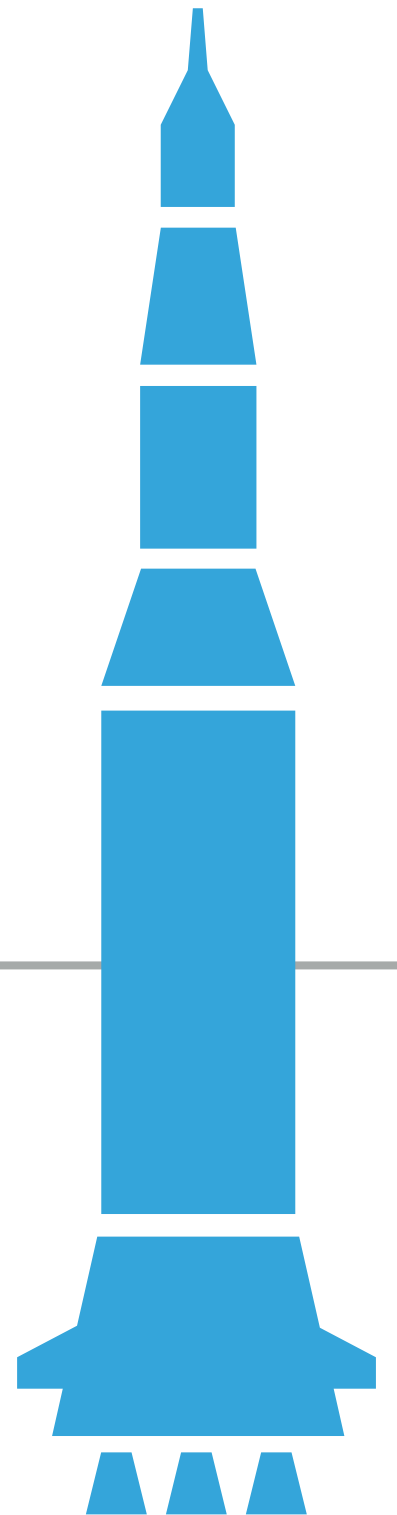
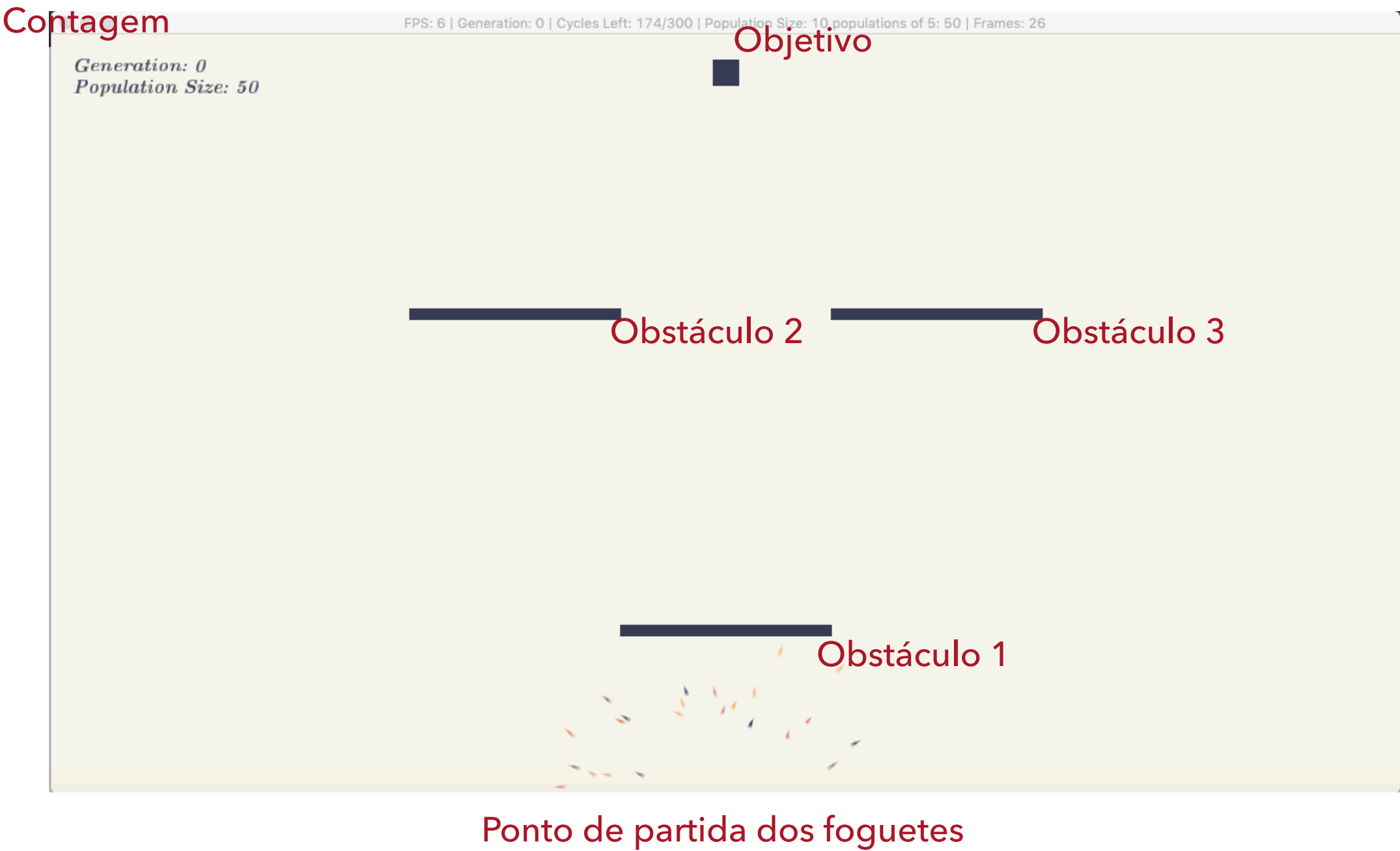


