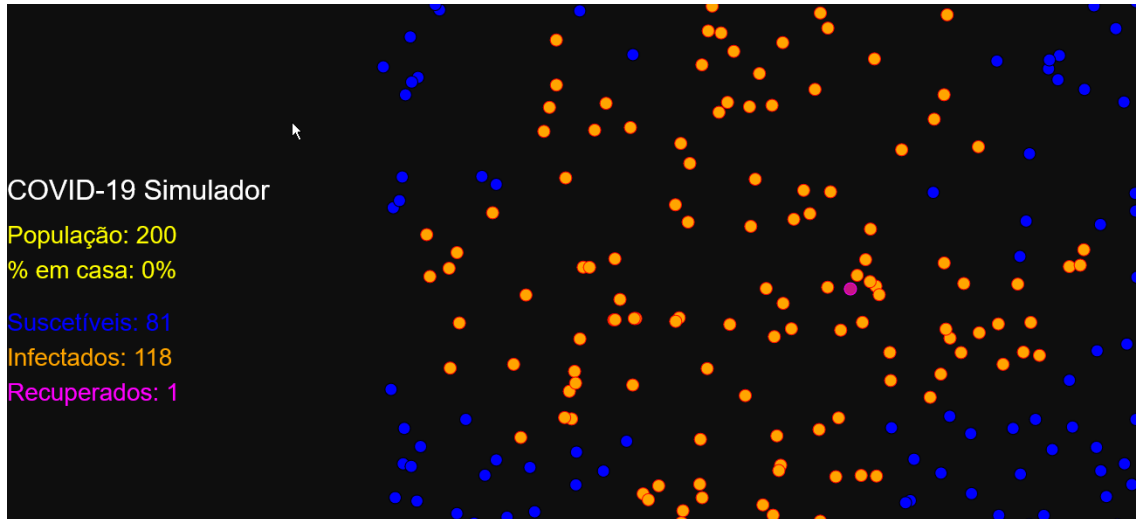


SIMULADOR COVID-19

ALUNOS: Vinicius Cavichioli, Thamyres e Carlos Turíbio.



OBSERVAÇÕES

Para que o programa possa funcionar, altere o valor da quantidade da população, para a quantidade desejada.

A taxa de isolamento é opcional, caso altere é definido uma probabilidade dos cidadãos ficarem parado, como se estivesse em casa.

VARIAVEIS DO PROGRAMA

Foram criadas 4 variáveis principais do sketch do programa, sendo elas:

- ✓ **População**: Variável que define a quantidade de pessoas;
- ✓ **TaxaIsolamento**: Variável que define a taxa de isolamento (taxa de pessoas que ficarão parados, como se estivesse dentro de casa);
- ✓ **W**: Variável que define o tamanho do canvas na horizontal;
- ✓ **H**: Variável que define o tamanho do canvas na vertical.

```
var Populacao = 200;  
var TaxaIsolamento = 0;  
var w = 1300;  
var h = 600;
```

CLASSE QUE DEFINE OS STATUS DA POPULAÇÃO

Esta classe cria os 3 possíveis status que cada cidadão possui, sendo elas suscetíveis (Não infectado), infectado e recuperado.

```
class SIR {  
  static get S() { //Status suscetíveis  
    return 0;  
  }  
  static get I() { //Status infectado  
    return 1;  
  }  
  static get R() { //status recuperado  
    return 2;  
  }  
}
```

FUNÇÃO DRAW () DO SKETCH

A função *Draw* do Sketch é onde está sendo criado os textos que receberá as informações do simulador, está sendo criada uma variável com valor inicial da posição vertical, utilizada para melhor posicionamento dos textos. E também onde é chamado as funções ***cidade.update()*** e ***cidade.draw()*** para desenhar e atualizar os dados.

```
function draw() {  
  background('#0e0e0e');  
  
  let y_start = 200; //Coordenada vertical inicial  
  
  // CRIAÇÃO DOS TEXTOS PARA INFORMAÇÃO  
  textSize(32);  
  fill('White');  
  noStroke();  
  text('COVID-19 Simulador', 0, y_start+30);  
  textSize(28);  
  fill('yellow');  
  text('População: ' + cidade.populacao, 0, y_start+80);  
  text('% em casa: ' + cidade.taxaDeIsolamento + '%', 0, y_start+120);  
  textSize(28);  
  
  fill('Blue');  
  text('Suscetíveis: ' + cidade.status[SIR.S], 0, y_start+180);  
  fill('Orange');  
  text('Infectados: ' + cidade.status[SIR.I], 0, y_start+220);  
  fill('Fuchsia');  
  text('Recuperados: ' + cidade.status[SIR.R], 0, y_start+260);  
  
  cidade.update();  
  cidade.draw();  
}
```

