

OpenGL: Introdução

Conceitualização, instalação e primeiros exemplos

P.A.E. Eric Macedo Cabral¹ Profa. Rosane Minghim²

¹cabral.eric@usp.br, ²rminghim@icmc.usp.br

Instituto de Computação e Matemática Computacional (ICMC)
Universidade de São Paulo (USP)

28 de Agosto de 2018

Sumário

- 1 Introdução
- 2 OpenGL

- 1 Introdução
 - Histórico
 - Aplicações
 - Instalação

- Foi introduzido em 1992 pela Silicon Graphics e atualmente é mantida pelo Khronos Group.
- Está na versão 4.6.
- <http://www.opengl.org>

O que é a OpenGL?

- **Open Graphics Library (OpenGL)** é uma especificação de uma Application Programming Interface (API) para a criação de aplicações gráficas.
- Implementa rotinas gráficas e de modelagem bidimensional e tridimensional.
 - Menor complexidade ao desenvolvedor.
 - Rotinas comunicam diretamente com a placa gráfica.
- Contém mais de 200 comandos distintos para objetos e operações necessárias para produzir aplicações gráficas.
- Foi construída para ser uma interface independente de hardware.
- A OpenGL é uma grande máquina de estados.

O que a OpenGL não é?

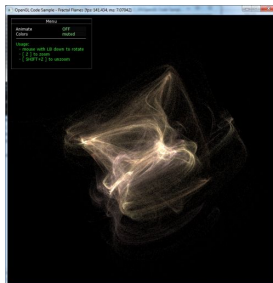
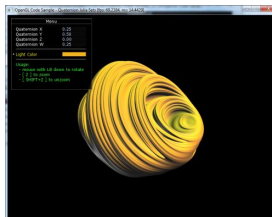
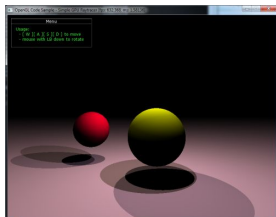
- Uma biblioteca para processar a entrada de usuários ou manipulações de janela.
 - Quem faz isso é a biblioteca GLUT.
- Não provê instruções alto nível que descrevem modelos geométricos tridimensionais.
 - O usuário pode definir modelos primitivos geométricos como: pontos, linhas e polígonos.

O que pode ser feito com a 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.
 - **Rasterização:** converte uma imagem vetorial para uma imagem matricial.



Aplicações



Cada Sistema operacional possui uma implementação da OpenGL e suas bibliotecas utilitárias, como também possui suas próprias implementações do compilador da linguagem de programação C/C++. Logo, cada ambiente possui um processo de instalação diferente. Siga o guia de instalação disponibilizado.

2 OpenGL

- Conceitos Preliminares
- Primitivas 2D
- GLUT

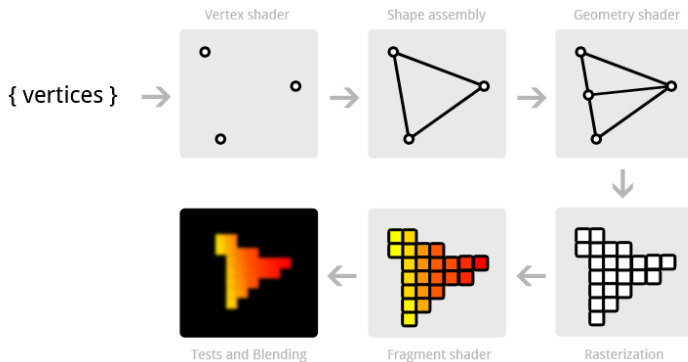


Figura: OpenGL Pipeline

Sintaxe

- Visa padronizar e facilitar a utilização da biblioteca.

