

Java Play

Sumário

- Etendendo o Básico
- Esqueleto dos Exemplos
- Exemplos

Sprites

O que é um Sprite?

É um objeto gráfico bi ou tridimensional que se move na tela sem deixar traços de sua passagem (como se fosse um "espírito"). No nosso caso até o fundo da tela será um sprite, só que esse não se moverá.

Exemplo de **Sprites**.

Se houver mais de uma imagem no sprite o fundo deve ser uma cor sólida ou transparente.

Exemplo **de Sprite com várias imagens**.

Classes de controle

- O framework javaPlay usa números inteiros para chamar alguma classe que é responsável por alguma coisa.
- Esse comando adiciona uma nova classe ao jogo com o número X sendo o de chamada.
- `GameEngine.getInstance().
 addGameStateController(X, new Classe());`

Classes de controle

Classe	Função	Nº Chamada
TelaInicial	Apresenta uma mensagem no COMEÇO do jogo	0
Jogar	Responsável por comandar o jogo	1
TelaFinal	Apresenta uma mensagem no FIM do jogo	2

Classes de controle

GameEngine.getInstance().

addGameStateController(0, new TelaInicial());

GameEngine.getInstance().

addGameStateController(1, new Jogar ());

GameEngine.getInstance().

addGameStateController(2, new TelaFinal ());

Classe de inicialização

- Informa ao javaPlay qual é a classe que primeiramente deve ser chamada quando o jogo começar a rodar.
- `GameEngine.getInstance().`
`setStartingGameStateController(numero X);`
- Numero X = número da classe adicionada em
`addGameStateController.`

Passando o controle para outra classe

GameEngine.getInstance().

setNextGameStateController(numero X);

Diga qual o número X de chamada da próxima classe e pronto.

Capturando teclas

- Quando uma tecla foi pressionada?
- Capturamos um instância da classe Keyboard
- `Keyboard k = GameEngine.getInstance()..getKeyboard();`

`if (k.keyDown(Keyboard.TECLA_ESCOLHIDA) == true)`
`ou`
`if (k.keyDown(Keyboard.TECLA_ESCOLHIDA))`
tanto
faz.

- Retorna true se a tecla foi pressionada e false ao contrário.
- **`k.keyDown(TECLA ESCOLHIDA)`**

Teclas

- O nosso Keyboard fornece as seguintes teclas:
- **up_key**: tecla seta pra cima
- **left_key**: tecla seta para esquerda
- **right_key**: tecla seta para direita
- **down_key**: tecla para baixo
- **escape_key**: tecla scape de espaço
- **space_key**: tecla barra de espaço
- **enter_key**:tecla enter

Coordenadas X e Y

Primeiro, todo **Sprite** tem suas **próprias** coordenadas X e Y.

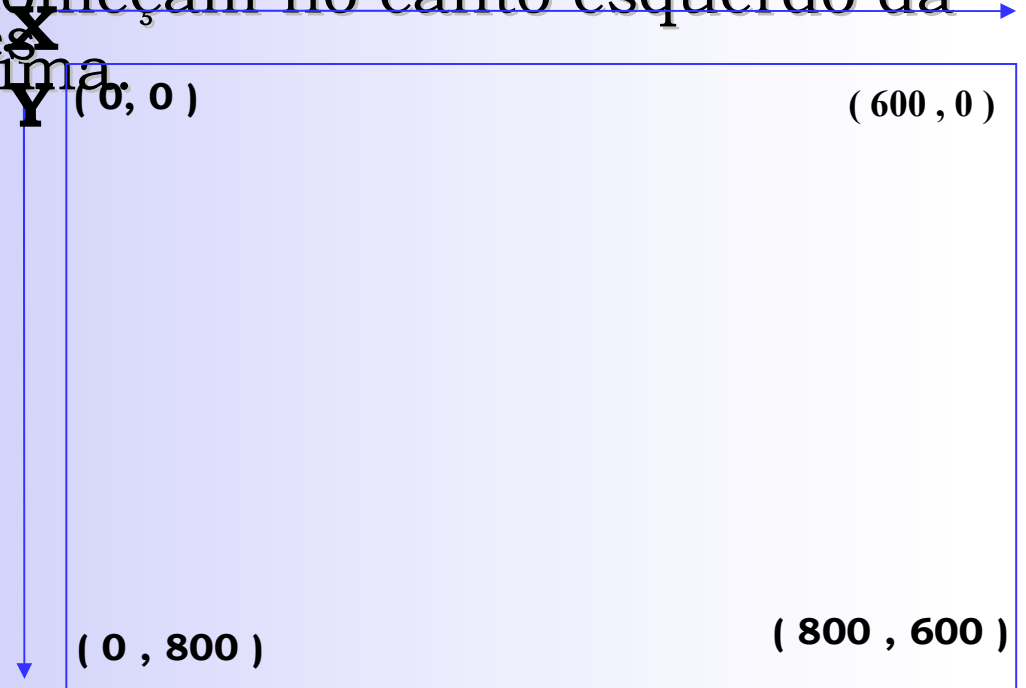
(X,Y)



Coordenadas X e Y

Java diz que a **TELA** tem as seguintes características:

As coordenadas começam no canto esquerdo da tela na parte de cima. Onde estão presentes o X=0 e o Y=0 e se estendem até o fim da tela que representando cada eixo.