CLASSE CIDADAO

Na classe cidadão, é criado um construtor passando por parâmetro um ID e taxa de isolamento. Será explicado as variáveis e suas funções conforme a imagem a baixo.

```

constructor(id, taxaIsolamento) {
  this.id = id;
  this.speed = 2;
  this.size = 14;
  this.status = SIR.S;
  this.dataInfeccao = 0;
  this.tempoRecuperacao = random(4, 15);
  this.estaEmCasa = random(0, 100) < taxaIsolamento;

  this.localizacao = createVector(random(width/3, width), random(0, height));
  this.velocity = createVector(this.speed, this.speed);
}

```

- ✓ **Speed:** declara a velocidade que a ellipse(cidadão) irá se mover;
- ✓ **Size:** declara o tamanho da elipse a ser criada;
- ✓ **Status:** declara o status inicial do cidadão, no caso como suscetível;
- ✓ **dataInfeccao:** variável que será utilizada futuramente para guardar o valor que cada cidadão foi infectado;
- ✓ **tempoRecuperacao:** declara um tempo aleatório entre 4 e 15 segundos para um cidadão se recuperar do vírus;
- ✓ **estaEmCasa:** declara um valor aleatório até 100 menor que a taxa de isolamento;
- ✓ **localização:** aqui é definido o vetor onde os cidadãos irão ser criados dentro do campo no canvas, nesse caso ele irá criar em uma parte do canvas no canto direito, ou seja, é o tamanho da tela onde o cidadão irá se mover;
- ✓ **Velocity:** vetor de velocidade que o cidadão irá se mover, isso de acordo com o valor declarado na variável **speed**.

Nesta classe também é criado a função que verifica a colisão entre as elipses. Aqui é verificado se o ID do cidadão é igual ao ID passado, é realizado a diferença entre as posições x e y do cidadão com o outro. Calculado a distancia total e a mínima de uma ellipse para outra para que possa ser definido se uma passou perto da outra, assim verificando se o status de uma ellipse que estava infectado ou não. Caso esteja infectado e a outra não é, ambas ficam infectadas.

```

verificaInteracoes(outrasPessoas) {
  for (let i = 0; i < outrasPessoas.length; i++) {
    if (this.id === i)
      continue;

    let dx = outrasPessoas[i].localizacao.x - this.localizacao.x;
    let dy = outrasPessoas[i].localizacao.y - this.localizacao.y;
    let distance = sqrt(Math.pow(dx, 2) + Math.pow(dy, 2));
    let minDist = this.size + outrasPessoas[i].size;

    if (distance <= minDist) {
      if (outrasPessoas[i].status === SIR.I && this.status === SIR.S) {
        this.status = SIR.I;
        this.dataInfeccao = floor(millis()/1000);
      }
    }
  }
}

```

Função update da classe cidadão

Essa função ficará atualizando o tempo todo para que os dados e o simulador funcionem corretamente. Primeiramente será adicionado duas funções para que a ellipse se movimente rotacionando.

```
this.localizacao.add(this.velocity);  
this.velocity.rotate(radians(random(-30, 30)));
```

Na primeira linha, é adicionado o vetor de velocidade dentro do vetor de localização, fazendo que a ellipse se movimenta conforme os valores definidos na variável. Na segunda linha, é adicionado uma rotação aleatoriamente para que a ellipse fique indo pra um lado e pro outro.

