**CS Design Project**

**Drag Race Simulator**

The game, *Drag Race Simulator*, was created by Leo, Rodrigo, and Esteban. The game we created is a simulation of driving a Porsche 911GT3RS, testing its speed. In the development of this project, we used the functions of: float, integer, boolean, sound and img.

Directions on how to play *Drag Race Simulator*

1. Accelerate the car with the letter “W” on your keyboard.
2. Turn the car any way you want. Using the letter “A” on your keyboard to turn to the left, and pressing the letter “D” to turn to the right.
3. To reduce speed or stop the car, press the “spacebar” on your keyboard.
4. To use turbo you can accelerate even more by pressing “T”.
5. To start and stop the stopwatch you use “O” and “P”
6. To reset the game you press “R”.

Code Explanation

float speed = 0;

float acceleration = 0.4; // the acceleration of the car

float deceleration = 0.1; // the decelaration of the car

float brakeDeceleration = 0.5; // deceleration of the car when the brake is pressed

int gear = 1;

float r;

float[] gearRatios = {2.5, 1.8, 1.2, 0.8};

PImage img;

PImage img2;

PImage car;

PImage logo;

PImage logo2;

PImage SM;

PImage backgroundImage;

PImage resizedImage; // Variable to hold the resized image

int bgX1, bgY1, bgX2, bgY2;

int scrollSpeed = 2; // Increase the value for faster movement

int newWidth = 800; // New width of the image

int newHeight; // New height of the image (calculated)

int gW = 20;

int mvmt = 200;

boolean restartButtonPressed = false;

boolean resetKeyPressed = false;

boolean turboMode = false;

float turboAcceleration = acceleration + 5;

int leftBoundary = 100;

int rightBoundary = 700;

int startTime = 0; // Variable to store the starting time in milliseconds

int elapsedTime = 0; // Variable to store the elapsed time in milliseconds

boolean isRunning = false; // Flag to indicate if the stopwatch is running

These variables decide the image, the sound, the gears, the boundaries, the start and stop time of the stopwatch and also the acceleration and deceleration of the car.

void setup() {

size(800, 800);

file = new SoundFile(this,"sound1.aiff");

file.play();

file.amp(.5);

img = loadImage("accelerator-removebg-preview.png");

car = loadImage ("Porshe\_911\_GT3rs-removebg-preview.png");

img2 = loadImage ("brake-removebg-preview.png");

logo = loadImage ("Porsche2.png");

logo2 = loadImage("gt3rs-removebg-preview.png");

SM = loadImage ("pngtree-speedometer-tachometer-and-fuel-gauge-png-image\_4773520-removebg-preview.png");

img.resize(50, 140);

img2.resize(120,140);

car.resize(75, 150);

logo.resize (40, 40);

logo2.resize (200, 60);

SM.resize (130, 110);

backgroundImage = loadImage("background.png");

// Resize the image to fit the canvas width and calculate the new height to maintain aspect ratio

//float aspectRatio = float(backgroundImage.height) / float(backgroundImage.width);

newHeight = int(newWidth );

resizedImage = backgroundImage.copy();

resizedImage.resize(newWidth, newHeight);

// Initialize the positions of the background images

bgX1 = 0;

bgY1 = 0;

bgX2 = 0;

bgY2 = newHeight; // Start the second image below the first image

}

In the void setup the images and sounds are loaded, and also the size of that the resize should be, specifically the new x, y coordinates.

void keyPressed() {

if (key == 'r' || key == 'R') {

resetKeyPressed = true;

isRunning = false;

}

if (key == 't' || key == 'T') {

turboMode = !turboMode;

}

if ((key == 'o' || key == 'O') && !isRunning) {

if (!isRunning) {

startTime = millis() - elapsedTime;

isRunning = true;

}

}

// Stop the stopwatch

if ((key == 'p' || key == 'P')&& isRunning) {

if (isRunning) {

elapsedTime = millis() - startTime;

isRunning = false;

}

}

}

In this void keyPressed it explains that when r is pressed then resetKey is true. Also if t is pressed then turbo mode is enabled. Finally o is used to start the stopwatch and p to stop it.

