<https://www.youtube.com/watch?v=LprJOTU37hk> link vídeo com programa Dino replica

var canvas = document.getElementById('game');

var ctx = canvas.getContext('2d');

let score; let scoreText; let highscore; let highscoreText; let player; let gravity;

let obstacles = [ ]; let gameSpeed; let keys = { };

 //Lista de eventos

    document.addEventListener('keydown', function (evt)  {  keys[evt.code] = true;  });

 document.addEventListener('keyup', function (evt)  {  keys[evt.code] = false;  });

  class Player {  constructor (x, y, w, h, c) {

 this.x = x;  this.y = w;  this.w = w;     this.h= h;  this.c = c;  this.dy = 0;  this.jumpForce = 15;

    this.originalHeigth = h;     this.grounded = false;  this.jumpTimer = 0;  }

    Animate( ) { // Pular  if(keys['space'] || keys['keyw']){  this.Jump(); } else  { this.jumpTimer = 0;  }

    this.y += this.dy;     if(keys['ShiftLeft'] || keys['keyS']){  } else {  this.h = this.originalHeigth;  }

    this.y += this.dy;   this.Draw();  }

//Gravidade

Animate() {  if (this.y + this.h < canvas.heigth){  this.dy += gravity;

 this.grounded=false ; } else {  this.dy = 0;  this.grounded = true;  this.y = canvas.heigth - this.h;

 this.y += this.dy;   this.Draw();  }     }

    Jump () {

    if (this.grounded && this.jumpTimer == 0) {  this.jumpTimer=1;

    this.dy =- this.jumpForce;  } else if  (this.jumpTimer> 0 && this.jumpTimer > 15 )  {   this.jumpTimer++;

    this.dy = -this.jumpForce - (this.jumpTimer / 50);  }    }

    Draw () {     ctx.beginPath();     ctx.fillStyle = this.c;     ctx.font = this.s + "px sans-serif";

    ctx.textAlign = this.a;   ctx.fillRect(this.x, this.y, this.w, this.h);     ctx.closePath();     }     }

    function Start () {

    canvas.width = window.innerWidth;     canvas.heigth = window.innerHeigth;   ctx.font = "20px sans-serif";

    gameSpeed = 3;     gravity = 1;     score = 0;     highscore = 0;

    player = new Player(25,150, 50, 50, '#FF5858');

    scoreText = new Text ("Score:" + score, 25, 25, "left", "#993333", "20");

    requestAnimationFrame(Update);    }

    //Obstáculos

    class Obstacle { constructor (x, y, w, h, c)  { this.x = x;   this.y = y;   this.w = w;  this.h = h;     this.c = c;    this.dx = -gameSpeed;  }

    Update ()  {  this.x += this.dx; this.Draw();   this.dx = -gameSpeed;   }

       Draw () {  ctx.beginPath();   ctx.fillStyle = this.c;  ctx.font = this.s + "px sans-serif";  ctx.textAlign = this.a;

       ctx.fillRect(this.x, this.y, this.w, this.h);    ctx.closePath();         }     }

    class Text {   constructor (t,x,y,a,c,s) {   this.t = t;   this.x = x;    this.y = y;   this.a = a;  this.c = c;  this.s = s;  }

    Draw () {   ctx.beginPath();   ctx.fillStyle = this.c;  ctx.font = this.s + "px sans-serif";     ctx.textAlign = this.a;

     ctx.fillText(this.text, this.x, this.y);       ctx.closePath();  } }

    //Função pontos Obstáculos

    function SpawnObstacle () {     let size = RandomIntInRange(20, 70);   let type  = RandomIntInRange(0, 1);

    let obstacle = new Obstacle(canvas.with + size, canvas.heigth - size,  size, size, '#CC6699');

        if (type == 1) {     obstacle.y -= player.originalHeigth - 10;    obstacles.push(obstacle);        }   }

    function RandomIntInRange (min, max) {      return Math.round(Math.random() \* (max - min) + min);     }

    function Start () {     canvas.width = window.innerWidth;    canvas.heigth = window.innerHeigth;

        ctx.font = "20px sans-serif";     gameSpeed = 3;  gravity = 1  score = 0;   highscore = 0;

