Desenvolvendo Space Invaders com o Flex SDK

Julio Rodrigues
Vítor Azevedo

Roteiro

- Introdução
- Game Engine
- MrEngine
- Estrutura de Pastas
- Desenhando na Tela
- Animações
- Verificando Inputs do Jogador
- Intervalo (20 mins)
- Verificação de Colisão
- Inteligência Artificial (Máquinas de Estados)
- Gerência de Fases
- HUD
- Livros

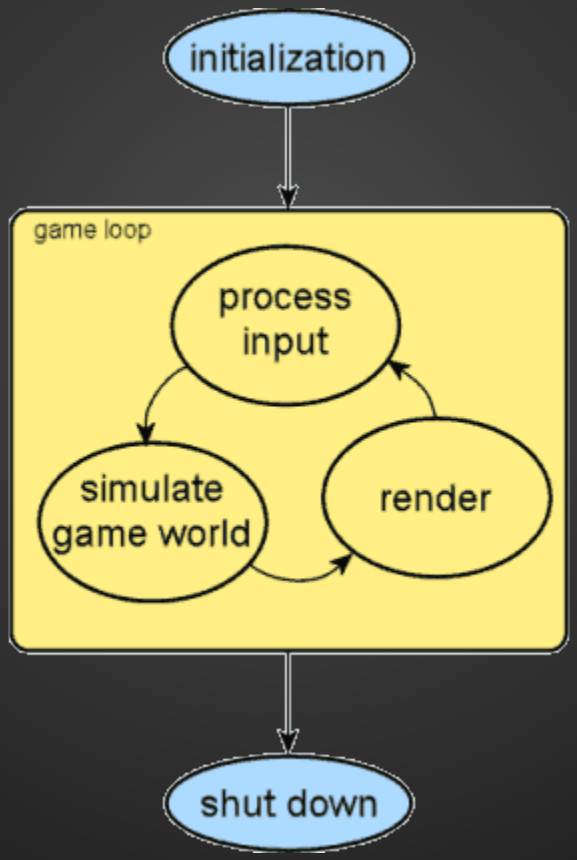
Quem Somos

Ciência da Computação UFES - 2008/1

Julio Rodrigues
juliobortolon.blogspot.com / @juliomarcos
as3 game-design python C/C++ web-dev

Vitor Azevedo
@vitorgomes
android java linux redes simple-3d-modeling

Game Engine


initialization
game loop
process input
simulate game world
render
shut down

MrEngine

Baseada em Flixel (www.flixel.org)