void draw() {

background(255);

updateSpeed();

drawSpeedometer();

drawGearRatio();

fill(255, 0, 0);

rect(700, 20, 80, 30); // Restart button position and size

textAlign(CENTER, CENTER);

fill(255);

textSize(16);

text("Restart", 740, 35);

// Check if the restart button is pressed

if (mousePressed && mouseX >= 700 && mouseX <= 780 && mouseY >= 20 && mouseY <= 50) {

reset();

}

if (resetKeyPressed) {

reset();

resetKeyPressed = false; // Reset the reset key press state

}

if (isRunning) {

elapsedTime = millis() - startTime;

int seconds = elapsedTime / 1000;

int milliseconds = elapsedTime % 1000;

// Display the stopwatch time

textSize(24);

fill(255);

textAlign(CENTER);

text(nf(seconds, 2) + ":" + nf(milliseconds, 3), 40, 25);

}

}

In this void draw the background is decided, and also the speed is updated. The speedometer, restart button,gear ratio and the digits of the stopwatch are drawn.

void reset() {

// Reset relevant variables to their initial values

speed = 0;

elapsedTime = 0;

// Add other variables that need to be reset

// Reset the restart button state

restartButtonPressed = false;

}

In the void reset the speed and elapsed time variables are reseted to 0.

void updateSpeed() {

if (turboMode) {

speed = constrain(speed, 0, 220);

} else {

speed = constrain(speed, 0, 180); // Regular speed constraint: 0 to 180 mph

}

if (mvmt + car.width > rightBoundary) {

mvmt = rightBoundary - car.width;

} else if (mvmt < leftBoundary) {

mvmt = leftBoundary;

}

if (mousePressed && mouseX >= 720 && mouseX <=770 && mouseY >= 655 && mouseY <= 780) {

speed += acceleration;

img.resize (40,118);

} else if (mousePressed && mouseX >= 580 && mouseX <= 700 && mouseY >= 750 && mouseY <= 780) {

speed -= brakeDeceleration;

img2.resize (115,130);

} else {

speed -= deceleration;

if (speed < 0) {

speed = 0;

}

{

{

if (keyPressed && (key == 'w' || key == 'W')) {

speed += acceleration;

} else if (keyPressed && (key == ' ' || key == ' ')) {

speed -= brakeDeceleration;

} else {

speed -= deceleration;

if (speed < 0) {

speed = 0;

}

speed = constrain(speed, 0, 180); // Limit the speed between 0 and 180 mph

}

{

if (keyPressed && (key == 'd' || key == 'D')) {

mvmt +=2;

speed += acceleration;

} else if (keyPressed && (key == 'a' || key == 'A')){

mvmt -=2;

speed += acceleration;

} else {

}

}

img.resize(50, 140);

img2.resize(120,140);

}

}

}

}

In void update speed the speed of the turbo mode is decided. Also the boundaries of the road are decided, which means that the car can’t go over the road barriers. Also when w is pressed the scroll speed should accelerate by 1. When the SPACE button is pressed the speed is decelerated by 1. When D is pressed the car goes right, and when a the car goes left. If this doesn’t happen then image1 and imag2 should resize.

void drawSpeedometer() {

// Draw the speedometer background

// Wrap-around effect for the first image

if (bgY1 >= newHeight) {

bgY1 = bgY2 - newHeight;

}

// Wrap around effect for the second image

if (bgY2 >= newHeight) {

bgY2 = bgY1 - newHeight;

}

// Draw the background images

image(resizedImage, bgX1, bgY1);

image(resizedImage, bgX2, bgY2);

//left side of car controls

fill (115,115,124);

quad (0, 640, 130, 640, 315, 800, 0, 800);

fill(60,74,157);

quad (5, 765, 35, 655, 85, 655, 115, 765);

textSize (14);

stroke (255,0,0);

strokeWeight (2);

fill (60,74,157);

rect (gW, 740, 12, 15);

stroke (0);

fill (255);

text (" N 1 2 3 4 5 6", 60, 745);

