

**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Câmpus Feliz

Imagens, eventos do teclado e mouse

Prof. Moser Fagundes

Programação II

Sumário

- Imagens
- Eventos do teclado
- Eventos do mouse
- Exercícios

Imagens

1. Crie o pacote **exemplo1**.
2. Crie uma pasta chamada **res** dentro do projeto.
3. Copie o arquivo **fundo.png** para a pasta **res**.
4. Crie uma tela em branco de **350x420**.

Imagens

```
package exemplo1;

import javax.swing.JFrame;

public class TesteCenario {

    public static void main(String[] args) {

        Cenario c = new Cenario();
        JFrame frame = new JFrame();
        frame.add(c);
        frame.setSize(350,420);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setResizable(false);
        frame.setVisible(true);
    }
}
```

Imagens

```
package exemplo1;
```

```
public class Cenario extends JPanel {
```

```
    public void paintComponent(Graphics g) {  
        super.paintComponent(g);
```

```
        try {  
            File arquivo = new File("res/fundo.png");  
            BufferedImage imagem = ImageIO.read(arquivo);  
            g.drawImage(imagem, 0, 0, 350, 400, null);  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }
```

```
}
```

Este código cria a imagem de fundo

Análise das linhas que criam a imagem na tela

Caminho e nome do arquivo da imagem

```
File arquivo = new File("res/fundo.png");  
BufferedImage imagem = ImageIO.read(arquivo);  
g.drawImage(imagem, 0, 0, 350, 400, null);
```

Lê o arquivo

Desenha

Altura da imagem

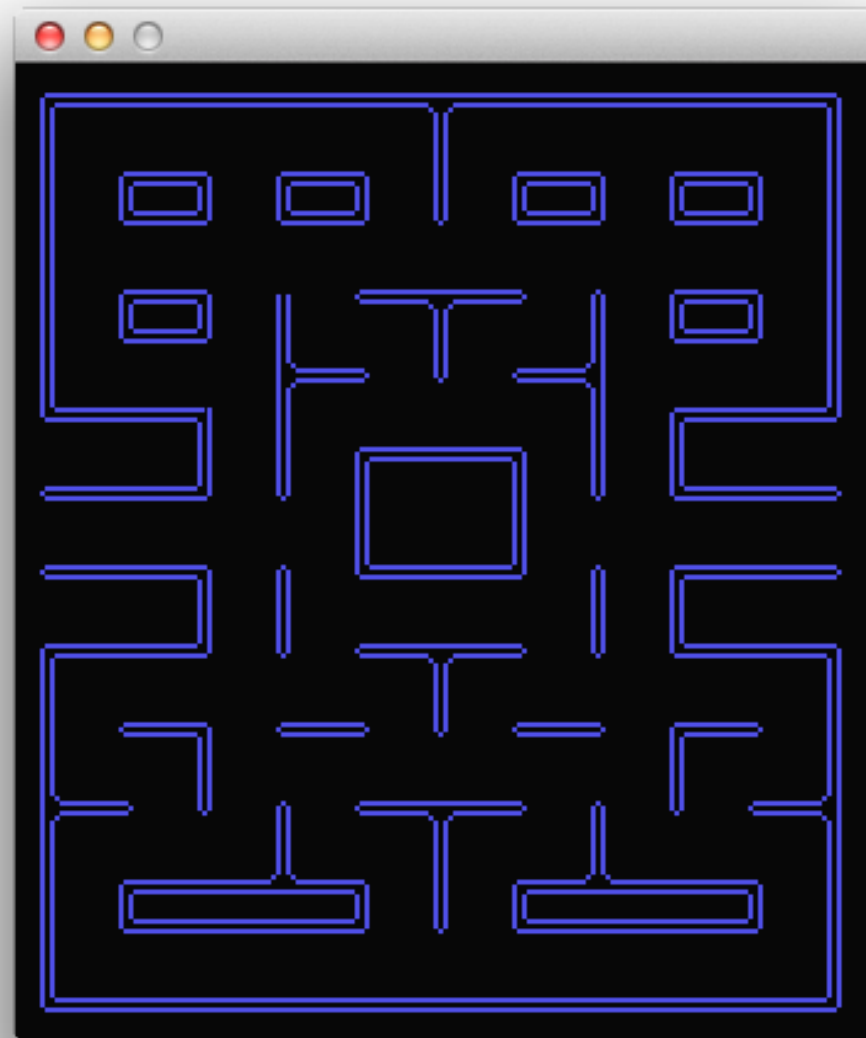
Largura da imagem

Coordenada Y (canto esquerdo superior da imagem)

