Universidade Federal do Ceará

Sistemas e Mídias Digitais

Matemática Aplicada à Multimídia

Conteúdo: Documento escrito sobre o Projeto Individual.

Tema: Máquina de Estados Finita (MEF)

Nome: Davi Lisboa de França (553865)

Prazo: 16/09/2024

Fortaleza, CE

2024

Apresentação do Tema

Uma Máquina de Estados Finita (FSM - do inglês Finite State Machine) ou autômato finito é usado para representar programas de computadores ou circuitos lógicos.

O conceito é concebido como uma máquina abstrata que deve estar em um de um número finito de estados.

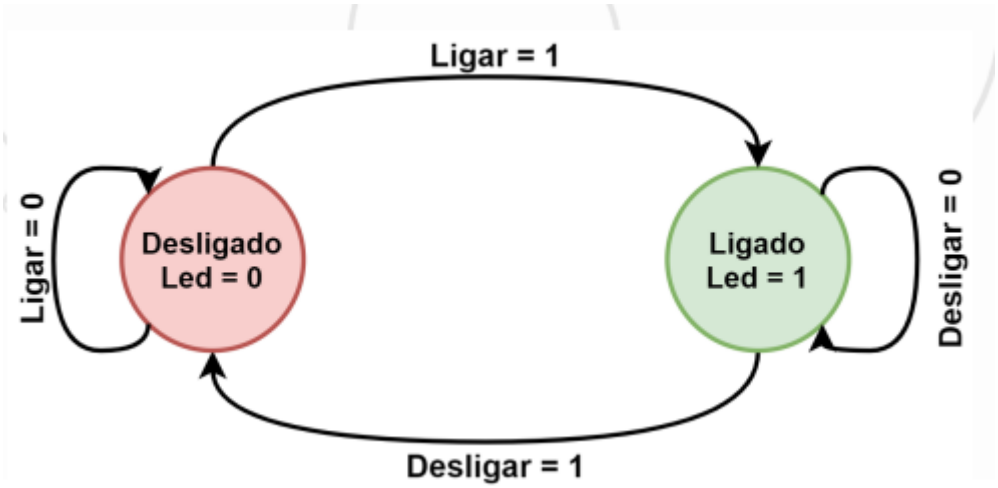
A máquina está em apenas um estado por vez, chamado de estado atual.

Um estado armazena informações de entradas e saídas.

Uma transição indica a condição necessária para uma mudança de estado.

Uma ação é a atividade realizada num determinado estado.

Exemplo de máquina de estados para o controle de um LED:



De forma simplificada uma máquina de estados é composta por:

• Estados: possíveis estados de operação do sistema ex: Desligado, Ligado.

• Transições: passagem de um estado a outro ex de Desligado para Ligado.

• Entradas: entradas do sistema, que disparam as transições ex: Ligar, Desligar.

• Saídas: são as saídas da máquina de estados que interagem com o resto do sistema.

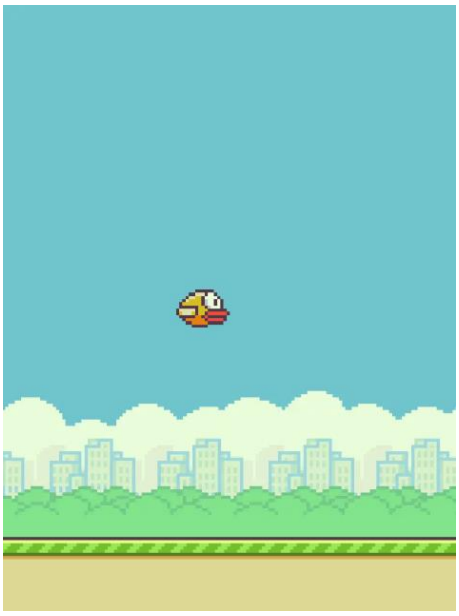
Cada estado monitora um conjunto de entradas para determinar se deve passar a outro estado ou permanecer onde está. Cada estado fornece uma determinada saída ao sistema.