SISTEMAS EVOLUTIVOS

**FOGUETES
INTELIGENTES**



FOGUETE ATINGIR O OBJETIVO



PROCESSING



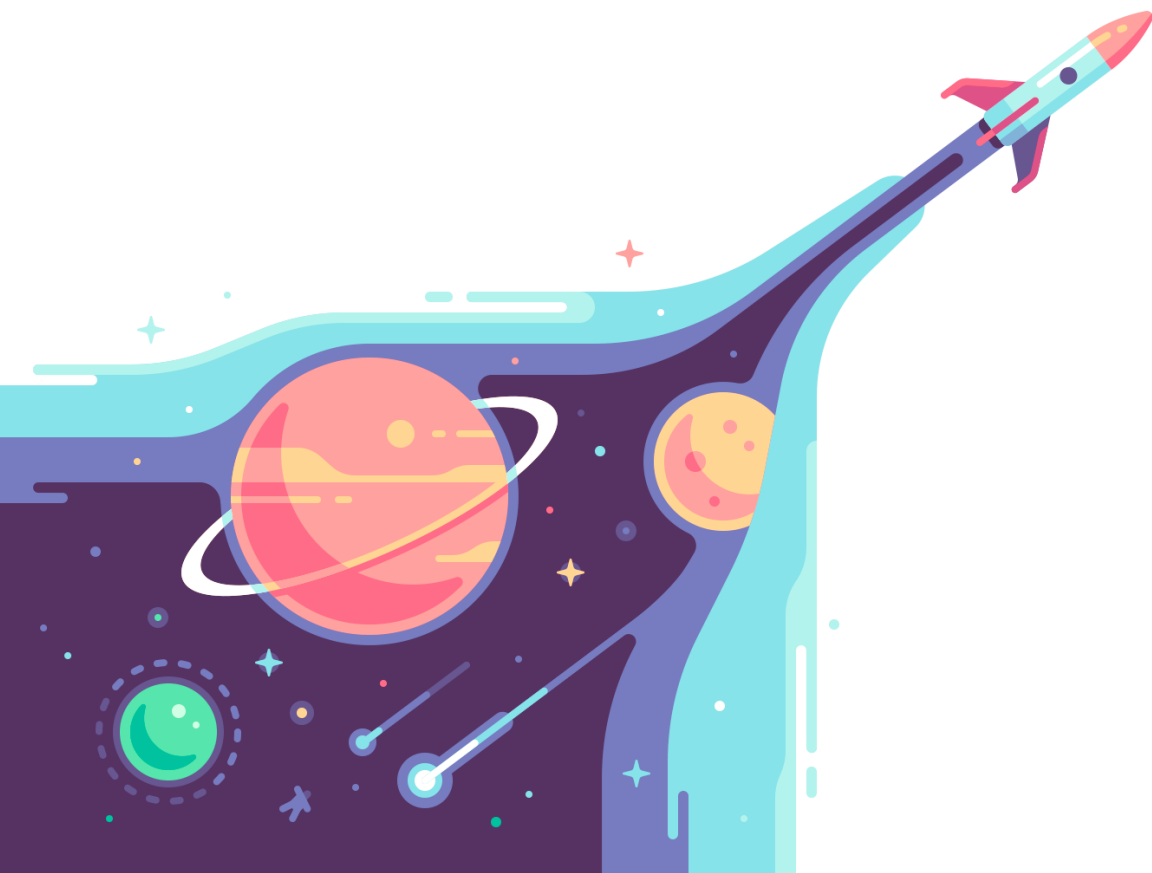
Setup: Onde é feito todo o pré-processamento: criação de desenho, tamanho da janela, carregar dados necessários, etc.

