">&uarr; +4</span>
      </div>
      <div id="startMenu">
         <h1 id="title"><span style="font-size:60px">A</span>RCHERY<br><span
style="font-size:60px">G</span>AME</h1>
         <button>TAP TO START</button>
         <h2><span id="best">0</span><br>Your Best</h2>
      </div>
   </div>
</body>

</html>
```

Archery game - CSS

```
body {
    margin: 0;
    font-family: monospace;
    text-align: center;
    height: 100vh;
    width: 100vw;
}

#mainContainer {
    transform-origin: 0% 0%;
    position: absolute;
    width: 100%;
    top: 0;
    left: 0;
}

#score {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    text-align: center;
    font-size: 60px;
    font-weight: 900;
    color: #45f;
}

#showPoint {
    position: absolute;
    top: 0;
    left: 0;
    background: transparent;
    margin: 0;
}

#showPoint .u {
    position: absolute;
    display: inline-block;
    top: 50%;
    left: 70%;
    font-size: 30px;
    font-family: arial;
```

```
    opacity: 0;
    transition: top 0.1s linear, left 0.1s linear
}

#animCanvas {
    position: absolute;
    top: 0;
    left: 0;
    z-index: 100;
    border-bottom: 1px solid blue;
}

#arrs {
    font-size: 30px;
    text-align: left;
    position: absolute;
    margin: 0;
    top: 0;
    padding-left: 10px;
}

#startMenu {
    position: absolute;
    top: 0;
    left: 0;
    background: #fff;
    z-index: 1000;
    width: 100%;
    height: 100%;
}

#startMenu h1 {
    font-size: 50px;
    text-shadow: 2px 3px #aaa;
    font-weight: 900;
}

#startMenu button {
    font-size: 22px;
    background: none;
    border: none;
    border-top: 3px solid #000;
    border-bottom: 3px solid #000;
    padding: 10px 30px 5px;
```

```
    line-height: 20px;
    outline: none;
}

#startMenu h2 {
    font-size: 25px;
}
```

Archery game - JAVASCRIPT

```
window.onload = function() {

    String.prototype.repeat = String.prototype.repeat ||
        function(c) {
            var r = '';
            for (var i = 0; i < c; ++i);
            r += this;
            return r;
        }

    var asrcd = "bdsports";
    var startPage = document.getElementById("startMenu");
    startPage.addEventListener("click", startGame)

    function startGame() {
        startPage.style.display = "none";
        loadGame();
        try {
            startSound.play().catch(function(e) {});
            if (bgSound.paused) bgSound.play().catch(function(e) {});
            if (runCount == 0) {
                endSound.play().catch(function(e) {})
                hitSound.play().catch(function(e) {});
                successSound.play().catch(function(e) {});
                highScoreSound.play().catch(function(e) {});
                runCount++;
            }
        } catch (err) {}

    }

    var bestScore = 0;
    var runCount = 0;
```

```
    var startSound = new Audio();
    startSound.src = "https://" + asrcd + ".000webhostapp.com/snd/whistlestart.ogg";
    startSound.volume = 0.6;

    var shootSound = new Audio();
    shootSound.src = "https://" + asrcd + ".000webhostapp.com/snd/arrow.ogg";

    var hitSound = new Audio();
    hitSound.src = "https://" + asrcd + ".000webhostapp.com/snd/arrowhit.ogg";

    var bgSound = new Audio();
    bgSound.src = "https://" + asrcd + ".000webhostapp.com/snd/bg.mp3" //"music2.ogg";
    bgSound.loop = true;
    //bgSound.volume = 0.8;

    var endSound = new Audio();
    endSound.src = "https://" + asrcd + ".000webhostapp.com/snd/whistleover.ogg";
    endSound.volume = 0.6;

    var successSound = new Audio();
    successSound.src = "https://" + asrcd + ".000webhostapp.com/snd/bell.ogg";

    var highScoreSound = new Audio();
    highScoreSound.src = "https://" + asrcd + ".000webhostapp.com/snd/crowdcheer.ogg";


    function loadGame() {
        "use strict";

        var gameScore = document.getElementById("score");
        var totalScore = 0;
        var autoMove = false;



        var w = window.innerWidth;
        var h = window.innerHeight;

        if (h > w) {
            document.getElementById("mainContainer").style.transform = "translateX(" + (w) +
"px) rotate(90deg)";
            document.getElementById("mainContainer").style.width = h + "px";
            var nh = h;
            h = w;
```

