

5 – Mouse

5.1 – Usando o mouse

A classe `Window` fornece uma instância da classe `Mouse`. Para usar a mesma faça do seguinte modo:

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

O método `Mouse getMouse()` existente na classe `Window` retorna uma instância da classe `Mouse`.

Assim como o teclado os botões do mouse também possuem comportamento e estes podem ser mudados. Porém, não é possível adicionar ou retirar botões da classe `Mouse`.

5.2 – Botões padrões do mouse

Eles são os seguintes: `BUTTON_LEFT`, `BUTTON_MIDDLE` e `BUTTON_RIGHT`.

Para acessar o código desses botões faça como se segue: `'Mouse.BUTTON_RIGHT'`.

Todos os botões padrões possuem o comportamento `DETECT_INITIAL_PRESS_ONLY`.

5.3 – Mudando o comportamento de um botão

Use o método `void setBehavior(int numberBotton, int behavior)`.

Os parâmetros a serem passados são: o código do botão a ser mudado e o comportamento pretendido do botão.

Exemplo: Mudando o comportamento do botão direito:

```
mouse.setBehavior(Mouse.BUTTON_RIGHT, Mouse.DETECT_EVERY_PRESS);
```

Agora, enquanto o botão direito estiver pressionado, o método `boolean isRightButtonPressed()`, descrito abaixo, irá retornar `true`.

5.4 – Botões pressionados

Para saber se um botão está pressionado use os seguintes métodos da classe `Mouse`:

`public boolean isLeftButtonPressed()` – retorna `true` se o botão esquerdo está pressionado, ao contrário, retorna `false`;

`public boolean isMiddleButtonPressed ()` – retorna `true` se o scroll do mouse está pressionado, ao contrário, retorna `false`;

`public boolean isRightButtonPressed ()` – retorna `true` se o botão direito do mouse está pressionado, ao contrário, retorna `false`;

5.5 – Recuperando a posição do mouse

Para recuperar a posição (x,y) do mouse na tela usamos o método `Point getPosition()`.

Exemplo:

```
Point posicaoMouse = mouse.getPosition();
```

A classe `Point` existente no `Java` serve para guardar os valores de (x,y) retornados por `mouse.getPosition()`. Esses valores são acessados da seguinte forma: `'posicaoMouse.x'` e `'posicaoMouse.y'`.

5.6 – Mouse sobre algum objeto do jogo

Para saber se o mouse está sobre algum objeto do jogo, use o método **boolean isOverObject(GameObject obj)**. Esse método retorna *true* se o mouse estiver sobre o objeto passado como parâmetro, caso contrário, retorna *false*;

O objeto passado como parâmetro pode ser um dos seguintes objetos: `GameObject`, `GameImage`, `Animation`, `Sprite` ou `Body`.

Exemplo:

```
Mouse mouse = janela.getMouse();
Sprite carro = new Sprite("carro.png");

if (mouse.isOverObject(carro))
    System.out.println("mouse está sobre o carro!")
else
    System.out.println("mouse NÃO está sobre o carro!")
```

5.7 – Mouse sobre uma determinada área

Para saber se o mouse está sobre uma determinada área use o **boolean isOverArea(Point start, Point end)**.

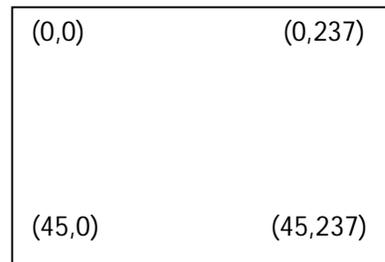
Ao lado está representada uma área com 45 pixels de altura e 237 pixels de largura.

Seus pontos mínimo e máximo são (0,0) e (45,237).

Para saber se o mouse está sobre esta área faça do seguinte modo:

```
Point pontoMinimo = new Point(0,0);
Point pontoMaximo = new Point(45,237).

if (mouse.isOverArea( pontoMinimo, pontoMaximo ) )
    imprime("Mouse está sobre a área");
else
    imprime("Não está sobre a área!")
```



O método **boolean isOverArea (int minX, int minY, int maxX, int maxY)**, poderia ser usado no lugar do **boolean isOverArea(Point start, Point end)**, ambas funcionam da mesma maneira.

O uso do método `boolean isOverArea (int minX, int minY, int maxX, int maxY)`, seria do seguinte modo:

```
if (mouse.isOverArea(0, 0, 45, 257) )
    imprime("Mouse está sobre a área");
```

Exemplo 05: Ao clicar com o mouse a imagem muda de posição

```
public class Exemplo05
{
    public static void main(String[] args)
    {
        Window janela = new Window(800,600);
        Keyboard keyboard = janela.getKeyboard();
        Mouse mouse = janela.getMouse();

        GameImage backGround = new GameImage("fundo.png");
        GameImage imagem = new GameImage("megaMan.png");
```

```
boolean executando = true;
while(executando)
{
    backGround.draw();
    imagem.draw();
    janela.display();

    if (mouse.isLeftButtonPressed() == true)//Se o mouse clicar muda a posição (x,y) da imagem
        imagem.setPosition( mouse.getPosition() );

    if ( keyboard.keyDown(Keyboard.ESCAPE_KEY) == true)
        executando = false;
}
janela.exit();
}
```

A ordem em que as imagens são desenhadas é importante. Se desenharmos primeiro a 'imagem' e depois o backGround, a 'imagem' será sobreposta.