

Computação Gráfica

OpenGL: visão geral

Prof. Gabriel M. Alves

2023-08-08

versão: 0707c5

- Lançada pela empresa *Silicon Graphics* em 1992.
- É uma API (*Application Programming Interface*) de padrão aberto, multiplataforma e *cross-language*.
- Pode ser acessada por programas escritos em diversas linguagens e diferentes sistemas operacionais.
- Fácil uso se comparada à antecessora PHIGS (*Programmer's Hierarchical Interactive Graphics System*)



- Trata apenas da parte gráfica e não possui recursos de GUI (gerenciamento de janelas, mouse, etc)
- Lançamento em 2003 da **OpenGL ES** (*Embedded System*) para dispositivos móveis
- Lançamento em 2011 da **WebGL** para aplicações de computação gráfica no browser (baseada em OpenGL ES)
- Desde 2006 a OpenGL é mantida pelo consórcio **Khronos Group**
- Outras API's gráficas: **Microsoft Direct3D**, **Vulkan**



- Versões da OpenGL

- 1.0 a 1.5: (1995 a 2003) - texturas 3D, extensões ARB (*Architecture Review Board*), *vertex arrays*, *vertex buffer objects*, GLSL (*OpenGL Shading Language*), texturas *cubemap* e *mipmapping*.
- 2.0 e 2.1: (2004 a 2008) - versões 1.1 e 1.2 da GLSL
- 3.0: (2008) - **ruptura**: divisão entre *pipeline* fixo e programável. Removeu muitas funções, por exemplo: `glBegin()` e `glEnd()`
- 3.1 a 3.3 (2009 e 2010) - aprimoramentos da OpenGL; incluiu novo estágio na pipeline, o processamento geométrico (*geometry shader*)
- 4.0 (2010) - incluiu outro estágio na pipeline, o *tessellation shader*
- 4.1 a 4.6 (2010 a 2017) - aprimoramentos da OpenGL

- Versão atual da OpenGL: **4.6**

Relembrando: O que é Computação Gráfica?

Transformar *dados* em imagens!

```
1 # 3ds Max Wavefront OBJ Exporter v0.9.7b - (c)2007 guruware
2 # File Created: 20.12.2011 11:12:50
3
4 mtlllib 12221_Cat_v1_l3.mtl
5
6 #
7 # object Cat
8 #
9
10 v 0.6134 -21.9357 31.4441
11 v 0.6205 -21.8541 31.4935
12 v 0.5534 -21.8682 31.5159
13 v 0.5464 -21.9516 31.4643
14 v 0.6207 -21.7610 31.5371
15 v 0.5544 -21.7724 31.5613
16 v 0.4816 -21.8847 31.5310
17 v 0.4743 -21.9703 31.4781
18 v 0.4838 -21.7858 31.5776
19 v 0.6156 -21.6618 31.5749
20 v 0.6070 -21.5619 31.6064
```

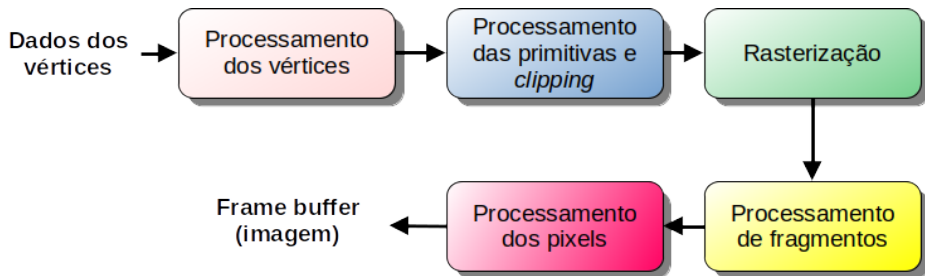


- Para isso é preciso realizar várias tarefas dispostas em um *pipeline*.
 - Um canal de informações, na qual resultados da etapa anterior serve como entrada para a próxima. Analogia: linha de produção.
 - Pipeline Definition & Meaning - Merriam-Webster

- antes da versão 3.0 o pipeline é chamado **fixo**, ou OpenGL legacy
- pipeline fixo limita a criação de efeitos novos
- permite utilizar apenas as funções prontas (*fixed-function*)
- os dados são processados imediatamente o que é ruim para cenas e objetos complexos (*performance*)
- a curva de aprendizado é menor (opinião pessoal)
- ainda é possível utilizar o pipeline fixo (modo de *compatibilidade*)