Em seguida é realizado uma verificação para que quando a ellipse bater no final da tela do canvas ela volte, ou seja, definir o final da tela como se fosse uma parede. Conforme mostra a imagem abaixo, é realizado a verificação da localização da ellipse na posição x e y, e verifica se ultrapassou a largura e altura definida na criação do canvas, caso a ellipse ultrapasse esses valores é invertido sua localização.

```
if (this.localizacao.x <= width/3 || this.localizacao.x > width || this.localizacao.y < 0 || this.localizacao.y >= height) {  
    this.velocity.y = this.velocity.y * -1;  
    this.velocity.x = this.velocity.x * -1;  
}
```

A função a baixo verifica se o cidadão está “em casa”, conforme os valores definidos nas variáveis no construtor. Caso seja verdadeiro, a ellipse, ou seja, o cidadão é criado no canvas parado, não irá se mover.

```
if (this.estaEmCasa) {  
    return;  
}
```

Para finalizar a função update, é realizado a verificação se o cidadão se recuperou da infecção do vírus COVID.

```
if (this.status === SIR.I) {  
    let hoje = floor(millis()/1000);  
    if (hoje - this.dataInfeccao >= this.tempoRecuperacao) {  
        this.status = SIR.R;  
    }  
}
```

Conforme a imagem a cima, é verificado se o status do cidadão é Infectado, se for verdadeiro, é verificado o valor de execução do programa atual menos a data que o cidadão foi infectado, caso seja maior que o tempo de recuperação definido o status é mudado para recuperado.

Função draw da classe cidadão

Na função draw é criado primeiramente as elipses de forma aleatória dentro do quadro definido na localização, conforme mostre na imagem a baixo.

```
ellipse(this.localizacao.x, this.localizacao.y, this.size);
```

Após criado as ellipses também é atribuído as cores conforme os status de cada cidadão.

```
strokeWeight(1);
if (this.status === SIR.S) {
  stroke(0);
  fill(0,0,255);
}
else if (this.status === SIR.I) {
  stroke(255,0,0);
  fill(255,165,0);
}
else if (this.status === SIR.R) {
  stroke(255,0,255);
  fill(199,21,133);
}
```

1. Primeiro é verificado se o status é suscetível, se for é atribuído a cor branca para a ellipse.
2. Segundo é verificado se o status é infectado, se for é atribuído a cor laranja para a ellipse.
3. Terceiro é verificado se o status é recuperado, se for é atribuído a cor rosa para a ellipse.

CLASSE CIDADE

Na classe cidade é criado o construtor com as seguintes definições;

- ✓ **Populacao**: é o valor da quantidade de população definido no sketch;
- ✓ **taxaDeIsolamento**: é o valor da taxa de isolamento definido no sketch;
- ✓ **cidadaos**: array para guardar os cidadãos criados;
- ✓ **status**: é guardado os valores dos status do cidadão;
- ✓ **isCoronaGone**: primeiramente atribuído como false, depois é realizado uma verificação se todos os cidadãos foram recuperados do vírus para que o programa possa parar a execução.

É criado um for para que realize a criação de cada cidadão dentro do array, e declaro a inserção de um cidadão aleatório infectado, para que possa começar a pandemia.

```

constructor() {
  this.populacao = Populacao;
  this.taxaDeIsolamento = TaxaIsolamento;
  this.cidadaos = [];

  this.status = {};
  this.status[SIR.S] = 0;
  this.status[SIR.I] = 0;
  this.status[SIR.R] = 0;

  this.isCorongaGone = false;

  //For que cria as pessoas para dentro do array cidadaos
  for (let i = 0; i < this.populacao; i++) {
    this.cidadaos.push(new Cidadao(i, this.taxaDeIsolamento));
  }

  // Uma pessoa infectada aleatoriamente
  random(this.cidadaos).status = SIR.I;
}

```

Função update da classe cidade

As condições a baixo verifica se a soma dos status de suscetível mais o status de recuperado é igual a população, se for verdadeiro, é a variável **isCorongaGone** é atribuído como verdadeiro e a execução do programa é parado.

```

// VERIFICA SE TODA A POLUIÇÃO FOI RECUPERADA E PARA A EXECUÇÃO
if (this.status[SIR.S] + this.status[SIR.R] === this.populacao) {
  this.isCorongaGone = true;
  return;
}

```

As outras funções declaradas dentro da função update, é para realizar a criação das movimentações dos cidadãos.

Função draw da classe cidade

Na função draw é realizado os desenhos das elipses no canvas;

```

draw() {
  // desenha os cidadaos
  for (let i = 0; i < this.populacao; i++) {
    this.cidadaos[i].draw();
  }
}

```