

# Computação Gráfica

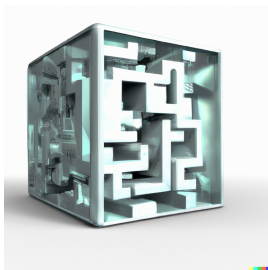
Guilherme Henrique de Souza Nakahata

Universidade Estadual do Paraná - Unespar

18 de Abril de 2024

# Imagem - Matriz

- Resolução - Largura x Altura;
- Exemplo: Imagem com 600 x 300;
- $600 \times 300 = 180.000$  pixels;
- 2D.



# Cores

- 0 - 255;
- 256 possibilidades por canal;
  - Vermelho;
  - Verde;
  - Azul
- $256^3$  cores;
- 16.777.216 possibilidades.

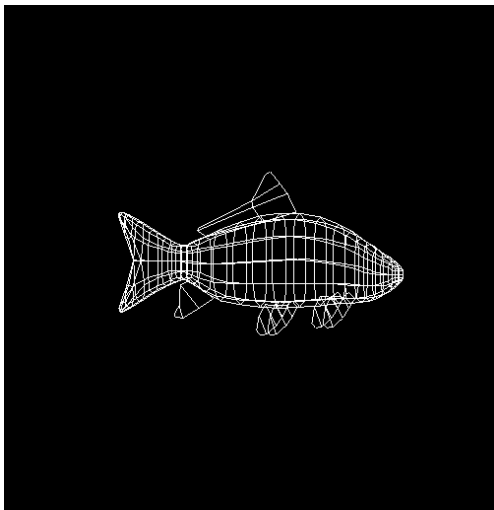
xxxxxx	FFxxxx	CCxxxx	99xxxx	66xxxx	33xxxx	00xxxx
xxFFFF	FFFFFF	CCFFFF	99FFFF	66FFFF	33FFFF	00FFFF
xxFFCC	FFFFCC	CCFFCC	99FFCC	66FFCC	33FFCC	00FFCC
xxFF99	FFFF99	CCFF99	99FF99	66FF99	33FF99	00FF99
xxFF66	FFFF66	CCFF66	99FF66	66FF66	33FF66	00FF66
xxFF33	FFFF33	CCFF33	99FF33	66FF33	33FF33	00FF33
xxFF00	FFFF00	CCFF00	99FF00	66FF00	33FF00	00FF00
xxCCFF	FFCCFF	CCCCFF	99CCFF	66CCFF	33CCFF	00CCFF
xxCCCC	FFCCCC	CCCCCC	99CCCC	66CCCC	33CCCC	00CCCC
xxCC99	FFCC99	CCCC99	99CC99	66CC99	33CC99	00CC99
xxCC66	FFCC66	CCCC66	99CC66	66CC66	33CC66	00CC66
xxCC33	FFCC33	CCCC33	99CC33	66CC33	33CC33	00CC33
xxCC00	FFCC00	CCCC00	99CC00	66CC00	33CC00	00CC00
xx99FF	FF99FF	CC99FF	9999FF	6699FF	3399FF	0099FF
xx99CC	FF99CC	CC99CC	9999CC	6699CC	3399CC	0099CC
xx9999	FF9999	CC9999	999999	669999	339999	009999
xx9966	FF9966	CC9966	999966	669966	339966	009966
xx9933	FF9933	CC9933	999933	669933	339933	009933
xx9900	FF9900	CC9900	999900	669900	339900	009900
xx66FF	FF66FF	CC66FF	9966FF	6666FF	3366FF	0066FF
xx66CC	FF66CC	CC66CC	9966CC	6666CC	3366CC	0066CC
xx6699	FF6699	CC6699	996699	666699	336699	006699
xx6666	FF6666	CC6666	996666	666666	336666	006666
xx6633	FF6633	CC6633	996633	666633	336633	006633
xx6600	FF6600	CC6600	996600	666600	336600	006600
xx33FF	FF33FF	CC33FF	9933FF	6633FF	3333FF	0033FF
xx33CC	FF33CC	CC33CC	9933CC	6633CC	3333CC	0033CC
xx3399	FF3399	CC3399	993399	663399	333399	003399
xx3366	FF3366	CC3366	993366	663366	333366	003366
xx3333	FF3333	CC3333	993333	663333	333333	003333
xx3300	FF3300	CC3300	993300	663300	333300	003300
xx00FF	FF00FF	CC00FF	9900FF	6600FF	3300FF	0000FF
xx00CC	FF00CC	CC00CC	9900CC	6600CC	3300CC	0000CC
xx0099	FF0099	CC0099	990099	660099	330099	000099
xx0066	FF0066	CC0066	990066	660066	330066	000066
xx0033	FF0033	CC0033	990033	660033	330033	000033
xx0000	FF0000	CC0000	990000	660000	330000	000000

