var PLAY = 1;

var END = 0;

var gameState = PLAY;

var trex, trex\_running, trex\_collided;

var ground, invisibleGround, groundImage;

var cloudsGroup, cloudImage;

var obstaclesGroup, obstacle1, obstacle2, obstacle3, obstacle4, obstacle5, obstacle6;

var score=0;

var gameOver, restart;

localStorage["HighestScore"] = 0;

function preload(){

trex\_running = loadAnimation("trex1.png","trex3.png","trex4.png");

trex\_collided = loadAnimation("trex\_collided.png");

groundImage = loadImage("ground2.png");

cloudImage = loadImage("cloud.png");

obstacle1 = loadImage("obstacle1.png");

obstacle2 = loadImage("obstacle2.png");

obstacle3 = loadImage("obstacle3.png");

obstacle4 = loadImage("obstacle4.png");

obstacle5 = loadImage("obstacle5.png");

obstacle6 = loadImage("obstacle6.png");

gameOverImg = loadImage("gameOver.png");

restartImg = loadImage("restart.png");

}

function setup() {

createCanvas(600, 200);

trex = createSprite(50,180,20,50);

trex.addAnimation("running", trex\_running);

trex.addAnimation("collided", trex\_collided);

trex.scale = 0.5;

ground = createSprite(200,180,400,20);

ground.addImage("ground",groundImage);

ground.x = ground.width /2;

ground.velocityX = -(6 + 3\*score/100);

gameOver = createSprite(300,100);

gameOver.addImage(gameOverImg);

restart = createSprite(300,140);

restart.addImage(restartImg);

gameOver.scale = 0.5;

restart.scale = 0.5;

gameOver.visible = false;

restart.visible = false;

invisibleGround = createSprite(200,190,400,10);

invisibleGround.visible = false;

cloudsGroup = new Group();

obstaclesGroup = new Group();

score = 0;

}

function draw() {

//trex.debug = true;

background(‘pink’);

text("Score: "+ score, 500,50);

if (gameState===PLAY){

score = score + Math.round(getFrameRate()/60);

ground.velocityX = -(6 + 3\*score/100);

if(keyDown("space") && trex.y >= 159) {

trex.velocityY = -12;

}

trex.velocityY = trex.velocityY + 0.8

if (ground.x < 0){

ground.x = ground.width/2;

}

trex.collide(invisibleGround);

spawnClouds();

spawnObstacles();

if(obstaclesGroup.isTouching(trex)){

gameState = END;

}

}

else if (gameState === END) {

gameOver.visible = true;

restart.visible = true;

//set velcity of each game object to 0

ground.velocityX = 0;

trex.velocityY = 0;

obstaclesGroup.setVelocityXEach(0);

cloudsGroup.setVelocityXEach(0);

//change the trex animation

trex.changeAnimation("collided",trex\_collided);

//set lifetime of the game objects so that they are never destroyed

obstaclesGroup.setLifetimeEach(-1);

cloudsGroup.setLifetimeEach(-1);

if(mousePressedOver(restart)) {

reset();

}

}

drawSprites();

}

function spawnClouds() {

//write code here to spawn the clouds

if (frameCount % 60 === 0) {

var cloud = createSprite(600,120,40,10);

cloud.y = Math.round(random(80,120));

cloud.addImage(cloudImage);

cloud.scale = 0.5;

cloud.velocityX = -3;

//assign lifetime to the variable

cloud.lifetime = 200;

//adjust the depth

cloud.depth = trex.depth;

trex.depth = trex.depth + 1;

//add each cloud to the group

cloudsGroup.add(cloud);

}

}

function spawnObstacles() {

if(frameCount % 60 === 0) {

var obstacle = createSprite(600,165,10,40);

//obstacle.debug = true;

obstacle.velocityX = -(6 + 3\*score/100);

//generate random obstacles

var rand = Math.round(random(1,6));

switch(rand) {

case 1: obstacle.addImage(obstacle1);

break;

case 2: obstacle.addImage(obstacle2);

break;

case 3: obstacle.addImage(obstacle3);

break;

case 4: obstacle.addImage(obstacle4);

break;

case 5: obstacle.addImage(obstacle5);

break;

case 6: obstacle.addImage(obstacle6);

break;

default: break;

}

//assign scale and lifetime to the obstacle

obstacle.scale = 0.5;

obstacle.lifetime = 300;

//add each obstacle to the group

obstaclesGroup.add(obstacle);

}

}

function reset(){

gameState = PLAY;

gameOver.visible = false;

restart.visible = false;

obstaclesGroup.destroyEach();

cloudsGroup.destroyEach();

trex.changeAnimation("running",trex\_running);

if(localStorage["HighestScore"]<score){

localStorage["HighestScore"] = score;

}

console.log(localStorage["HighestScore"]);

score = 0;

}