

# Teclado e Mouse

## Programação de Jogos

Judson Santos Santiago

# Introdução

- ▶ Um **jogo é interativo** por natureza
  - Sem interatividade ele se torna uma:
    - Brincadeira
    - Estória
    - Filme
- ▶ O **dispositivo de interação** varia de acordo com:
  - O tipo de jogo  
corrida, simulação de voo, tiro em primeira pessoa
  - A plataforma para a qual ele foi desenvolvido  
PC, Tablet, Smartphone, Console

# Introdução

- Os dispositivos mais comuns são:



Teclado & Mouse  
PC



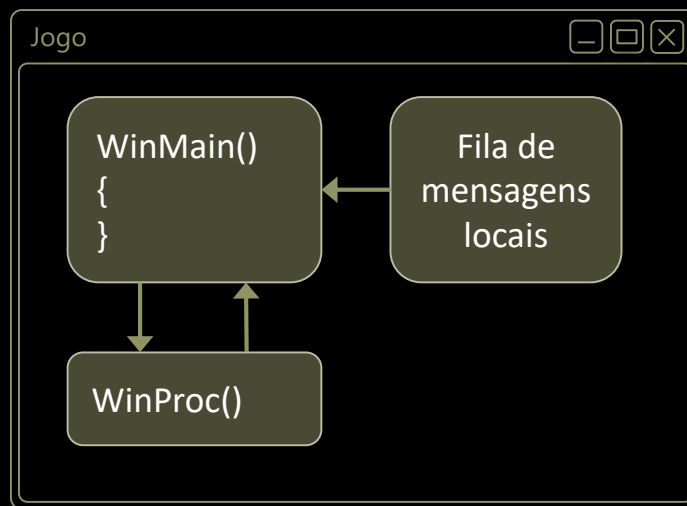
Tela de toque  
Smartphones & Tablets



Controle  
Consoles de Videogame

# Teclado

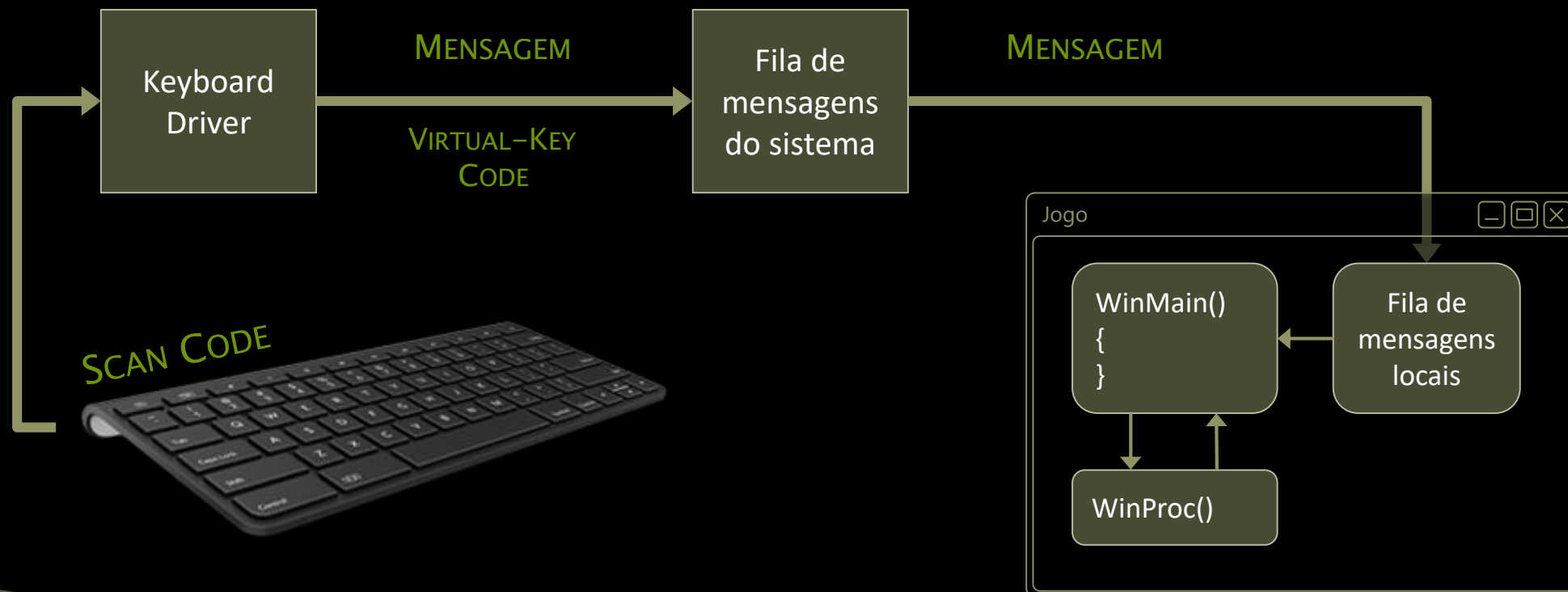
- ▶ Uma aplicação recebe dados do teclado na forma de **mensagens postadas** para sua janela



O Windows fornece suporte **independente de dispositivo** através de um driver

# Teclado

- ▶ O driver do teclado interpreta o **SCAN CODE** gerado e o traduz em uma **VIRTUAL-KEY CODE**



# Teclado

- ▶ Quando apropriado, a **VIRTUAL-KEY CODE** é traduzida em caracteres por TranslateMessage



