Introdução ao JPlay

Bruno Schettino

O que é JPlay?

É uma biblioteca que ajuda a criar jogos 2D usando a linguagem Java.

Baixando o JPlay

Acesse o link:
http://www2.ic.uff.br/jplay/jplay_download.html

Escolha a opção mais nova. (Hoje é a 3.0)

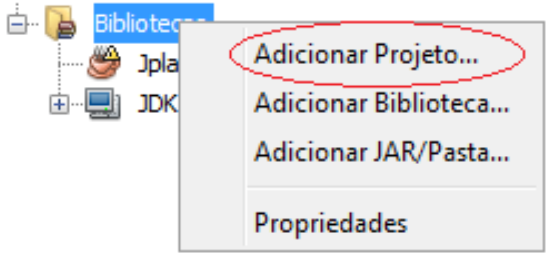
Criando o projeto no NetBeans

1. Clique em “Novo Projeto” 
2. Selecione “Aplicação Java” e clique em “Próximo”
3. Digite o nome do Projeto.
4. Selecione a opção “Usar Pasta Dedicada para Armazenar Bibliotecas”
5. Clique em “Finalizar”

Adicionando o JPlay no seu projeto

1. Extraia o arquivo do JPlay dentro da pasta “lib” do seu projeto

2. No NetBeans, clique com o botão direito em “Bibliotecas” e em seguida “Adicionar Projeto”



3. Procure a pasta do JPlay que extraímos dentro da pasta “lib” do projeto.
4. Clique em “Adicionar arquivos JAR de projeto”

Como funciona o JPlay?

É baseado em um loop. Em em cada iteração, devemos atualizar todos os elementos a serem desenhados e capturar as ações necessárias.

Criando a janela do jogo

- Na classe principal, gerada automaticamente na criação do projeto, insira a importação *import jplay.Window;*

- Para criar uma janela com dimensões 800px por 600px:

```
public static void main(String[] args) {
    Window janela = new Window(800, 600);
}
```

Capturando o teclado

- Insira a importação *import jplay.Keyboard;*
- A classe Keyboard interpreta a entrada do teclado.
- Ao criar a classe Window, uma instância da classe Keyboard é criada. Para acessar essa instância, utilizamos o método *getKeyboard().*
- Para checkar se uma tecla específica foi pressionada, utilizamos o método *keyDown* de uma instância da classe Keyboard.

Capturando o teclado

- Se desejarmos terminar o jogo ao pressionar a tecla ESC:

```
public static void main(String[] args) {
    Window janela = new Window(800, 600);
    Keyboard keyboard = janela.getKeyboard();
    boolean executando = true;
    while (executando) {
        janela.update();
        if (keyboard.keyDown(Keyboard.ESCAPE_KEY)) {
            executando = false;
        }
    }
    janela.exit();
}
```

Criando o plano de fundo

- A imagens do jogo são gerenciadas pela classe GameImage.
- Para criar o plano de fundo do jogo, vamos baixar uma imagem exatamente do mesmo tamanho da nossa janela. No caso, 800x600.
- Importante: O Java só aceita imagens dos tipos png, jpeg e gif!
- Sugestão: Crie uma pasta chamada “images” na raiz do projeto para guardar as imagens.

Criando o plano de fundo

- Insira a importação *import jplay.GameImage;*

- Para instanciar a imagem de fundo:

```
GameImage bg = new GameImage("images/bg.jpg");
```

- Para desenhar a imagem na janela:

```
while (executando) {
    bg.draw();
    janela.update();
    if (keyboard.keyDown(Keyboard.ESCAPE_KEY)) {
        executando = false;
    }
}
```

Criando o plano de fundo

Importante: A chamada método draw do plano de fundo deve vir antes do comando janela.update()!

Capturando o mouse

- Insira a importação *import jplay.Mouse;*
- A classe Mouse interpreta a entrada do mouse.
- Analogamente à classe Keyboard, ao criar a classe Window, uma instância da classe Mouse é criada. Para acessar essa instância, utilizamos o método *getMouse().*
- Para checkar se o mouse está sobre um objeto:

```
mouse.isOverObject(obj);
```

Capturando o mouse

- Para saber se o botão esquerdo foi pressionado:

```
mouse.isLeftButtonPressed();
```

- Para saber a posição atual do mouse:

```
mouse.getPosition();
```

Sprites

- São imagens ou coleções de imagens que usamos quando queremos fazer animações ou movimentações.
- Para criar:

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

- 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);
```

Sprites

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

```
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);
```

Querem mais?

www2.ic.uff.br/jplay/tutoriais.html