        player = new Player(25,150, 50, 50, '#99FF00');

        scoreText = new Text ("Score:" + score, 25, 25, "left", "#CC6699", "20");

        requestAnimationFrame(Update);  }  let initialspawnTimer = 200;  let spawnTimer =  initialspawnTimer;

 function Update()  {  requestAnimationFrame(Update);    ctx.clearRect(0, 0, canvas.width, canvas.heigth);

        spawnTimer--;        if (spawnTimer <= 0 ) {   SpawnObstacle();   spawnTimer = initialspawnTimer - gameSpeed \* 8;

        if (spawnTimer < 60) {   spawnTimer = 60;    }    }

        for (let i = 0; i < obstacles.length; i++) {    let o = obstacles[i];       if (o.x + o.w < 0) {    obstacles.splice(i,1);

        o.Update();     }

        player.Animate();

        score++;   scoreText.t = "Score:" + score;  scoreText.Draw();   player.Draw();     player.x++;   }

    player = new Player(25, 0, 50,50, '#FF5858');

    scoreText = new Text("Score:" + score, 25, 25, 'left', '#99FF00', '20');

    highscoreText = new Text ("Highscore" + highscore, canvas.with - 25, 25, 'rigth' , '#FFCC66', '20');

    requestAnimationFrame(Update);  for (let i = 0; i < obstacles.length; i++)  {  let o = obstacles[i];

    if (o.x + o.w < 0) {  obstacles.splice(i,1);   }   if (player.x < o.x + o.w &&    player.x + player.w > o.x &&

    player.y < o.y + o.h &&   player.y + player.h > o.y) {     obstacle = [ ];     score = 0;     spawnTimer = initialSpawnTimer;

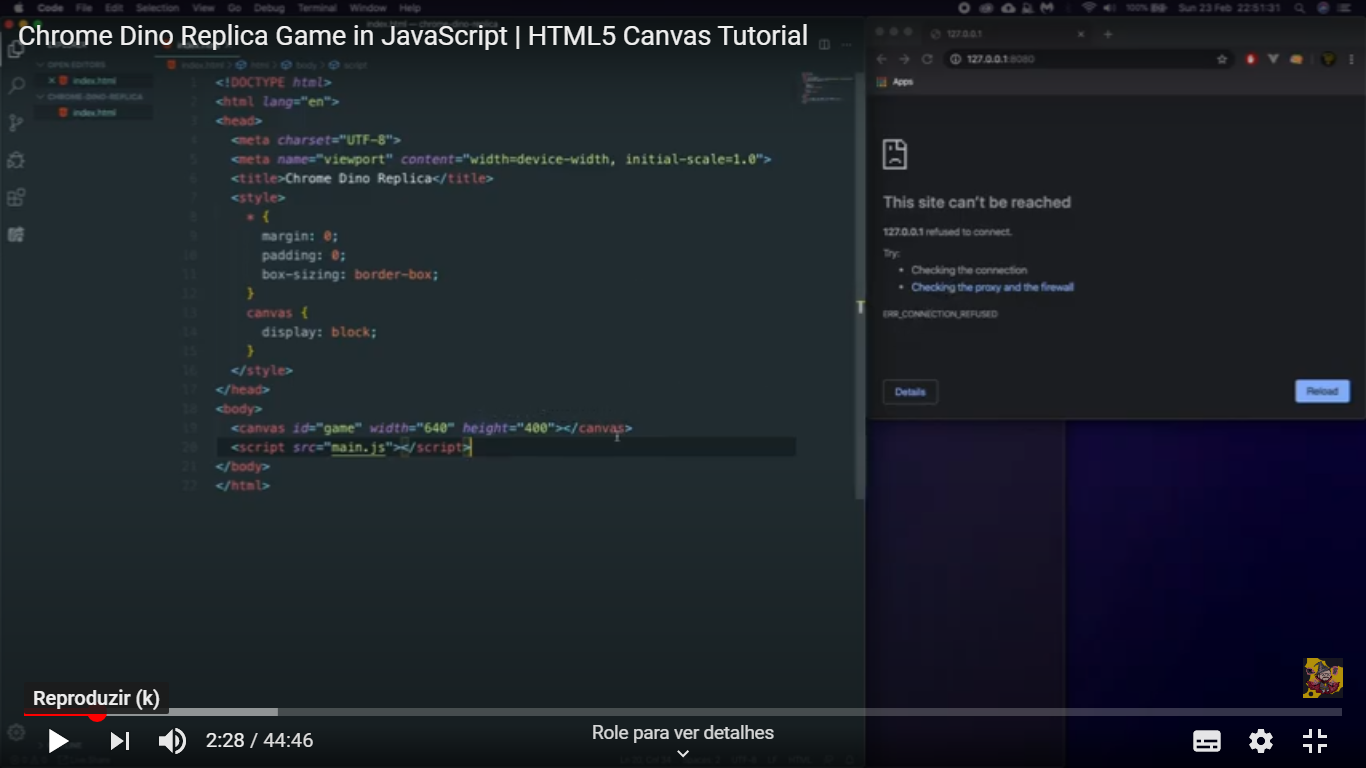
    gameSpeed = 3;   window.localStorage.setItem('highscore' , highscore);     }

