

4 – Usando o teclado – comportamento das teclas

Na parte 3 dessa apostila vimos um pequeno exemplo de como usar o teclado. Agora, veremos como adicionar novas teclas e mudar o comportamento das teclas já existentes.

4.1 – Teclas Padrões

As teclas *defaults* do teclado do JPlay são as seguintes:

`DOWN_KEY`, `ENTER_KEY`, `ESCAPE_KEY`, `LEFT_KEY`, `RIGHT_KEY`, `SPACE_KEY`, `UP_KEY`.

4.2 – Comportamento das teclas

As teclas presentes no teclado do JPlay possuem um dos dois comportamentos: `DETECT_EVERY_PRESS` ou `DETECT_INITIAL_PRESS_ONLY`.

Antes de vermos o que cada comportamento significa, tenha em mente o seguinte: quando uma tecla está pressionada o Java dispara um evento de tecla pressionada. Esse é um detalhe importante para entender os comportamentos citados:

`DETECT_EVERY_PRESS` - a cada iteração do loop o método `keyDown()` do teclado retorna `true` se a tecla estiver pressionada, ou seja, *retorna `true` enquanto a tecla estiver pressionada*.

`DETECT_INITIAL_PRESS_ONLY` - se a tecla tiver esse comportamento, o método `keyDown()` só irá *retornar `true` no momento do pressionamento da tecla*, diferentemente do `DETECT_EVERY_PRESS` que retorna `true` enquanto a tecla estiver pressionada. Isso significa que o método `keyDown()` só irá retornar `true` outra vez quando a tecla for liberada e novamente pressionada.

O comportamento `DETECT_EVERY_PRESS` pode ser usado para a movimentação de um boneco. O `DETECT_INITIAL_PRESS_ONLY` pode ser usado para fazer um boneco ou nave atirar.

4.3 – Adicionando teclas

Para adicionar uma tecla ao teclado do JPlay, você necessita saber qual o código da tecla que será adicionada, para isso use a classe `KeyEvent`, que está presente no Java e ela guarda os códigos de teclas. Para acessar esses códigos digite `'KeyEvent.VK_'`.

Para adicionar uma tecla use o método `void addKey(int keyCode)` ou `void addKey(int keyCode, int behavior)` presentes na classe `Keyboard`.

Em `void addKey(int keyCode, int behavior)` os parâmetros são o código da tecla que se deseja adicionar e o comportamento da mesma.

Exemplo: adicionar a tecla *control* ao teclado do JPlay.

```
teclado.addKey( KeyEvent.VK_CONTROL )
```

Usando o método `addKey(int keyCode)` adicionamos uma tecla e seu comportamento será `DETECT_INITIAL_PRESS_ONLY`.

Se você quiser que o comportamento seja `DETECT_EVERY_PRESS`, use o método `addKey(int keyCode, int behavior)` do seguinte modo:

```
teclado.addKey( KeyEvent.VK_CONTROL, Keyboard.DETECT_EVERY_PRESS )
```

Obs.: Se for feita a tentativa de adicionar uma tecla já existente, a mesma será substituída por aquela que estiver sendo adicionada, pois, há a possibilidade da nova tecla ter um comportamento diferente da anterior.

4.4 – Comportamentos das teclas padrões

As teclas `UP_KEY`, `LEFT_KEY`, `RIGHT_KEY` e `DOWN_KEY` possuem o comportamento `DETECT EVERY PRESS`.

As teclas `ESCAPE_KEY`, `SPACE_KEY` e `ENTER_KEY` possuem o comportamento `DETECT_INITIAL_PRESS_ONLY`.

4.5 – Mudando o comportamento de uma tecla

Para mudar o comportamento de uma tecla utilize o método `void setBehavior(int key, int behavior)`, presente na classe `Keyboard` do `JPlay`.

Exemplo: Mudando o comportamento da tecla `UP_KEY`:

```
teclado.setBehavior(Keyboard.UP_KEY, Keyboard.DETECT_INITIAL_PRESS_ONLY);
```

Os parâmetros a serem passados são: o código da tecla e o novo comportamento.

4.6 – Removendo uma tecla

Para remover uma tecla do teclado do `JPlay` use o método `void removeKey(int key)`.

Exemplo 4: Adiciona a tecla G que será usada para encerrar o jogo e remove a tecla `ESCAPE`.

```
public class Exemplo04
{
    public static void main(String[] args)
    {
        Window janela = new Window(800,600);
        Keyboard keyboard = janela.getKeyboard();
        GameImage backGround = new GameImage("fundo.png");

        keyboard.addKey(KeyEvent.VK_G); //Adiciona a tecla G com o comportamento DETECT_INITIAL_PRESS_ONLY
        keyboard.removeKey(KeyEvent.VK_ESCAPE);

        boolean executando = true;
        while(executando)
        {
            backGround.draw();
            janela.display();
            if ( keyboard.keyDown(KeyEvent.VK_G) )
                executando = false;
        }
        janela.exit();
    }
}
```