Projeto Freeway

Iniciando o projeto

Queremos recriar o jogo Freeway, onde uma galinha tenta atravessar uma rua com várias faixas e carros passando usando a linguagem Javascript.

Para editar nosso código, vamos utilizar o editor online do P5 Js. Porém, assim que abrimos a página p5, temos apenas um fundo cinza, conforme ilustra a imagem abaixo, e não é isso que queremos.

let imagemDaEstrada;

let imagemDoAtor;

let imagemCarro;

*// carro*

let xCarro = 600;

*// ator*

let yAtor = 366;

function preload() {

  imagemDaEstrada = loadImage("image/estrada.png");

  imagemDoAtor = loadImage("image/ator-1.png");

  imagemCarro = loadImage("image/carro-1.png");

}

function setup() {

  createCanvas(500, 400);

}

function draw() {

  background(imagemDaEstrada);

  mostraAtor();

  mostraCarro();

  movimentaCarro();

  movimentaAtor();

  verificaColisaoCarro();

}

function mostraAtor() {

  image(imagemDoAtor, 100, yAtor, 30, 30);

}

function mostraCarro() {

  image(imagemCarro, xCarro, 40, 50, 40);

}

function movimentaCarro() {

  xCarro -= 2;

*if* (xCarro < -50) {

    xCarro = 600; *// Reseta a posição do carro quando ele sai da tela*

  }

}

function movimentaAtor() {

*if* (keyIsDown(UP\_ARROW)) {

*if* (yAtor > 0) { *// Limita o movimento para cima*

      yAtor -= 3;

    }

  }

*if* (keyIsDown(DOWN\_ARROW)) {

*if* (yAtor < height - 30) { *// Limita o movimento para baixo*

      yAtor += 3;

    }

  }

}

function verificaColisaoCarro() {

*// Aqui você pode adicionar a lógica de colisão, caso queira adicionar essa funcionalidade*

}

O arquivo que contém o código do jogo, o sketch.js está com 48 linhas, com:

* código do ator e do carro,
* carregamento de imagens,
* funções referentes ao ator e outras referentes ao carro.

Isto não está legal...

À medida que o nosso jogo vai ganhando novas funcionalidades, ficará difícil manter ou editar o código.

Pensando nisso, divida o código criando mais arquivos js, como:

* carro.js para manter todo código referente ao carro,
* imagens.js para manter o código relacionado ao carregamento das imagens
* ator.js para o código do ator do jogo.

Além disso, adicione mais 2 carros e faça com que eles reapareçam na tela após o término da faixa em que estão!

* sketch.js:
* function setup() {
* createCanvas(500, 400);
* }
* function draw() {
* background(imagemDaEstrada);
* mostraAtor();
* mostraCarro();
* movimentaCarro();
* movimentaAtor();
* voltaPosicaoInicialDoCarro();
* }
* ator.js:
* //código do ator
* **let** xAtor = 100;
* **let** yAtor = 366;
* function **mostraAtor**(){
* image(imagemDoAtor, xAtor, yAtor, 30, 30);
* }
* function **movimentaAtor**(){
* **if** (keyIsDown(UP\_ARROW)){
* yAtor -= 3;
* }
* **if** (keyIsDown(DOWN\_ARROW)){
* yAtor += 3;
* }
* }
* carro.js:
* //código do carro
* //carro 1
* **let** xCarro = 600;
* **let** yCarro = 40;
* **let** velocidadeCarro1 = 2;
* //carro 2
* **let** xCarro2 = 600;
* **let** yCarro2 = 96;
* **let** velocidadeCarro2 = 2.5;
* //carro 3
* **let** xCarro3 = 600;
* **let** yCarro3 = 150;
* **let** velocidadeCarro3 = 3.2;
* function **mostraCarro**(){
* image(imagemCarro, xCarro, yCarro, 50, 40);
* image(imagemCarro2, xCarro2, yCarro2, 50, 40);
* image(imagemCarro3, xCarro3, yCarro3, 50, 40);
* }
* function **movimentaCarro**(){
* xCarro -= velocidadeCarro1;
* xCarro2 -= velocidadeCarro2;
* xCarro3 -= velocidadeCarro3;
* }
* function **voltaPosicaoInicialDoCarro**(){
* **if** (xCarro < -50){
* xCarro = 600
* }
* **if** (xCarro2 < -50){
* xCarro2 = 600
* }
* **if** (xCarro3 < -50){
* xCarro3 = 600
* }
* }
* imagens.js:
* // imagens do jogo
* **let** imagemDaEstrada;
* **let** imagemDoAtor;
* **let** imagemCarro;
* **let** imagemCarro2;
* **let** imagemCarro3;
* function **preload**(){
* imagemDaEstrada = loadImage("imagens/estrada.png");
* imagemDoAtor = loadImage("imagens/ator-1.png");
* imagemCarro = loadImage("imagens/carro-1.png");
* imagemCarro2 = loadImage("imagens/carro-2.png");
* imagemCarro3 = loadImage("imagens/carro-3.png");
* }
* index.html:
* <!DOCTYPE **html**><html><head>
* <script src="p5.min.js"></script>
* <script src="p5.dom.min.js"></script>
* <script src="p5.sound.min.js"></script>
* <link rel="stylesheet" type="text/css" href="style.css">
* <meta charset="utf-8">
* </head>
* <body>
* <script src="sketch.js"></script>
* <script src="imagens.js"></script>
* <script src="ator.js"></script>
* <script src="carro.js"></script>
* </body></html>

**REFATURANDO O JOGO COM LISTA**

Para movimentar os 3 carros durante a execução do nosso jogo, 3 variáveis são necessárias:

* posição x,
* posição y,
* velocidade de cada carro.