Aplicação Escolhida

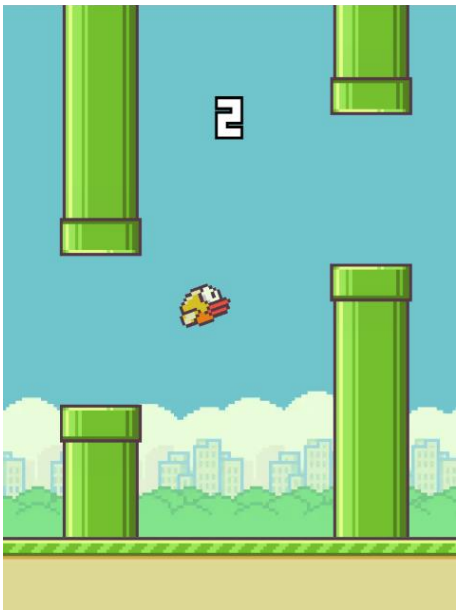
O produto/aplicação escolhido trata-se do famoso jogo **Flappy Bird**.

O jogo se comporta como uma Máquina de Estados Finitos ao transitar sempre entre 3 estados.

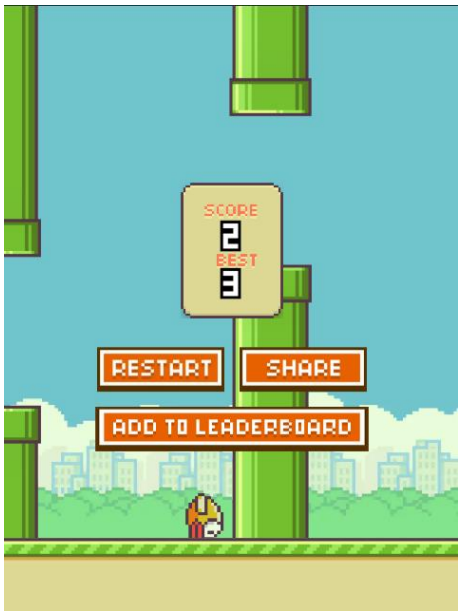
1 – Tela de início



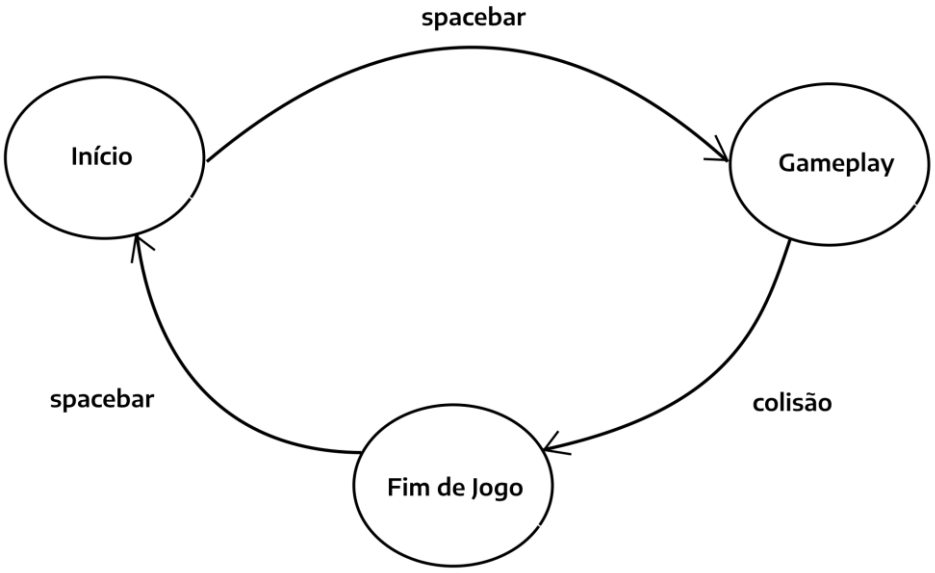
2 – Gameplay



3 – Tela de fim de jogo



O game alterna entre esses três estados finitos conforme determinadas condições são atendidas:



Repare que, para ir da Tela de Início para o Gameplay, basta que a barra de espaço seja pressionada. O mesmo vale da tela de Fim de Jogo para a Tela de Início.

