

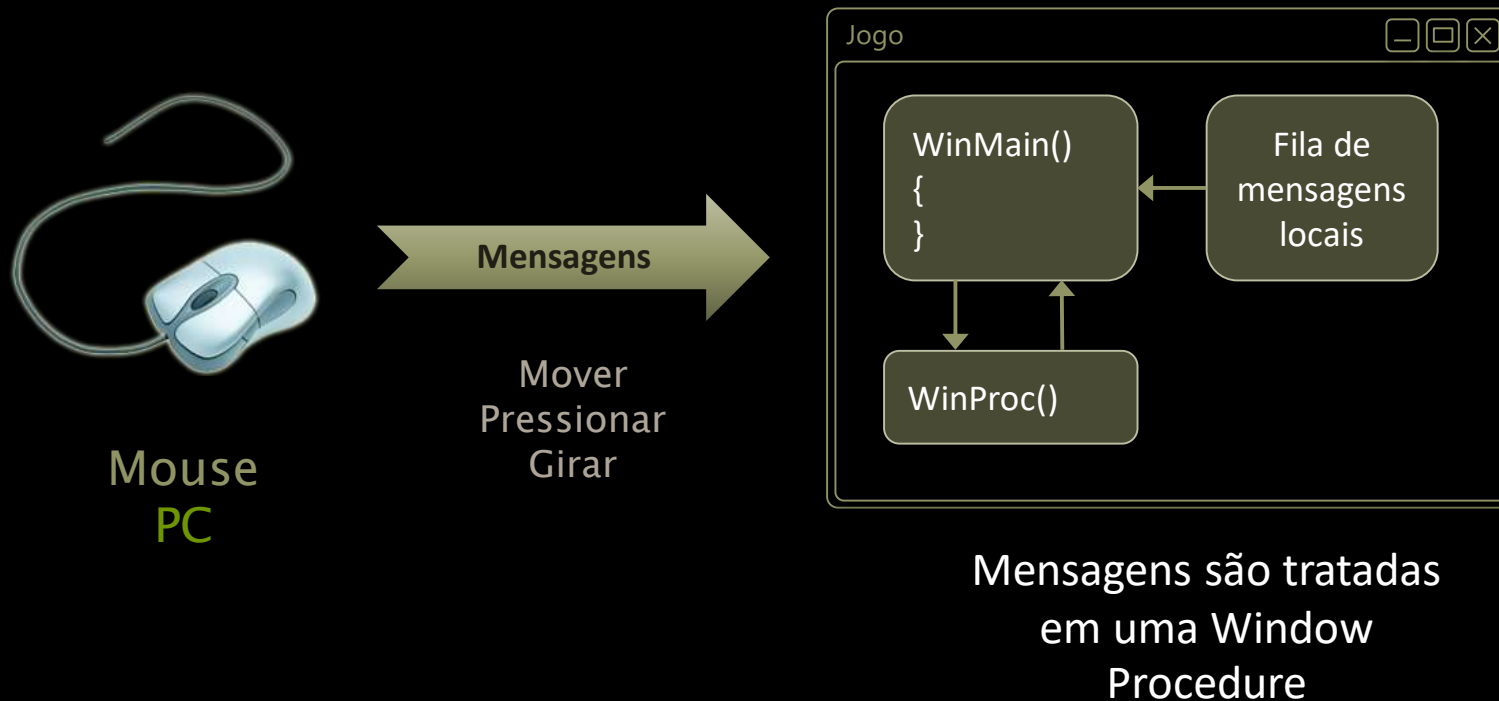
# Interação com o Mouse

Programação de Jogos

Judson Santos Santiago

# Introdução

- ▶ O jogo obtém **informações** do mouse **através do S.O.**



# Introdução

## ► Tratando mensagens do Windows

```
LRESULT CALLBACK Window::WinProc(HWND hWnd, UINT msg, WPARAM wParam, LPARAM lParam)
{
    switch(msg)
    {
        case WM_MOUSEMOVE: // movimento do mouse registrado
            windowMouseX = GET_X_LPARAM(lParam);
            windowMouseY = GET_Y_LPARAM(lParam);
            return 0;

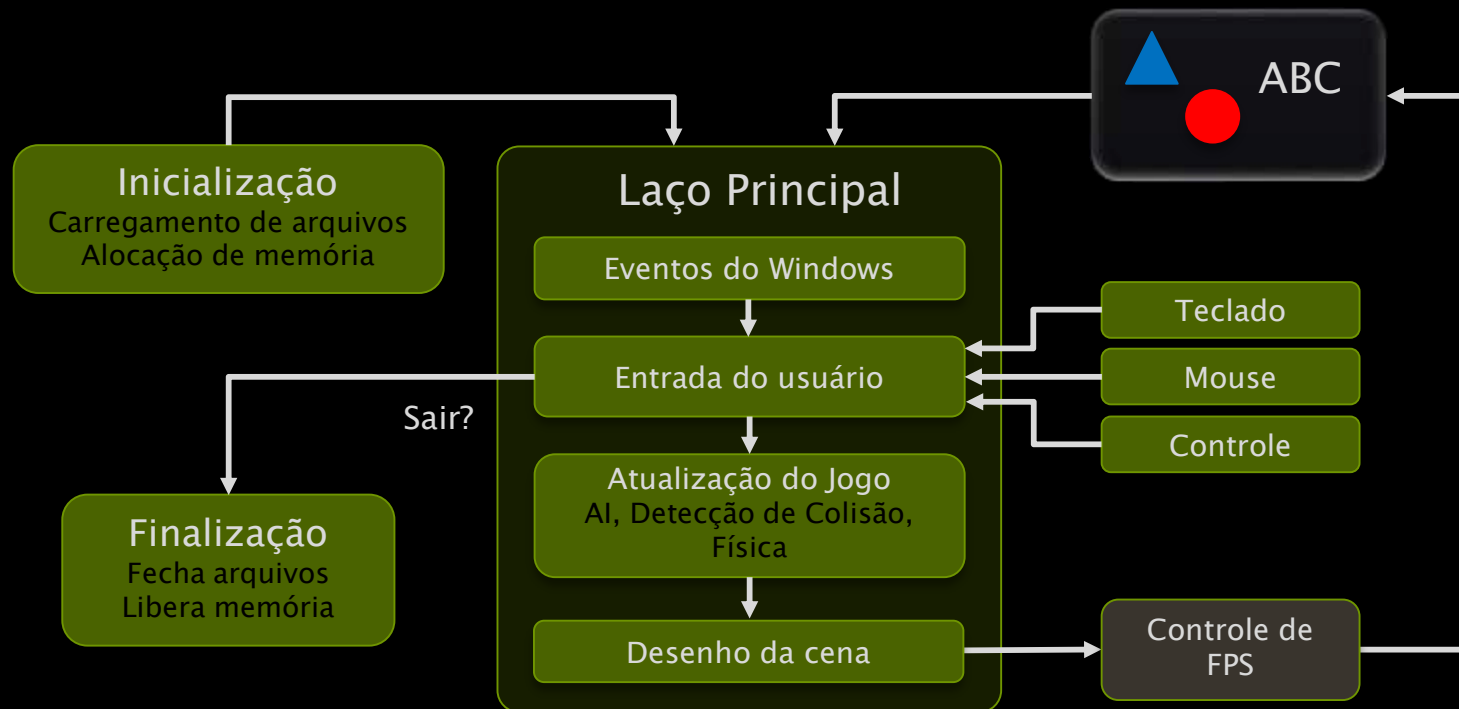
        case WM_KEYDOWN: // tecla do teclado pressionada
            windowKeys[wParam] = true;
            return 0;

        case WM_LBUTTONDOWN: // botão esquerdo do mouse pressionado
        case WM_LBUTTONDBLCLK:
            windowKeys[VK_LBUTTON] = true;
            return 0;

        ...
    }
}
```

# Introdução

- Um jogo é um **laço de tempo real**



O computador é mais rápido que nossas reações

Vários ciclos se passam em cada pressionamento de tecla ou botão

# Controle de Teclas/Botões

- ▶ Para evitar que um pressionamento seja **registrado várias vezes** é preciso fazer algum tipo de controle

```
static bool windowKeys[256];    // estado do teclado e botões do mouse  
static bool windowCtrl[256];    // controle de liberação de teclas
```

- Pode ser feito **guardando um booleano** para cada tecla/botão
  - Indica se a tecla ou botão **foi liberado**
  - O Windows captura 256 Virtual Key Codes (VK\_)
    - Cobrem as teclas do teclado
    - Os botões do mouse

# Controle de Teclas/Botões

- ▶ O método **KeyPress** consulta o estado de controle