//right side controls

fill (115,115,124);

quad (150, 800, 600, 640, 800,640, 800, 800);

fill (50,50,50);

quad (550, 795, 600, 655, 785, 655, 785,795);

//border linement

fill (160, 160, 160);

rect (0, 625,130, 15);

quad (130, 640, 130, 625, 200, 685, 200, 700);

rect (200, 685, 240, 15);

quad (440,700, 440, 685, 600, 625, 600, 640);

rect (600, 625, 200, 15);

fill (232,232,232);

strokeWeight (0);

quad (200, 685, 250, 660, 390, 660, 440, 685);

//center controls

strokeWeight (1);

fill (78, 80, 90);

quad (140,800, 190, 700, 450, 700, 500, 800);

fill (70, 60, 80);

quad (140,800, 130, 800, 180, 700, 190, 700);

quad (450, 700, 460, 700, 510, 800, 500, 800);

fill(60,74,157);

strokeWeight (4);

ellipse (400, 800, 150, 150);

strokeWeight (0);

fill (203,51,51);

ellipse (400,800, 10,10);

stroke (0);

strokeWeight (1);

fill (18,61,88);

rect (185, 730, 130, 470 );

fill (0);

rect (190, 740, 120, 50);

fill (63,67,70);

strokeWeight (2);

quad (185, 730, 205, 720, 295, 720, 315, 730);

image (logo, 380, 750);

image (logo2, 220, 680);

image (SM, 255, 610);

// Draw the speedometer needle

pushMatrix();

translate(400,800);

float angle = map(speed, 0, 180, -1 \* PI , PI / 4);

rotate(angle);

strokeWeight(3);

stroke (203,51,51);

line(0, 0, 50, 0);

popMatrix();

stroke (0);

strokeWeight (3);

// Display the speed value

textAlign(CENTER, CENTER);

textSize(36);

fill(255,255,255);

text( + int(speed) + " mph", 250, 750);

image(img, 720, 655);

image (car, mvmt, 430);

image (img2, 580, 655);

//speed numbers

textSize (12);

textAlign (CENTER, CENTER);

text ("10", 337, 782);

text ("30", 350, 760);

text ("50", 370, 740);

text ("70", 400, 732);

text ("90", 430, 740);

text ("110", 451, 760);

text ("130", 460, 782);

stroke (255);

strokeWeight (3);

line (335, 769, 340, 771); //20mph

line (353, 745, 358, 751); //40mph

line (383, 730, 385, 740); //60mph

line (417, 730, 415, 740); //80mph

line (447, 745, 442, 751); //100mph

line (335, 769, 340, 771); //120mph

line (465, 769, 460, 771); //14mph

stroke (0);

strokeWeight (1.5);

}

In the void draw speedometer the wrap around effect of the image is done. Also all the controls that aren’t loaded images are drawn like the gears, the control panel, the speedometer needle, the speed value and also the speed numbers.

void drawGearRatio() {

// Calculate the gear ratio based on the speed

if (speed < 1) {

gear = 0;

bgY1 += scrollSpeed \*0;

bgY2 += scrollSpeed \*0;

textSize (48);

textAlign (CENTER, CENTER);

fill (255);

text ("N", 60, 700);

gW = 17;

} else if (speed < 5) {

gear = 1;

bgY1 += scrollSpeed;

bgY2 += scrollSpeed;

textSize (48);

textAlign (CENTER, CENTER);

text ("1", 60, 700);

gW = 30;

} else if (speed < 10) {

gear = 1;

bgY1 += scrollSpeed\*2;

bgY2 += scrollSpeed\*2;

textSize (48);

textAlign (CENTER, CENTER);

text ("1", 60, 700);

gW = 30;

}else if (speed < 15) {

gear = 1;

bgY1 += scrollSpeed\*3;

bgY2 += scrollSpeed\*3;

textSize (48);

textAlign (CENTER, CENTER);

text ("1", 60, 700);

gW = 30;