`glVertex3f(x,y,z)`

Nome da função → `glVertex3f`
dimensões → `3`
pertence a biblioteca GL → `gl`
`x,y,z` são floats → `x,y,z`

`glVertex3fv(p)`

`p` é um ponteiro para um vetor → `p`

Tipos de Dados

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

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

Tabela: Sufixos e tipos de dados equivalentes

Errado!

```
void drawDot(int x, int y)
{
    glBegin(GL_POINTS);
    glVertex2i(x, y);
    glEnd();
}
```

Correto!

```
void drawDot(GLint x, GLint y)
{
    glBegin(GL_POINTS);
    glVertex2i(x, y);
    glEnd();
}
```

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.
- O valor corrente permanece ativo até ser alterado.

Ambiente Visual

- Uma imagem consiste em uma matriz de pontos, já um modelo é uma representação computacional de um objeto.
- O modelo corresponde a uma estrutura de dados com a descrição geométrica da cena.
- Na OpenGL os objetos são representados no Sistema de Referência do Universo (SRU).
- Todos os comandos e modelos são definidos em relação a este sistema de referência.

Ambiente Visual 2D

- No caso 2D, é necessário definir a porção o universo que desejamos mapear na tela.
- Essa área é chamada de janela de seleção, ou window.
- Na OpenGL, definimos essa porção de tela por meio do comando:

```
gluOrtho2D(  
    GLdouble left,  
    GLdouble right,  
    GLdouble bottom,  
    GLdouble top  
);
```

Ambiente Visual 2D

- É necessário definir também em que parte do monitor deseja-se exibir o conteúdo da window.
- Chamamos essa região de viewport (ou janela de exibição).
- Na OpenGL, definimos essa porção de tela por meio do comando:

```
glViewport(  
    GLint x,  
    GLint y,  
    GLsizei width,  
    GLsizei height  
);
```

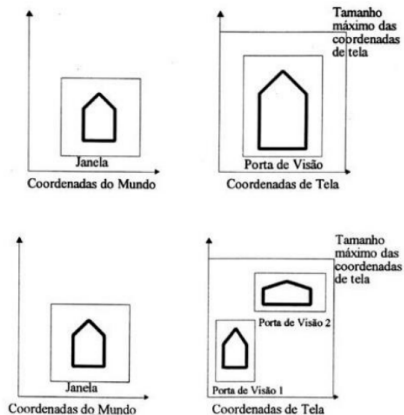


Figura: Viewport na OpenGL

OpenGL – Primitivas 2D

Ponto

```
glBegin(GL_POINTS);  
    glVertex2f(x1, y1);  
    glVertex2f(x2, y2);  
    glVertex2f(x3, y3);  
glEnd();
```

Linha

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

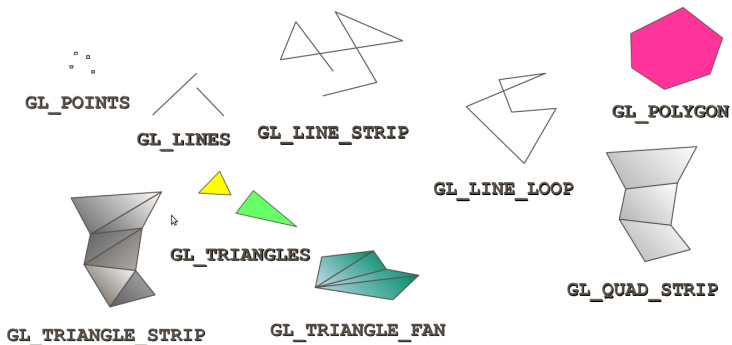
Triângulo

```
glBegin(GL_TRIANGLES | GL_TRIANGLE_STRIP | GL_TRIANGLE_FAN);  
    glVertex2f(x1, y1);  
    glVertex2f(x2, y2);  
    glVertex2f(x3, y3);  
glEnd();
```

Quadriláteros

```
glBegin(GL_QUADS | GL_QUAD_STRIP);  
    glVertex2f(x1, y1);  
    glVertex2f(x2, y2);  
    glVertex2f(x3, y3);  
    glVertex2f(x4, y4);  
glEnd();
```

OpenGL – Primitivas 2D



GLUT – OpenGL Utility Toolkit

- Baseada em eventos.
 - Callbacks.
 - O pipeline de manipulação de eventos já é implementado por padrão, o desenvolvedor apenas indica qual a função callback.
- Instancia elementos de interface gráfica.
 - Janelas.
 - Pop-ups.