    o.Update();    player.Animate();   score++;  scoreText.t = "Score:" +score;  scoreText.Draw();

    if (score > highscore)  {    highscore = score;  highscoreText.t = "Highscore:" + highscore;  }  highscoreText.Draw();

    gameSpeed += 0.003; }   Start();

}



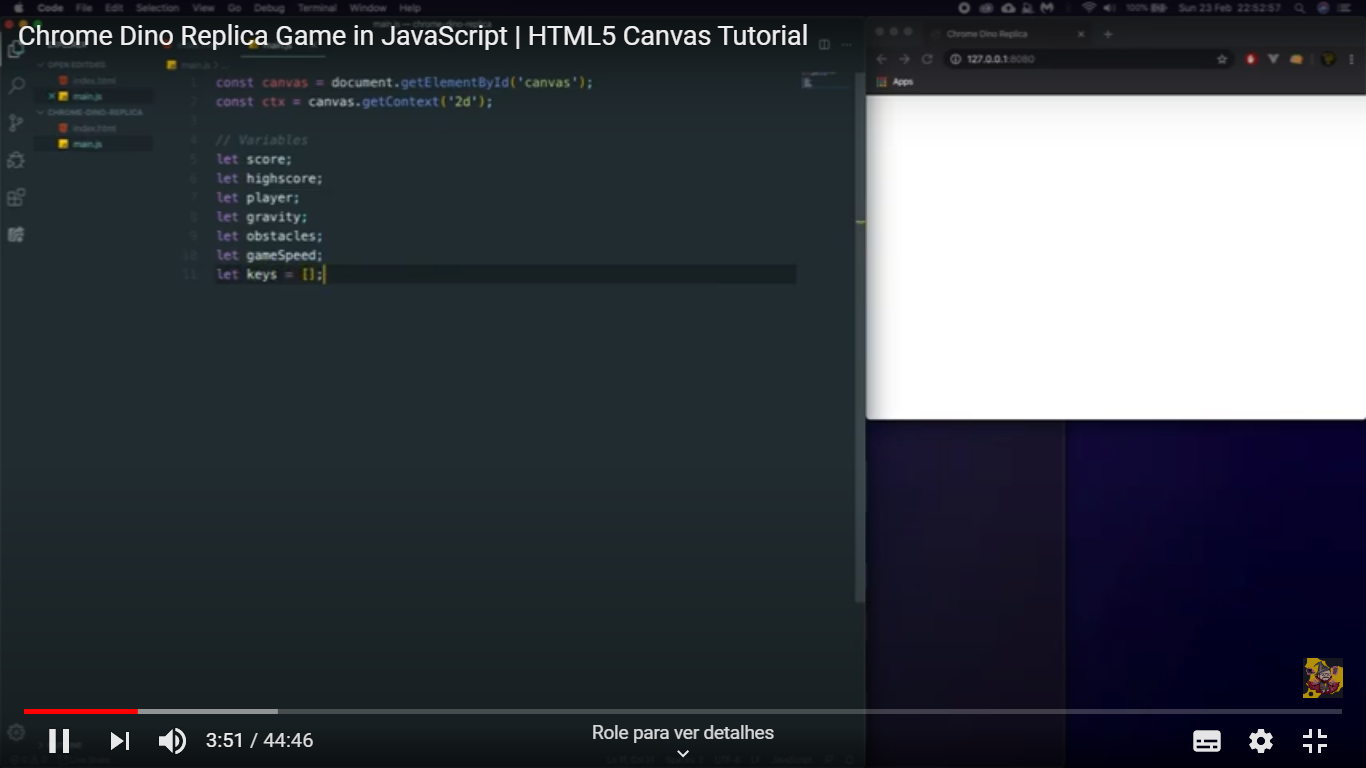
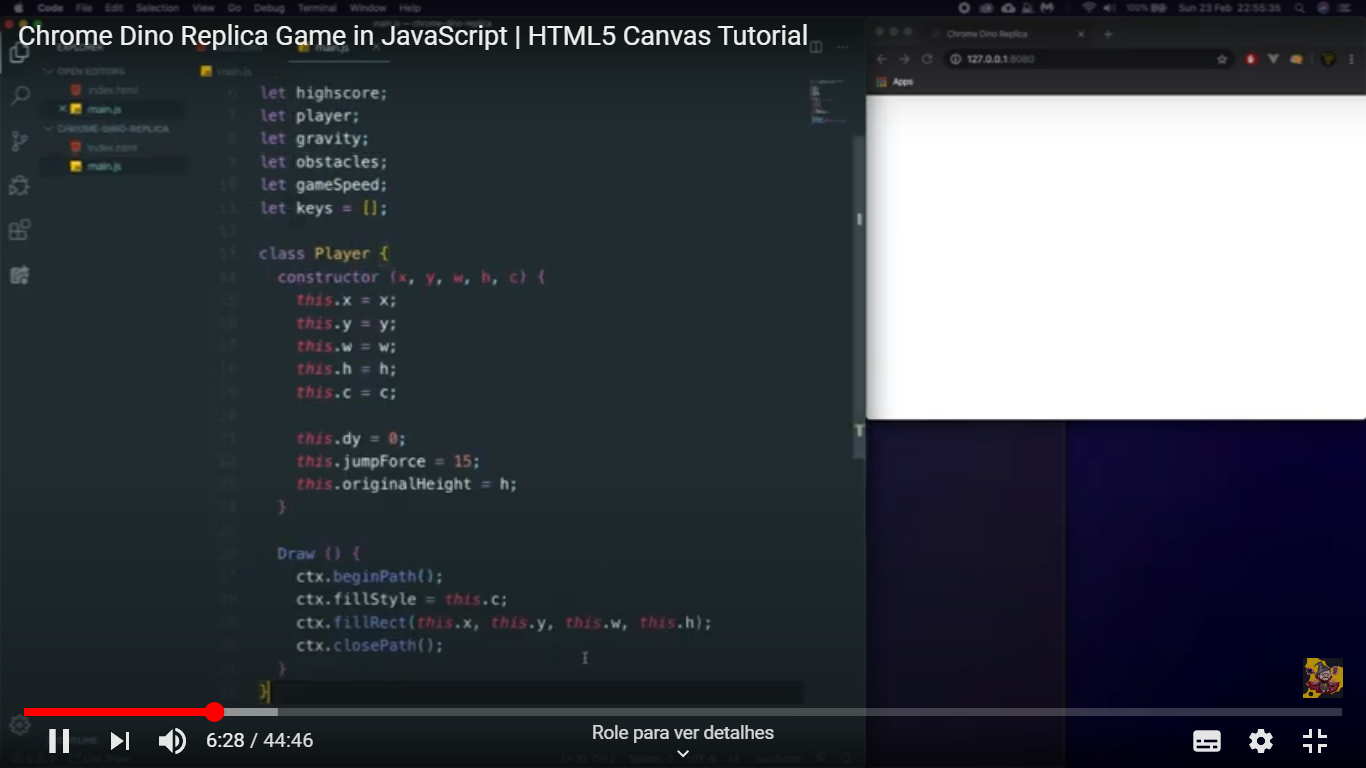


Figura trocar ("canvas") por ("game")



# 