Draw: Loop de desenho. Essa função roda indefinidamente renderizando quaisquer instruções de desenho dentro dela, até o programa ser fechado.

COMO FUNCIONA

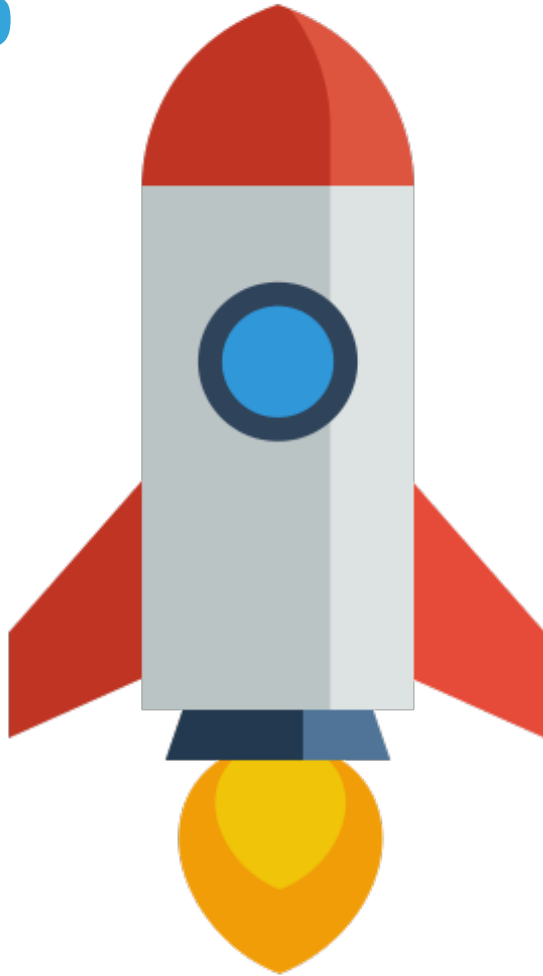
O princípio é fazer com que os foguetes (indivíduos) mais aptos gerem descendentes, que se tornarão aptos e farão o mesmo para as próximas gerações.

Esses foguetes serão colocados em uma *mating pool*, uma piscina genética, de onde serão sorteados os pares reprodutivos.



Função para alteração de direção (genes) de maneira aleatória, simulando mutações.