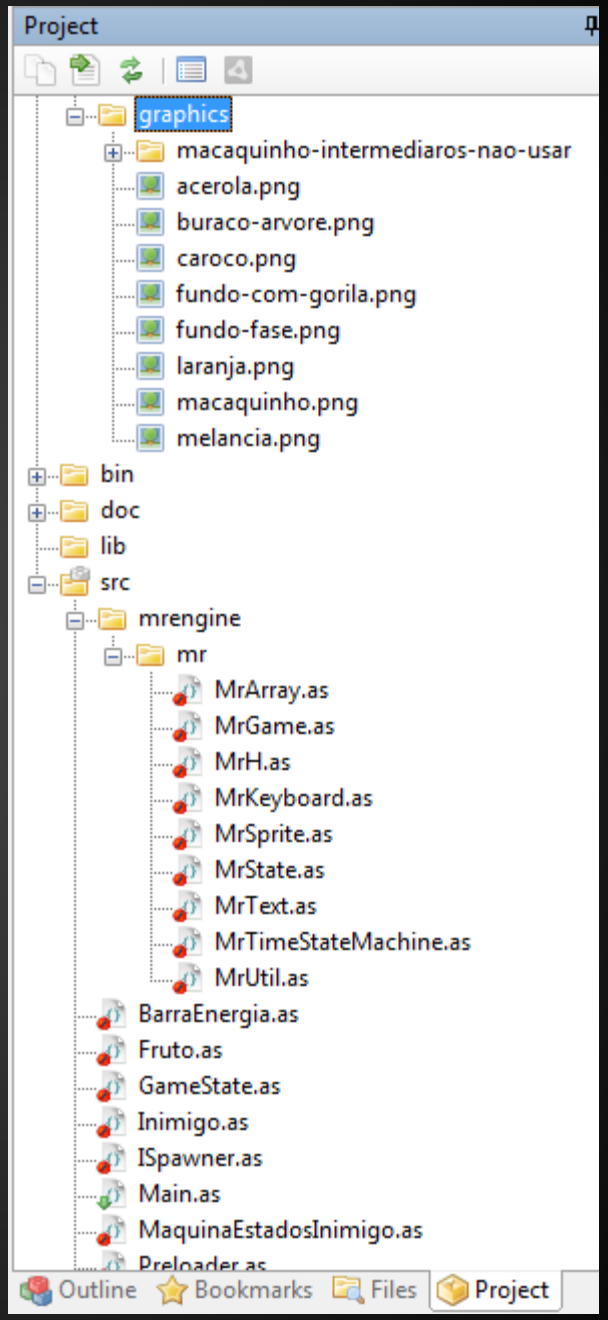
Motivação

- Uso da API do Flex (flash)
- Transição entre fases, menus, etc
- Animação por Sprite Sheet
- Utilizar o teclado sem lidar com o sistema de eventos do flash diretamente

https://bitbucket.org/juliomarcos/mrengine

Estrutura de Pastas

```
/assets
  /graphics
  /sfx
  /music
/lib
  /engineExterna
/src
  /engineInterna
/bin
/docs
```