Criação de Janelas

- **void glutInit(int *argc, 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.
- **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.

GLUT Main Loop

- **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

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

int main (int argc, char *argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition(400, 400);
    glutInitWindowSize(300, 300);
    glutCreateWindow("Hello World");

    // glutDisplayFunc(Draw);

    glutMainLoop();

    return EXIT_SUCCESS;
}
```

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`.

```
//valores variam entre [01]  
void glClearColor(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha);  
  
//GL_MODELVIEW (padrao), GL_PROJECTION, GL_TEXTURE, GL_COLOR  
void glMatrixMode(GLenum mode);  
  
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.

```
//GL_COLOR_BUFFER_BIT, GL_DEPTH_BUFFER_BIT, ↔  
GL_ACCUM_BUFFER_BIT, GL_STENCIL_BUFFER_BIT  
void glClear(GLbitfield mask);  
  
//Cor do objeto que esta sendo desenhado  
void glColor3f(GLfloat red, GLfloat green, GLfloat blue);  
  
//Diametro do objeto que esta sendo desenhado  
void glPointSize(GLfloat size);  
  
//Forca execucao esvaziamento dos buffers e execucao das funcoes OpenGL  
void glFlush();
```

Funções de exibição

- *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.

Funções de evento: Teclado

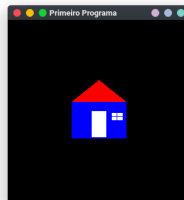
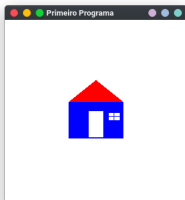
- `void glutKeyboardFunc(handle_KeyboardFunc)`
 - invoca a função *callback* quando teclas como letras e números são pressionadas.
 - combinações de teclas podem ser gerenciadas utilizando também a função `glutGetModifiers()`.
 - a função `handle_KeyboardFunc` tem assinatura:
`void handle_KeyboardFunc(unsigned char key, GLint x, GLint y)`

Funções de eventos: Mouse

- `void glutMouseFunc(mouse_callback)`
 - invoca a função *callback* quando algum botão do **mouse** é clicado/liberado.
 - a função `mouse_callback` tem a assinatura:
`void mouse_callback(int button, int state, int x, int y)`

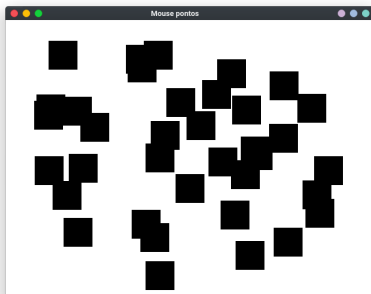
Tarefa 1: Desenhar Casa

- Desenhar uma casa simples em OpenGL
- Ao pressionar a tecla espaço do teclado, o plano de fundo escurece.
- Se o fundo estiver escuro e o usuário pressionar qualquer tecla, o plano de fundo torna-se branco.



Tarefa 2: Pontos com o Mouse

- Criar uma tela branca.
- Detectar cliques com o botão esquerdo e desenhar um ponto preto no local que foi clicado de tamanho 50 na tela.
- Ao pressionar o botão direito, limpa a tela.



- **Hoário de atendimento:** Quarta-feira, das 14 às 16h.
 - Laboratório VICG (Bloco 1, sala 007)
 - Enviar email com antecedência.
- **Email :** cabral.eric@usp.br
 - **Subject :** [CG2018_2]

● 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.

• 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>