FOGUETES

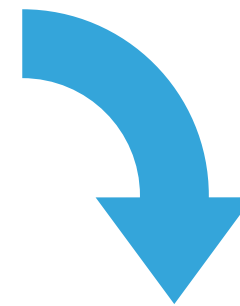


Vetor de velocidade

Vetor de posição

Vetor de aceleração

Função de Fitness



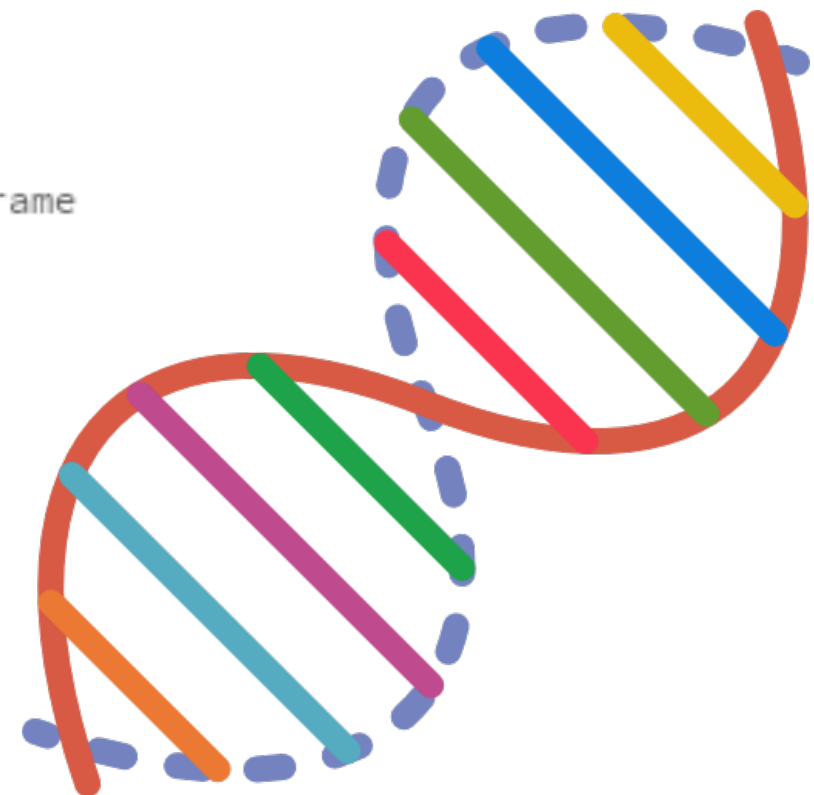
Cálculo da pontuação de
cada indivíduo

```
void fitness() {  
  if (recordDist < 1) recordDist = 1;  
  
  // Reward finishing faster and getting close  
  fitness = (1/(finishTime*recordDist));  
  
  // Make the function exponential  
  fitness = pow(fitness, 4);  
  
  if (hitObstacle) fitness *= 0.1; // lose 90% of fitness hitting an obstacle  
  if (hitTarget) fitness *= 2; // twice the fitness for finishing!  
}
```

DNA

Possuem DNA → Possuem Genes → Vetor de direções

```
DNA() {  
    genes = new PVector[lifetime];  
    for (int i = 0; i < genes.length; i++) {  
        float angle = random(TWO_PI);  
        genes[i] = new PVector(cos(angle), sin(angle));  
        genes[i].mult(random(0, maxforce));  
    }  
  
    // Let's give each Rocket an extra boost of strength for its first frame  
    genes[0].normalize();  
}
```



POPULAÇÃO

Vetor de foguetes → Reprodução → Novos indivíduos



Crossover

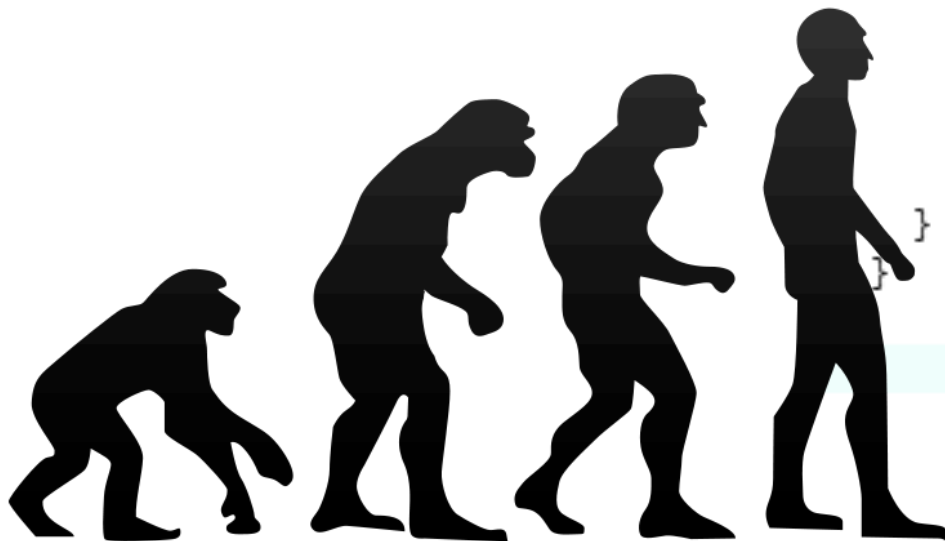
Parte dos genes da mãe e parte dos genes do pai para gerar o filho. Além disso, quando o código genético do filho está pronto, há a chance de existir uma mutação