Enquanto a condição de derrota é uma colisão durante a Gameplay, que aciona a Tela de Fim de Jogo.

Prototipação - MEF

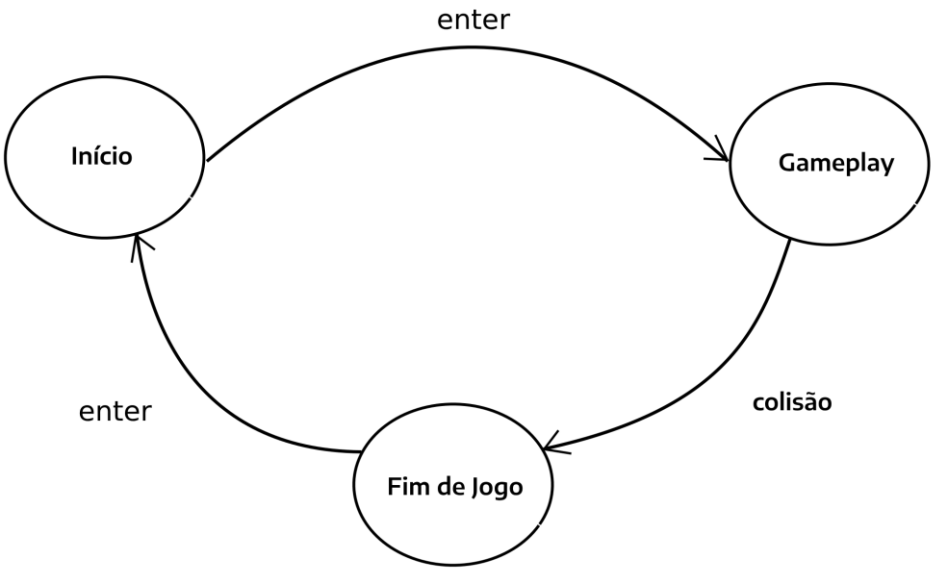
No meu protótipo em p5js de Flappy Bird, os estados do jogo são gerenciados por meio do trecho de código a seguir:

```
if (gameState === 'start')
{
  displayStartScreen();

} else if (gameState === 'playing')
{
  updateGame();

} else if (gameState === 'gameOver')
{
  displayGameOverScreen();
}
```

A alternância entre os estados se dá pela tecla Enter, diferente do jogo original, onde a tecla de espaço era usada, observe:



Prototipação - Colisão

Dentro do código do meu protótipo, três trechos de código são responsáveis por fazer a colisão funcionar, o primeiro deles sendo uma função **hits** dentro da classe dos Canos/Pipes/Obstáculos.

```
hits(bird) {
  if (bird.y < this.top || bird.y > height - this.bottom) {
    if (bird.x > this.x && bird.x < this.x + this.width) {
      return true;
    }
  }
  return false;
}
```

Repare que a função acima retorna true quando há colisão entre o pássaro e as coordenadas **top** ou **bottom** dos Canos.

O método a seguir vem da classe Bird( ) e retorna **true**/verdadeiro caso o pássaro saia da tela.

```
offscreen( ) {
  return this.y > height || this.y < 0;
}
```

As condições a seguir são definidas dentro da função updateGame( ), e trocam o estado do jogo caso as funções **hits( )** ou **offscreen( )** retornem **true.** Ou seja, o jogo termina se o pássaro colidir com os obstáculos ou com as bordas da tela.

```
if (pipes[i].hits(bird)) {
  gameState = 'gameOver';
}


if (bird.offscreen()) {
  gameState = 'gameOver';
}
```

Prototipação - Gravidade

Um aspecto visto na aula de MRU e MRUV foi a gravidade. Diferente da abordagem em sala de aula, onde foram usadas fórmulas matemáticas, meu código simula a gravidade de maneira de diferente.