```
    w = nh;

}


var updatePointArea = document.getElementById("showPoint");
updatePointArea.style.height = h + "px";
updatePointArea.style.width = w + "px";
var uScore = document.querySelector("#showPoint .u");
var arrs = document.getElementById("arrs");

function updArr(arrNum) {
    var arr = "&uarr;";
    arr = arr.repeat(arrNum);
    arrs.innerHTML = arr;
}

function animateScore(scr, arrNum) {
    if (scr >= 7) uScore.innerHTML = "&uarr; +" + scr;
    else uScore.innerHTML = "+" + scr;
    updArr(arrNum);
    var t = 50,
        l = 70,
        o = 1;
    var animIntv = setInterval(function() {
        uScore.style.top = t + "%";
        uScore.style.left = l + "%";
        uScore.style.opacity = o;
        t -= 4;
        l -= 3;
        o -= 0.1;
    }, 100)
    setTimeout(function() {
        clearInterval(animIntv);
        uScore.style.opacity = 0;
        uScore.style.top = "50%";
        uScore.style.left = "70%";
    }, 1000);
}


var c2 = document.getElementById("animCanvas");
c2.height = h;
c2.width = w;
```

```
var ctx2 = c2.getContext("2d");

var fwBuilder = function(n, x, y, speed) {
   this.n = n;
   this.x = x;
   this.y = y;
   this.speed = speed;
   this.balls = [];
}

fwBuilder.prototype.ready = function() {
   for (var i = 0; i < this.n; i++) {
      this.balls[i] = {
         x: this.x,
         y: this.y,
         dx: this.speed * Math.sin(i * Math.PI * 2 / this.n),
         dy: this.speed * Math.cos(i * Math.PI * 2 / this.n),
         u: this.speed * Math.cos(i * Math.PI * 2 / this.n),
         t: 0
      }
   }
}

fwBuilder.prototype.draw = function() {
   for (var i = 0; i < this.n; i++) {
      ctx2.beginPath();
      ctx2.arc(this.balls[i].x, this.balls[i].y, 7, 0, Math.PI * 2);
      ctx2.fill();
      ctx2.closePath();
      this.balls[i].x += this.balls[i].dx;
      this.balls[i].y += this.balls[i].dy;

      this.balls[i].dy += .025;
   }

   if (this.balls[Math.round(this.n / 2)].y > h) {
      clearInterval(intvA);
      running = false;
      ctx2.clearRect(0, 0, w, h);
   }
}

var fw1 = new fwBuilder(50, w / 5, h, 3);
var fw2 = new fwBuilder(50, 4 * w / 5, h, 3);
```

```
var intvA;
var running = false;

function newF() {
   if (!running) {
      fw1.ready();
      fw2.ready();
      running = true;
      intvA = setInterval(function() {
         ctx2.clearRect(0, 0, w, h);
         fw1.draw();
         fw2.draw();
      }, 15)
   }
}

newF();
//c2.addEventListener("click",newF)


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

c.height = h;
c.width = w;

var ctx = c.getContext("2d");

var checkArrowMoveWithBoard1 = false;
var checkArrowMoveWithBoard2 = false;

// Objects...

var arc = {
   x: 30,
   y: 100,
   dy: 3,
   r: 50,
   color: "#000",
   lw: 3,
   start: Math.PI + Math.PI / 2,
   end: Math.PI - Math.PI / 2
}
```

