

Introdução à OpenGL

SCC0250 - Computação Gráfica

Prof. Rosane Minghim

[https://sites.google.com/site/computacaograficaicmc2017t2/
rminghim@icmc.usp.br](https://sites.google.com/site/computacaograficaicmc2017t2/rminghim@icmc.usp.br)

P.A.E. Nicolas Roque nrsantos@usp.br

Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP)

15 de março de 2017



Introdução

Histórico

- Open Graphics Library (OpenGL) é uma especificação de uma Application Programming Interface (API) para a criação de aplicações gráficas.
- Foi introduzido em 1992 pela Silicon Graphics e atualmente é mantida pelo Khronos Group.
- Está na versão 4.5.

Introdução

Histórico

- Open Graphics Library (OpenGL) é uma especificação de uma Application Programming Interface (API) para a criação de aplicações gráficas.
- Foi introduzido em 1992 pela Silicon Graphics e atualmente é mantida pelo Khronos Group.
- Está na versão 4.5.

O que é

- Coleção de rotinas que o programador pode chamar para gerar elementos gráficos.
- Oferece suporte para gerar e exibir cenas 2D e 3D.
- Contém mais de 100 comandos distintos que podem ser utilizados para produzir aplicações gráficas.

Introdução

O que é

- Não possui funções para processar entrada de usuários ou manipulações de janela.
 - Podem ser utilizadas as funções específicas de cada plataforma.
 - Ou usar uma biblioteca independente (GLUT ou SDL, por exemplo).
- Portável.
 - Mesmo código pode ser compilado em diferentes sistemas operacionais.
 - Aplicação não precisa ser alterada.

Variáveis de Estado

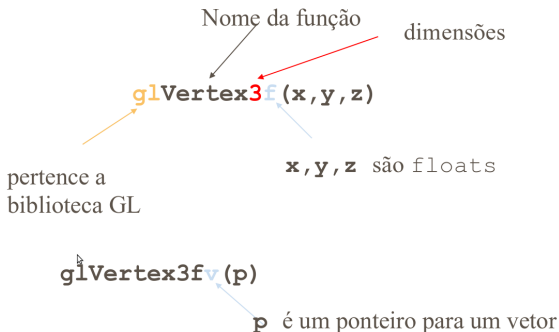
Variáveis de Estado

- A OpenGL rastreia inúmeras variáveis de estado.
 - Tamanho de um ponto, cor de fundo da janela, cor do desenho, entre outras.
- O valor corrente permanece ativo até ser alterado.
 - Cor de Fundo: `glClearColor(r, g, b, alpha)`
 - Tamanho do Ponto: `glPointSize(3.0);`

Nomenclatura de Funções

Nomenclatura de Funções

- Visa padronizar e facilitar a utilização.
- Possibilita identificar a qual biblioteca a função pertence, quantidade de argumentos e quais os tipos dos argumentos.



Tipos de Dados

Tipos de Dados

- A OpenGL possui tipos de dados próprios.
- Tornam o programa-fonte portátil.

Tabela: Tipo de dados

Sufixo	Tipo	Tipo C	Nome
b	inteiro 8 bits	signed char	GLbyte
s	inteiro 16 bits	short	GLshort
i	inteiro 32 bits	int/long	GLint
f	float 32 bits	float	GLfloat
...

Tipos de Dados

- Tomar cuidado

```
1 void drawDot(int x, int y)
2 {
3     glBegin(GL_POINTS);
4     glVertex2i(x, y); // Função espera inteiro de 32 bits ←
                        // mas recebe um int.
5     glEnd();
6 }
```

- Correto

```
1 void drawDot(GLint x, GLint y)
2 {
3     glBegin(GL_POINTS);
4     glVertex2i(x, y);
5     glEnd();
6 }
```


Primitivas Geométricas 2D

- Linha

```
1   glBegin(GL_LINES);  
2       glVertex2f(x1, y1);  
3       glVertex2f(x2, y2);  
4   glEnd();
```

- Triângulo

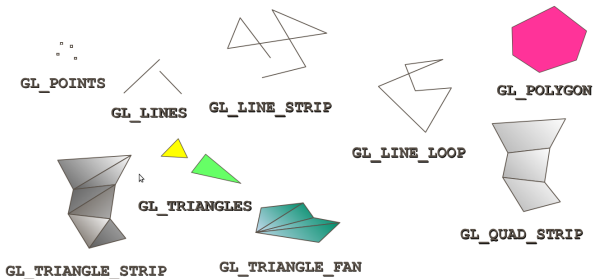
```
1   glBegin(GL_TRIANGLES);  
2       glVertex2f(x1, y1);  
3       glVertex2f(x2, y2);  
4       glVertex2f(x3, y3);  
5   glEnd();
```

Primitivas Geométricas 2D

- Quadriláteros

```
1  glBegin(GL_QUADS);  
2      glVertex2f(x1, y1);  
3      glVertex2f(x2, y2);  
4      glVertex2f(x3, y3);  
5      glVertex2f(x4, y4);  
6  glEnd();
```

