

Introdução à OpenGL

SCC0250/0650 - Computação Gráfica

Prof. Rosane Minghim

<https://edisciplinas.usp.br/course/view.php?id=61213>

<https://edisciplinas.usp.br/course/view.php?id=61210>

P.A.E. Diego Cintra e Fábio Felix

diegocintra@usp.br, f_diasfabio@usp.br

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

12 de abril de 2018



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 4.6.

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 4.6.

O que é

- Coleção de rotinas que o programador pode utilizar para gerar elementos gráficos.
- Oferece suporte para gerar e exibir cenas 3D interativas.
- Contém mais de 200 comandos distintos para objetos e operações necessárias para produzir aplicações gráficas.

Introdução

O que não é

- Não possui funções para processar entrada de usuários ou manipulações de janela.
 - Foi construída para ser uma interface independente de hardware.
 - Podem ser utilizadas as funções específicas de cada plataforma, ou usar uma biblioteca independente (GLUT ou SDL, por exemplo).
- Não provê instruções alto nível que descrevem modelos geométricos tridimensionais.
 - Motores gráficos realizam esse tipo de modelagem, como Blender, Maya e 3DS Max.
 - Apesar de poder definir modelos complexos, isso somente é construído com um conjunto de primitivas geométricas: **pontos, linhas e polígonos.**

Introdução

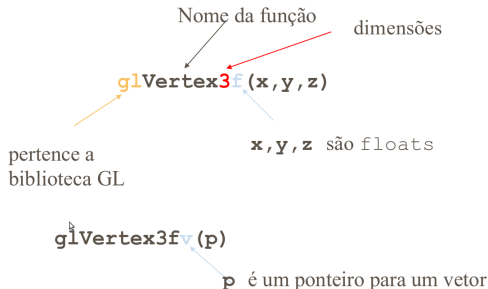
Operações da OpenGL

- Construção de modelos ou formas com primitivas geométricas, descrevendo modelos matemáticos de objetos.
- Definir pontos no espaço tridimensional e selecionar um ponto para visualização da cena.
- Cálculo de cor das formas e modelos, seja de maneira automática ou com condições específicas de iluminação.
- Conversão dos modelos matemáticos e as cores dos objetos para pixels na tela, através da *rasterização*.

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: Sufixos e tipos de dados equivalentes

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);
5          glEnd();
6      }
```

- Correto

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


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.
 - Valores correntes dessas variáveis podem ser consultados com o uso de funções com sufixo `glGet*()`.
 - Também existe a função `glutGet()` que retorna estados armazenados da biblioteca GLUT.

Exemplos de Inicialização

O valor corrente permanece ativo até ser alterado. Exemplos:

- Cor de Fundo: `glClearColor(r, g, b, alpha)`.
- Tamanho do Ponto: `glPointSize(3.0)`.

Exemplos de Recuperação

```
void glGetBooleanv(GLenum pname, GLboolean *data).  
int glutGet(GLUT_SCREEN_WIDTH).
```

Pipeline da OpenGL

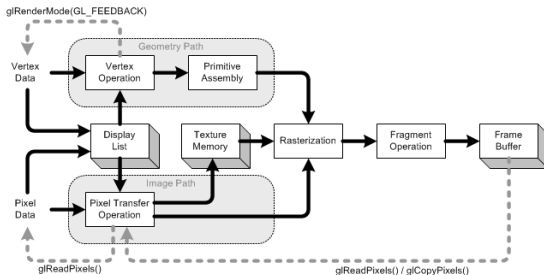


Figura: http://www.songho.ca/opengl/gl_pipeline.html.

- A sequência de operações da Figura 9 não é uma regra restrita de implementação da OpenGL, mas é um guia confiável de sequência adotada pela biblioteca.

Primitivas Geométricas 2D

- Ponto

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

- Linha

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

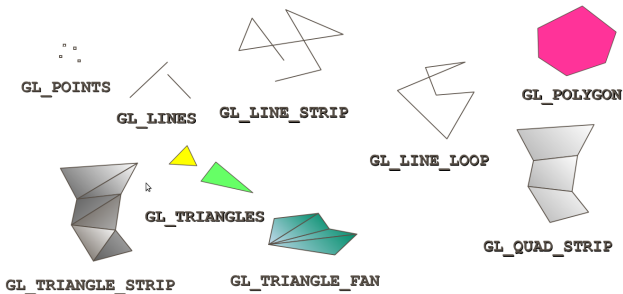
- Triângulo

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

Primitivas Geométricas 2D

- Quadriláteros

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



Bibliotecas

- OpenGL *basic library* ou OpenGL *core library* (GL)
- OpenGL *Utility* (GLU)
- Gerenciamento de janelas
 - OpenGL *Extension to the X Window System* (GLX)
 - *Apple GL* (AGL)
 - *Windows-to-OpenGL* (WGL)
 - *Presentation manager to OpenGL* (PGL)
 - **OpenGL Utility Toolkit** **OpenGL (GLUT)** , *Simple DirectMedia Layer* (SDL), *Graphics Library Framework* (GLFW)

Difícil