Definindo as variáveis de posição em lista

//código do carro

**let** xCarros = [600, 600, 600, 600, 600, 600];

**let** yCarros = [40, 96, 150, 210, 270, 318];

**let** velocidadeCarros = [2, 2.5, 3.2, 5, 3.3, 2.3];

**let** comprimentoCarro = 50;

**let** alturaCarro = 40;

**function** **mostraCarro**(){

**for** (**let** i = 0; i < imagemCarros.length; i++){

**image**(imagemCarros[i], xCarros[i], yCarros[i], comprimentoCarro, alturaCarro);

}

}

**function** **movimentaCarro**(){

**for** (**let** i = 0; i < imagemCarros.length; i++){

xCarros[i] -= velocidadeCarros[i];

}

}

**function** **voltaPosicaoInicialDoCarro**(){

**for** (**let** i = 0; i < imagemCarros.length; i++){

**if** (**passouTodaATela**(xCarros[i])){

xCarros[i] = 600;

}

}

}

**function** **passouTodaATela**(xCarro){

**return** xCarro < - 50;

}

**Adicionar a lista de carros no arquivo:**

Imagens.js

function preload(){

  imagemDaEstrada = loadImage("imagens/estrada.png");

  imagemDoAtor = loadImage("imagens/ator-1.png");

  imagemCarro = loadImage("imagens/carro-1.png");

  imagemCarro2 = loadImage("imagens/carro-2.png");

  imagemCarro3 = loadImage("imagens/carro-3.png");

  imagemCarros = [imagemCarro, imagemCarro2, imagemCarro3, imagemCarro, imagemCarro2, imagemCarro3]

**adicionando colisão ao jogo e pontuação:**

baixar os arquivos de implementação

adicione a [biblioteca p2 collide 2D](https://github.com/bmoren/p5.collide2D/archive/master.zip) no seu projeto e utilize a função [collideRectCircle](https://github.com/bmoren/p5.collide2D" \l "colliderectcircle" \t "_blank) para detectar as colisões. Adicione também carros em todas as pistas e crie um placar, para marcar quantas vezes conseguimos atravessar a rua!

No **p5**, clicaremos na seta para baixo, ao lado de "*project folder*" e clicaremos em "*Add file*" para adicionar um arquivo. Clicaremos no espaço da janela que se abrirá e selecionaremos "p5.collide2d.js" para abri-lo. Clicando em "*Open*", ele aparecerá como o último arquivo de nossa lista na aba lateral esquerda da tela.

Para executar de fato esse código dentro de nosso projeto, vamos em "index.html" e assim como fizemos para a imagem, o ator e o carro, faremos também para nosso arquivo: adicionaremos uma linha de <script src="p5.collide2d.js"></script>.

  <body>

    <script src="sketch.js"></script>

    <script src="imagens.js"></script>

    <script src="ator.js"></script>

    <script src="carro.js"></script>

    <script src="p5.collide2d.js"></script>

</body></html>

Arquivo:

Ator.js

*//código do ator*

let xAtor = 85;

let yAtor = 366;

let colisao = false;

let meusPontos = 0;

function mostraAtor(){

  image(imagemDoAtor, xAtor, yAtor, 30, 30);

}

function movimentaAtor(){

*if* (keyIsDown(UP\_ARROW)){

    yAtor -= 3;

  }

*if* (keyIsDown(DOWN\_ARROW)){

*if*(podeSeMover()){

      yAtor += 3;

    }

  }

}

function verificaColisao(){

*//collideRectCircle(x1, y1, width1, height1, cx, cy, diameter)*

*for* (let i = 0; i < imagemCarros.length; i++){

    colisao = collideRectCircle(xCarros[i], yCarros[i], comprimentoCarro, alturaCarro, xAtor, yAtor, 15)

*if* (colisao){

      voltaAtorParaPosicaoInicial();

      somDaColisao.play();

*if* (pontosMaiorQueZero()){

        meusPontos -= 1;

      }

    }

  }

}

function voltaAtorParaPosicaoInicial(){

  yAtor = 366;

}

function incluiPontos(){

  textAlign(CENTER);

  textSize(25);

  fill(color(255, 240, 60))

  text(meusPontos, width / 5, 27);

}

function marcaPonto(){

*if* (yAtor < 15){

    meusPontos += 1;

    somDoPonto.play();

    voltaAtorParaPosicaoInicial();

  }

}

function pontosMaiorQueZero(){

*return* meusPontos > 0;

}

function podeSeMover(){

*return* yAtor < 366;

}

Chamando as funções no sketch.js

function setup() {

createCanvas(500, 400);

}

function draw() {

background(imagemDaEstrada);

mostraAtor();

mostraCarro();

movimentaCarro();

movimentaAtor();

voltaPosicaoInicialDoCarro();

verificaColisao();

incluiPontos();

marcaPonto();

}