- OpenGL suporta uma ampla gama de recursos gráficos
  - Geometria 3D;
  - Iluminação;
  - Sombreamento;
  - Texturas;
- OpenGL é compatível com uma ampla gama de plataformas
  - Windows;
  - Mac OS;
  - Linux;
  - iOS;
  - Android;

- Altamente otimizado para gráficos 3D em tempo real;
- OpenGL também suporta o uso de shaders personalizados;
- OpenGL é amplamente usado na indústria de jogos e em outras aplicações gráficas:
  - Visualização científica;
  - Modelagem 3D;
  - Animação;
  - Design gráfico
- Pode ser utilizada com diversas linguagens como:
  - C++;
  - Python;
  - Java;
  - C;
  - Rust;
  - Ruby;
  - Lua.
- Frameworks GLFW e Qt.

- Tipo básico de um objeto gráfico;
- Série de vértices;
- Usado para construir modelos mais complexos;
  - Polígonos;
  - Malhas;
- Exemplo:





# Primitivas

- Algumas primitivas OpenGL;
  - **GL\_POINTS**: Pontos individuais, cada um definido por um vértice.
  - **GL\_LINES**: Conjunto de linhas conectando pares de vértices adjacentes.
  - **GL\_TRIANGLES**: Conjunto de triângulos, conectando cada grupo de três vértices adjacentes.
  - **GL\_QUADS**: Conjunto de quadrados, conectando cada grupo de quatro vértices adjacentes.

Primitivas mais avançadas:

- **GL\_TRIANGLE\_STRIP**;
- **GL\_TRIANGLE\_FAN**.

Desenhos com formas mais complexas;

Caracteres de texto;

Gráficos 2D;

Outros tipos de objetos.



<b>Tipo de dado OpenGL</b>	<b>Representação interna</b>	<b>Tipo de dado C equivalente</b>
GLbyte	8-bit integer	signed char
GLshort	16-bit integer	short
GLint GLsizei	32-bit integer	int ou long
GLfloat GLclampf	32-bit floating-point	float
GLdouble GLclampd	64-bit floating-point	double
GLubyte GLboolean	8-bit unsigned integer	unsigned char
GLushort	16-bit unsigned integer	unsigned short
GLuint GLenum GLbitfield	32-bit unsigned integer	unsigned long ou unsigned int

- Teclado;
- Mouse;
- Câmeras;
- Scanners;
- Mesa digitalizadora;
- JoyStick;
- Entre outros.

- Janelas
  - Mover;
  - Redimensionar;
  - Fechar;
  - Abrir;
- Eventos;
  - Clique do mouse;
  - Teclado;
  - Mesa digitalizadora;

- Listeners:
  - Mouse;
  - Teclado;
- MovedMouse;
- MouseDraged;
- OnKeyPressed;
- MouseFunc;

# Exemplo eventos

```
1 int main(int argc, char** argv) {  
2     glutInit(&argc, argv);  
3     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);  
4     glutInitWindowSize(500, 500);  
5     glutInitWindowPosition(120, 150);  
6     glutCreateWindow("Atividade 02/2");  
7     gluOrtho2D(0, 500, 500, 0);  
8     glutDisplayFunc(Desenha);  
9     glutMouseFunc(VerificarMouse);  
10    glutSpecialFunc(ListenerTeclado);  
11    glutMainLoop();  
12 }
```