MUTAÇÕES

Alterações no vetor de direções de maneira aleatória, após a mistura de genes de dois indivíduos

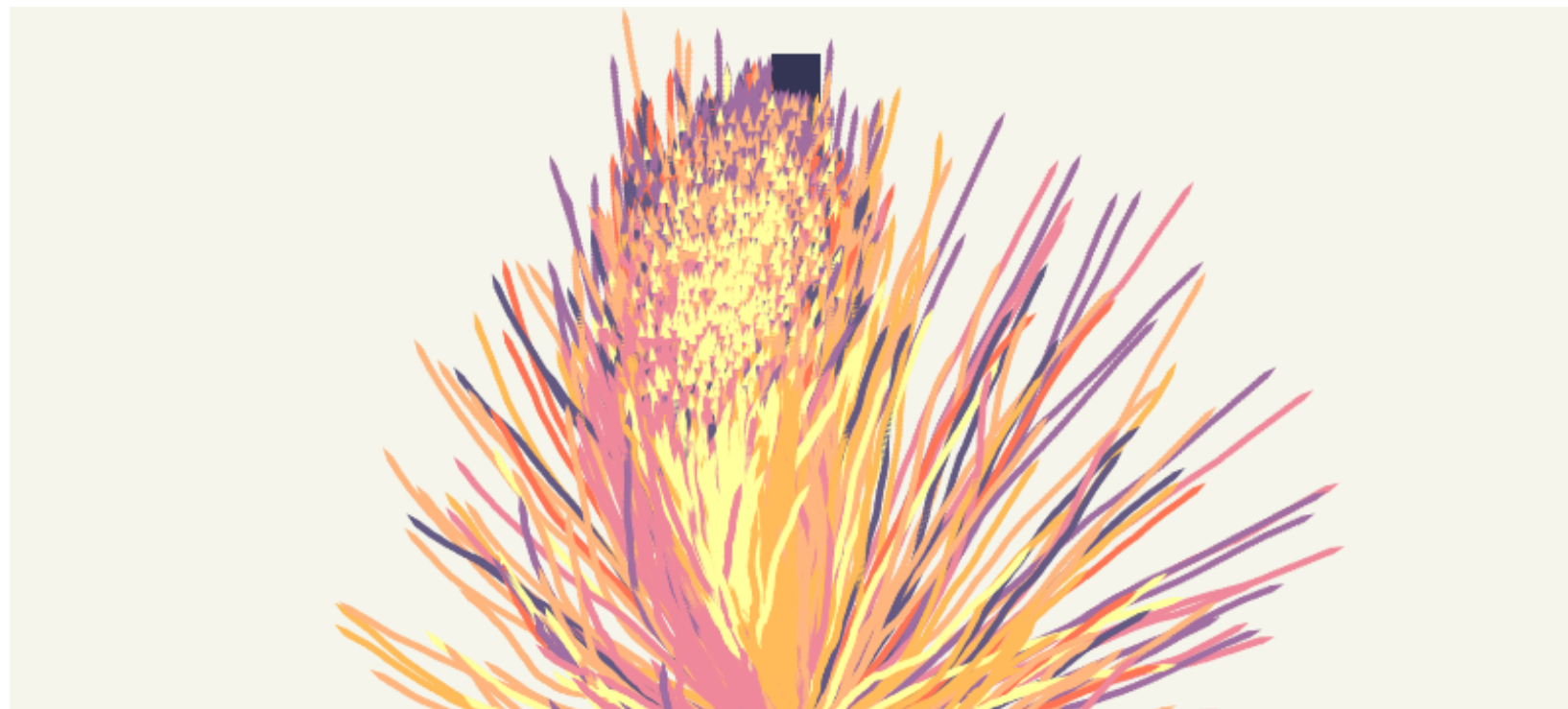
```
// Based on a mutation probability, picks a new random Vector
void mutate(float m) {
    for (int i = 0; i < genes.length; i++) {
        if (random(1) < m) {
            float angle = random(TWO_PI);
            genes[i] = new PVector(cos(angle), sin(angle));
            genes[i].mult(random(0, maxforce));
            // float angle = random(-0.1,0.1);
            // genes[i].rotate(angle);
            // float factor = random(0.9,1.1);
            // genes[i].mult(factor);
            if (i == 0) genes[i].normalize();
        }
    }
}
```