Coordenada X (canto esquerdo superior da imagem)

The diagram illustrates the components of the provided Java code. Red arrows point from descriptive labels to specific parts of the code: 'Caminho e nome do arquivo da imagem' points to the string 'res/fundo.png'; 'Lê o arquivo' points to the 'read' method call; 'Desenha' points to the 'drawImage' method call; 'Altura da imagem' points to the value '400'; 'Largura da imagem' points to the value '350'; 'Coordenada Y (canto esquerdo superior da imagem)' points to the first '0' in the 'drawImage' call; and 'Coordenada X (canto esquerdo superior da imagem)' points to the second '0' in the 'drawImage' call.

Exemplo



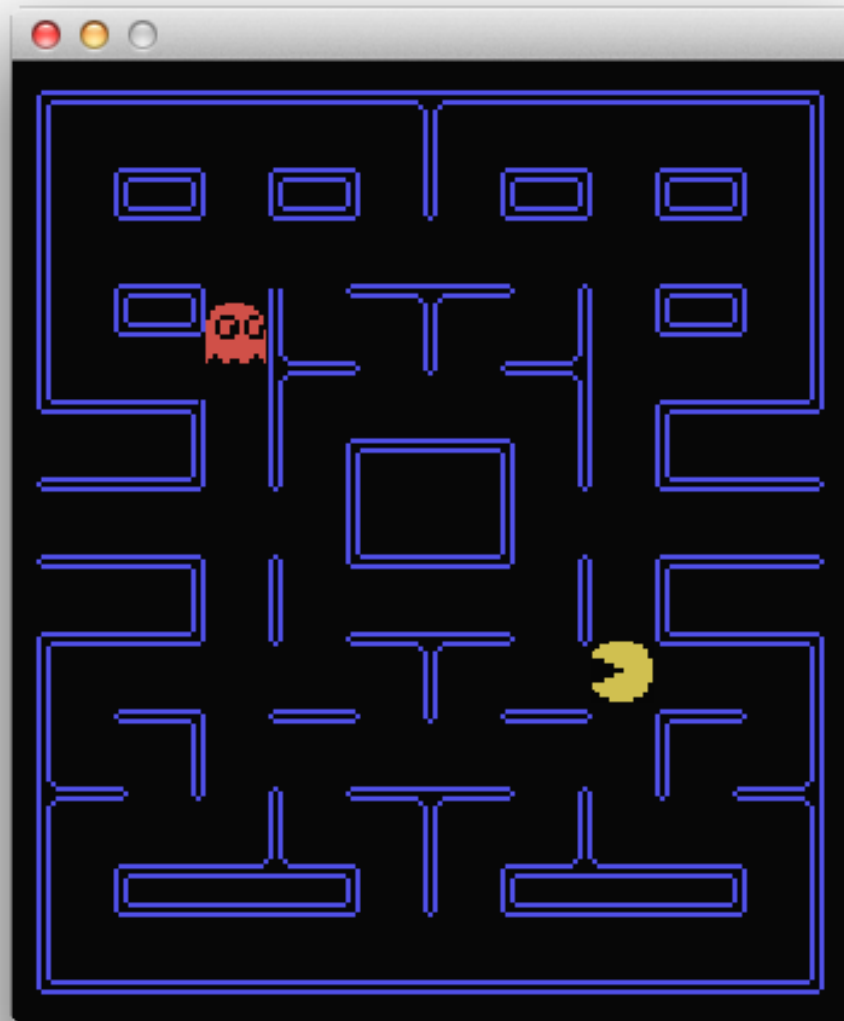
Para vocês fazerem...