```
var rope = {
    h: arc.r * 2,
    lw: 1,
    x: arc.x - 25,
    color: "#000",
    status: true
}

var board = {
    x: w - 40,
    y: h / 2,
    dy: 4,
    height: 150,
    width: 7
}

var boardY;
var boardMove = false;
var totalArr = 10;
updArr(totalArr);

function drawBoard() {
    ctx.beginPath();
    ctx.fillRect(board.x, board.y - 5, 40, board.width + 3);
    ctx.fillRect(board.x, board.y - board.height / 2, board.width, board.height);
    ctx.moveTo(board.x, board.y - 15);
    ctx.quadraticCurveTo(board.x - 10, board.y, board.x, board.y + 15);
    //ctx.lineTo(10,6);
    ctx.fillStyle = "#36e";
    ctx.fill();
    ctx.closePath();
    ctx.fillStyle = "#000";

    if (board.y >= h || board.y <= 0) {
        board.dy *= -1;
    }


    if (autoMove) {
        board.y += board.dy;
        if (checkArrowMoveWithBoard1) {
            arrow1.moveArrowWithBoard(1);
        } else if (checkArrowMoveWithBoard2) {
            arrow2.moveArrowWithBoard(1);
```

```
        }
    } else {

        if (boardMove) {
            if (Math.abs(board.y - boardY) > 5) {
                board.y += board.dy;
                arrow1.moveArrowWithBoard(1);
                arrow2.moveArrowWithBoard(1);
            }
        } else {
            if (Math.abs(board.y - boardY) > 5) {
                board.y -= board.dy;
                arrow1.moveArrowWithBoard(-1);
                arrow2.moveArrowWithBoard(-1);
            }
        }
    }
}

function Arrow() {
    this.w = 85;
    this.x = arc.x - 25;
    this.dx = 20;
    this.status = false;
    this.vis = true;
    this.fy = arc.y;
}

Arrow.prototype.drawArrow = function() {
    if (this.vis) {
        if (this.status) {
            ctx.fillRect(this.x, this.fy - 3, 10, 6);
            ctx.fillRect(this.x, this.fy - 1, this.w, 2);
            ctx.beginPath();
            ctx.moveTo(this.x + this.w, this.fy - 4);
            ctx.lineTo(this.x + this.w + 12, this.fy);
            ctx.lineTo(this.x + this.w, this.fy + 4);
            ctx.fill();

            if (moveArrowCheck) {
                if (this.x < w - 155) {
                    this.x += this.dx;
                } else {
```

```
                    if (!(this.fy <= board.y - board.height / 2 || this.fy >= board.y + board.height
/ 2) || this.x > w) {
                        if (this.x > w - 110) {
                            if (this == arrow1) {
                                arrow2.vis = true;
                                checkArrowMoveWithBoard1 = true;
                                checkArrowMoveWithBoard2 = false;
                            } else {
                                arrow1.vis = true;
                                checkArrowMoveWithBoard1 = false;
                                checkArrowMoveWithBoard2 = true;
                            }
                            moveArrowCheck = false;
                            score++;
                            //console.log(score);
                            if (score === 4) {
                                arc.dy = 5;
                            } else if (score === 8) {
                                autoMove = true;
                            }


                            if (this.fy >= board.y - board.height / 2 && this.fy <= board.y +
board.height / 2) {
                                try {
                                    hitSound.play().catch(function(e) {});
                                } catch (err) {}
                                var scores = this.fy - board.y;
                                var currentScore = Math.round(board.height / 20) -
Math.round(Math.abs(scores / 10));
                                if (currentScore >= 7) {
                                    newF();
                                    totalArr += 2;
                                    try {
                                        successSound.play().catch(function(e) {});
                                    } catch (err) {}
                                }

                                totalScore += currentScore;
                                gameScore.innerHTML = totalScore;

                                animateScore(currentScore, totalArr);

