Interfaces Event-Driven
Técnica de Rubberband

Sistemas Gráficos/

Computação Gráfica e Interfaces

Interfaces Event-Driven

Modelo de Programa Event-Driven:

Neste modelo o programa está, por defeito, inactivo (“idle”) a aguardar a ocorrência de algum evento.

→ Os eventos são provocados pelo utilizador quando selecciona algum botão da interface, move o rato, pressiona teclas, etc.

Ciclo principal do programa:

```
Do
        wait for an event to arrive
        process the event
Until a stop event occurs
```

Interfaces Event-Driven

Event generator: objecto que origina o evento. Por exemplo um botão da interface.

Event Handler: Objecto ou função que processa a tarefa associada a um dado evento.

Como é que o um dado *Event Handler* sabe da ocorrência do evento que lhe interessa ?

Notar que as ocorrências são assíncronas, porque dependem das acções do utilizador.

Interfaces Event-Driven

O objecto que gera o evento tem de registar qual o objecto/função que processa esse evento, i.e. o *Event Handler*.

Exemplo em Java:

```
buttonLine = new javax.swing.JButton ();
buttonLine.addMouseListener (new java.awt.event.MouseAdapter () {
    public void mousePressed (java.awt.event.MouseEvent evt) {
        buttonLineMousePressed (evt);
        }
    }
);
...
private void buttonLineMousePressed (java.awt.event.MouseEvent evt)
       // Add your handling code here:
         changeTool(LINETOOL);
 }
```

Interfaces Event-Driven

Exemplo em C/Glut:

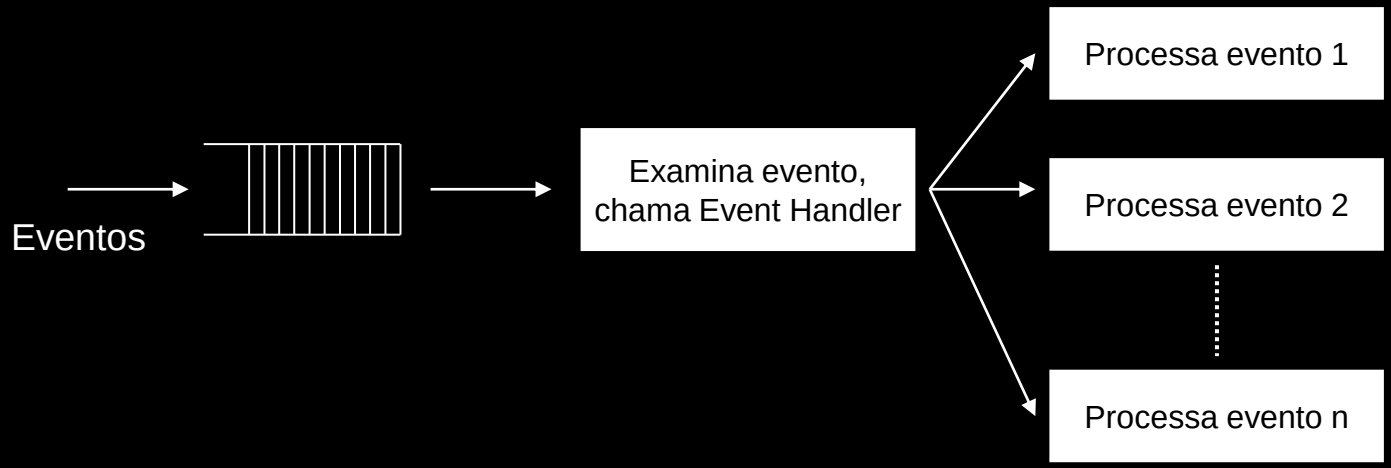
```
void display(void)
{ ...}

void processMouse(int button, int state, int x, int y)
{
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
    {     ... }
    if (button == GLUT_RIGHT_BUTTON && state == GLUT_DOWN)
    {     printf("x=%d y=%d \n", x, y);      }
    glutPostRedisplay();
}

int main(int argc, char** argv)
{
    glutDisplayFunc(display);
    glutKeyboardFunc(keyboard);
    glutMouseFunc(processMouse);
    glutMotionFunc(processMouseMoved);
     ...
}
```