```
1 #include <GL/gl.h>
2 #include <GL/glu.h>
3 #include <GL/glx.h>
```

Simples

```
1 #include <GL/glut.h>
```

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 - Criação de Janelas I

Funções básicas

- *void glutInit(int *argcp, char **argv);*
 - Inicializa a biblioteca GLUT e negocia uma seção com o sistema de janelas.
- *void glutInitDisplayMode(unsigned int mode);*
 - Define qual será o modo inicial de display.
- *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.

GLUT - Criação de Janelas II

Funções básicas

- *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 glutMainLoop();*
 - Apresenta o gráfico inicial e coloca o programa em um laço infinito que analisa as entradas dos dispositivos como *mouse*, teclado, *touch pad* etc.

Programa OpenGL básico

```
1      #include <cstdlib>
2      #include <GL/glut.h> //ou #include <GLUT/glut.h> no macOS
3
4      int main (int argc, char *argv[])
5      {
6          glutInit(&argc, argv);
7          glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
8          glutInitWindowPosition(400, 400);
9          glutInitWindowSize(300, 300);
10         glutCreateWindow("Hello World");
11
12         //
13
14         glutMainLoop();
15
16         return EXIT_SUCCESS;
17     }
```

Compilação

```
g++ -o hello_world hello_world.cpp -lGL -lglut -lGLU
```

OpenGL básico

Antes de começar a desenhar algo na tela, é necessário definir a cor de fundo da tela *glClearColor* e *glClear*, a matriz OpenGL que estará sendo utilizada *glMatrixMode* e os parâmetros da projeção *gluOrtho2D*

```
1 //valores variam entre [0-1]
2 void glClearColor(GLfloat red, GLfloat green, GLfloat blue, GLfloat ↵
   alpha);
3
4 //GL_MODELVIEW (padrao), GL_PROJECTION, GL_TEXTURE, ↵
   GL_COLOR
5 void glMatrixMode(GLenum mode);
6
7 void gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, ↵
   GLdouble top);
```

OpenGL básico

Para cada objeto desenhado podem ser definidas características como cor, tamanho etc. Além disso, para que tudo seja realmente executado pelo OpenGL a função *glFlush* ou *glFinish* precisa ser chamada ao final de qualquer processo.

```
1  //GL_COLOR_BUFFER_BIT, GL_DEPTH_BUFFER_BIT, ←  
   GL_ACCUM_BUFFER_BIT, GL_STENCIL_BUFFER_BIT  
2  void glClear(GLbitfield mask);  
3  
4  //Cor do objeto que esta sendo desenhado  
5  void glColor3f(GLfloat red, GLfloat green, GLfloat blue);  
6  
7  //Diametro do objeto que esta sendo desenhado  
8  void glPointSize(GLfloat size);  
9  
10 //Forca execucao esvaziamento dos buffers e execucao das funcoes OpenGL  
11 void glFlush();
```

GLUT - Tratamento de Eventos

Funções básicas

- *void glutDisplayFunc(handle_display);*
 - Define qual a função responsável por redesenhar a janela quando necessário.
- *void glutReshapeFunc(handle_reshape);*
 - Define a função responsável por tratar o evento referente ao redimensionamento da tela.