Adicione o **Fantasma** e um **Pacman** no cenário.

Use os arquivos **fantasma.png** e **pacman.png**.
Os arquivos têm dimensão 25x25.



Resultado



Eventos do teclado

- Os **eventos do teclado** são capturados por um objeto de uma classe que contém métodos que são chamados quando uma tecla é pressionada ou solta.
 - Esta classe implementa a interface **KeyListener**
 - Devemos associar o objeto desta classe com o componente da interface gráfica que irá gerar os eventos (por exemplo, um **JFrame** ou **JPanel**)
 - Exemplo nos próximos slides...

Como capturar eventos do teclado?

- Criar uma tela usando um **JFrame**
- Criar um **JPanel** no qual vamos desenhar
- Criar os métodos da interface **KeyListener**
- Desenvolver os métodos para responder aos eventos do teclado


Criando uma tela com JFrame

```
package exemplo2;
```

```
import javax.swing.JFrame;
```

Logo estudaremos estas linhas

```
public class ExemploTeclado {  
    public static void main(String[] args) {  
        MeuPanel c = new MeuPanel();  
        c.setFocusable(true);  
        c.requestFocus();  
        JFrame frame = new JFrame();  
        frame.add(c);  
        frame.setSize(500,500);  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setVisible(true);  
    }  
}
```



Criando um JPanel

```
package exemplo2;

import javax.swing.JPanel;

public class MeuPanel extends JPanel {

    private static final long serialVersionUID = 1L;

    public MeuPanel () {
        super();
    }
}
```

Adicione implements KeyListener

```
package exemplo2;
```

```
import java.awt.event.KeyListener;
```

```
import javax.swing.JPanel;
```

```
public class MeuPanel extends JPanel implements KeyListener {
```

```
    private static final long serialVersionUID = 1L;
```

```
    public MeuPanel () {
```

```
        super();
```

```
        addKeyListener(this);
```

```
    }
```

```
}
```

Adiciona o listener

Clique para adicionar
os métodos não
implementados.

```
package exemplo3;
```

```
import java.awt.event.KeyEvent;
```

```
import java.awt.event.KeyListener;
```

```
import java.util.Random;
```

```
import javax.swing.JPanel;
```

```
public class MeuPanel extends JPanel implements KeyListener {
```

```
    private s
```

```
    public Me
```

```
        super
```

```
        addKe
```

```
    }
```

```
}
```

➤ Add unimplemented methods

➤ Make type 'MeuPanel' abstract

📁 Rename in file (%2 R)

📁 Rename in workspace (\%2 R)

3 methods to implement:

- java.awt.event.KeyListener.keyTyped()
- java.awt.event.KeyListener.keyPressed()
- java.awt.event.KeyListener.keyReleased()

Formato dos métodos KeyListener

Os seguintes métodos serão inseridos:

```
public void keyPressed(KeyEvent e) {  
    // Chamado para quando uma tecla é pressionada  
    // É o primeiro na sequência  
}
```

```
public void keyTyped(KeyEvent e) {  
    // Chamado quando o caracter é enviado  
    // É o segundo na sequência  
}
```

```
public void keyReleased(KeyEvent e) {  
    // Chamado para quando uma tecla é solta  
    // É o último na sequência  
}
```

Formato dos métodos KeyListener

Os seguintes métodos serão inseridos:

```
public void keyPressed(KeyEvent e) {  
    System.out.println("Pressed – Desceu!");  
}
```

```
public void keyTyped(KeyEvent e) {  
    System.out.println("Typed – Caracter enviado!");  
}
```

```
public void keyReleased(KeyEvent e) {  
    System.out.println("Released – Subiu!");  
}
```