Interfaces Event-Driven

Exemplo em C/Glut:

```
void keyboard(unsigned char key, int x, int y)
{
   switch (key) {
      case 27:         // tecla de escape termina o programa
         exit(0);
         break;
   }
}
```

Interfaces Event-Driven



Todas as acções do utilizador geram eventos que são enviados para o programa principal e guardados numa fila de espera, sendo atendidos sequencialmente. No processamento de cada evento o controlo passa temporariamente para o *Event Handler*.

Nota: o evento (mensagem) identifica o componente que o gerou de modo a que seja chamado o *Event Handler* correspondente.

Tipos de eventos: keyPressed, keyReleased, keyTyped
mouseDragged, mouseMoved, mouseClicked, mousePressed, mouseReleased, etc.

Interfaces Event-Driven

No java existem várias classes abstractas que facilitam o registo de *Event Handler* para cada componente. Só é necessário implementar os métodos correspondentes aos eventos que se pretendem atender. Por defeito são definidos vazios.

Class KeyAdapter Para processar eventos do teclado: keyPressed, keyReleased, keyTyped.

Class mouseAdapter Para processar eventos do rato: mouseClicked, mouseEntered, mouseExited, mousePressed, mouseReleased.

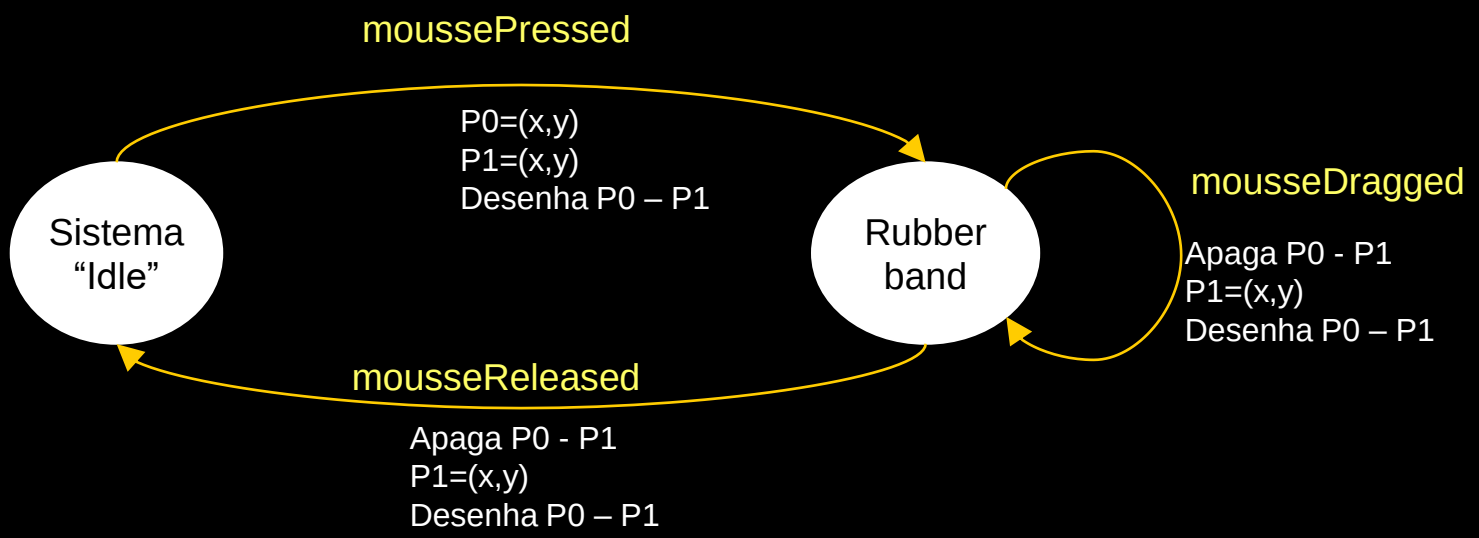
Class mouseMotionAdapter Para processar movimentos do rato: mouseMoved e mouseDragged.