```
1  void handle_display()
2  {
3
4  }
5  void handle_reshape(GLuint width, GLuint heigh)
6  {
7
8  }
```

GLUT - Mais funções

- *void glutPostRedisplay()*
 - Sinaliza para que a tela seja redesenhada na próxima iteração do *glutMainLoop*.
- *void glutFullScreen(void)*
- *void glutReshapeWindow(int width, int height)*

GLUT - Interação com o Teclado I

Funções básicas

- *void glutKeyboardFunc(handle_KeyboardFunc)*
 - invoca a função *callback* quando teclas como letras e números são pressionadas.
 - combinações de teclas podem ser gerenciadas utilizando também a função *glutGetModifiers()*.
- *void glutSpecialFunc(handle_SpecialFunc)*
 - invoca a função *callback* quando teclas como *Ctrl*, *Shift*, teclas de função (F1, F2, ...), teclas direcionais etc. são pressionadas.

GLUT - Interação com o Teclado II

```
1  void handle_KeyboardFunc(GLubyte key, GLint x, GLint y)
2  {
3      GLint m = glutGetModifiers();
4
5      if (key == 'a')
6          executarAlgumaCoisa();
7      else if(m == GLUT_ACTIVE_CTRL && key == 'c')
8          copiarAlgumaCoisa();
9  }
10 void handle_SpecialFunc(GLint key, GLint x, GLint y)
11 {
12     if (key == GLUT_KEY_F11)
13         doSomething();
14 }
```

GLUT - Interação com o Mouse I

Funções básicas

- *void glutMouseFunc(handle__mouse)*
 - invoca a função *callback* quando algum botão do **mouse** é clicado/liberado.
- *void glutPassiveMotionFunc(handle__passivemotion)*
 - invoca a função *callback* quando o **mouse** é movido na tela (com botão LIBERADO).
- *void glutMotionFunc(handle__motion)*
 - invoca a função *callback* quando o **mouse** é arrastado na tela (com botão PRESSIONADO).

GLUT - Interação com o Mouse II

```
1  void handle_mouse(GLint button, GLint action, GLint x, GLint y);  
2  {  
3      if(button == GLUT_LEFT_BUTTON && action == GLUT_DOWN)  
4          desenha_ponto(x, y);  
5  }  
6  void handle_passivemotion(GLint x, GLint y)  
7  {  
8  
9  }  
10 void handle_motion(GLint x, GLint y)  
11 {  
12  
13 }
```

GLUT - Criação e interação com menus I

Funções básicas

- *int glutCreateMenu(handle_menu).*
- *void glutAddMenuEntry(char *menu_name, int menu_index).*
- *void glutAddSubMenu(char *menu_name, int sub_menu_index).*
- *void glutAttachMenu(int button)* e *void glutDetachMenu(int button).*

GLUT - Criação e interação com menus II

```
1  void create_menu()
2  {
3      GLint menu_id = glutCreateMenu(handle_menu);
4      glutAddMenuEntry("Item 1", 1);
5      glutAddMenuEntry("Item 2", 2);
6
7      GLint submenu_id = glutCreateMenu(handle_menu);
8      glutAddMenuEntry("Subitem 1", 4);
9      glutAddMenuEntry("Subitem 2", 5);
10     glutAddMenuEntry("Subitem 3", 6);
11
12     glutAddSubMenu("Item 3", submenu_id);
13
14     glutAttachMenu(GLUT_RIGHT_BUTTON);
15 }
16 void handle_menu(GLint menu_index);
17 {
18
19 }
```

Bibliografia

- **Básica:**

- Hearn, D. Baker, M. P. Computer Graphics with OpenGL, Prentice Hall, 2004. **(livro texto)**
- Neider, J. Davis, T. Woo, M. OpenGL programming guide, 2007. **(livro base para aulas práticas)**
- 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.
- Material do ano passado: <https://sites.google.com/site/computacaograficaicmc2017t2/>
- Apostilas antigas da disciplina Computação Gráfica
 - <http://www.gbdi.icmc.usp.br/material?q=system/files/apostilas.pdf>