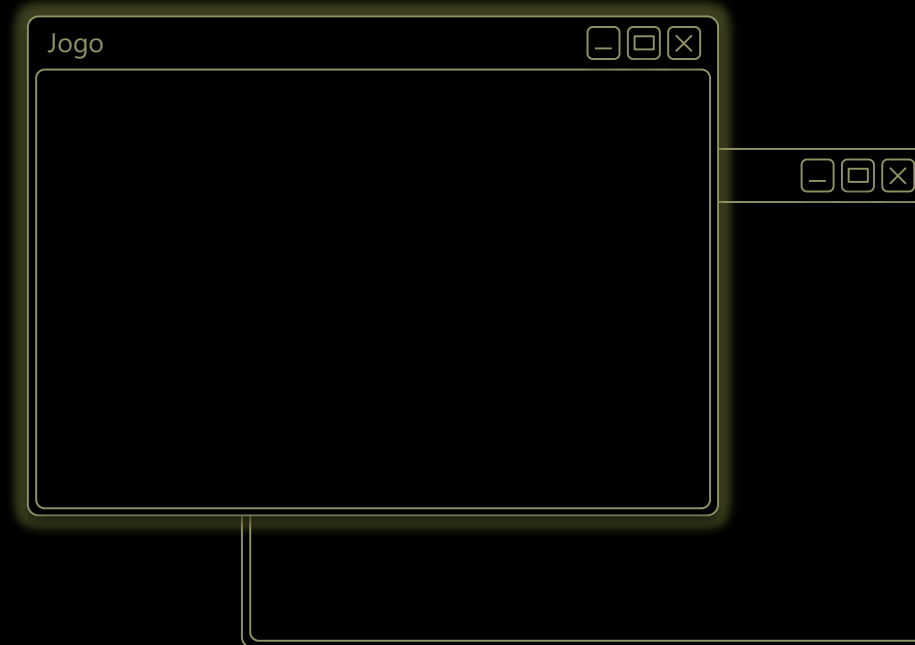
# Teclado

## ► Foco e Ativação:

- O sistema compartilha o teclado através do foco
- O foco está relacionado com a janela ativa

A janela que tem o foco  
recebe todas as mensagens  
de teclado

O sistema envia notificações sobre  
a mudança de foco através das  
mensagens **WM\_KILLFOCUS** e  
**WM\_SETFOCUS**





# Teclado

- ▶ O jogo pode tratar mensagens de **foco e ativação**

```
LRESULT CALLBACK WinProc (HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    switch (message)
    {
        case WM_KILLFOCUS:
            strcpy(msg, "Até Logo!");
            InvalidateRect(hwnd, NULL, TRUE);
            return 0;

        case WM_SETFOCUS:
            strcpy(msg, "Bem Vindo!");
            InvalidateRect(hwnd, NULL, TRUE);
            return 0;
    }

    return DefWindowProc(hwnd, message, wParam, lParam);
}
```



# Teclado

- ▶ Capturando o pressionamento de teclas
  - O pressionamento de qualquer tecla **gera uma mensagem**
    - Sistema: WM\_SYSKEYDOWN e WM\_SYSKEYUP (usando Alt)
    - Normal: **WM\_KEYDOWN** e **WM\_KEYUP**



WM\_KEYUP

# Teclado

- ▶ O código da tecla está no **wParam** da mensagem
  - É responsabilidade do jogo guardar o **estado das teclas**
    - Pode-se utilizar um vetor de 256 booleanos

```
LRESULT CALLBACK WinProc (HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    switch (message)
    {
        case WM_KEYDOWN:
            vkKeys[wParam] = true;
            return 0;

        case WM_KEYUP:
            vkKeys[wParam] = false;
            return 0;
    }

    return DefWindowProc(hwnd, message, wParam, lParam);
}
```



# Teclado

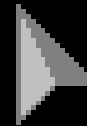
## ► Capturando Caracteres

- As mensagens **WM\_KEYDOWN** e **WM\_KEYUP** não informam o código ASCII da tecla pressionada
- A função **TranslateMessage**
  - Obtém o **código ASCII** e gera uma mensagem **WM\_CHAR** (apenas para teclas que representam caracteres)
  - O parâmetro **wParam** contém o valor do caractere

```
while (GetMessage(&msg, NULL, 0, 0))  
{  
    TranslateMessage(&msg);  
    DispatchMessage(&msg);  
}
```

# Mouse

- ▶ Ao mover o mouse, o sistema move uma **imagem** na tela chamada de cursor do mouse
  - Efetivamente o **cursor do mouse** é um único ponto na tela (*hot spot*)
- ▶ O mouse gera mensagens para:
  - Movimento do mouse
  - Pressionamento de um botão
  - Liberação de um botão
  - Giro do botão de rolagem



# Resumo

- ▶ Teclado e Mouse são lidos a partir do **tratamento de mensagens** encaminhadas para a janela
  - Teclado
    - Pode ser consultado sobre o pressionamento de teclas
    - **WM\_KEYDOWN**, **WM\_KEYUP**
    - Usado para obter os caracteres digitados  
**WM\_CHAR**
  - Mouse
    - Pode ser consultado sobre sua posição e estado dos botões  
**WM\_MOUSEMOVE**, **WM\_LBUTTONDOWN**, **WM\_LBUTTONUP**,  
**WM\_RBUTTONDOWN**, **WM\_RBUTTONUP**, **WM\_MBUTTONDOWN**, **WM\_MBUTTONUP**