OpenGL
Animação e Texturas

Renato Rodrigues Oliveira da Silva

Outubro 2013

Animação

- Animação Tradicional
  - Sequência de imagens em alta velocidade
  - Velocidade de exibição (*frame rate*) varia de acordo com a mídia utilizada
- Importante garantir que a aparência da imagem não mude muito rapidamente
  - *Aliasing* Temporal

Animação - OpenGL

- Redesenhar a cena continuamente
- Cada quadro deve ser ligeiramente diferente, criando a ilusão de movimento
  - Movimento dos objetos
  - Cor
  - Forma, etc.
- Em 3D pode-se também alterar a posição do observador

Animação - OpenGL

- É preciso forçar uma atualização contínua da tela
  - Necessário invocar função que é chamada em intervalos de tempo predeterminados
- Deve-se evitar que a imagem fique “piscando” com o contínuo redesenho
  - Utiliza-se 2 *buffers* de exibição:

glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB)

Animação - OpenGL

- Um dos *buffers* é preenchido enquanto o outro é exibido
  - São trocados pelo comando **glutSwapBuffers()**

Animação - OpenGL

- A GLUT oferece uma função *callback* que é invocada em um tempo predeterminado

```
void glutTimerFunc(int msecs,
                   void (*func)(int value),
                   int value)
```

Animação - OpenGL

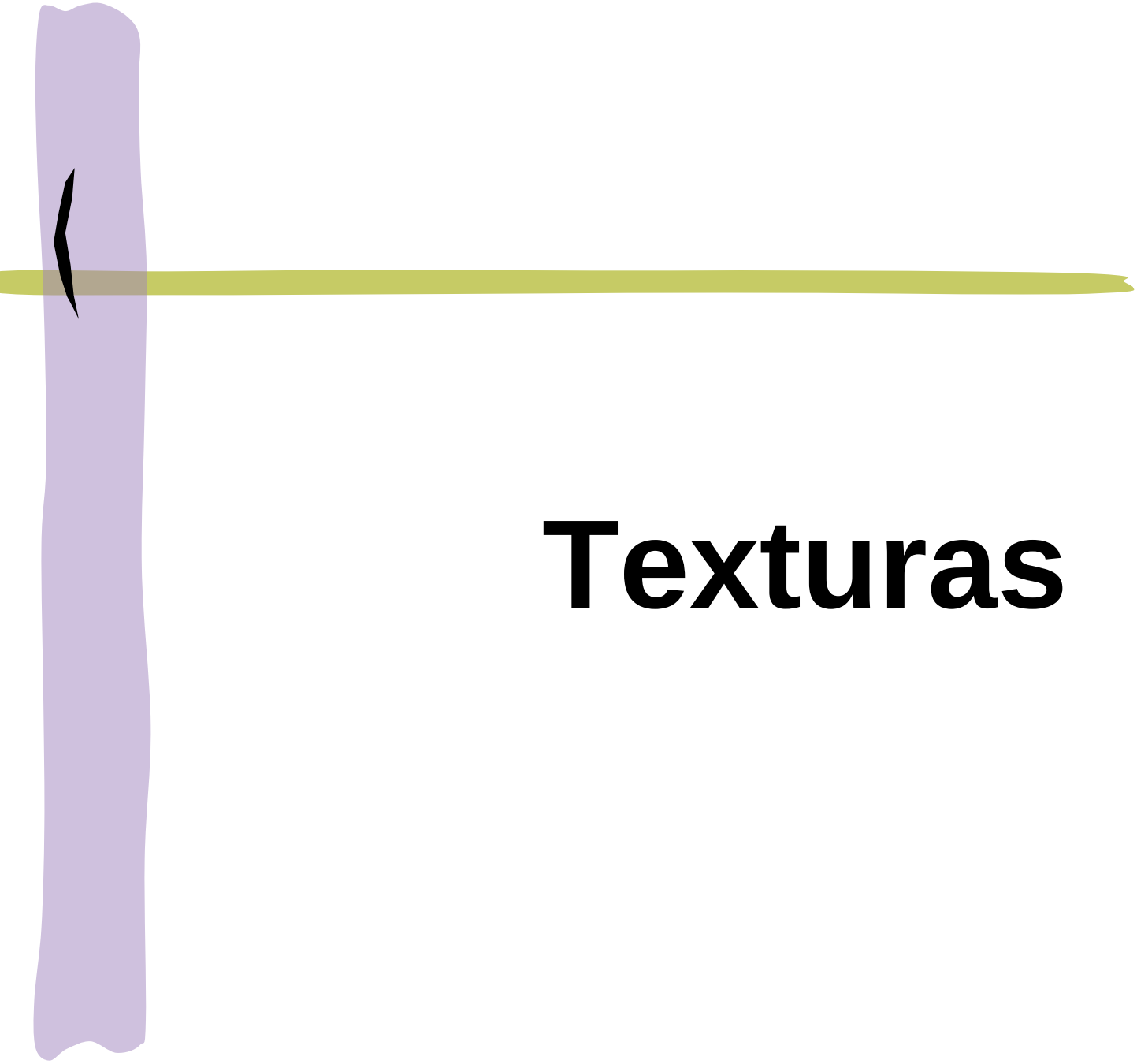
- void glutTimerFunc (int msecs,
  void *timer(int value),
  int value)
  - int msecs: Tempo em milisegundos
  - void *timer(int value): Função de animação
  - int value: Valor passado para a função timer