Primeiro, o valor dela é setado no construtor da classe Bird( )

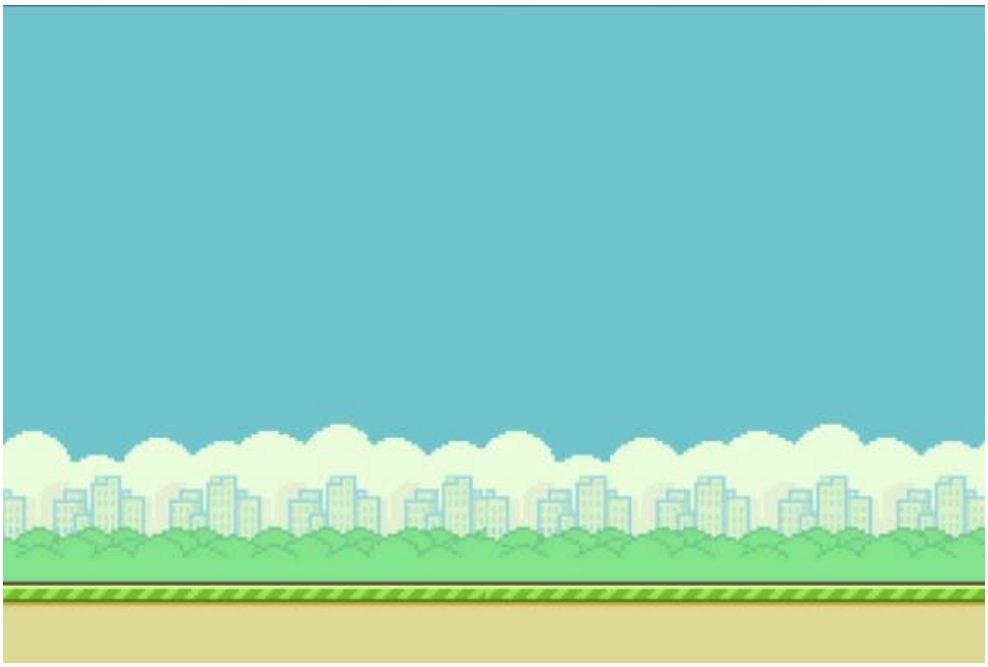
```
class Bird {
  constructor() {
    this.y = height / 2;
    this.x = 64;
    this.gravity = 0.6;
```

Em seguida, ela entra em ação sobre o Pássaro através do método update( )

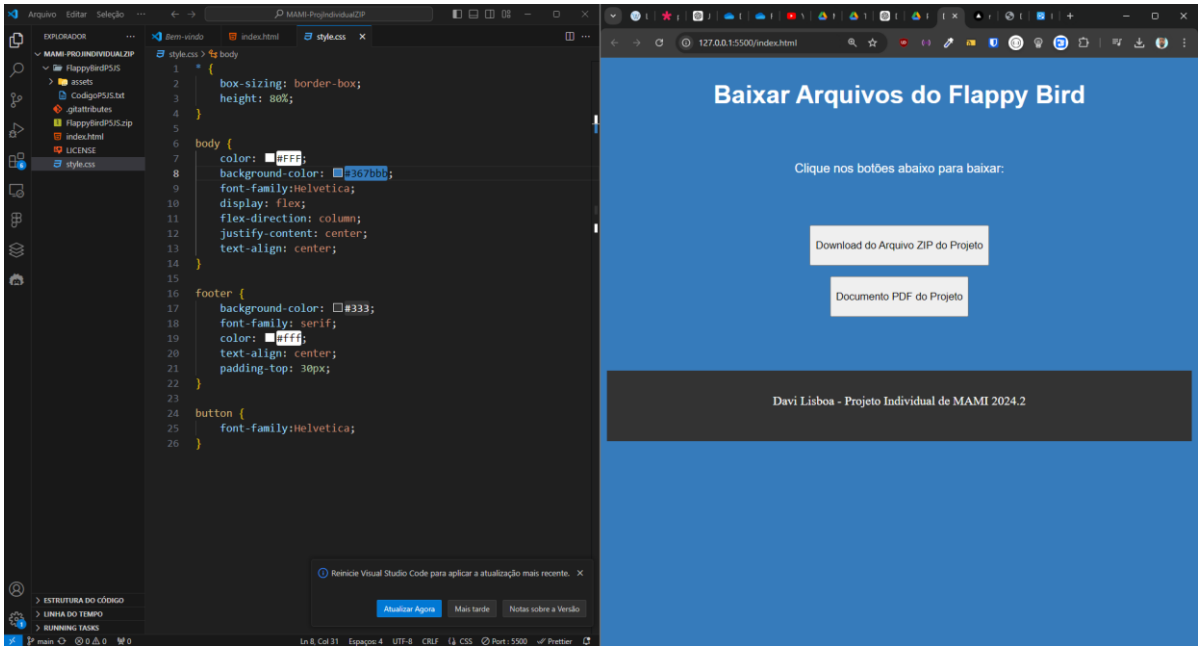
```
update() {
  this.velocity += this.gravity; //O pássaro ganha velocidade
  this.y += this.velocity; // O pássaro cai
  this.velocity *= 0.9;
}
```