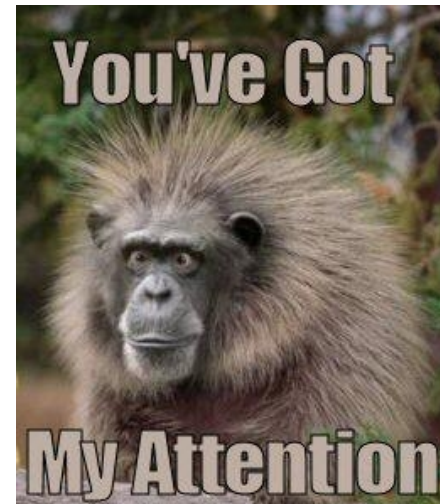

Request focus

```
package exemplo2;

import javax.swing.JFrame;

public class ExemploTeclado {
    public static void main(String[] args) {
        MeuPanel p = new MeuPanel();
        p.setFocusable(true);
        p.requestFocus();
        JFrame frame = new JFrame();
        frame.add(p);
        frame.setSize(500,500);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

No caso de associar um `KeyListener` com um `JPanel`, você terá que solicitar o foco da aplicação para o `JPanel`, caso contrário os métodos do listener nunca serão chamados e nada acontecerá quando você clicar no teclado.



Exemplo com a tecla M

Adicione o seguinte código:

```
public void keyPressed(KeyEvent e) {
```

```
    int key = e.getKeyCode();
```

Obtém o código
da tecla pressionada.

```
    switch(key) {
```

```
        case KeyEvent.VK_M : {
```

```
            JOptionPane.showMessageDialog(null, "Teclou M");  
            break;
```

```
        }
```

```
    }
```

```
}
```

Verifica qual tecla foi
pressionada pelo usuário.


Neste caso foi a tecla M.

Exemplo com a tecla Q

Adicione o seguinte código:

```
public void keyPressed(KeyEvent e) {  
    int key = e.getKeyCode();  
  
    switch(key) {  
        case KeyEvent.VK_M : {  
            JOptionPane.showMessageDialog(null, "Teclou M");  
            break;  
        }  
        case KeyEvent.VK_Q : {  
            Random r = new Random();  
            int x = r.nextInt(400);  
            int y = r.nextInt(400);  
            int width = 10 + r.nextInt(90);  
            int height = 10 + r.nextInt(90);  
            Graphics g = getGraphics();  
            g.drawRect(x, y, width, height);  
            break;  
        }  
    }  
}
```

Desenha o retângulo
chamando o método
drawRect().



Eventos do mouse

- Os **eventos do mouse** são capturados por um objeto de uma classe que contém métodos que são chamados quando um botão do mouse é pressionado ou liberado.
 - Esta classe implementa a interface **MouseListener**
 - Devemos associar o objeto desta classe com o componente da interface gráfica que irá gerar os eventos (por exemplo, um **JFrame** ou **JPanel**)
 - Exemplo nos próximos slides...



Como capturar eventos do mouse?

- Criar uma tela usando um **JFrame**
- Criar um **JPanel** no qual vamos desenhar
- Criar os métodos da interface **MouseListener**
- Desenvolver os métodos para responder aos eventos do mouse



Criando uma tela com JFrame

```
package exemplo3;

import javax.swing.JFrame;

public class ExemploMouse {
    public static void main(String[] args) {
        MeuPanel c = new MeuPanel();
        c.setFocusable(true);
        c.requestFocus();
        JFrame frame = new JFrame();
        frame.add(c);
        frame.setSize(500,500);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

Criando um JPanel

```
package exemplo3;

import javax.swing.JPanel;

public class MeuPanel extends JPanel {

    private static final long serialVersionUID = 1L;

    public MeuPanel () {
        super();
    }
}
```

Adicione implements MouseListener

```
package exemplo3;
```

```
import java.awt.event.MouseListener;
```

```
import javax.swing.JPanel;
```

```
public class MeuPanel extends JPanel implements MouseListener {
```

```
    private static final long serialVersionUID = 1L;
```

```
    public MeuPanel() {
```

```
        super();
```

