LEI / LCC
Departamento de Informática
Universidade do Minho

Interactividade Básica com GLUT e Primitivas Geométricas

GLUT – Rato, Teclado e Popup Menus
OpenGL – Desenho e Transformações Geométricas

Interactividade Básica com GLUT

- O GLUT suporta de forma simples um conjunto alargado de dispositivos de entrada:
  - Rato
  - Teclado
  - Trackball
  - Tablet
- A utilização de dispositivos de entrada em GLUT implica a definição de funções para processamento dos eventos gerados.
- Implica ainda o registo dessas funções no GLUT.

Registo de Callbacks - Teclado

- Teclas Normais (letras, números, etc...)

Registo da função:

```
glutKeyboardFunc(nome_função);
```

Assinatura da função registada:

```
void nome_função(unsigned char tecla, int x, int y);
```

Esta função será invocada sempre que for premida uma tecla "normal".

A função é invocada com a indicação da tecla, e a posição do rato em coordenadas relativas da janela.

Registo de Callbacks - Teclado

- Teclas Especiais (F1..F12, Home, End, setas, etc...)

Registo da função:

```
glutSpecialFunc(nome_função);
```

Assinatura da função registada:

```
void nome_função(int tecla, int x, int y);
```

Os códigos das teclas são constantes definidas em GLUT com o prefixo GLUT_KEY, por exemplo: GLUT_KEY_F1.

Registo de Callbacks - Rato

- Rato: Botão premido ou solto

Registo da função:

```
glutMouseFunc(nome_função);
```

Assinatura da função registada:

```
void nome_função(int botão, int estado, int x, int y);
```

A função registada devolve informação sobre:

- qual o botão (`GLUT_LEFT_BUTTON`, `GLUT_MIDDLE_BUTTON`, `GLUT_RIGHT_BUTTON`);
- qual o estado do botão (`GLUT_UP`, `GLUT_DOWN`).
- posição do rato (x,y) em coordenadas relativas da janela;

Registo de Callbacks - Rato

- Rato: Movimento passivo (sem botão premido) ou activo

Registo da função:

```
glutMotionFunc(nome_função);
glutPassiveMotionFunc(nome_função);
```

Assinatura da função registada:

```
void nome_função(int x, int y);
```

A função registada devolve informação sobre a posição do rato (em coordenadas relativas da janela);

Interactividade com GLUT

- Através do GLUT é ainda possível definir pop-up menus de uma forma simples.

- Os items dos pop-up menus podem por sua vez também ser menus.

Interactividade com GLUT - Menus

- Primeiro passo: Criar um menu.
  - No processo de criação de um menu é necessário especificar qual a função que irá processar as opções do menu.
    - `int glutCreateMenu(nome_função);`
  - O valor devolvido por `glutCreateMenu` é o identificador do menu.
  - A função registada receberá como parâmetro o identificador da opção selecionada no menu.
  - Assinatura da função registada:
    - `void nome_função(int id_op);`

Interactividade com GLUT - Menus

- Segundo passo: Adicionar opções ao menu.

```
void glutAddMenuEntry(char *op, int id_op);
```

  - Ex: `glutAddMenuEntry("Vermelho",1);`
  - As opções são adicionadas ao fim do menu. Não é possível inserir a meio ou no princípio.

- Terceiro passo: Associar o menu a um botão do rato

```
glutAttachMenu(int botão);
```

  - botão = `GLUT_LEFT_BUTTON`, `GLUT_RIGHT_BUTTON`, ou `GLUT MIDDLE_BUTTON`

Interactividade com GLUT - Menus

- Notas:
  - Ao adicionar elementos a um menu não se especifica qual o menu. Quando se cria um menu, este passa a ser o menu actual, e é neste menu que as opções são inseridas.
  - Pode-se no entanto, a qualquer altura, actualizar um menu criado anteriormente, tornando-o o menu actual
    - `glutSetMenu(int menuId);`

Gestão de Recursos

- Ao utilizar a idleFunc, o GLUT está a redesenhar a cena continuamente.
- Em cenas estáticas, só há necessidade de redesenhar quando a câmara ou o objecto se move.
- Para evitar a utilização desnecessária de recursos do sistema, sempre que a câmara, ou algo na cena se mova, utiliza-se

```
glutPostRedisplay()
```

- Esta função irá pedir ao GLUT que refresque o conteúdo da janela logo que possível.

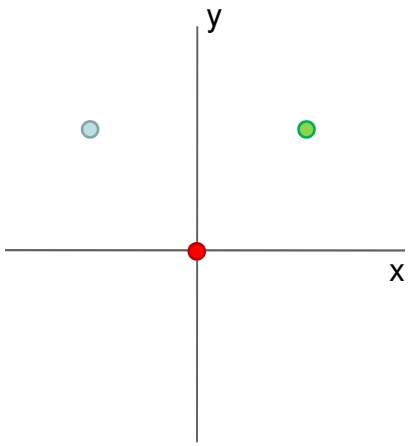
Modelação 3D

- Definição de um ponto em 3D

```
glVertex3f(x,y,z);
```

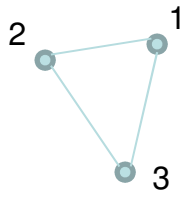
- Para desenhar triângulos:

```
glBegin(GL_TRIANGLES);
     glVertex3f(0.0f,  0.0f,  0.0f);
     glVertex3f(1.0f,   1.0f,  0.0f);
     glVertex3f(-1.0f,  1.0f,  0.0f);
glEnd();
```

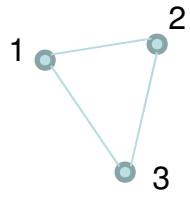


Modelação 3D

- Orientação dos polígonos
  - Por omissão, os pontos devem ser definidos pela ordem inversa à do movimento dos ponteiros do relógio



Polígono virado para a frente



Polígono virado para trás

Modelação 3D

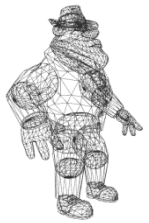
- *Back Face Culling*
  - `glEnable(GL_CULL_FACE);`
  - `glCullFace(GL_FRONT ou GL_BACK);`
- Podemos definir a orientação dos polígonos
  - `glFrontFace(GL_CW ou GL_CCW);`

Modelação 3D

- Modo de Preenchimento dos polígonos

```
glPolygonMode(face,modo);
```

- face:
  - `GL_FRONT, GL_BACK, GL_FRONT_AND_BACK`
- modo:
  - `GL_FILL, GL_LINE, GL_POINT`



Algumas Funções Necessárias

- Funções OpenGL e GLU

```
glTranslatef(x,y,z); // move o objecto

glRotatef(ângulo,x,y,z); // ângulo em graus

glColor3f(r,g,b); // define a cor a utilizar

gluLookAt(px,py,pz,  lx,ly,lz,  ux,uy,uz);
```

`px,py,pz` - posição da câmara
`lx,ly,lz` - ponto para onde a câmara está a olhar
`ux,uy,uz` - inclinação da câmara, por omissão utilizar (`0.0, 1.0, 0.0`)

Exercício

- Completar o esqueleto fornecido de modo a criar uma aplicação interactiva que desenhe uma pirâmide (uma face de cada cor).
- O teclado deve permitir mover a pirâmide no plano XZ, rodá-la em torno do eixo dos YY, e ainda alterar a altura da pirâmide.
- Utilizar glutPostRedisplay;
- Criar um menu para selecionar o modo de preenchimento dos polígonos.