

Aula 4: Captura de eventos - teclado e mouse

Apresentação

Quando interagimos com um computador, seja este PC, videogame, smartphone ou SmartTV (entre outros que nos cercam), estamos frequentemente solicitando que ele capture eventos.

A captura de eventos permite que sua solicitação para o equipamento, seja para a atuação do joystick, uma troca de canal ou o acesso a um aplicativo, possa ser atendida de forma quase imediata.

Nesta aula, veremos como essa eficiência para atender à solicitação feita é possível e como programá-la em C para captura de eventos de teclado e mouse. Veremos que essa captura, em sistemas operacionais mais antigos, fazem o acesso a registros, demonstrando uma característica expressiva da programação de software básico.

Objetivos

- Identificar bibliotecas para captura de eventos de teclado e mouse;
- Desenvolver programas em C com captura de eventos;
- Reconhecer os conceitos de animação de objetos gráficos.

Captura de eventos e interrupção em programação

Capturar eventos é uma das formas mais comuns de estabelecer uma interface entre os usuários e um programa desenvolvido por você. Em programação, um evento é algo que ocorre como resultado de uma ação do usuário ou de outra fonte, como clicar em um botão do mouse ou pressionar uma tecla no teclado.

Um manipulador de eventos é uma rotina usada para lidar com o evento, permitindo que um programador escreva o código que será executado quando o evento ocorrer.

Além de eventos de teclado e mouse, outros exemplos de eventos comuns são:

- Um navegador da web carregando completamente uma página da web.
- Um arquivo sendo criado ou modificado em um sistema de arquivos.
- Um sensor de hardware, como uma webcam ou um microfone, recebendo entrada sensorial.
- A chegada do tráfego de rede recebido.
- A ocorrência de um erro no nível do programa ou sistema.

A figura 1 ilustra a manipulação de um evento em um programa.