Movendo um Sprite

Mover player para Cima

```
If ( k.keyDown(Keyboard.UP_KEY) == true )  
    sprite.y--;
```

Mover player para Baixo

```
if ( k.keyDown(Keyboard.DOWN_KEY) == true )  
    sprite.y++;
```

Mover player para Esquerda

```
If ( k.keyDown(Keyboard.LEFT_KEY) == true )  
    sprite.x--;
```

Mover player para Direita

```
If ( k.keyDown(Keyboard.RIGHT_KEY) == true )  
    sprite.x++;
```

Métodos de Controle

Métodos que fazem parte da classe **GameStateController** e são responsáveis por isso.

public void load() { } – carrega tudo na memória.

public void unload() { } – descarrega tudo da memória.

public void start() { } - chamado quando somente algumas coisas devem ser modificadas na memória.

public void step(int) { } – responsável por cada etapa do jogo. Passando o tempo em milissegundos decorrente entre o frame atual e o frame anterior, permitindo assim animações baseadas em tempo.

public void draw() { } – responsável pelo desenho dos objetos do jogo na tela.

Criando um Sprite

Pra criarmos um sprite devemos fornecer o **caminho** onde o sprite se localiza.

A **quantidade de imagens** em que o sprite é dividido e as suas **dimensões** em pixels.

Podemos utilizar imagens do tipo jpg e png nas imagens dos sprites.

Criando um Sprite

```
sprite = new Sprite ( caminho,  
                        Quantidade de Imagens,  
                        alturaPixels,  
                        larguraPixels );
```

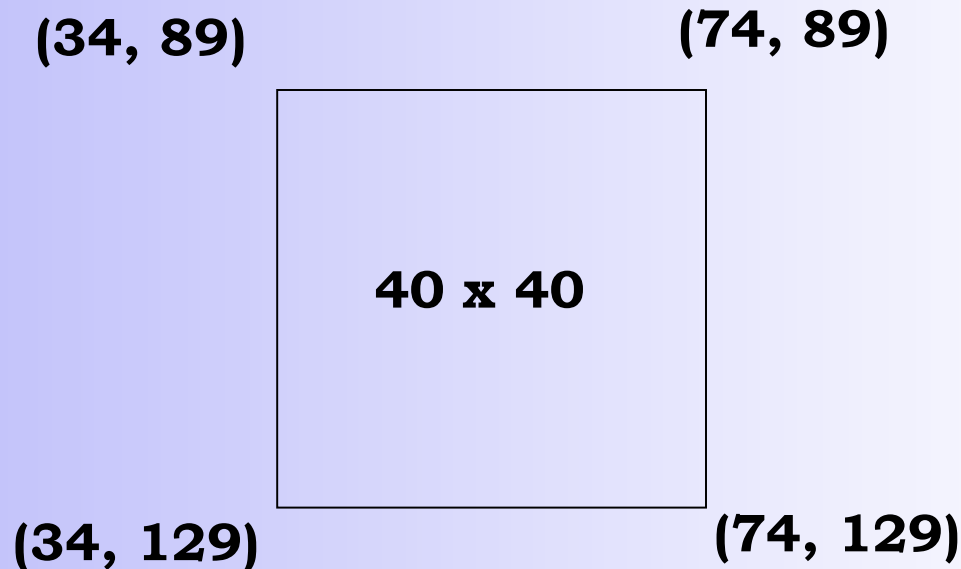
```
sprite = new Sprite( "imagem.png", 1, 250, 600 );
```


Desenhando o Sprite

- `Sprite.draw(g, coordenadaX, coordenadaY) ;`
- As coordenadas X e Y são pontos onde o sprite começará a ser desenhado.
- O g apresentado é fornecido pela Java, portanto, não precisamos nos preocupar com ele.

Exemplo

Ex.: tamanho do sprite = 40x40, X=34, Y=89.

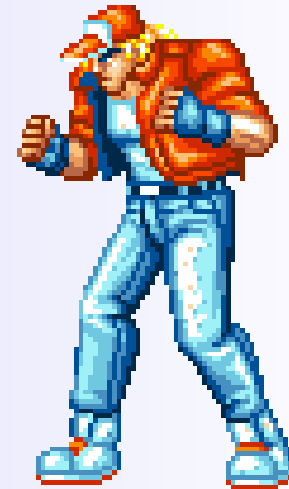


Fechando o jogo

Para fechar um jogo basta usar o seguinte comando em qualquer parte do código:

```
GameEngine.getInstance()  
    .requestShutdown();
```

Exemplos de Sprites

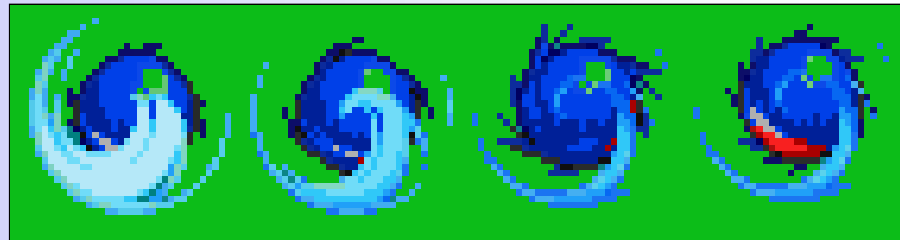


[Voltar](#)

Exemplos de Sprites



1 imagem
Fundo transparente



4 imagens
Fundo Verde



2 imagens
Fundo Laranja

[Voltar](#)