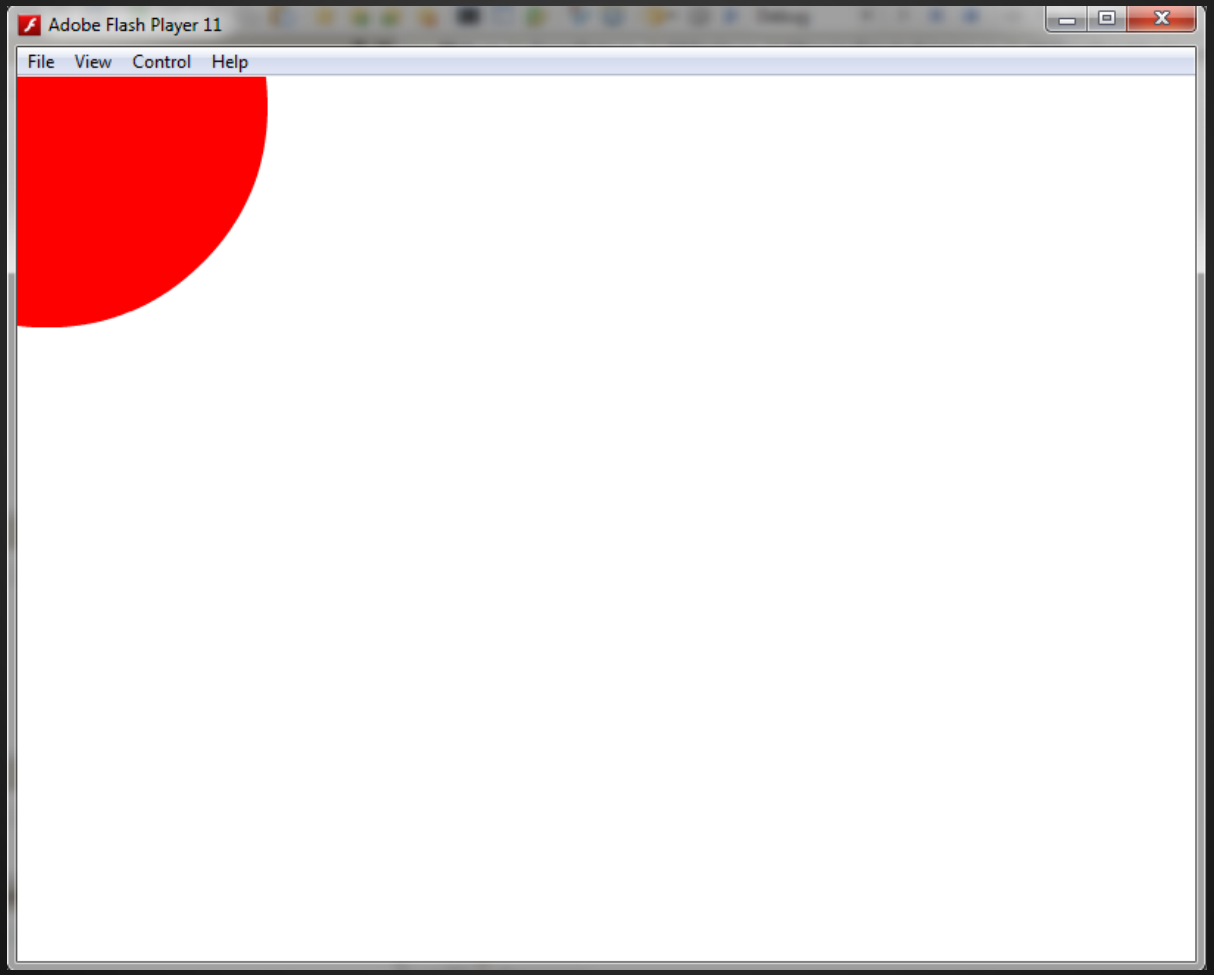


Desenhando na Tela - 1

Desenhos Vetoriais por Código

```
private function init(e:Event=null):void{
    ...
    var desenho:Sprite = new Sprite();
    desenho.graphics.beginFill(0xFF0000);
    desenho.graphics.drawCircle(20, 20, 50);
    desenho.graphics.endFill();
    addChild(desenho);
    ...
}
```

Desenhando na Tela - 2

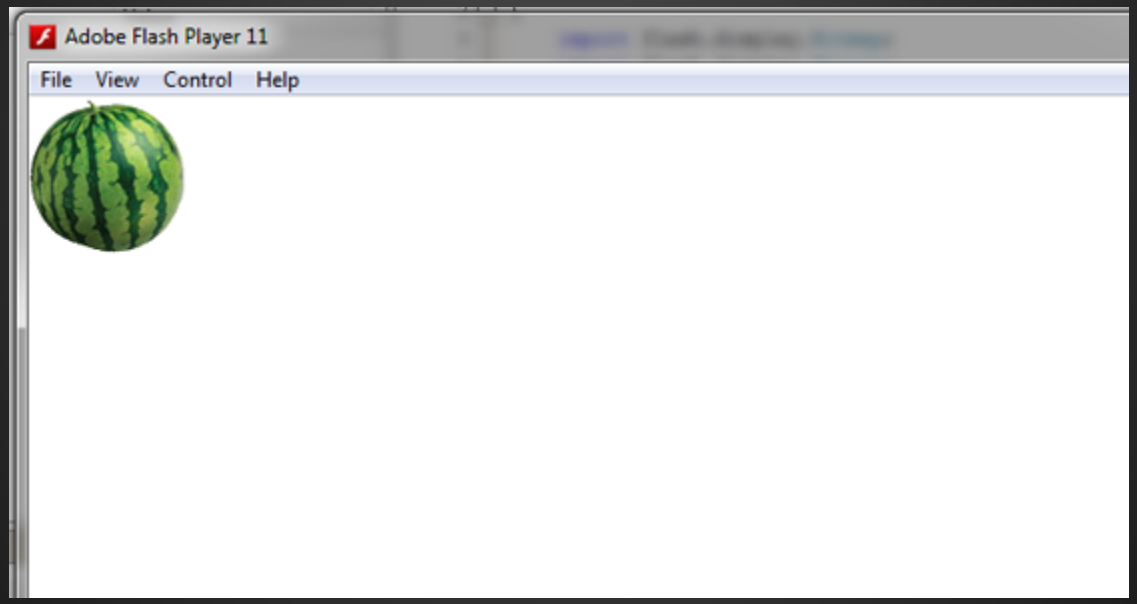


Desenhando na Tela - 3

Desenhos de imagens PNG, JPEG, etc

```
...
    [Embed(source="../assets/graphics/melancia.png")]
    public static var melanciaIMG:Class;

    private function init(e:Event = null):void
    {
        var melanciaNoJogo:Bitmap = new melanciaIMG();
        addChild(melanciaNoJogo);
    }
```