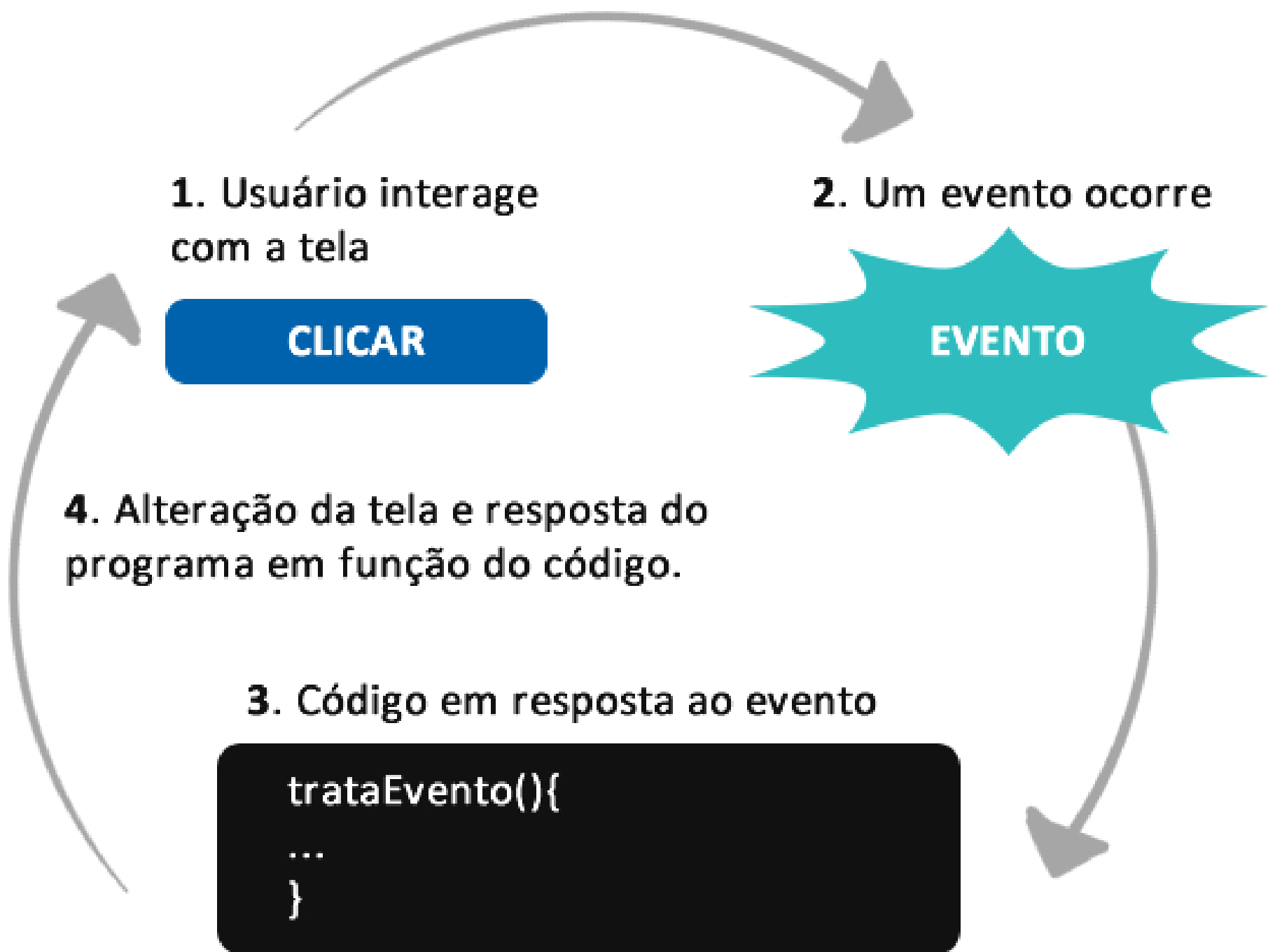


 Figura 1: Manipulação de evento em um programa.

Em alguns sistemas, como no antigo MS-DOS, um evento de teclado ou mouse pode provocar uma interrupção. Uma interrupção é um sinal enviado ao processador que interrompe o processo atual. Além de dispositivos de hardware, um software pode gerar uma interrupção.

Exemplo

Se você estiver usando um processador de texto e pressionar uma tecla, o programa deverá processar a entrada imediatamente. Digitar “olá” cria cinco solicitações de interrupção, o que permite ao programa exibir as letras digitadas. Da mesma forma, sempre que você clica em um botão do mouse ou toca em uma tela sensível ao toque, há o envio de um sinal de interrupção para o dispositivo.

Uma interrupção é enviada ao processador como uma solicitação de interrupção ou InterruptRequestLine (IRQ). Cada dispositivo de entrada tem uma configuração ou prioridade exclusiva de IRQ. Isso evita conflitos e garante que dispositivos de entrada comuns, como teclados e mouses, sejam priorizados.

As interrupções de software são usadas para manipular erros e exceções que ocorrem enquanto um programa está em execução.

Exemplo

Se um programa espera que uma variável seja um número válido, mas o valor é nulo, pode ser gerada uma interrupção para impedir que o programa trave. Isso permite que o programa mude de rumo e lide com o erro antes de continuar.

Da mesma forma, uma interrupção pode ser usada para interromper um loop infinito, o que pode criar um vazamento de memória ou fazer com que um programa não responda.

As interrupções de hardware e software são processadas por um manipulador de interrupções, também chamado de Interrupt Service Routine(ISR) – ou Rotina de Serviço de Interrupção.

Saiba mais

Quando um programa recebe uma solicitação de interrupção, o ISR lida com o evento, e o programa é retomado. Como as interrupções costumam ser tão breves quanto o pressionamento de tecla ou o clique do mouse, elas geralmente são processadas em poucos milissegundos.

Evento e interrupção

Os conceitos de evento e interrupção oferecem maneiras para o sistema/programa lidar com várias condições que ocorrem durante o desenrolar normal de algum programa e que podem exigir que o sistema/programa faça outra coisa antes de retornar para a tarefa original.

Comentário

No entanto, apesar dessa semelhança funcional, eles são conceitos muito distintos, usados em contextos distintos, em níveis distintos.

Diferença entre evento e interrupção

Fornecem um dispositivo de baixo nível para quebrar o desenrolamento normal de qualquer parte do programa que a CPU esteja trabalhando em um determinado momento e para que a CPU comece a processar as instruções em outro endereço.