Animação - OpenGL

- Exemplo:
  - **04-01-ExemploPongOpenGL.c**
  - Corrigir o *aliasing* temporal
  - Utilizar dois buffers de exibição

Animação - OpenGL

- Exercício:
  - Alterar o código **04-02-Cubo.c**
  - Fazer uma animação para o cubo rotacionar em torno de um eixo



Texturas

Texturas

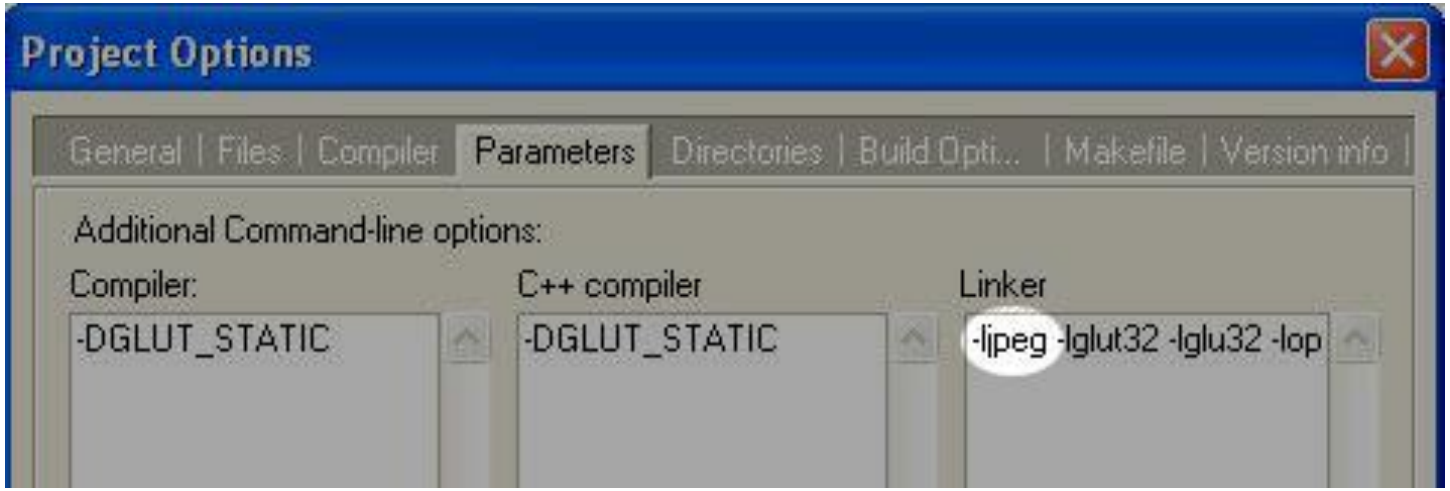
- O termo refere-se a uma imagem bidimensional que é “colada” à superfície de um objeto
- Podem ter 1, 2 ou 3 dimensões

Texturas

- É necessário utilizar uma biblioteca para manipular imagens
- O Dev-CPP possui a biblioteca **libjpeg** em seu repositório
  - Instalar essa biblioteca antes de manipular as texturas
  - Selecionar o pacote **libjpeg-9** no gerenciador de pacotes do Dev-CPP

Texturas

- Os projetos que usarem a libjpeg devem adicionar o parâmetro **-ljpeg** no linker
  - ALT+P → Aba Parameters → Linker



Texturas

- Inicialmente deve-se gerar uma identificação única para a textura

void glGenTextures(GLsizei n,
GLuint *textures)

- **n:** quantos identificadores devem ser gerados
- **textures:** vetor de inteiros que guardam o valor dos identificadores gerados