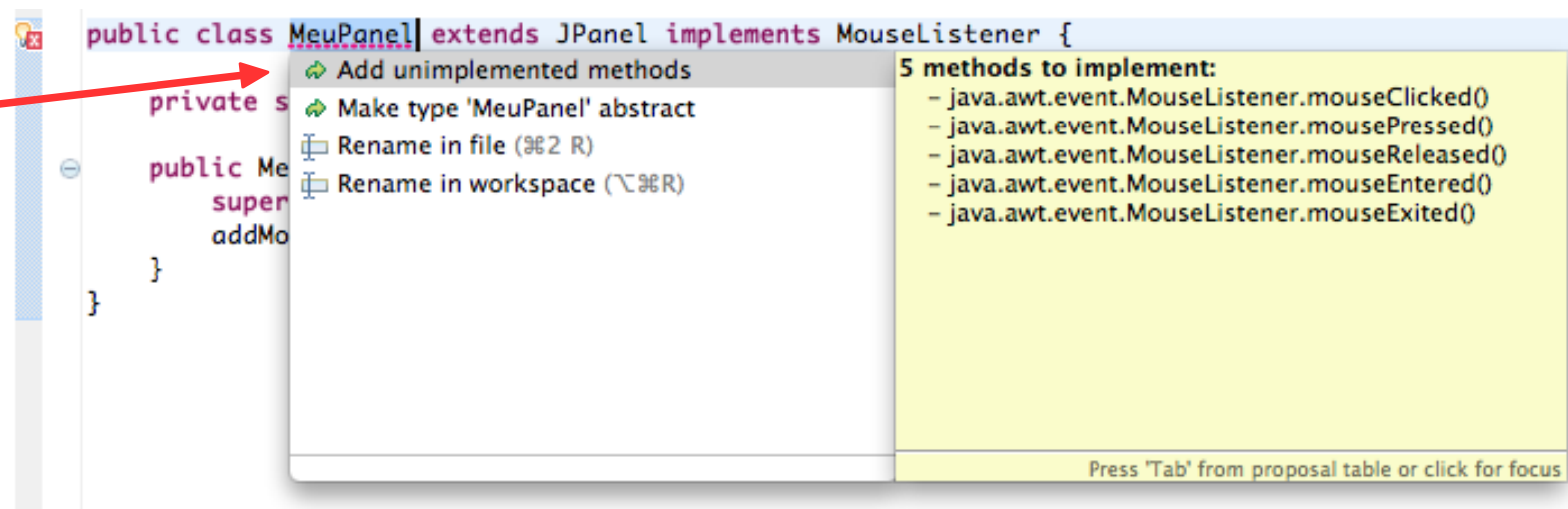
```
        addMouseListener(this);
```

```
    }
```

```
}
```

Adiciona o listener

Clique para adicionar os métodos não implementados.



Formato dos métodos **MouseListener**

```
public void mousePressed(MouseEvent e) {  
    // Chamado quando um botao do mouse é PRESSIONADO  
}
```

```
public void mouseClicked(MouseEvent e) {  
    // Chamado quando um botao do mouse é PRESSIONADO E LIBERADO,  
    // ou seja, quando há um clique completo e continuo  
}
```

```
public void mouseReleased(MouseEvent e) {  
    // Chamado quando um botao do mouse é LIBERADO  
}
```

```
public void mouseEntered(MouseEvent e) {  
    // Chamado quando o mouse ENTRA nos limites do componente  
}
```

```
public void mouseExited(MouseEvent e) {  
    // Chamado quando o mouse SAI nos limites do componente  
}
```

Adicione o seguinte código

```
public void mousePressed(MouseEvent e) {  
    setBackground(Color.RED);  
    System.out.println("Press - X:" + e.getX() + "Y:" + e.getY());  
}  
  
public void mouseReleased(MouseEvent e) {  
    setBackground(Color.BLUE);  
    System.out.println("Release - X:" + e.getX() + "Y:" + e.getY());  
}  
  
public void mouseClicked(MouseEvent e) {  
  
}  
  
public void mouseEntered(MouseEvent e) {  
  
}  
  
public void mouseExited(MouseEvent e) {  
  
}
```

Exercícios

Lista de exercícios no Moodle.