<html>

  <body>

    <canvas id="gameCanvas" style="border: 10px solid black;" width="1600" height="900"></canvas>

  </body>

  <script>

    let skyImg = new Image();

    skyImg.src = "sky.jpg";

    let birdImg = new Image();

    birdImg.src = "bird.png";

    let towerImg = new Image();

    towerImg.src = "tower.png";

    let score = 0;

    // Wait for background to load before starting

    skyImg.onload = () => {

      main();

    };

    async function main() {

      let canvas = document.getElementById("gameCanvas");

      let ctx = canvas.getContext("2d");

      let y = 0;

      let yVelocity = 0;

      let yAcc = 9.8;

      const TOWER\_WIDTH = 80;

      const GAP\_HEIGHT = 250;

      let bird = { x: 200, y: y, width: 60, height: 60 };

      // Towers array

      let towers = [{ x: 800, gapY: 300, passed: false }];

      function resetGame() {

        y = 0;

        yVelocity = 0;

        score = 0; // reset score

        towers = [{ x: 800, gapY: 300, passed: false }];

      }

      document.onkeydown = function () {

        yVelocity = -50;

      };

      for (let i = 0; i < 100000000; i++) {

        // Bird physics

        yVelocity += 0.33 \* yAcc;

        y += yVelocity \* 0.33;

        bird.y = y;

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

        // Draw background

        ctx.drawImage(skyImg, 0, 0, canvas.width, canvas.height);

        // Draw bird

        ctx.drawImage(birdImg, bird.x, bird.y, bird.width, bird.height);

        // Draw score

        ctx.fillStyle = "white";

        ctx.font = "50px Arial";

        ctx.fillText("Score: " + score, 50, 80);

        // --- Towers ---

        for (let t of towers) {

          // Top tower

          ctx.drawImage(towerImg, t.x, 0, TOWER\_WIDTH, t.gapY);

          // Bottom tower

          ctx.drawImage(towerImg, t.x, t.gapY + GAP\_HEIGHT, TOWER\_WIDTH, 900 - (t.gapY + GAP\_HEIGHT));

          // Move tower left

          t.x -= 3;

          // Check if bird passed the tower

          if (!t.passed && bird.x > t.x + TOWER\_WIDTH) {

            score++;

            t.passed = true;

          }

          // Collision

          if (

            (bird.x < t.x + TOWER\_WIDTH && bird.x + bird.width > t.x && bird.y < t.gapY) ||

            (bird.x < t.x + TOWER\_WIDTH && bird.x + bird.width > t.x && bird.y + bird.height > t.gapY + GAP\_HEIGHT)

          ) {

            resetGame();

          }

        }

        // Add new tower

        if (towers[towers.length - 1].x < 600) {

          let gapY = Math.floor(Math.random() \* 500) + 100;

          let lastTower = towers[towers.length - 1];

          towers.push({ x: lastTower.x + TOWER\_WIDTH + 220, gapY: gapY, passed: false });

        }

        // Remove off-screen towers

        if (towers[0].x + TOWER\_WIDTH < 0) {

          towers.shift();

        }

        // Ceiling and floor collision

        if (bird.y < 0 || bird.y + bird.height > 900) {

          resetGame();

        }

        await new Promise(r => setTimeout(r, 0.03333333 \* 1000));

      }

    }

  </script>

</html>