São úteis para lidar com várias situações que exigem o processamento imediato da CPU.

São normalmente despachadas por meio de tabelas vetoriais, nas quais a CPU tem um local específico na memória contendo uma matriz de endereços, onde residem determinados manipuladores de interrupção. Modificando o conteúdo da tabela de interrupções, se for permitido, um programa pode redefinir qual manipulador específico será chamado para um determinado número de interrupção.

São mensagens no nível do sistema/linguagem que podem ser usadas para significar várias situações de hardware ou software, como cliques no mouse e entradas do teclado, mas também no nível do aplicativo, em situações como “Novo registro inserido no banco de dados”.

Diferentemente das interrupções, com seu comportamento relativamente simples, que é totalmente definido pela CPU, existem vários sistemas de eventos, no nível do sistema operacional, bem como várias estruturas, como estruturas do MS Windows, JavaScript, .NET etc.

Todos os sistemas de eventos, embora diferentes em suas implementações, geralmente compartilham propriedades comuns, como o conceito de um manipulador, que é uma função específica do programa designada para lidar com tipos específicos de eventos de fontes diversas.

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Captura de eventos em programas 16 bits

Como principais periféricos de entrada de um computador, teclado e mouse têm diversas bibliotecas em C para acesso aos eventos de cliques em suas teclas e botões. Em programas do tipo console, a conhecida função scanf() da biblioteca-padrão é um exemplo de acesso a dados provenientes de cliques em um teclado.



No caso do mouse, essa interação não é tão usual nas aplicações tipo console. Aplicações para o console são aquelas executadas em uma janela de texto (também conhecida como Prompt do MS-DOS). Mesmo sendo obsoleto, o acesso ao mouse também pode ser visto nessas condições para um bom entendimento dos programas de software básico.

Acesso a mouse em programas tipo console

1

O acesso ao mouse em programas tipo console não é direto, mas por meio do driver. Usam-se interrupções para obter acesso a esse driver. Cada dispositivo fornecido por um computador tem uma porta exclusiva, que é um valor hexadecimal projetado para ser independente da capacidade de portabilidade do programa. Essas portas são acessadas pelos programas.

2

O mouse tem a porta 0x33 anexada a ele. Também fazemos uso de registros de endereços. Estes são basicamente Union do tipo REGS definido no cabeçalho“dos.h”. Usam-se dois registradores para se comunicar a um driver de dispositivo: Um para entrada e outro para saída.

3 Enviamos um valor ao driver de dispositivo, por meio do registro de entrada, e recebemos informações nele embutidas no registro de saída. Podemos acessar várias funções do mouse usando diferentes valores de entrada do Registro AX e passando esses valores para a porta do mouse utilizando uma interrupção.

A figura 2 mostra as estruturas da Union REGS in, out (entrada, saída) para acesso aos registros do mouse em aplicações console.