REPRODUÇÃO

Os indivíduos considerados mais aptos recebem mais bilhetes para entrar na *mating pool*, que representam proporcionalmente suas chances de se reproduzir.

Assim, garantimos que os genes melhores continuem se propagando para a próxima geração.



RODANDO O ALGORITMO

```
POPULAÇÃO 48 - OBJETIVO CONCLUÍDO. Geração: 42
POPULAÇÃO 49 - OBJETIVO CONCLUÍDO. Geração: 38
POPULAÇÃO 50 - OBJETIVO CONCLUÍDO. Geração: 29
POPULAÇÃO 51 - OBJETIVO CONCLUÍDO. Geração: 34
POPULAÇÃO 52 - OBJETIVO CONCLUÍDO. Geração: 37
POPULAÇÃO 53 - OBJETIVO CONCLUÍDO. Geração: 24
POPULAÇÃO 54 - OBJETIVO CONCLUÍDO. Geração: 28
POPULAÇÃO 55 - OBJETIVO CONCLUÍDO. Geração: 41
POPULAÇÃO 56 - OBJETIVO CONCLUÍDO. Geração: 33
POPULAÇÃO 57 - OBJETIVO CONCLUÍDO. Geração: 30
POPULAÇÃO 58 - OBJETIVO CONCLUÍDO. Geração: 25
POPULAÇÃO 59 - OBJETIVO CONCLUÍDO. Geração: 21
POPULAÇÃO 60 - OBJETIVO CONCLUÍDO. Geração: 15
POPULAÇÃO 61 - OBJETIVO CONCLUÍDO. Geração: 47
POPULAÇÃO 62 - OBJETIVO CONCLUÍDO. Geração: 41
POPULAÇÃO 63 - OBJETIVO CONCLUÍDO. Geração: 31
POPULAÇÃO 64 - OBJETIVO CONCLUÍDO. Geração: 34
POPULAÇÃO 65 - OBJETIVO CONCLUÍDO. Geração: 24
POPULAÇÃO 66 - OBJETIVO CONCLUÍDO. Geração: 25
POPULAÇÃO 67 - OBJETIVO CONCLUÍDO. Geração: 34
POPULAÇÃO 68 - OBJETIVO CONCLUÍDO. Geração: 19
POPULAÇÃO 69 - OBJETIVO CONCLUÍDO. Geração: 32
POPULAÇÃO 70 - OBJETIVO CONCLUÍDO. Geração: 31
POPULAÇÃO 71 - OBJETIVO CONCLUÍDO. Geração: 39
POPULAÇÃO 72 - OBJETIVO CONCLUÍDO. Geração: 23
POPULAÇÃO 73 - OBJETIVO CONCLUÍDO. Geração: 35
POPULAÇÃO 74 - OBJETIVO CONCLUÍDO. Geração: 41
POPULAÇÃO 75 - OBJETIVO CONCLUÍDO. Geração: 25
POPULAÇÃO 76 - OBJETIVO CONCLUÍDO. Geração: 34
POPULAÇÃO 77 - OBJETIVO CONCLUÍDO. Geração: 24
POPULAÇÃO 78 - OBJETIVO CONCLUÍDO. Geração: 27
POPULAÇÃO 79 - OBJETIVO CONCLUÍDO. Geração: 33
POPULAÇÃO 80 - OBJETIVO CONCLUÍDO. Geração: 47
POPULAÇÃO 81 - OBJETIVO CONCLUÍDO. Geração: 24
POPULAÇÃO 82 - OBJETIVO CONCLUÍDO. Geração: 28
POPULAÇÃO 83 - OBJETIVO CONCLUÍDO. Geração: 33
POPULAÇÃO 84 - OBJETIVO CONCLUÍDO. Geração: 28
```

