Computer Graphics (MIEIC)

Practical Work 4

Applying textures

Goals

- texture coordinates set appropriately
- Explore the different texture of application modes.
- Combine the use of materials with textures for a realistic appearance

Practical work

Over the following points are described various tasks to accomplish. Some of them are noted

with the icon  (Image capture) .Nestes points should, with the program running, capturaruma image execution. Should nomearas images captured following the format **"CGFImage-TP4-TtGgg-xypng"** ,on what **TtGgg** refers to the class and group number and **x** and **y** match point and sub-point corresponding to the task (eg **"CGFImage-TP4-T3G10-2.4.png"** , or **"CGFImage-TP4-T2G08-extra.jpg"**).

The tasks marked with the icon  (Code) must create a .zip file of your project, and nomeálo as**"CGFCode-TP4-TtGgg-xyzip"**, (with **TtGgg,** **x** and **y** identifying the class, group and task as described above).

When the icon  arise, it is expected to execute the program and observe the results. At the end, should submit all files via Moodle through the link provided for this purpose.

They should also include a file***ident.txt***with the list of group members (name and number). Only a member of the group must submit the work.

Preparation Desktop

This work should be based on a copy of the previous work (a classroom with two flat, two tables, two walls, the floor, a cylinder and a prism, TP3 and TP2 respectively). They should descarregaro code available for this work and add Moodle the folder ***resources*** (Containing the images to use as textures) at the same level the folder***lib*** . Replace Also the file ***plane.js*** previously provided by provided in the new code.

IMPORTANT: Make sure you have the latest version of WebCGF (the version number appears on the console to open the page should be the same as is in the downloads page WebCGF)

1. Application of textures

The texture mapping is a way of assigning information stored in bitmap format different areas of 3D surfaces drawn. One of its most common uses is to map parts or all of an image with a geometry in order to add visual detail without aumentaro number of vertices and without adding complex geometry (other mapping include, for example,*bump mapping* and*normalmapping,* but that will not be explored in this job).

In the context of OpenGL / WebGL a two-dimensional texture can result load a bitmap image, and it is loaded to a buffer, which can later be accessed using two dimensions commonly identified as SET (Or in other contexts as u and v) and whose coordinates are normalized between 0 and 1 (see Fig. 1).

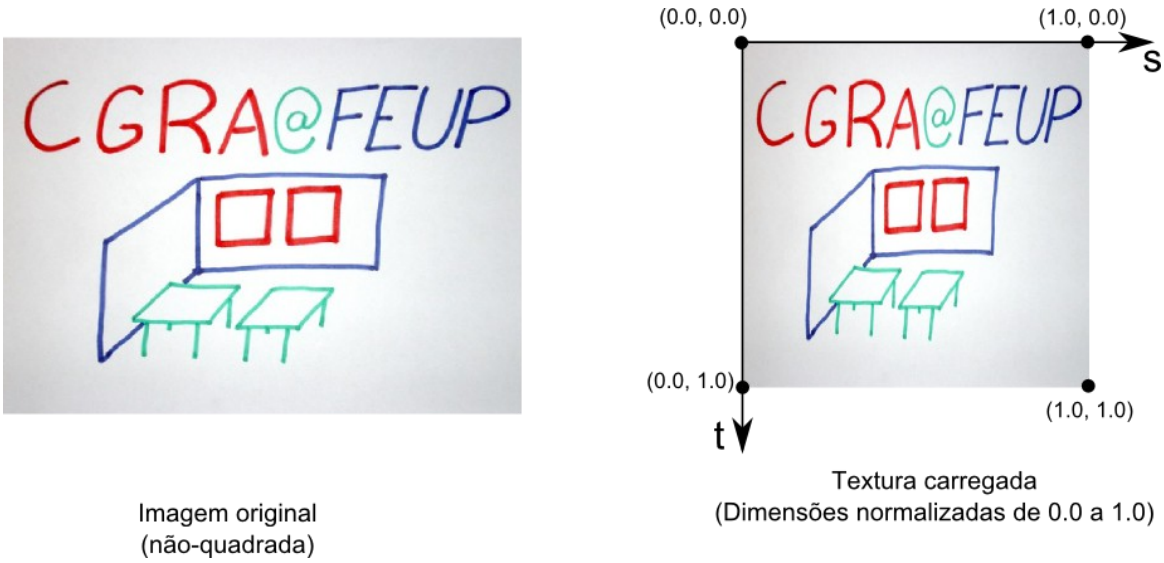

Figure 1: Image and corresponding texture charged

A preloaded texture may be applied to a given geometry in the most basic case a triangle making the mapping between the vertices of the geometry and the image points which are associated with them defining for each vertex one texture coordinate (see Fig. 2, a) and b)).