Texturas

- A seguir deve-se especificar qual é a textura corrente

void glBindTexture (GLenum target,
GLuint texture)

- **target**: indica o tipo de textura (para 2D usar GL_TEXTURE_2D)
- **texture:** identificador da textura

Texturas

- Especificar o modo de aplicação da textura corrente

```
void glTexEnvi(GLenum target,
               GLenum pname,
               Glint param);
```

  - **target:** GL_TEXTURE_ENV
  - **pname:** GL_TEXTURE_ENV_MODE
  - **param =** GL_MODULATE (com iluminação), GL_REPLACE (sem iluminação)

Texturas

- Pode-se utilizar a biblioteca **bibutil** para facilitar
- Possui uma estrutura para especificar os parâmetros das imagens
- Possui funções para aplicar os parâmetros

Texturas

- Estrutura TEX:

```
typedef struct {
    int ncomp; // numero de componentes
    int dimx; // largura
    int dimy; // altura
    Gluint texid; //identificação
    unsigned char *data; // conteúdo imagem
} TEX;
```

Texturas

- Função para carregar a imagem na memória

TEX* CarregaTextura(char *path,
boolean mipmaps);

  - **path:** caminho para o arquivo
  - **mipmaps:** indica se fará ou não o uso da técnica de mipmaps (qualidade)

Texturas

- Deve-se também ativar a variável de estado GL_TEXTURE_2D
  - glEnable(GL_TEXTURE_2D)
- O mapeamento é realizado pela associação das coordenadas de textura a cada vértice da superfície
  - De forma semelhante à associação de cores por meio de glColor3f

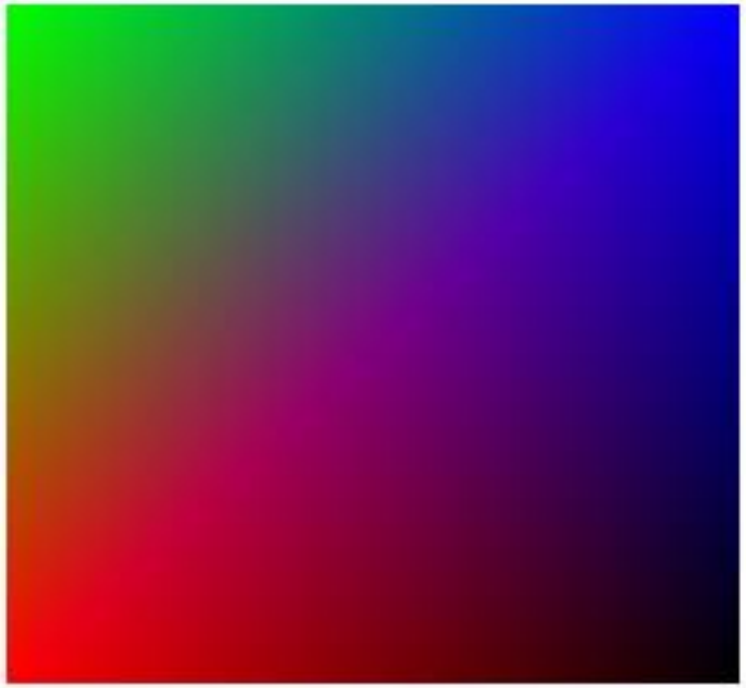
Texturas

- Mapeamento de quadrilátero com cores

```
glBegin(GL_QUADS)
   glColor3f(1.0f, 0.0f, 0.0f); //vermelho
   glVertex2f(-45.0f, -15.0f);
   glColor3f(0.0f, 1.0f, 0.0f); //verde
   glVertex2f(-45.0f, 15.0f);
   glColor3f(0.0f, 0.0f, 1.0f); //azul
   glVertex2f(-15.0f, 15.0f);
   glColor3f(0.0f, 0.0f, 0.0f); //preto
   glVertex2f(-15.0f, -15.0f);
glEnd();
```

