

Computação Gráfica 65081

Março 2004

Aula número 02 – Parte II

Christian Hofsetz

chofsetz@acm.org

Parte II

GLUT

Baseado no livro *OpenGL Programming Guide*, por Mason Woo, Jackie Neider, Tom Davis, Dave Shreiner e nas anotações do Prof. Marcelo Walter

GLUT

- Graphics Library Utility Toolkit
- Biblioteca de software que se integra com OpenGL (autor: Mark J. Kilgard - SGI)
- Habilita acesso aos eventos do sistema operacional necessários em programas interativos
 - mouse, teclado, display, etc.

OpenGL + Glut:

Redesenhar a cada passo

- A imagem precisa ser redeseenhada constantemente mesmo se não houver nenhuma mudança
- Portanto, deve-se evitar processamentos do tipo alocação de memória para estruturas e inicialização dentro da rotina de visualização
- Ou seja, no trabalho do grau A não se deve ler o arquivo, criar a lista de vértices e faces a cada renderização!

Filosofia de Projeto

- Até onde possível, independente do sistema de janelas nativo
- Programas de tamanho pequeno e médio
- Não apresenta muitos recursos de interface (apenas menus)
- Programação orientada a eventos

Programação Interativa

- Programas gráficos são na maioria dos casos interativos
 - Usuário controla a execução do programa
- Controle é executado via dispositivos de entrada (mouse, teclado, etc.)
- 2 maneiras básicas para implementação
 - Amostragem
 - Eventos