Conceptually, we can consider that we are defining the "image clipping" to be applied to the triangle in question, and if the "cut" does not have the same proportions of the original triangle, the image will be distorted accordingly (see Fig. 2, c) and d)).

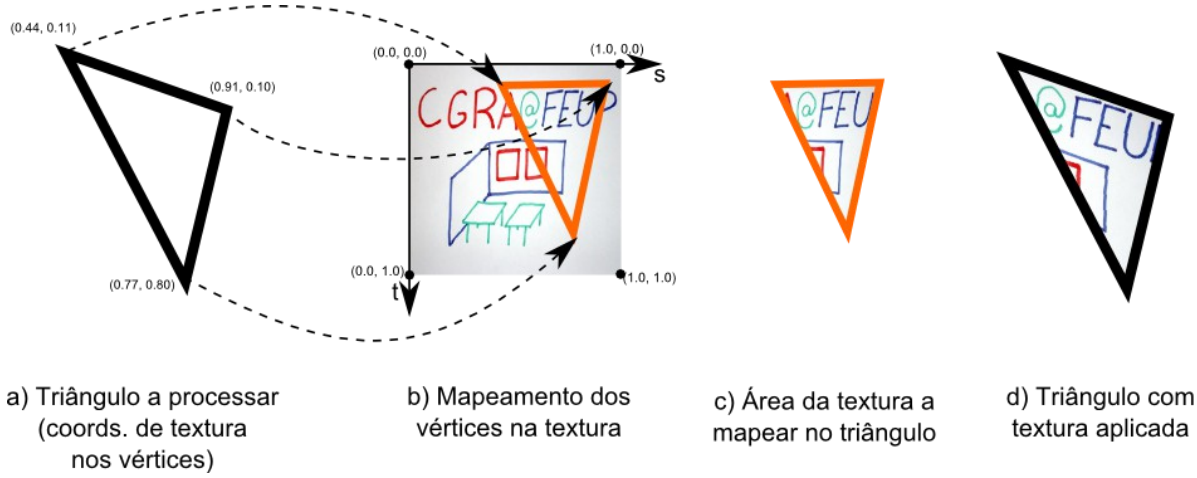

Figure 2: Mapping between triangle and texture defining texture coordinates per vertex.

In this first exercise is necessary to declare texture coordinates to objects in which want to apply textures, class criarobjetos ***CGFappearance*** that include textures, and aplicálos to other elements of the scene texturing.

1.Edite method ***initBuffers*** from class ***MyQuad*** In order to set texture coordinates for its four vertices. For this purpose should add, like he did for normal one *array* Texture coordinates (only two coordinates per vertex):

```
this.texCoords = [
        ... ... .
        .. ... ..
        . ... ...
        ...

];
```

With this, all objects that use ***MyQuad*** Textures may be applied (in particular ***MyUnitCubeQuad*** and by extension the ***MyTable***).

2.Na function init scene, enable the use of textures:

```
this.enableTextures (true);
```

3.Crie an object **tableAppearance** from class **CGFappearance** describing the material of the top Wooden tables (specular component little low brightness, strong diffuse component) assign it the texture **table.png** (use the method ***CGFappearance.loadTexture*** and keep in Note that the path to the respective file is relative to the folder where the file **index.html**). Apply this material to the top of both tables.

(1.3) (1.3)

2. Modes *wrapping* Textures

In the example exploited until now, the texture coordinates associated with each vertex are the normalized range of 0.0 to 1.0. However, you can indicate values outside this range, when we intend, for instance, have several replicates in a polygon of the same image or map the entire image only in a part of the polygon. The way the values outside the range [0..1] are used in the implementation of a texture is controlled by setting the mode ***wrapping***. The methods of *wrapping*Supported vary slightly between OpenGL versions, in the case of modes WebGL 1.0 possíveissão **'REPEAT'**,**'CLAMP_TO_EDGE'** and **'MIRRORED_REPEAT'**. In Figure 3 are illustrated some examples of how to manipulate the texture coordinates of each mode to obtain different effects. Note that the mode*wrapping* It can be different in the two dimensions ***s*** and ***t***