Curiosidades Adicionais

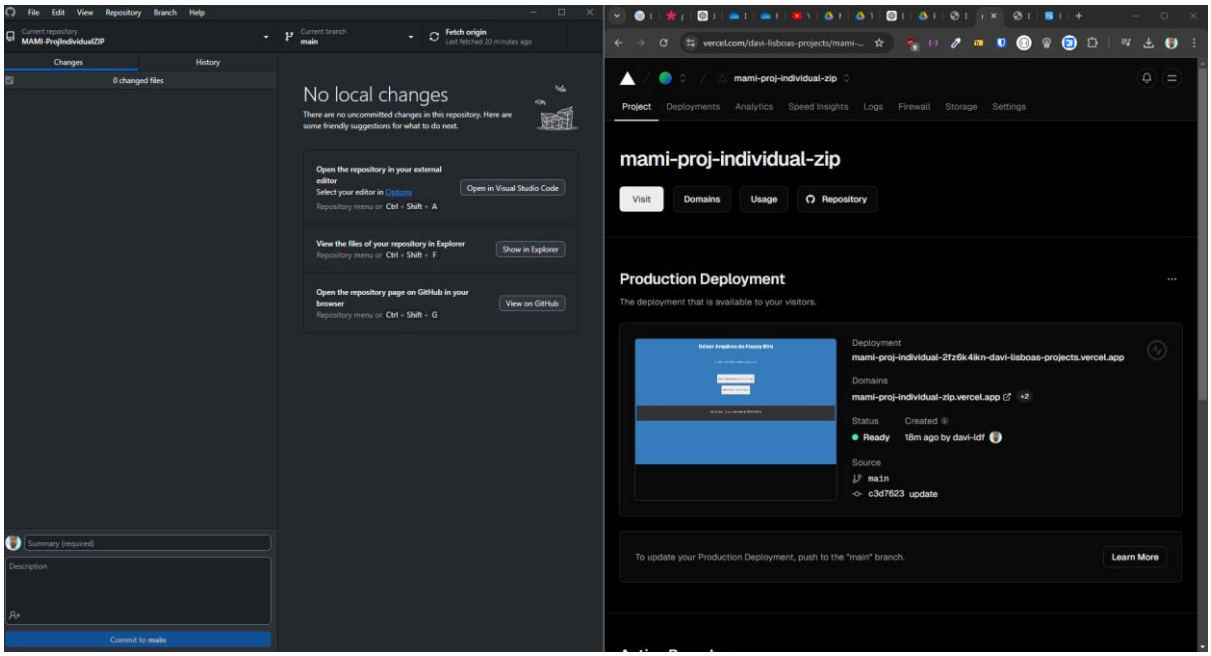
A versão “Ultrawide” do cenário de Flappy Bird disponível no meu protótipo em p5js foi feita por mim, a partir da duplicação do cenário do jogo original e do uso do tamanho do meu Canvas em p5 como base.