}else if (speed < 20) {

gear = 1;

bgY1 += scrollSpeed\*4;

bgY2 += scrollSpeed\*4;

textSize (48);

textAlign (CENTER, CENTER);

text ("1", 60, 700);

gW = 30;

}else if (speed < 25) {

gear = 1;

bgY1 += scrollSpeed\*5;

bgY2 += scrollSpeed\*5;

textSize (48);

textAlign (CENTER, CENTER);

text ("1", 60, 700);

gW = 30;

}else if (speed < 30) {

gear = 1;

bgY1 += scrollSpeed\*6;

bgY2 += scrollSpeed\*6;

textSize (48);

textAlign (CENTER, CENTER);

text ("1", 60, 700);

gW = 30;

}else if (speed < 35) {

gear = 2;

bgY1 += scrollSpeed\*7;

bgY2 += scrollSpeed\*7;

textSize (48);

textAlign (CENTER, CENTER);

text ("2", 60, 700);

gW = 43;

}else if (speed < 40) {

gear = 2;

bgY1 += scrollSpeed\*8;

bgY2 += scrollSpeed\*8;

textSize (48);

textAlign (CENTER, CENTER);

text ("2", 60, 700);

gW = 43;

}else if (speed < 45) {

gear = 2;

bgY1 += scrollSpeed\*9;

bgY2 += scrollSpeed\*9;

textSize (48);

textAlign (CENTER, CENTER);

text ("2", 60, 700);

gW = 43;

}else if (speed < 50) {

gear = 2;

bgY1 += scrollSpeed\*10;

bgY2 += scrollSpeed\*10;

textSize (48);

textAlign (CENTER, CENTER);

text ("2", 60, 700);

gW = 43;

}else if (speed < 55) {

gear = 2;

bgY1 += scrollSpeed\*11;

bgY2 += scrollSpeed\*11;

textSize (48);

textAlign (CENTER, CENTER);

text ("2", 60, 700);

gW = 43;

}else if (speed < 60) {

gear = 2;

bgY1 += scrollSpeed\*12;

bgY2 += scrollSpeed\*12;

textSize (48);

textAlign (CENTER, CENTER);

text ("2", 60, 700);

gW = 43;

}else if (speed < 65) {

gear = 3;

bgY1 += scrollSpeed\*13;

bgY2 += scrollSpeed\*13;

textSize (48);

textAlign (CENTER, CENTER);

text ("3", 60, 700);

gW = 56;

}else if (speed < 70) {

gear = 3;

bgY1 += scrollSpeed\*14;

bgY2 += scrollSpeed\*14;

textSize (48);

textAlign (CENTER, CENTER);

text ("3", 60, 700);

gW = 56;

}else if (speed < 75) {

gear = 3;

bgY1 += scrollSpeed\*15;

bgY2 += scrollSpeed\*15;

textSize (48);

textAlign (CENTER, CENTER);

text ("3", 60, 700);

gW = 56;

}else if (speed < 80) {

gear = 3;

bgY1 += scrollSpeed\*16;

bgY2 += scrollSpeed\*16;

textSize (48);

textAlign (CENTER, CENTER);

text ("3", 60, 700);

gW = 56;

}else if (speed < 85) {

gear = 3;

bgY1 += scrollSpeed\*17;

bgY2 += scrollSpeed\*17;

textSize (48);

textAlign (CENTER, CENTER);

text ("3", 60, 700);

gW = 56;

}else if (speed < 90) {

gear = 3;

bgY1 += scrollSpeed\*18;

bgY2 += scrollSpeed\*18;

textSize (48);

textAlign (CENTER, CENTER);

text ("3", 60, 700);

gW = 56;

}else if (speed < 95) {

gear = 4;

bgY1 += scrollSpeed\*19;

bgY2 += scrollSpeed\*19;

textSize (48);

textAlign (CENTER, CENTER);

text ("4", 60, 700);

gW = 69;

}else if (speed < 100) {

gear = 4;

bgY1 += scrollSpeed\*20;

bgY2 += scrollSpeed\*20;

textSize (48);

textAlign (CENTER, CENTER);

text ("4", 60, 700);

gW = 69;