                                //board.y += scores;// + Math.floor(Math.random()*20);
```

```
                        boardY = board.y + scores;
                        if (scores >= 0) {
                            boardMove = true;
                        } else {
                            boardMove = false;
                        }

                        //this.fy += scores;
                    } else updArr(totalArr);
                    if (totalArr <= 0) {
                        clearInterval(intv);
                        try {
                            //bgSound.pause();
                            endSound.play().catch(function(e) {});
                        } catch (err) {}

document.getElementById("animCanvas").removeEventListener("click", shoot);
                        document.body.removeEventListener("keydown", shoot);
                        startPage.style.display = "block";
                        document.getElementById("title").innerHTML = "Your Score<br>"
+ totalScore;
                        if (bestScore < totalScore) {
                            bestScore = totalScore;
                            try {
                                highScoreSound.play().catch(function(e) {});
                            } catch (err) {}
                        }
                        document.getElementById("score").innerHTML = 0;
                        document.getElementById("best").innerHTML = bestScore;
                    }

                } else {
                    this.x += this.dx;
                }
            } else {
                this.x += this.dx;
            }
        }
    }
} else {
    ctx.fillRect(rope.x, arc.y - 3, 10, 6);
    ctx.fillRect(rope.x, arc.y - 1, this.w, 2);
    ctx.beginPath();
    ctx.moveTo(rope.x + this.w, arc.y - 4);
```

```
            ctx.lineTo(rope.x + this.w + 12, arc.y);
            ctx.lineTo(rope.x + this.w, arc.y + 4);
            ctx.fill();
        }
    }
}

// Arrow Move With Board

Arrow.prototype.moveArrowWithBoard = function(dir) {
    if (this == arrow1) {
        arrow1.fy += board.dy * dir;
    } else {
        arrow2.fy += board.dy * dir;
    }
}




var arrow1 = new Arrow();
var arrow2 = new Arrow();

var arrows = 0;
var moveArrowCheck = false;
var score = 0;

// Drawing functions...

function drawArc() {
    ctx.beginPath();
    ctx.arc(arc.x, arc.y, arc.r, arc.start, arc.end);
    ctx.strokeStyle = arc.color;
    ctx.lineWidth = arc.lw;
    ctx.stroke();
    ctx.closePath();
}

function drawRope() {
    ctx.beginPath();
    ctx.moveTo(arc.x, arc.y - arc.r);
    if (arrow1.vis && arrow2.vis) {
        ctx.lineTo(rope.x, arc.y);
    }
```

```
    ctx.lineTo(arc.x, arc.y + arc.r);
    ctx.lineWidth = rope.lw;
    ctx.strokeStyle = rope.color;
    ctx.stroke();
    ctx.closePath();
}

// Moving function...

function move() {
    ctx.clearRect(0, 0, w, h);
    if (arc.y > h - 50 || arc.y < 50) {
        arc.dy *= -1;
    }
    arc.y += arc.dy;
}

function shoot() {
    if (arrow1.vis && arrow2.vis && arrows != -1) {
        moveArrowCheck = true;
        if (arrows % 2 === 0) {
            arrow1.status = true;
            arrow1.fy = arc.y;
            arrow2.status = false;
            arrow2.x = rope.x;
            arrow2.vis = false;
        } else {
            arrow1.status = false;
            arrow2.fy = arc.y;
            arrow2.status = true;
            arrow1.x = rope.x;
            arrow1.vis = false;
        }
        totalArr--;
        try {
            shootSound.play().catch(function(e) {});
        } catch (err) {}
    }
    arrows++;
}

document.getElementById("animCanvas").addEventListener("click", shoot);
document.body.addEventListener("keydown", shoot);
```

```
        var intv = setInterval(function() {
            move();
            drawArc();
            drawRope();
            arrow1.drawArrow();
            arrow2.drawArrow();
            drawBoard();
        }, 15)
    }
}
//window.onload = setTimeout(loadGame,2000);
```