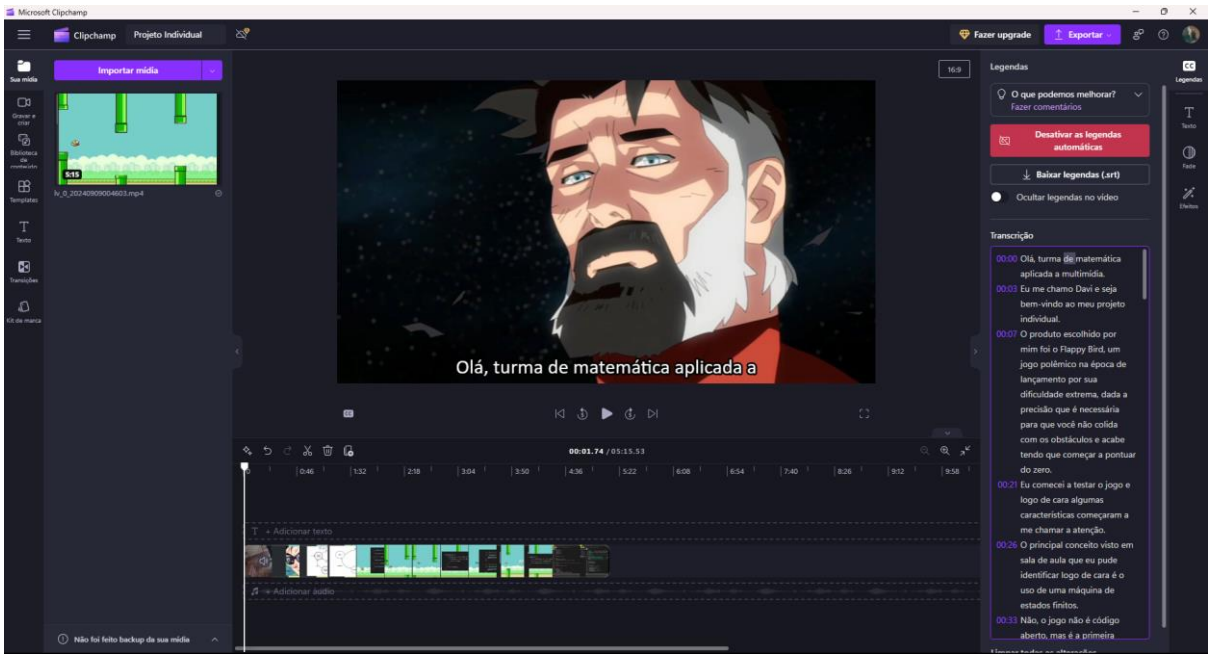
Eu mesmo criei um site interativo onde disponibilizo os Downloads ZIP e PDF do meu Projeto Individual, utilizando meus conhecimentos prévios de Desenvolvimento Web e GitHub.



A hospedagem se deu pela plataforma gratuita Vercel, que suporta páginas Web estáticas.



O vídeo de apresentação do Projeto Individual foi legendado por meio do programa Microsoft Clipchamp, que utiliza de IA para gerar legendas dinamicamente de forma gratuita (A edição foi pelo CapCut).



Bibliografia

https://professor.luzerna.ifc.edu.br/ricardo-kerschbaumer/wp-content/uploads/sites/43/2021/04/Maquinas-de-Estado.pdf

Circuitos Digitais: Aula 10 (ufpe.br)

https://homepages.dcc.ufmg.br/~nvieira/cursos/tl/a17s2/livro/cap2.pdf

https://www.facom.ufu.br/~abdala/sd/MEFs.pdf

Links Importantes

Vídeo de Apresentação:

https://youtu.be/eMY2saySvPM

Download do ZIP com o Código e seus Assets:

https://mami-proj-individual-zip.vercel.app/