}else if (speed < 105) {

gear = 4;

bgY1 += scrollSpeed\*21;

bgY2 += scrollSpeed\*21;

textSize (48);

textAlign (CENTER, CENTER);

text ("4", 60, 700);

gW = 69;

}else if (speed < 110) {

gear = 4;

bgY1 += scrollSpeed\*22;

bgY2 += scrollSpeed\*22;

textSize (48);

textAlign (CENTER, CENTER);

text ("4", 60, 700);

gW = 69;

}else if (speed < 115) {

gear = 4;

bgY1 += scrollSpeed\*23;

bgY2 += scrollSpeed\*23;

textSize (48);

textAlign (CENTER, CENTER);

text ("4", 60, 700);

gW = 69;

}else if (speed < 120) {

gear = 4;

bgY1 += scrollSpeed\*24;

bgY2 += scrollSpeed\*24;

textSize (48);

textAlign (CENTER, CENTER);

text ("4", 60, 700);

gW = 69;

}else if (speed < 125) {

gear = 5;

bgY1 += scrollSpeed\*25;

bgY2 += scrollSpeed\*25;

textSize (48);

textAlign (CENTER, CENTER);

text ("5", 60, 700);

gW = 82;

}else if (speed < 130) {

gear = 5;

bgY1 += scrollSpeed\*26;

bgY2 += scrollSpeed\*26;

textSize (48);

textAlign (CENTER, CENTER);

text ("5", 60, 700);

gW = 82;

}else if (speed < 135) {

gear = 5;

bgY1 += scrollSpeed\*27;

bgY2 += scrollSpeed\*27;

textSize (48);

textAlign (CENTER, CENTER);

text ("5", 60, 700);

gW = 82;

}else if (speed < 140) {

gear = 5;

bgY1 += scrollSpeed\*28;

bgY2 += scrollSpeed\*28;

textSize (48);

textAlign (CENTER, CENTER);

text ("5", 60, 700);

gW = 82;

}else if (speed < 145) {

gear = 5;

bgY1 += scrollSpeed\*29;

bgY2 += scrollSpeed\*29;

textSize (48);

textAlign (CENTER, CENTER);

text ("5", 60, 700);

gW = 82;

}else if (speed < 150) {

gear = 5;

bgY1 += scrollSpeed\*30;

bgY2 += scrollSpeed\*30;

textSize (48);

textAlign (CENTER, CENTER);

text ("5", 60, 700);

gW = 82;

}else if (speed < 155) {

gear = 6;

bgY1 += scrollSpeed\*31;

bgY2 += scrollSpeed\*31;

textSize (48);

textAlign (CENTER, CENTER);

text ("6", 60, 700);

gW = 95;

}else if (speed < 160) {

gear = 6;

bgY1 += scrollSpeed\*32;

bgY2 += scrollSpeed\*32;

textSize (48);

textAlign (CENTER, CENTER);

text ("6", 60, 700);

gW = 95;

}else if (speed < 165) {

gear = 6;

bgY1 += scrollSpeed\*33;

bgY2 += scrollSpeed\*33;

textSize (48);

textAlign (CENTER, CENTER);

text ("6", 60, 700);

gW = 95;

}else if (speed < 170) {

gear = 6;

bgY1 += scrollSpeed\*34;

bgY2 += scrollSpeed\*34;

textSize (48);

textAlign (CENTER, CENTER);

text ("6", 60, 700);

gW = 95;

}else if (speed < 175) {

gear = 6;

bgY1 += scrollSpeed\*35;

bgY2 += scrollSpeed\*35;

textSize (48);

textAlign (CENTER, CENTER);

text ("6", 60, 700);

gW = 95;

} else {

gear = 6;

bgY1 += scrollSpeed\*36;

bgY2 += scrollSpeed\*36;

textSize (48);

textAlign (CENTER, CENTER);

text ("6", 60, 700);

gW = 95;

}

}

In the void draw gear ratio function the type of gear used is decided.

.