

---

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS  
DEPARTAMENTO DE COMPUTAÇÃO  
COMPUTAÇÃO GRÁFICA CMP 1170 – 2019/1  
PROF. MSC. GUSTAVO VINHAL

# Aula 17

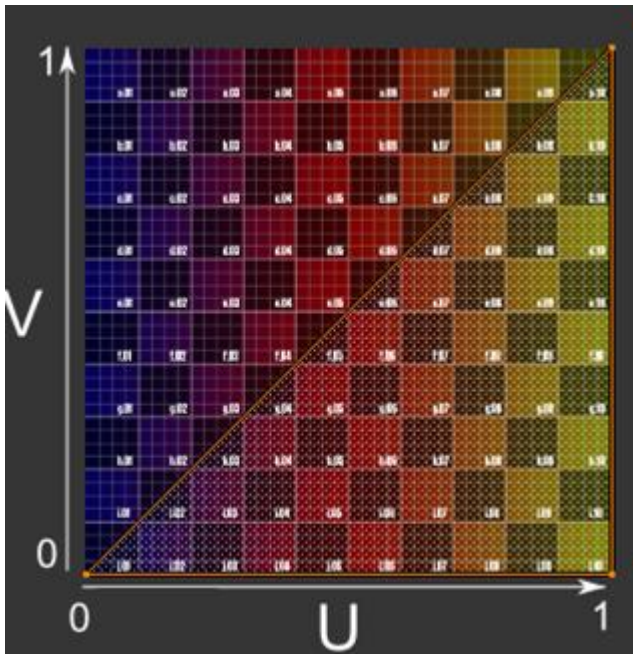
## Textura em OpenGL

## Texturas

- Quando se deseja um realismo maior nas cenas, utiliza-se imagens cobrindo a superfície de cada polígono.
- Essa imagem é chamada de textura.
- A textura é uma imagem 2D (podendo ser 3D) usada para adicionar detalhes em objetos.
  - Como se fosse um papel de parede.

## Texturas em OpenGL

- Os vértices de um polígono possui algumas propriedades, como:
  - Coordenadas de posicionamento no eixo de coordenadas (x,y,z)
  - Coordenadas da textura: também chamada de coordenadas UV (s,t,r)



(0,1)

(1,1)

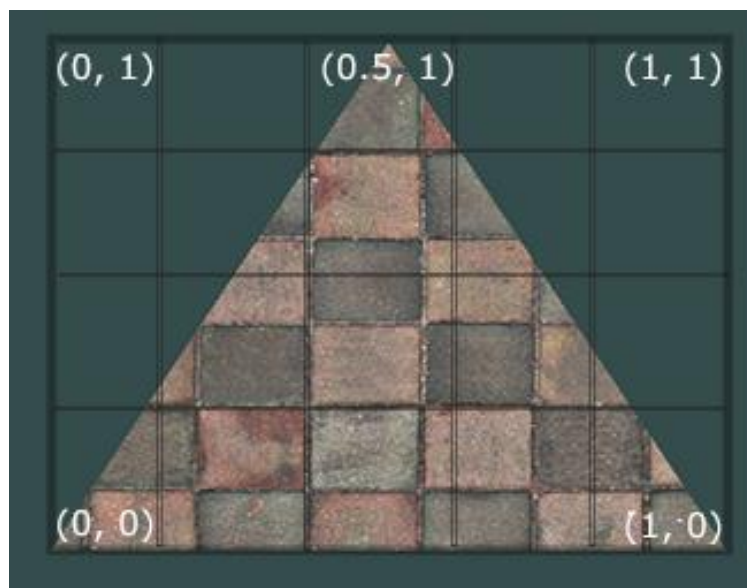
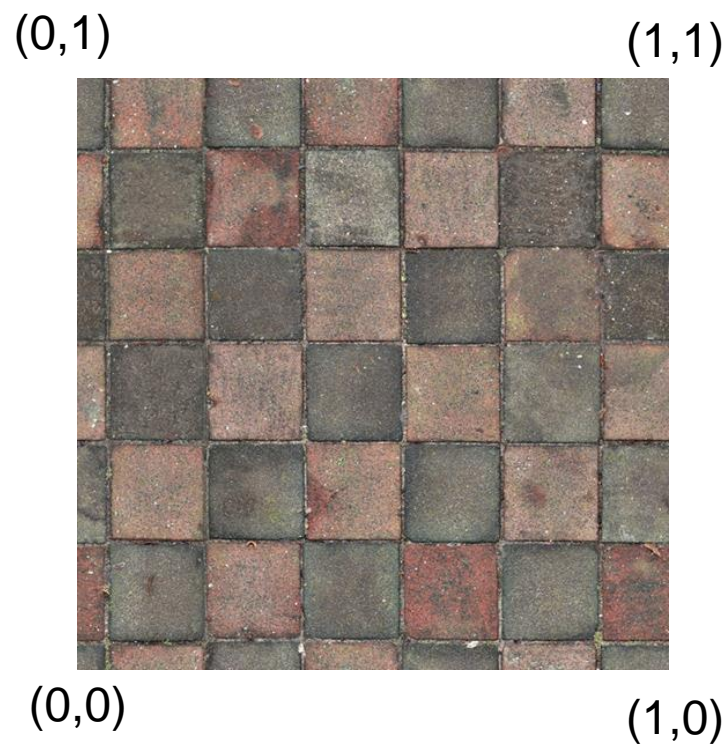


(0,0)

(1,0)

## Texturas em OpenGL

- Cada vértice do polígono deve possuir uma coordenada da textura.