Desenhando na Tela - 4



Animações - 1

Engines como Flixel e FlashPunk

Necessário Sprite Sheet

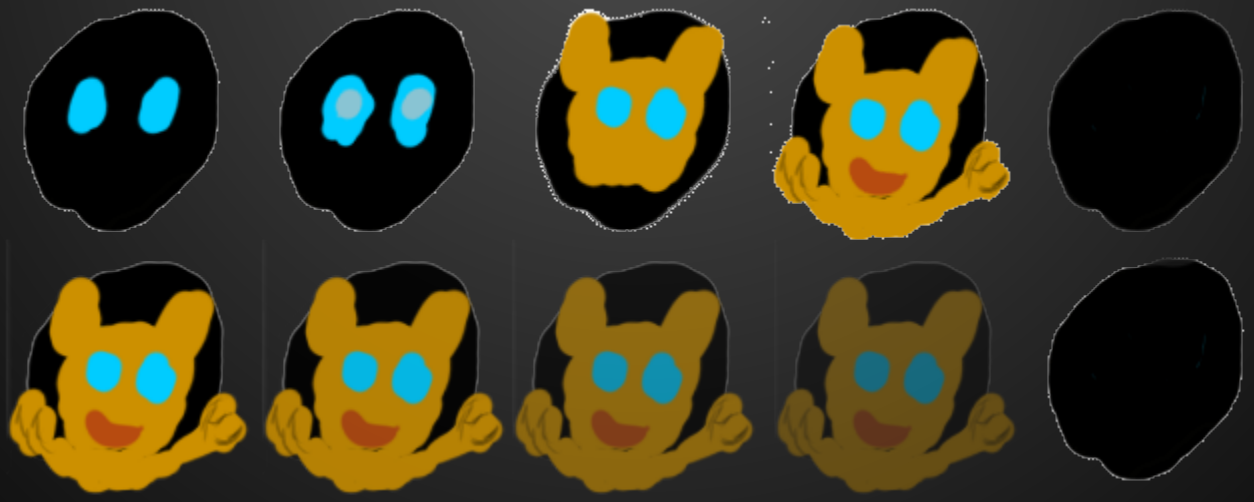
Animações - 2

A MrEngine possui suporte a animações por Sprite Sheet

```
[Embed(source = "../assets/graphics/macaquinho.png")]
static private var MacaquinhoIMG:Class;

public function Macaquinho(...){
    // tamanho de cada quadro
    super(Macaquinho, 128, 128);

    ...
```

Animações - 3

```
// (nome, [frames], loop, fps)
addAnimation("idle", [4]);
addAnimation("aparecendo", [0, 1, 2, 3], false);
addAnimation("esperando", [3]);
addAnimation("saindo", [5, 6, 7, 8, 9], false, 10);

play("idle");
```



Verificando Inputs do Jogador - 1

```
private function input():void
{
    if (keyboard.Down("left"))
    {
        nave.x -= nave.velocidade;
    }
    if (keyboard.Down("right"))
    {
        nave.x += nave.velocidade;
    }
    ...
```

Verificando Inputs do Jogador - 2

```
    ...
    if (keyboard.JustPressed("X"))
    {
        var tiro:Tiro = tiros[iTiroAtual] as Tiro;

        tiro.fire(nave.x + (nave.frameWidth >> 1),
    nave.y - (nave.frameHeight) + 15);

    }
}
```