## Exemplo eventos

```
1 void VerificarMouse(int button, int state, int x, int y) {  
2     if (button == GLUT_LEFT_BUTTON) {  
3         if (state == GLUT_DOWN) {  
4             listaPontos.push_back(x,y);  
5         }  
6     } else if (button == GLUT_RIGHT_BUTTON) {  
7         if (state == GLUT_DOWN) {  
8             MenorDistancia(x, y);  
9         }  
10    }  
11    glutPostRedisplay();  
12  
13 }
```

- Mouse:
  - **GLUT\_LEFT\_BUTTON;**
  - **GLUT\_MIDDLE\_BUTTON;**
  - **GLUT\_RIGHT\_BUTTON;**
  - **GLUT\_UP** ou **GLUT\_DOWN;**

# Exemplo Eventos

- Teclado:
  - Key;
  - GLUT\_KEY\_UP;
  - GLUT\_KEY\_DOWN;
  - GLUT\_KEY\_RIGHT;
  - GLUT\_KEY\_LEFT;

```
1 void ListenerTeclado(int key, int x, int y) {  
2     if (key == GLUT_KEY_UP) {  
3         // Rotina A  
4     } else if (key == GLUT_KEY_DOWN) {  
5         // Rotina B  
6     } else if (key == GLUT_KEY_RIGHT) {  
7         // Rotina C  
8     } else if (key == GLUT_KEY_LEFT) {  
9         // Rotina D  
10    }  
11    glutPostRedisplay();  
12 }
```



# Exemplo Eventos

- Teclado:
  - Key;
  - GLUT\_KEY\_UP;
  - GLUT\_KEY\_DOWN;
  - GLUT\_KEY\_RIGHT;
  - GLUT\_KEY\_LEFT;

```
1 void ListenerTeclado(int key, int x, int y) {  
2     if (key == GLUT_KEY_UP) {  
3         // Rotina A  
4     } else if (key == GLUT_KEY_DOWN) {  
5         // Rotina B  
6     } else if (key == 'C') {  
7         // Rotina C  
8     } else if (key == 'A') {  
9         // Rotina D  
10    }  
11    glutPostRedisplay();  
12 }
```

# Exemplo Eventos

- Teclado:
  - Key;
  - GLUT\_KEY\_UP;
  - GLUT\_KEY\_DOWN;
  - GLUT\_KEY\_RIGHT;
  - GLUT\_KEY\_LEFT;

```
1 void ListenerTeclado(int key, int x, int y) {  
2  
3     switch(key):  
4  
5         case 'a':  
6             // Rotina A  
7             break;  
8         case 'b':  
9             // Rotina B  
10            break;  
11        case 'c':  
12            // Rotina C  
13            break;  
14        case 'd':  
15            // Rotina D  
16            break;  
17        glutPostRedisplay();  
18 }
```

- Espessura;
- Angulação;
- Comprimento;



- Necessário dois pontos para realizar uma linha;
- Ponto Inicial;
- Ponto Final.

```
glBegin(GL_LINES);  
glVertex2i(x0, y0);  
glVertex2i(x1, y1);  
glEnd();
```

# Primitivas - Linhas

```
1 void Desenho() {  
2  
3     // Define que o buffer de cores deve ser limpo antes da renderizacao de cada imagem  
4     glClearColor(GL_COLOR_BUFFER_BIT);  
5  
6     // Define a cor  
7     glColor3f(1.0, 0.0, 0.0);  
8  
9     // Define a largura  
10    glLineWidth(3);  
11  
12    // Desenha a linha entre os pontos (0,0) e (1,0)  
13    glBegin(GL_LINES);  
14        glVertex2f(0.0, 0.0);  
15        glVertex2f(1.0, 1.0);  
16    glEnd();  
17  
18    glFlush();  
19 }
```

# Obrigado! Dúvidas?

Guilherme Henrique de Souza Nakahata

[guilhermenakahata@gmail.com](mailto:guilhermenakahata@gmail.com)

<https://github.com/GuilhermeNakahata/UNESPAR-2024>