OpenGL: pipelines

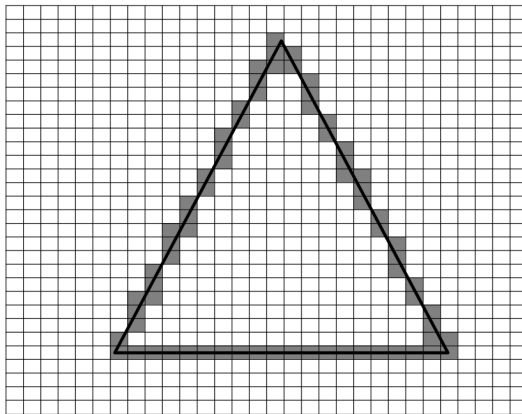
- Visão simplificada do **pipeline fixo**



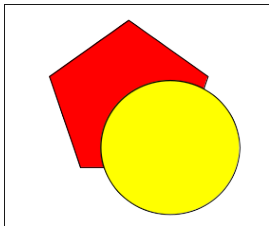
- Dados dos vértices:
 - informações geométricas (coordenadas x, y, z)
 - atributos: cor, textura, faces, etc.
- Processamento de vértices:
 - geração dos vértices a partir das informações geométricas;
 - projeção dos vértices (modelo, câmera)
- Processamento de primitivas e *clipping*:
 - geração das primitivas geométricas (triângulos, quadrados) a partir dos vértices gerados;
 - aplicar recortes (*clipping*) no volume de visualização

OpenGL: pipelines

- Rasterização:
 - Converte a representação 3D de um objeto para um conjunto de pontos chamados de **fragmentos** que, posteriormente, serão tratados como os pixels da imagem.
 - Leva em consideração as coordenadas do dispositivo.



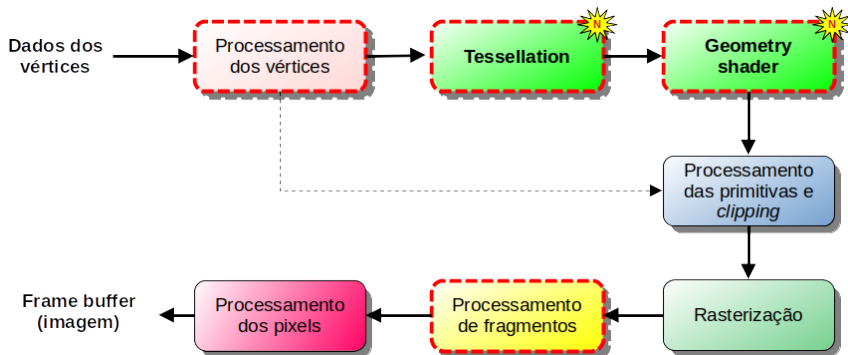
- Processamento de fragmentos:
 - os atributos dos vértices (cores, texturas, etc) são atribuídos aos fragmentos;
 - é realizado o cálculo dos valores dos fragmentos que estão entre os vértices.
- Processamento dos pixels:
 - Uma imagem é formada por pixels;
 - cada pixel pode conter vários fragmentos de diferentes objetos;
 - considera a posição e orientação da câmera.



- Motivações para o pipeline programável:
 - possibilidade de incluir novas funções no pipeline da OpenGL
 - explorar poder computacional das placas gráficas (GPU - *Graphical Processing Unit*)
 - fabricantes de placas gráficas começaram a oferecer extensões OpenGL
 - surgimento dos *shaders*, fragmentos de código que permitem programar etapas da pipeline
- *OpenGL Shading Language* (GLSL)
 - linguagem similar a C e C++
 - possui características similares à *RenderMan Shading Language*
 - gera programa para GPU
 - possui suporte nativo a operações matriciais
- Pipeline fixo (modo imediato) e o pipeline programável (modo *core profile*)

OpenGL: pipelines

- Visão simplificada do **pipeline** programável



- processamento dos vértices (*vertex shader*):
 - é executado uma vez por vértice e processa os dados associados;
 - programador deve especificar as matrizes de transformação e não apenas as transformações como no pipeline fixo.
- processamento de fragmentos (*fragment shader*):
 - é executado uma vez por fragmento e configura ou descarta a cor dele.
 - calcula os valores das cores interpoladas por fragmento.
 - caso contenha textura, o cálculo da cor deve considerá-la.

