



Apresentação JPlay

Alessandro Campello Silva

Monitor de Programação de Computadores I

O que é JPlay?

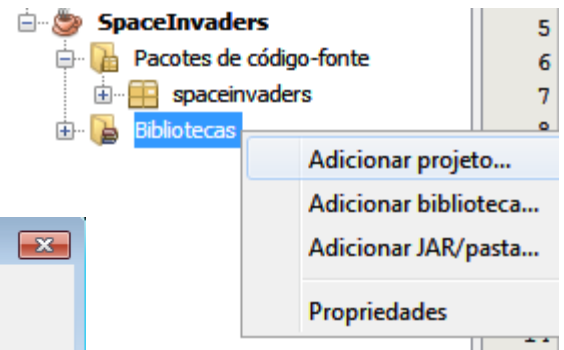
- Framework com funções para facilitar a manipulação de imagens na tela.
- Ajudar a criar jogos em 2D.

Como adicionar o JPlay

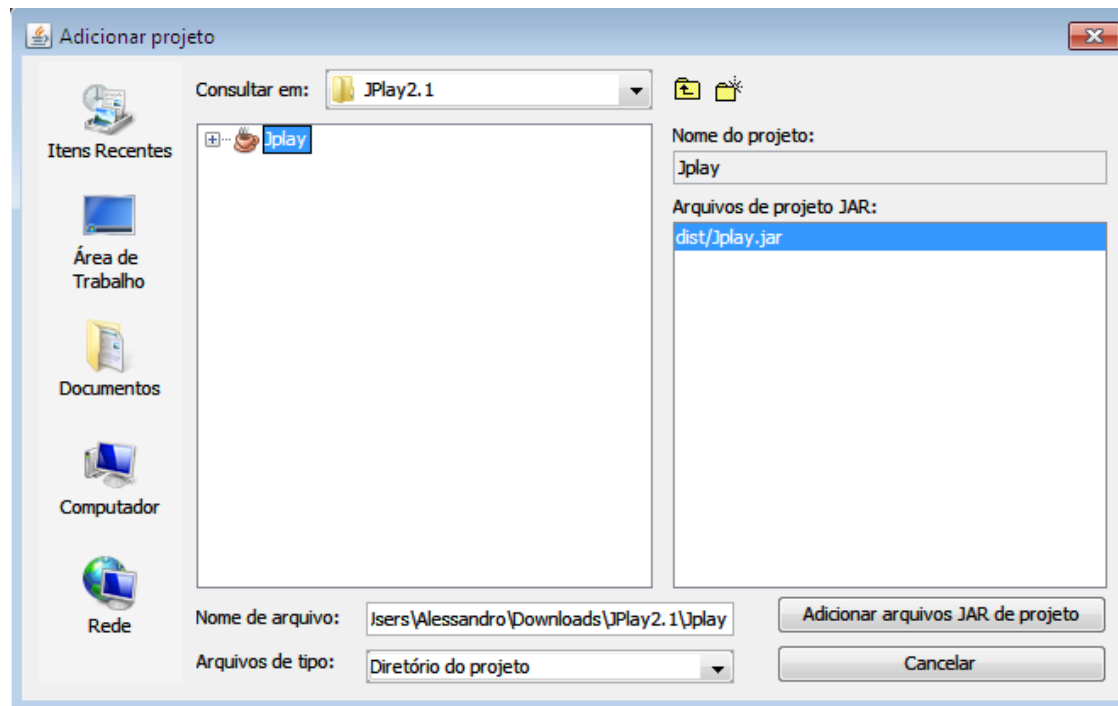
Baixar o JPlay v2.1 no site:

http://www.ic.uff.br/jplay/jplay_download.html

Após descompactá-lo, clicar com o botão direito na aba “Bibliotecas” (Libraries) do seu projeto:



E selecioná-lo no local que foi salvo:



Classes JPlay

- Window

- É a janela onde será feito o jogo
- Construtor:

```
Window janela = new Window(800, 600);
```

- Para atualizar a janela:

```
janela.update();
```

- GameImage

- É usada como fundo (background) do jogo
- Para criar a imagem: (é necessário salvá-la dentro da pasta do projeto)

```
GameImage fundo = new GameImage("fundo.jpg");
```

- Para desenhar a imagem na tela:

```
fundo.draw();
```

Classes JPlay

- Keyboard
 - É a classe que interpreta o que o usuário digita no teclado.
 - Construtor:

```
Keyboard keyboard = janela.getKeyboard();
```

- Saber se uma tecla está pressionada: (retorna verdadeiro ou falso)

```
keyboard.keyDown(Keyboard.UP_KEY)
```

- Para verificar constantemente se alguma tecla está pressionada, usaremos essa função dentro de uma repetição. Essa repetição é chamada de GameLoop, dentro dela ficará todas as ações do jogo

Classes JPlay

- Mouse

- Classe que detecta os movimentos do mouse. Para usá-la, criamos dessa forma:

```
Mouse mouse = janela.getMouse();
```

- Para saber se o mouse está sobre um objeto:

```
mouse.isOverObject(img)
```

- Para saber se algum botão do mouse está pressionado:

```
mouse.isLeftButtonPressed()
```

- Para saber a posição do mouse no momento:

```
mouse.getPosition()
```

Classes JPlay

- Sprite

- Essa classe tem uma imagem (ou várias), pode se mover pela tela, e outros métodos úteis para o jogo. Para criá-la:

```
Sprite boneco = new Sprite("nave.png");
```

- Ou caso tenha mais de uma imagem

```
Sprite boneco = new Sprite("nave.png", 2);
```

- Para alterar suas coordenadas x e y:

```
boneco.x = 50;  
boneco.y = 500;
```

- Para mover o Sprite horizontalmente ou verticalmente:

```
boneco.moveX(5); boneco.moveY(5);
```

Classes JPlay

- Sprite

- Mover um Sprite até um ponto, sem intervenção do usuário:

```
boneco.moveTo(boneco.x, 0, 10);
```

- Para atualizar a imagem do Sprite:
`boneco.update();`

- Para testar se aconteceu alguma colisão entre dois Sprites, basta usar o método

```
sprite1.collided(sprite2);
```