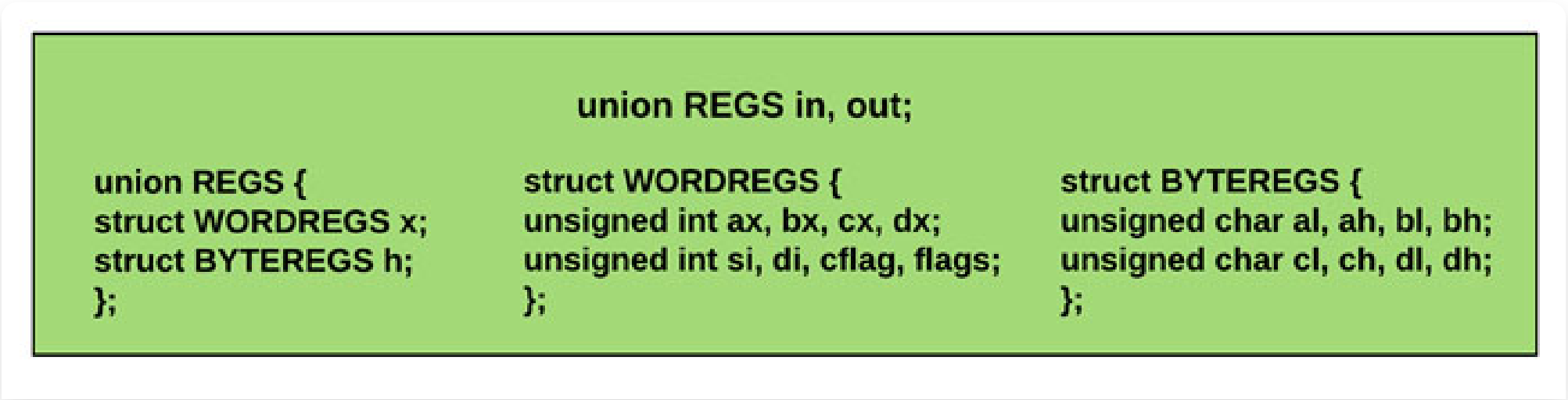


Figura 2: Estruturas da Union REGS para acesso aos registros do mouse em aplicações console. (Fonte: autor).

As funções estão listadas abaixo. Aqui, AX, BX, CX e DX são membros do Union REGS.

- **Entrada:** Funções executadas com parâmetro de retorno, se houver
- **AX = 0:** Obtém o status do mouse (valor AX = suporte FFFFh está disponível/valor AX = 0, o suporte não está disponível)
- **AX = 1:** Mostrar o ponteiro do mouse
- **AX = 2:** Ocultar o ponteiro do mouse
- **AX = 3:** Posição do mouse (CX = coordenada x do mouse/DX = coordenada y do mouse)
- **AX = 3:** Botão do mouse pressionado (BX = 0 – nenhuma tecla é pressionada –, BX = 1 – botão esquerdo é pressionado –, BX = 2 – botão direito é pressionado – e BX = 3 – botão central é pressionado).
- **AX = 7**
CX = MaxX1
DX = MaxX2: Definir limite horizontal
- **AX = 8**
CX = MaxX1
DX = MaxX2: Definir limite vertical

Comentário

Somente programas para Windows 16-bits e DOS permitem o acesso a esses registros. Os compiladores C atuais, de 32 e 64 bits, não terão acesso dessa forma aos registros e dados do mouse. Logo, os testes dos códigos a seguir só poderão ser feitos no Turbo C++.

Detectando o mouse

Observe o programa seguinte:

```
#include <dos.h>
union REGS in, out;

void detecta_mouse ()
{
    in.x.ax = 0;
    int86 (0x33,&in,&out);//Invoca a interrupção
    if (out.x.ax == 0)
        printf ("\nMouse falhou a inicializacao");
    else
        printf ("\nMouse foi inicializado com sucesso.");
}
int main ()
{
    detecta_mouse ();
    getch ();
    return 0;
}
```

O programa acima declara duas variáveis do tipo Union REGS, que é declarada em dos.h. A Union REGS contém duas estruturas (struct WORDREGS x struct BYTEREGS h), como visto na figura 2.

Essas duas estruturas contêm algumas variáveis de 1 byte e 2 bytes de comprimento que, indiretamente, representam registros da CPU. Ao colocar 0 no registrador ax e invocar a interrupção do mouse (33h), podemos verificar se o driver do mouse está carregado ou não.

Usa-se a função `int86()` para invocar o método `interrupt.int86()`, que recebe três argumentos:

Número da interrupção (0x33) e duas variáveis do tipo Union REGS.

Mostrando o mouse

Observe o programa seguinte:



O cursor do mouse pode ser usado tanto no modo de texto quanto no modo gráfico. No modo de texto, ele parece com um quadrado e, no modo gráfico, com o ponteiro do Windows.