```
bool Window::KeyPress(int vkcode)
{
    if (windowCtrl[vkcode])
    {
        if (KeyDown(vkcode))
        {
            windowCtrl[vkcode] = false;
            return true;
        }
    }
    else if (KeyUp(vkcode))
    {
        windowCtrl[vkcode] = true;
    }

    return false;
}
```

# Usando o Mouse

- ▶ Para **trabalhar com o mouse**, dispomos de:

```
bool KeyDown(int vkcode);    // verifica se uma tecla/botão está pressionado
bool KeyUp(int vkcode);      // verifica se uma tecla/botão está liberado
bool KeyPress(int vkcode);   // registra um pressionamento por vez

float MouseX();              // retorna posição x do mouse
float MouseY();              // retorna posição y do mouse
```

- Com os seguintes VK Codes:

```
VK_LBUTTON - Botão esquerdo
VK_RBUTTON - Botão direito
VK_MBUTTON - Botão do meio
```

# Usando o Mouse

- ▶ Existem duas formas de **usar o mouse em um jogo**
  - O mouse **cria objetos**:
    1. Capturar a posição do mouse
    2. Ler o estado dos botões
    3. Usar essa informação para criar objetos que interagem com a cena
  - O mouse **é um objeto**:
    1. Criar um objeto “Mouse”, que se mantém na posição do cursor
    2. Inserir esse objeto na cena
    3. Tratar a colisão dele com outros objetos



# Resumo

- ▶ A **Win32** fornece tratamento para o mouse
  - Mensagens são geradas sempre que há:
    - Movimento do mouse
    - Pressionamento de botões
- ▶ Um jogo é um **laço de tempo real**
  - Um método de controle precisa ser criado para evitar que o pressionamento de um botão seja registrado várias vezes
- ▶ O mouse pode **criar** objetos **ou ser um objeto** do jogo