Texturas

- Mapeamento de quadrilátero com cores



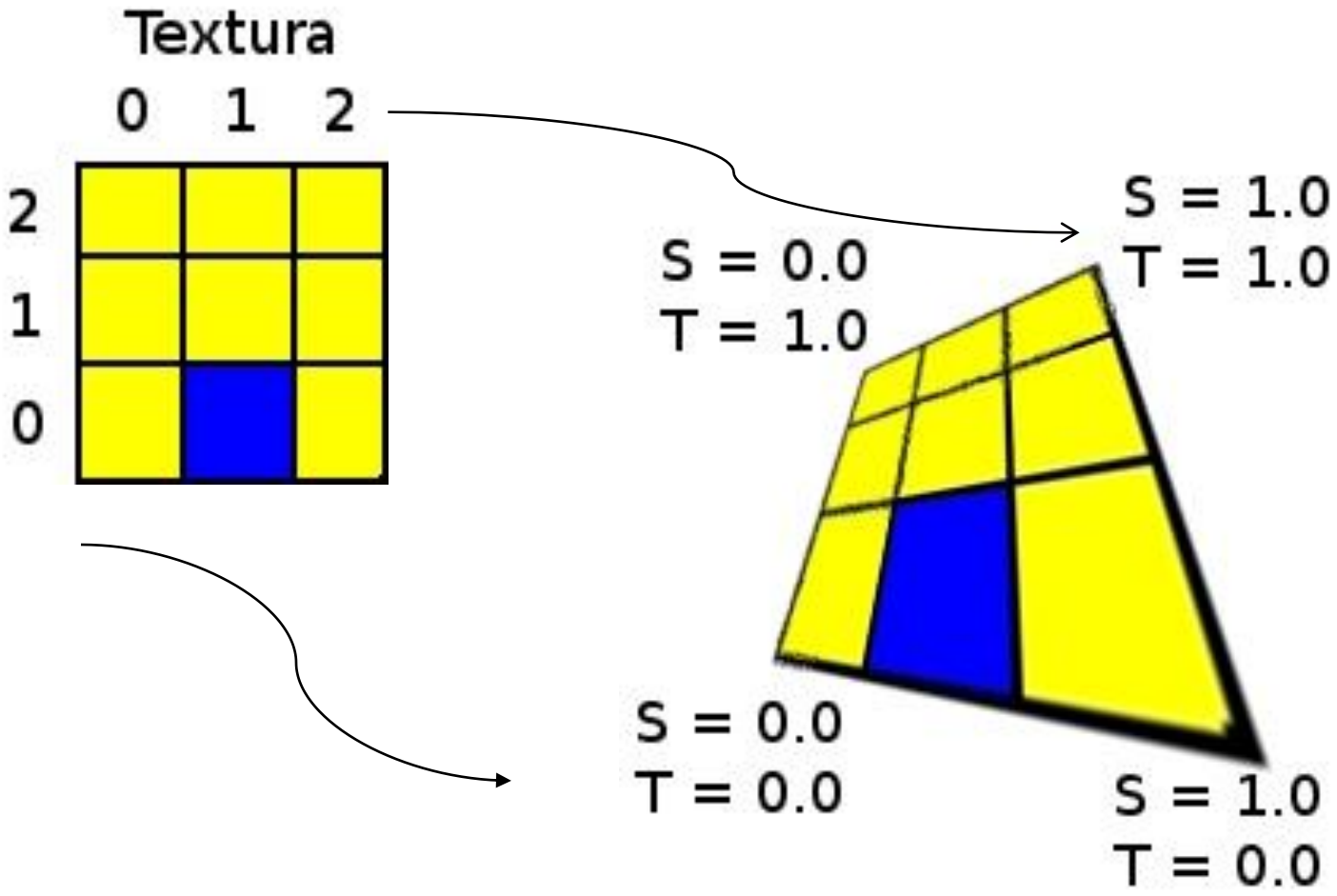
Texturas

- void glTexCoord2f (GLfloat s, GLfloat t)

// Coordenadas da textura a serem associadas com o vértice corrente

  - **s:** correspondente ao eixo X na imagem (varia de 0 a 1)
  - **t:** correspondente ao eixo Y na imagem (varia de 0 a 1)

Texturas



Texturas

- Mapeamento de quadrilátero com textura

```
glBegin(GL_QUADS);
    glTexCoord2f(0,0);
    glVertex3f(-1,-1,0);
    glTexCoord2f(1,0);
    glVertex3f(1,-1,0);
    glTexCoord2f(1,1);
    glVertex3f(1, 1,0);
    glTexCoord2f(0,1);
    glVertex3f(-1, 1,0);
glEnd();
```

Texturas

- Mapeamento de quadrilátero com textura

Texturas

- Exemplo:
  - **04-03-ExemploTextura2D.cpp**

Texturas

- Exercício:
  - Alterar **04-04-CuboColorido.cpp**
  - Utilizar textura na face frontal do cubo

- Resolução:
  - **04-05-Cubo3DTextura.cpp**

Bibliografia

- Cohen, M. , Manssour, I. H, OpenGL – Uma Abordagem Prática e Objetiva, 2006, Novatec Editora