- Etapas opcionais:

- *Tessellation shader*: introduzida na OpenGL 4.0, essa etapa permite gerenciar o nível de detalhes de uma cena. Ela pode ajustar a malha do objeto, por exemplo: adicionar mais e menores triângulos quando o objeto se aproxima da câmera para visualizar mais detalhes.
- *Geometry shader*: introduzida na OpenGL 3.2, essa etapa permite transformar a geometria original, por exemplo trocar triângulos por linhas ou novos triângulos de diferentes tamanhos. Essas alterações radicais de um objeto geralmente são feitas para algum efeito específico.

- A OpenGL é uma biblioteca que interage com várias linguagens de programação, por exemplo: Java, Python, JavaScript.
- No entanto, geralmente uma aplicação que utiliza a OpenGL é escrita em C ou C++.
- Principais bibliotecas:
 - GL É o núcleo do OpenGL responsável por ler e escrever dados no *framebuffer*, ou seja, renderizar.
 - glew The OpenGL Extension Wrangler é uma biblioteca que provê mecanismos eficientes, em tempo de execução, para determinar quais extensões OpenGL são suportadas para uma determinada plataforma. Site: [OpenGL SDK: GLEW - The OpenGL Extension Wrangler Library](#).

- Bibliotecas adicionais, incluindo para interface gráfica:

GLU biblioteca que oferece funções utilitárias para utilizar a biblioteca GL de modo mais fácil. Está em desuso com a nova versão do OpenGL.

freeglut é uma biblioteca livre/aberta que substitui a OpenGL Utility Toolkit (GLUT) e visa tornar portáteis os programas OpenGL. Ela é responsável por todas as tarefas específicas do sistema operacional necessárias para criar janelas, inicializar contextos OpenGL e lidar com eventos de entrada. Site: [The freeglut Project](#).

GLFW *Graphics Library Framework* é uma biblioteca para construir interfaces gráficas que utilizam a API OpenGL. Site: [GLFW: An OpenGL library](#).

- Outras bibliotecas/frameworks para desenvolvimento de interfaces gráficas de aplicações que utilizam a OpenGL:
 - **SFML** *Simple and Fast Multimedia Library*. Site: <https://www.sfml-dev.org/>
 - **SDL** *Simple DirectMedia Layer*. Site: <https://www.libsdl.org/>
 - **Qt** Framework para desenvolvimento de aplicações multi-plataforma. Site: <https://www.qt.io/>
- JOGL (*Java OpenGL*) é um *bind* em Java para OpenGL e permite escrever aplicações de computação gráfica utilizando Java e OpenGL. Site: [JOGL - Java Binding for the OpenGL API](#)
 - Confirmar a situação do projeto.

OpenGL: bibliotecas

- **Mesa3D** é uma implementação *open-source* da OpenGL.
- AMD e NVidia possuem implementações próprias.
- **Nouveau** é uma implementação *open-source* do driver Nvidia.

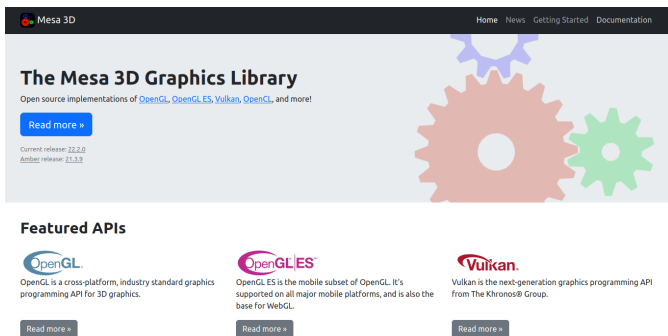


Figure: <https://mesa3d.org/>

- Khronos Group. *History of OpenGL*. Disponível em: https://www.khronos.org/opengl/wiki/History_of_OpenGL. Último acesso em outubro de 2022.
- Wikipedia. *OpenGL*. Disponível em: <https://en.wikipedia.org/wiki/OpenGL>. Último acesso em outubro de 2022.
- Jonas Gomes, Luiz Velho. *Fundamentos da Computação Gráfica*, IMPA: Instituto de Matemática Pura e Aplicada. 2015.
- Dave Shreiner, Graham Sellers, John M. Kessenich, Bill Licea-Kane. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3*, Addison-Wesley. 2013.

- Dúvidas?
- Comentários?

Contato

Gabriel Marcelino Alves
gabriel.marcelino@ifsp.edu.br



This work is licensed under Attribution-NonCommercial-ShareAlike 4.0 International. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/>