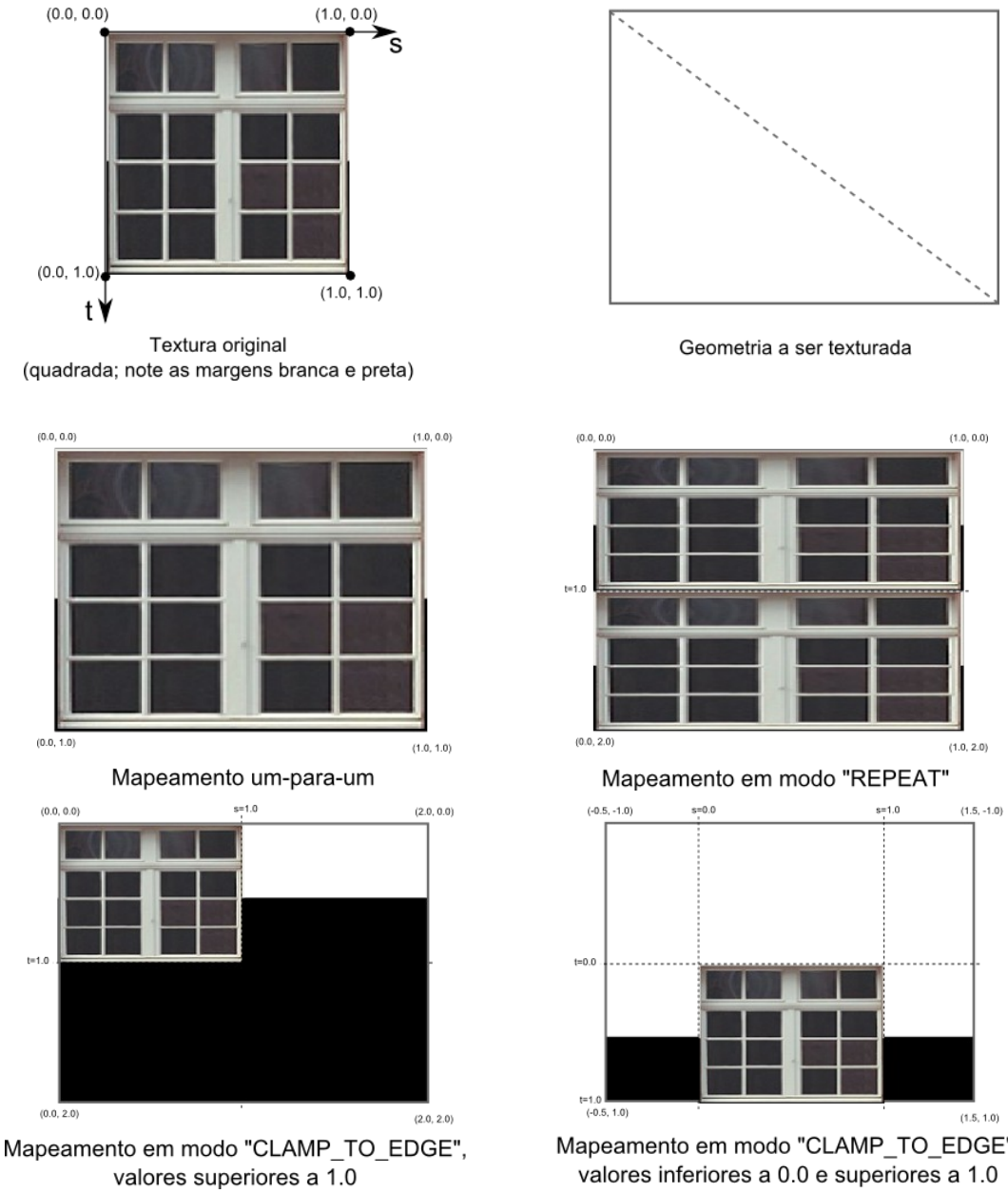

Figure 3: Application of a texture using repetition or *clamping* .

Note how the clamping of the edges of the image are extended along the outside of the range coordinate areas [0..1]

if 1.Pretende that change the constructor of the class MyQuad to allow indicate how parameters the values ***Mins, MaxS, minT, Maxton*** (Minimum and maximum ***s*** it's from ***t***) So that it can parameterize the texture coordinates. For example, the last case of Figure 3 could be obtained by:

```
new MyQuad (scene, 0.5, 1.5, 1.0, 1.0);
```

2.Crie the CGFappearance **floorAppearance** with the texture **floor.png** . Apply it to the floor so this is repeated 10 times in the direction ***s*** and 12 times in the direction ***t*** (Note: You may have to adjust initialization of ***MyQuad*** in ***floor*** for this effect)

3. Replace the left wall for ***MyQuad*** instead of ***Plane*** . Create CGFappearance **windowAppearance** with the texture **window.png.** Apply to left in the wall so that the window to appear only once, mascentrada in height and width, using the mode '**CLAMP_TO_EDGE** '(consulteadocumentaçãoda***CGFappearance.setTextureWrap***). Add a light source in the window position.