Rubberband

A técnica de Rubberband consiste no desenho interactivo de objectos.
Linha:

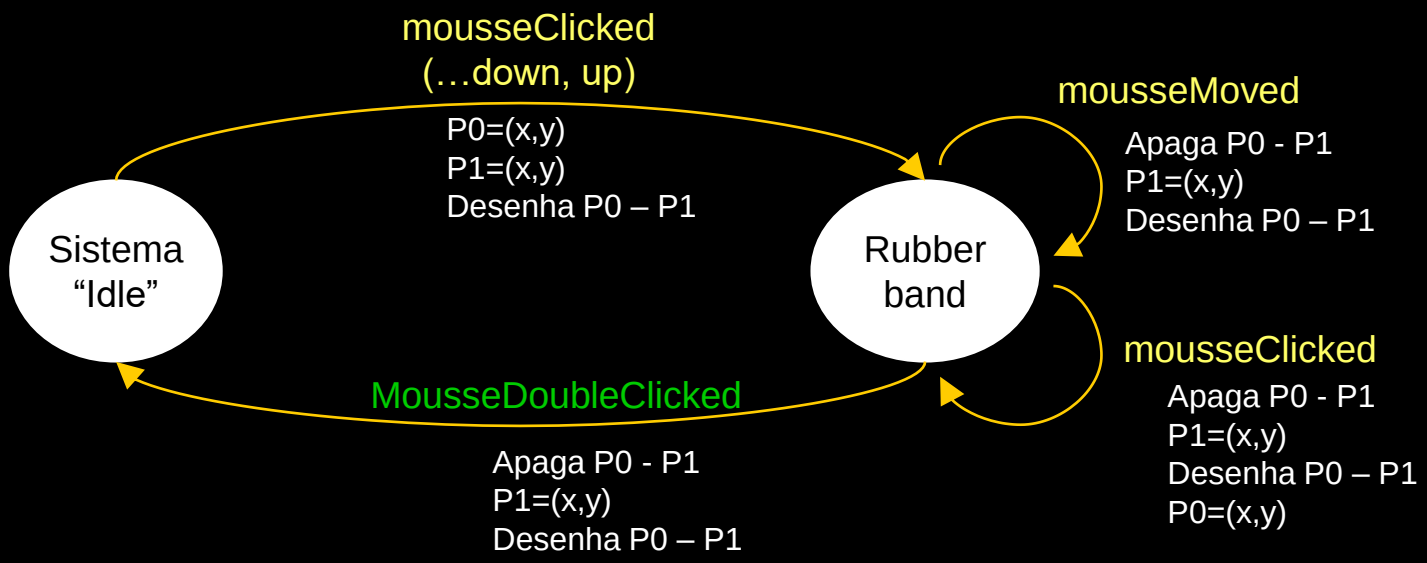
1. Pressionando o botão esquerdo selecciona-se o ponto inicial da linha
2. Enquanto se mantém botão pressionado, com o movimento do rato é desenhada uma linha entre ponto inicial e actual (mouseDragged)
3. Quando solta o botão, a posição actual é definida como a posição final
4. Desenhar a Linha definitiva entre os dois pontos

Diagrama de estados para o desenho de linha:



Rubberband

Diagrama de estados para o desenho de *polyline*:



MouseDoubleClick

```
private void myMouseClicked(java.awt.event.MouseEvent evt) {
// Add your handling code here:
    if( SwingUtilities.isLeftMouseButton(evt))
    {
      if(evt.getClickCount() == 1 )
        // click simples
      else if(evt.getClickCount() == 2 )
        // click duplo
      repaint();
    }
  }
```

Detecção do DoubleClick