

# OpenGL

Ricardo Dutra da Silva  
Elisa de Cássia Silva Rodrigues

Universidade Tecnológica Federal do Paraná  
Universidade Federal de Itajubá

2020

# Criando uma janela

- Tutoriais sobre OpenGL:
  - <https://learnopengl.com/>
  - <https://openglbook.com/>
  - <http://www.opengl-tutorial.org/>

# Criando uma janela

- Bibliotecas para a criação de janelas e gerenciamento de eventos (sinais de entrada, etc.):
  - GLUT: usada nos nossos códigos de exemplo;
  - GLFW: alternativa que pode ser encontrada nos tutoriais.
- Os códigos básicos para criação de uma janela podem ser encontrados no arquivo `window.{cpp,py}`.
- As funções do GLUT e OpenGL serão descritas na sequência.

# Criando uma janela

`glutInit`

Inicia a biblioteca GLUT.

- Primeira função encontrada no método `main`.
- Função deve ser sempre chamada no início do programa.

# Criando uma janela

- Funções para criar um “contexto”, que permite a comunicação com o hardware.

```
glutInitContextVersion(X,Y)
```

Informa que o contexto é compatível com a versão OpenGL X.Y.

- No código usamos a versão 3.3.

```
glutInitContextProfile(valor)
```

Informa a compatibilidade do contexto.

- No código usamos GLUT\_CORE\_PROFILE: não queremos compatibilidade com algumas características de versões anteriores do OpenGL.

# Criando uma janela

- Pesquise e procure saber qual versão OpenGL é compatível com o seu sistema.
- No Linux você pode usar o comando: `glxinfo`.
- Exemplo de saída:

```
$ glxinfo | grep "OpenGL"
OpenGL vendor string: NVIDIA Corporation
OpenGL renderer string: GeForce GT 740M/PCIe/SSE2
OpenGL core profile version string: 4.6.0 NVIDIA 440.100
OpenGL core profile shading language version string: 4.60 NVIDIA
OpenGL core profile context flags: (none)
OpenGL core profile profile mask: core profile
OpenGL core profile extensions:
OpenGL version string: 4.6.0 NVIDIA 440.100
OpenGL shading language version string: 4.60 NVIDIA
OpenGL context flags: (none)
OpenGL profile mask: (none)
OpenGL extensions:
OpenGL ES profile version string: OpenGL ES 3.2 NVIDIA 440.100
OpenGL ES profile shading language version string: OpenGL ES GLSL ES 3.20
OpenGL ES profile extensions:
$
```

- Indica a compatibilidade com a versão OpenGL 4.6.

## `glutInitDisplayMode`

Define opções de como uma cena deve ser renderizada (desenhada). Opções são separadas por '|'.

- No código usamos:
  - GLUT\_DOUBLE: habilita o uso de double-buffering;
  - GLUT\_RGBA: habilita a composição de cores usando canais red, green, blue e alpha (cores mais opacidade).
- Pesquise e descreva outras opções para a função.

# Criando uma janela

- Na sequência é definida a dimensão e o nome da janela.

`glutInitWindowSize`

Define a dimensão da janela que será mostrada.

`glutCreateWindow`

Cria e define o nome da janela.



# Criando uma janela

`glewInit()`

Inicia a compatibilidade de funções do OpenGL em diferentes sistemas operacionais.

- Deve ser chamada nos códigos em C++.

# Criando uma janela

- Definição das callbacks, funções responsáveis por gerenciar eventos.
- Pelo menos deve ser definida a função de desenho (display).

## `glutReshapeFunc`

Define a função para redimensionamento da janela.

## `glutDisplayFunc`

Define a função para desenho.

## `glutKeyboardFunc`

Define a função para tratar entradas do teclado.

# Criando uma janela

- Última função do main.

## `glutMainLoop`

Loop que executa e gerencia as funções/callbacks que devem ser chamadas para cada evento que ocorre no programa.

# Criando uma janela

- As funções definidas para as callbacks no exemplo:
  - display,
  - reshape,
  - keyboard.

# Criando uma janela

- A função `display` define o que será desenhado na janela.

## `glClearColor`

Define a cor para “apagar” a tela antes de desenhar.

## `glClear`

Apaga/pinta a tela com a cor definida. Usada antes de desenhar a cena.

## `glutSwapBuffers`

Pede para que os desenhos sejam de fato exibidos na tela. Usada para double-buffering.

# Criando uma janela

- No exemplo, a janela é pintada com a cor dada pelos valores RGBA (0.2, 0.3, 0.3, 1.0).
- Pesquise os parâmetros RGBA para a cor de sua preferência e mude o código para desenhá-la.
- Pesquise e descreva outros parâmetros para a função `glClear`.
- `glFlush` é uma alternativa para `glutSwapBuffers`, pesquise quando ela é usada.

# Criando uma janela

- A função `reshape` define o que fazer quando a janela é redimensionada.

## `glViewport`

Define o tamanho do retângulo de desenho: a área da janela que mostra primitivas desenhadas.

## `glPostRedisplay`

Envia um evento para chamar a função de desenho. Deve ser chamada sempre que o desenho muda.

# Criando uma janela

- A função `keyboard` define o que fazer quando uma tecla é pressionada.
- No exemplo são definidas teclas que terminam o programa.



# Criando uma janela

- Para compilar o código C++:

```
$ make
```

```
g++ window.cpp -o window -lglut -lGLEW -lGL
```

- Note as flags necessárias para as bibliotecas.
- Para executar o código C++:

```
$ ./window
```

- Para executar o código python:

```
$ ./window.py
```

# Criando uma janela

- A janela.