INTERVALO

Colisão - 1

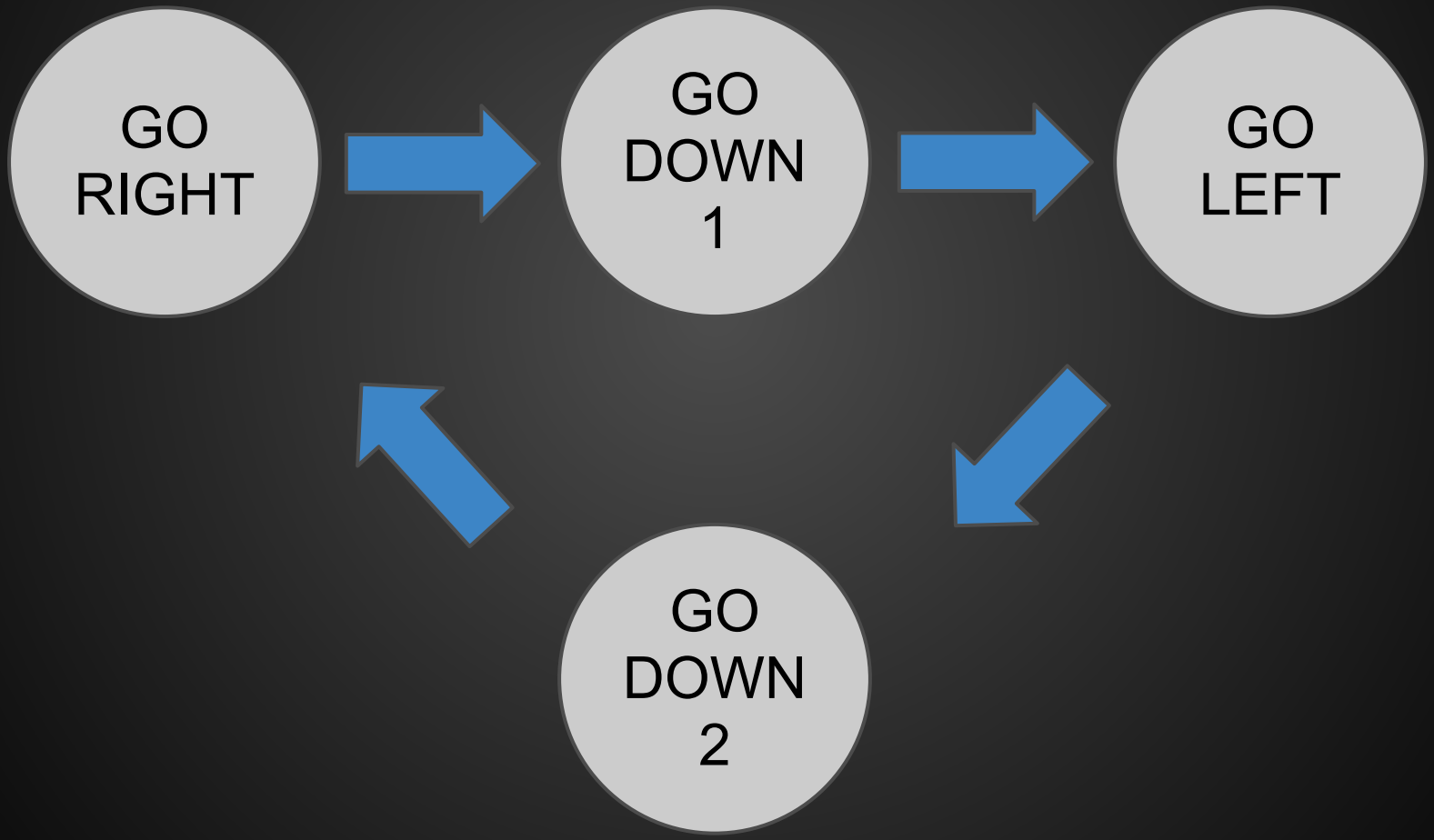
Bounding Boxes

Colisão - 2

Pseudo Código

```
para todo monstro m{
    para todo tiro t{
        se m.colide(t){
            m.matar()
            t.matar()
            pontuar(m.pontos)
        }
    }
}
```

Máquina de Estados - 1



Máquina de Estados - 2

```
if (estado == 'GO_RIGHT') {
   velocity.x = SPEED;
   if (stateCounter > STATE_MAX_HORIZONTAL_COUNTER){
      mudaEstado('GO_DOWN1');
   }
   if (estado == 'GO_DOWN1') {
      velocity.y = SPEED;
      if (stateCounter > STATE_MAX_VERTICAL_COUNTER){
               mudaEstado('GO_LEFT');
      }
   }
...
```

Gerência de Fases

```
//Em detecção de colisão:
   if(m.colide(t))
      numDead++;


...
if (numDead==MAX_MONSTROS)
{
   MrH.SwitchState(GameState);
}
```

HUD - 1

Score

```
class HUD{
    ...

    levelDisplay = new MrText(String(Registry.level), "Fixedsys", 30,
0xFFFFFF);
    levelDisplay.x = -520;
    addChild(levelDisplay);

    updateLevel();
    }

    public function updateLevel():void
    {
        levelDisplay.updateText(String("LEVEL " + Registry.level));
    }
```

HUD - 2

```
// na função update
if (numDead==MAX_MONSTROS)
{
    passarDeFase();
}

private function passarDeFase():void
{
    Registry.level++;
    hud.updateLevel();
    MrH.SwitchState(GameState);
}
```

Resultado Final

Melhorias

Adição de

- Powerups
- Game Over
- Inimigos com comportamentos e velocidades diferentes
- Efeitos Sonoros
- Limitar a área onde o jogador pode se mover
- Menu Inicial
- Ranking Online no Facebook (avançado)

Livros

1. Real-World Flash Game Development
   (Christopher Griffith)
2. Flash Game Development by Example
   (Emanuele Feronato)
3. Game Engine Architecture
   (Jason Gregory)
4. Programming Game AI by Example
   (Mat Buckland)