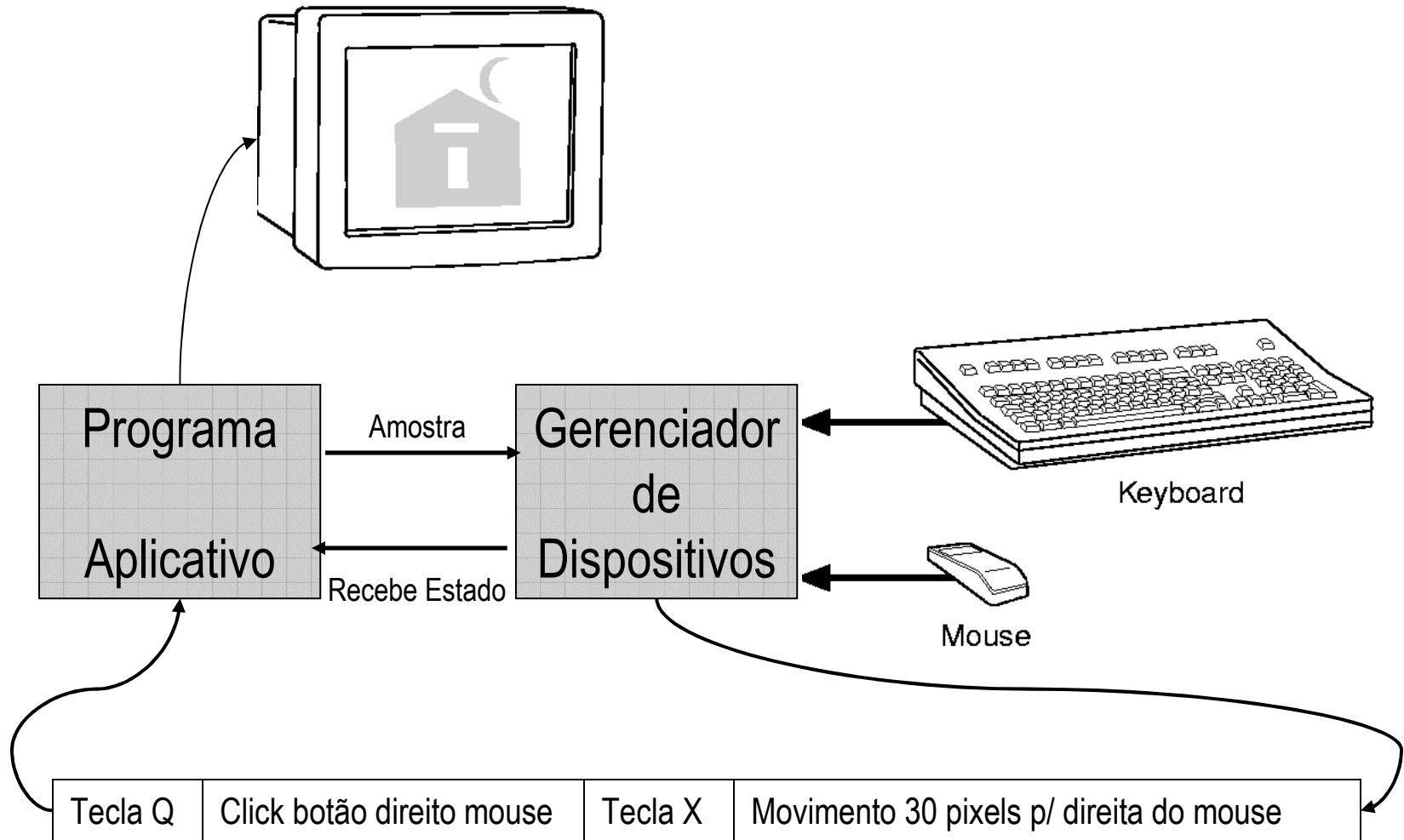
Amostragem

- O programa aplicativo *amostra* o estado atual dos dispositivos sem ser interrompido
- A amostragem é feita *independentemente* do estado ter mudado ou não
- Frequência da Amostragem

Eventos

- Um gerenciador monitora em *background* a ocorrência de *eventos* e os coloca numa fila de eventos
- *Evento*: mudança no estado do dispositivo causada pela ação do usuário
- Programa principal, à sua conveniência, verifica a fila de eventos

- Primeiro evento da fila é removido
- Controle da execução é transferido para execução de uma rotina associada ao evento
- Se a fila estiver vazia, programa principal segue execução (rotina *idle*)
- Exemplos de eventos: click do mouse, tecla pressionada, janela redimensionada, etc.



FILA DE EVENTOS

Rotinas

- Iniciam com prefixo `glut`
- `(0,0)` canto superior esquerdo da janela
- Incluir diretiva `#include <GL/glut.h>`
- Inicialização
 - `glutInit(int *argc, char **argv)`
 - `glutInitWindowPosition`
 - `glutInitWindowSize`

- **Processamento de Eventos**

- `void glutMainLoop(void)`

- **Gerenciamento de janelas**

- `int glutCreateWindow(char *name)`

- `void glutPostRedisplay(void)`

- **Registro de rotinas de callback**

- `glutDisplayFunc`

- `glutReshapeFunc`

- `glutKeyboardFunc`

- `glutMouseFunc`

- **Objetos Pré-definidos**

- `glutSolidSphere, glutWireCube, glutSolidTeapot`

Estrutura de um programa

- main
 - inicializa GLUT/OpenGL e cria janela
 - *while* houver eventos
 - *if* evento = exibir janela
 - modifica janela de exibição (se necessário)
 - redesenha
 - *else /** evento de mouse, teclado, etc **/*
 - executa rotina associada ao evento

Redesenha

- Limpa janela

```
glClear (GL_COLOR_BUFFER_BIT)
```

- Modifica estados (se necessário)

- desenha

- ...

- Troca buffers

```
glutSwapBuffers ()
```

Exemplo: *OpenGL* + Glut

- Programa principal:

```
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize (250, 250);
    glutInitWindowPosition (100, 100);
    glutCreateWindow ("Ola mundo");
    init (); /* Ver próximo slide */
    glutDisplayFunc(display); /* Ver proxs. Slides */
    glutMainLoop();
    return 0;
}
```

Exemplo: *OpenGL* + Glut

```
void init (void)
{
    /* Cor de fundo */
    glClearColor (0.0, 0.0, 0.0, 0.0);

    /* Inicializa valores da câmera */
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);
}
```

```
void display(void)
{
/* Apagar todos os pixels */
   glClear (GL_COLOR_BUFFER_BIT);

/* Desenhar um quadrilatero de cor branca */
   glColor3f (1.0, 1.0, 1.0);
   glBegin(GL_QUAD);
       glVertex3f (0.25, 0.25, 0.0);
       glVertex3f (0.75, 0.25, 0.0);
       glVertex3f (0.75, 0.75, 0.0);
       glVertex3f (0.25, 0.75, 0.0);
   glEnd();

/* Iniciar o processamento das rotinas OpenGL
   imediatamente */
   glFlush ();
}
```