(2.3 ) (2.3 ) 

3. Partitioning of a texture by various primitives

The flexibility in assigning texture coordinates to manipulate different ways to applying textures. For example, if we have a surface consisting of multiple polygons on qualqueremos to apply a single texture, as the plane used for tables, we can do it distributing the texture coordinates appropriately for their vertices, as shown in Fig. 4.

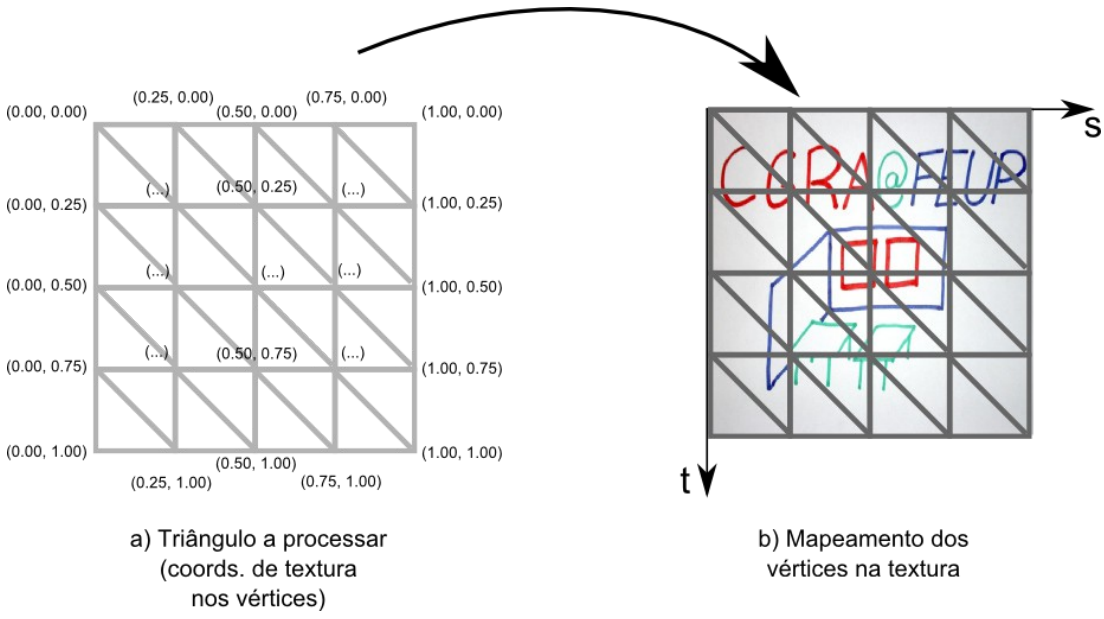

Figure 4: Applying a texture on various primitive

1. Edite class ***Plane*** provided with the code of this work, in order to add texture coordinates for vertices form similar to that shown in Fig. 4.

2. Crie now two materials: **slidesAppearance** and **boardAppearance**. The first will describe a projected screen (represented by texture **slides.png**) With little specular component and low brightness, strong diffuse component. The second will be a whiteboard (represented by texture **board.png**) With some specular component, enough brightness and a lower diffuse component. Apply the material on the frames (viewed from the front, the left will be the projection screen).

3. Note that the texture has a different proportion of the frame. Correct this level the application of texture, and for this purpose to pass additional parameters to the constructor Plane class to allow to change the ratio of texture.

(3.3) (3.3)

Extra: Complete the scene

Complete the scene with more textures and materials. Create columns with the cylinder made in TP3 and apply a suitable texture.

(extra) (Extra)

Check list

Until finaldo work must submit the following images and versions of code via Moodle, strictly respecting the rule of names, and the ident.txt file with the identification group members:

- Pictures (4): 1.3 2.3 3.3 extra (Nomesdotipo " CGFImage-TP4-TtGgg-xypng ")
- Códigoem arquivozip (4): 1.3,2.3,3.3, extra (nomesdotipo "CGFCode-TP4-TtGgg-xyzip")