RODANDO O ALGORITMO

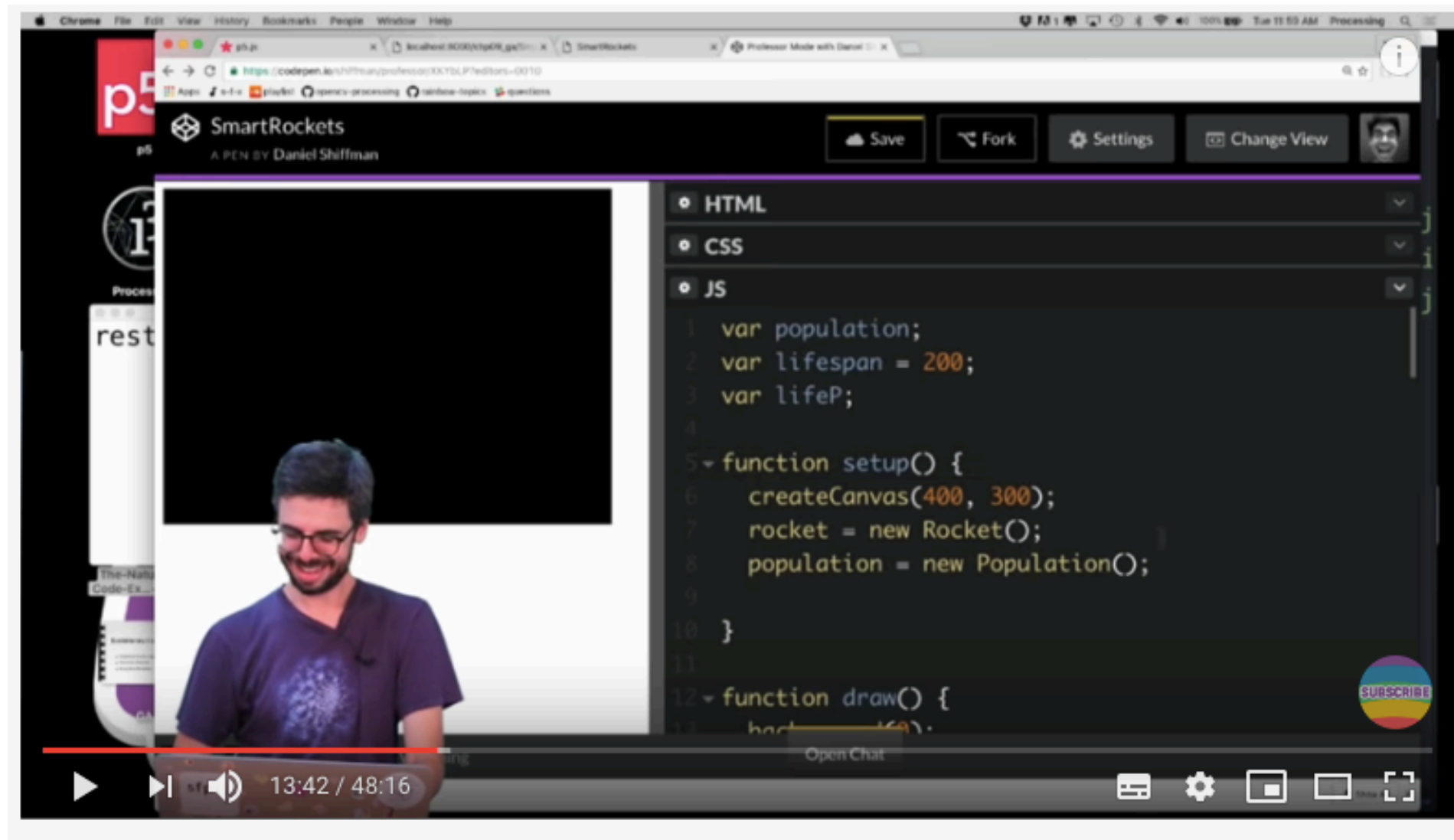
Generation: 0
Population Size: 90000



MAIS GERAÇÕES



THE CODING TRAIN



Coding Challenge #29: Smart Rockets in p5.js