No programa abaixo, usamos a função de biblioteca-padrão `initgraph ()` para inicializar o sistema gráfico. Essa função recebe três argumentos: driver de gráficos, modo de gráfico e caminho para o arquivo do driver. Usando `DETECT`, que é uma macro definida em `graphics.h`, dizemos à função para selecionar um driver adequado. Quando `DETECT` é usado, não é necessário atribuir nada ao modo gráfico.

```
#include <dos.h>
#include <graphics.h>

union REGS in, out;

void showmouse_graphics ()
{
    int gdriver = DETECT, gmode, errorcode;
    initgraph(&gdriver, &gmode, "c:\\turboc3\\bgi");
    in.x.ax = 1;
    int86 (0X33,&in,&out);
    getch ();
    closegraph ();
}

void detecta_mouse ()
{
    in.x.ax = 0;
    int86 (0X33,&in,&out); //Invoca a interrupção
    if (out.x.ax == 0)
        printf ("\nMouse falhou a inicializacao");
    else
        printf ("\nMouse foi inicializado com sucesso.");
}

int main ()
{
    detecta_mouse ();
    showmouse_graphics ();
    getch ();
    return 0;
}
```

Saiba mais

Leia o texto [Verificando que botão do mouse foi pressionado <galeria/aula4/anexo/PDF1_Programacao_Aula04_convertido.pdf>](#).

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Biblioteca em programas 32 bits

A captura de evento associado ao pressionar o teclado ou ao clicar em um botão do mouse é a forma preferencial atualmente de receber essas informações.

Atenção

Para compiladores como o Dev C++, a biblioteca descrita em windows.h conta com funções diversas para capturar esses eventos, como a função `GetKeyState()`. O parâmetro passado é um [Virtual-Key Code <https://docs.microsoft.com/pt-br/windows/desktop/inputdev/virtual-key-codes>](https://docs.microsoft.com/pt-br/windows/desktop/inputdev/virtual-key-codes) que estipula várias teclas especiais, caracteres e botões do mouse para verificação.

O código abaixo verifica se o botão esquerdo ou direito do mouse foi pressionado e se a tecla A foi pressionada.

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>

int main(){

while(1){
//Verifica se o botão esquerdo do mouse está pressionado
if ((GetKeyState(VK_LBUTTON) & 0x80) != 0)//
{
    Sleep(200);
    printf( "Botao da esquerda apertado\n");
}

//Verifica se o botão direito do mouse está pressionado
if ((GetKeyState(VK_RBUTTON) & 0x80) != 0)
{
    Sleep(200);
    printf( "Botao da direita apertado\n");
}