## Textura em OpenGL

- A textura em OpenGL vem desabilitada por padrão. Para utilizá-la, deve-se executar o seguinte comando:

`glEnable(GL_TEXTURE_2D)`

- O comando anterior habilita a textura em 2D. Uma vez ativada, deve-se executar dois passos distintos:
  - Carga da textura;
  - Aplicação da textura;

## Textura em OpenGL

### Carga da textura

- Deve-se configurar a quantidade de texturas a serem utilizadas, assim como ativar a textura corrente e carregar a imagem do disco.

/\* Definir quantas texturas serão utilizadas no programa \*/

GLuint texture\_id[MAX\_NO\_TEXTURES]; // vetor com o número de ID das texturas

/\* Define quantas texturas serão geradas e sua respectiva textura (ID) \*/

/\* 1 = uma textura \*/

/\* texture\_id = vetor que guarda o número das texturas (ID) \*/

glGenTextures (1, texture\_id);

glBindTexture (GL\_TEXTURE\_2D, texture\_id[0]) ; // Definir a textura corrente

## Textura em OpenGL

### Carga da textura

- Deve-se configurar a quantidade de texturas a serem utilizadas, assim como ativar a textura corrente e carregar a imagem do disco.
  - Para leitura da imagem no disco, pode-se utilizar qualquer biblioteca para carregamento da imagem ou criar um código para isso.

```
#define STB_IMAGE_IMPLEMENTATION  
#include "stb_image.h"
```

```
int width, height, nrChannels;  
unsigned char *data = stbi_load("wall.jpg", &width, &height, &nrChannels, 0);
```

## Textura em OpenGL

### Carga da textura

- Deve-se configurar a quantidade de texturas a serem utilizadas, assim como ativar a textura corrente e carregar a imagem do disco.
  - Uma vez carregada a imagem, deve-se atribuí-la a textura criada:

```
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, width, height, 0, GL_RGB,  
             GL_UNSIGNED_BYTE, data);
```

- **GL\_TEXTURE\_2D**: especifica a textura alvo;
- **0**: especifica o nível mipmap;
- **GL\_RGB**: especifica o formato que se deseja na textura;
- **width**: largura da imagem que foi carregada;
- **height**: altura da imagem que foi carregada;
- **0**: sempre será 0 (implementação antiga do OpenGL);
- **GL\_RGB**: formato da imagem de origem (que está sendo guardada na textura);
- **GL\_UNSIGNED\_BYTE**: tipo dos dados da imagem de origem;
- **Data**: imagem de origem.



## Textura em OpenGL

### Aplicação da textura

- Uma vez carregada, a textura deve ser aplicada associando cada ponto dela com o vértice do polígono.

```
glBegin(GL_POLYGON); // Quadrado
    glTexCoord2d(1.0, 1.0); glVertex3d(-1.0, 8.0, 0.0); // superior direito
    glTexCoord2d(0.0, 1.0); glVertex3d(-8.0, 8.0, 0.0); // superior esquerdo
    glTexCoord2d(0.0, 0.0); glVertex3d(-8.0, 1.0, 0.0); // inferior esquerdo
    glTexCoord2d(1.0, 0.0); glVertex3d(-1.0, 1.0, 0.0); // inferior direito
glEnd();
```

## Textura em OpenGL

### Considerações

- Se, por algum motivo, as coordenadas UV (da textura) não coincidirem com os vértices do polígono (coordenadas fora dos limites (0,0) até (1,1)) deve-se estipular como o OpenGL tratará a textura. As opções disponíveis são:

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_MIRRORED_REPEAT);  
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_MIRRORED_REPEAT);
```



GL\_REPEAT



GL\_MIRRORED\_REPEAT



GL\_CLAMP\_TO\_EDGE



GL\_CLAMP\_TO\_BORDER

## Textura em OpenGL

### Considerações

- Dependendo do tamanho da textura é necessário realizar uma filtragem na imagem.

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);  
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
```

- **GL\_TEXTURE\_MIN\_FILTER**: filtro aplicado quando a textura for maior (ela vai ser reduzida);
- **GL\_TEXTURE\_MAG\_FILTER**: filtro aplicado quando a textura for menor (ela vai ser ampliada);

## Textura em OpenGL

### Considerações

- Dependendo do tamanho da textura é necessário realizar uma filtragem na imagem.

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);  
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
```

GL_NEAREST	GL_LINEAR
Vizinho mais próximo	Bilinear



## REFERÊNCIAS BIBLIOGRÁFICAS:

AZEVEDO, Eduardo; CONCI, Aura. **Computação gráfica: teoria e prática**. Rio de Janeiro: Campus, 2003.