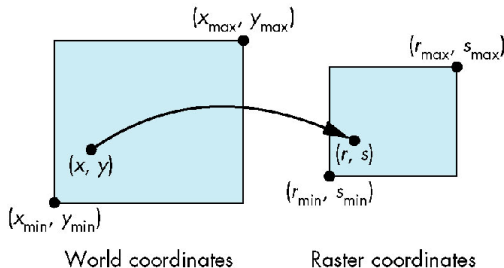
- Outras primitivas



3D para 2D

3D para 2D

- O pipeline da OpenGL é sempre 3D, mas é possível criar desenhos 2D, definindo
 - `glMatrixMode(GL_PROJECTION)`
 - `gluOrtho2D(left, right, bottom, top)`



Sistema de Coordenadas

Sistema de Coordenadas

- OpenGL permite desenhar gráficos de modo independente do dispositivo
- É possível especificar o Sistema de Coordenadas do Mundo (ponto flutuante) onde os objetos são definidos.
- Os elementos são traçados no Sistema de Coordenadas do Dispositivo ou Sistema de Coordenadas da Tela. (inteiro)
 - O mapeamento entre esses sistemas de coordenadas é feito de forma transparente pela OpenGL.

GLUT

GLUT - OpenGL Utility Toolkit

- Direcionada a eventos (Event-driven).
- Programa responde a eventos: clique do mouse, tecla pressionada, redimensionamento da janela, entre outros.
- Biblioteca que permite a criação de elementos de interface gráfica.
 - Criação de janelas e menus pop-up
 - Gerenciamento de eventos de mouse e teclado
- Gerenciamento de eventos é realizado por meio de funções "callback".
- Quem chama a função para o tratamento de um evento é a GL Utility Toolkit (GLUT) e não o programador.
- O papel do programador neste caso é definir a função a ser chamada.

GLUT

Tratamento de Eventos

- void glutDisplayFunc(funçãoCallback)
 - Define qual a função responsável por redesenhar a janela quando necessário.
 - A função deve ser do seguinte protótipo:
 - void nomeDaFunção(void)
- void glutReshapeFunc(AlteraTamanhoJanela)
 - Define a função responsável por tratar o evento referente ao redimensionamento da tela.
 - A função deve ser do seguinte protótipo:
 - void nomeDaFunção(int largura, int altura)

GLUT

Interação com o Teclado

- `void glutKeyboardFunc(funcaoDeTratamento)`
 - Define qual a função responsável por tratar o pressionamento de teclas.
 - A função deve ser do seguinte protótipo:
 - `void nomeDaFunção(unsigned char tecla, int x, int y)`
 - O primeiro parâmetro é o código ASCII da tecla pressionada.
 - Os parâmetros `x` e `y` representam as coordenadas da posição do mouse no momento em que a tecla foi pressionada.

```
1 void Teclado(unsigned char tecla, int x, int y)
2 {
3     if (tecla == 27)
4         doSomething();
5 }
```

GLUT

Interação com o Mouse

- `void glutMouseFunc(funcaoDeTratamento)`
 - Define qual a função responsável por tratar o pressionamento/liberação de botões do mouse.
 - A função deve ser do seguinte protótipo:
 - `void nomeDaFunção(int botao, int estado, int x, int y)`
 - O primeiro parâmetro é o botão pressionado, podendo ser: `GLUT_LEFT_BUTTON`, `GLUT_MIDDLE_BUTTON` OU `GLUT_RIGHT_BUTTON`.
 - O parâmetro estado indica se a tecla foi pressionada ou liberada, sendo que os valores possíveis são: `GLUT_UP` e `GLUT_DOWN`.
 - Os parâmetros `x` e `y` representam as coordenadas da posição do mouse no momento em que o botão foi pressionado.

```
1 void Mouse(int botao, int estado, int x, int y)
2 {
3     if (botao == GLUT_LEFT_BUTTON && estado == GLUT_DOWN)
4     {
5         glColor3f(1.0f, 0.0f, 0.0f);
6     }
7 }
```


Introdução

GLUT - Criação de Janelas

- `void glutInitWindowPosition(int x, int y)`
 - Utilizada para definir a posição inicial da janela, sendo que os parâmetros representam a posição do canto superior esquerdo.
- `void glutInitWindowSize(int width, int height)`
 - Define a largura e altura da janela.
- `int glutCreateWindow(char *string)`
 - Cria a janela, sendo que o parâmetro será o título dela.
- `void glutInitDisplayMode(unsigned int mode)`
 - Define qual será o modo inicial de display.

Exemplo

```
1  int main (int argc, char *argv)
2  {
3      glutInit(&argc, argv);
4      glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
5      glutInitWindowPosition(400, 400);
6      glutInitWindowSize(300, 300);
7      glutCreateWindow("Programa exemplo");
8      glutMainLoop();
9  }
```

Bibliografia

- **Básica:**

- Hearn, D. Baker, M. P. Computer Graphics with OpenGL, Prentice Hall, 2004. **(livro texto)**
- Angel, E. Interactive computer graphics: a top-down approach with OpenGL, Addison Wesley, 2000.
- Foley, J. et. al - Introduction to Computer Graphics, Addison-Wesley, 1993.
- Kessenich, J., Sellers, G., Shreiner, D. OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.5 with SPIR-V - Ninth Edition.

Bibliografia

- **Complementar:**

- Computer Graphics Comes of Age: An Interview with Andries van Dam. CACM, vol. 27, no. 7. 1982
- The RenderMan – And the Oscar Goes to... IEEE Spectrum, vol. 38, no. 4, abril de 2001.