//Verifica se a tecla A está pressionada
if ((GetKeyState(65) & 0x80) != 0) // 64 em decimal = 0x41 em hexadecimal
{
    Sleep(200);
    printf( "Tecla A pressionada\n");
}
}
return 0;}
```


OpenGL para gerenciar eventos de mouse e teclado

A biblioteca GLUT, utilizada com OpenGL, nos permite criar aplicativos que detectam a entrada do teclado usando as teclas “normais” ou as teclas especiais, bem como toques no mouse.

Sempre que quisermos controlar o processamento de um evento, devemos informar antecipadamente ao GLUT qual função executará essa tarefa. Até agora, usamos o GLUT para informar ao sistema do Windows quais funções queríamos executar quando a janela precisava ser redesenhada, por exemplo.

As principais funções para realizar essas tarefas estão descritas a seguir.

[glutKeyboardFunc](#)



É usada para informar ao sistema Windows em qual função queremos processar os eventos de toque em teclas “normais”. Por teclas normais, queremos dizer letras, números, qualquer coisa que tenha um código ASCII. A função usada como argumento para glutKeyboardFunc precisa ter três argumentos.

O primeiro fornece o código ASCII da tecla pressionada. Os dois argumentos restantes fornecem a posição do mouse quando a tecla é pressionada. A posição do mouse é relativa ao canto superior esquerdo da área do cliente da janela.

[glutSpecialFunc](#)



Embora as de teclas ASCII sejam muitas, há algumas teclas que você pode usar – por exemplo, as teclas de seta – que podem ser representadas de maneira diferente em vários sistemas. Felizmente, o GLUT nos permite usar essas teclas. Para isso, devemos usar a função glutSpecialFunc, que verifica o pressionamento de teclas especiais.

Por exemplo, se quisermos verificar se a seta esquerda foi pressionada, podemos usar GLUT_KEY_LEFT. De forma semelhante, usamos GLUT_KEY_F1 para a tecla F1, e assim também para as demais. A posição do mouse também pode ser identificada.

[glutMouseFunc](#)



Estabelece eventos de toques no mouse. Os valores possíveis do primeiro parâmetro de entrada, que identifica o botão clicado, são: GLUT_LEFT_BUTTON, GLUT_MIDDLE_BUTTON e GLUT_RIGHT_BUTTON. O segundo parâmetro, do estado do botão, pode ser GLUT_UP ou GLUT_DOWN. Os dois parâmetros restantes indicam a localização do mouse quando o evento de clique ocorreu.

[glutMotionFunc](#)



Identifica movimento no mouse, chamando a função para tratar esse evento, quando esse movimento ocorre na janela da aplicação enquanto um ou mais botões do mouse estão pressionados.

[glutPassiveMotionFunc](#)



Também identifica o movimento no mouse, só que enquanto nenhum de seus botões está pressionado.

Exemplo

Veja um [exemplo <galeria/aula4/anexo/PDF2_Programacao_Aula04_convertido.pdf>](#) de interações de teclados que alteram a imagem de um retângulo desenhado.

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Atividade

1. Vimos que o valor 3 para o registro ax invoca a interrupção do mouse, e que essa subfunção retorna informações de posição do mouse nos registros cx (coordenada x) e dx (coordenada y). Escreva uma função que imprima a coordenada do mouse quando se aperta seu botão esquerdo. Teste no Turbo C++.
2. A função `line(x1,y1,x2,y2)` é usada para desenhar uma linha de um ponto (x1, y1) para o ponto (x2, y2), isto é, (x1, y1) e (x2, y2) são os pontos inicial e final da linha. Desenvolva um programa em C, baseando-se nos exemplos dados de acesso ao registro ax e utilizando a biblioteca `graphics.h`, que desenhe uma linha entre dois pontos escolhidos pelo usuário quando este mantém o botão esquerdo do mouse pressionado. Ou seja, um programa básico que se assemelha ao desenhar de uma linha no Paint ou outro programa de desenho.
3. No programa que usa OpenGL para capturar evento de teclado e mouse, altere para acrescentar a função que faz com que o retângulo desenhado siga o mouse quando este estiver pressionado (botão esquerdo).
4. Como podemos diferenciar **interrupções** de **eventos** em programas?
5. Como a função `GetKeyState` opera para se saber se uma tecla foi pressionada ou não?

Notas

Título modal ¹

Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos.

Título modal ¹

Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos.

Referências

MANSSOUR, Isabel Harb. **Introdução à OpenGL**. Disponível em: <https://www.inf.pucrs.br/~manssour/OpenGL/Utilizacao.html>
<<https://www.inf.pucrs.br/~manssour/OpenGL/Utilizacao.html>>. Acesso em: 26 dez. 2019.

SELLERS, Graham; WRIGHT, Richard S.; HAEMEL, Nicholas. **OpenGL Superbible**: Comprehensive Tutorial and Reference. 7th ed. Addison-Wesley Professional, 2015.

VRIES, Joey de. **Learn OpenGL**. Disponível em: <https://learnopengl.com/book/offline%20learnopengl.pdf>
<<https://learnopengl.com/book/offline%20learnopengl.pdf>>. Acesso em: 26 dez. 2019.

Próxima aula

- Bibliotecas em C para funções de entrada e saída, por console e arquivo;
- Acesso à porta serial;
- Controle de dispositivos com microcontroladores.

Explore mais

- Realizar o controle de elementos gráficos é o básico para construir jogos. A OpenGL ainda é muito utilizada para essa função. O material [Introdução à computação gráfica com OpenGL](https://www.ft.unicamp.br/~magic/opengl/animacao-2D.html#ANIMACAO-DESCRICAO) <<https://www.ft.unicamp.br/~magic/opengl/animacao-2D.html#ANIMACAO-DESCRICAO>> mostra mais sobre a construção de um jogo simples.
- Existem outras bibliotecas para realizar elementos gráficos e jogos. Uma delas é a Allegro. Os materiais a seguir apresentammais conhecimentosobre essa biblioteca.
 - [Aprendendo Allegro 5](https://aprendendoallegro.tk/index.php) <<https://aprendendoallegro.tk/index.php>>;
 - [C - Allegro Aula 1 \(Instalação\)](https://www.youtube.com/watch?v=NHq58TIWLe0) <<https://www.youtube.com/watch?v=NHq58TIWLe0>>.