Texture Mapping

Texture mapping for a GL quad object. In this exercise, we are going to use the WINAPI functions to load a texture from file to the memory. Then we can create a texture object in OpenGL.

1. Step 1: Variable declaration

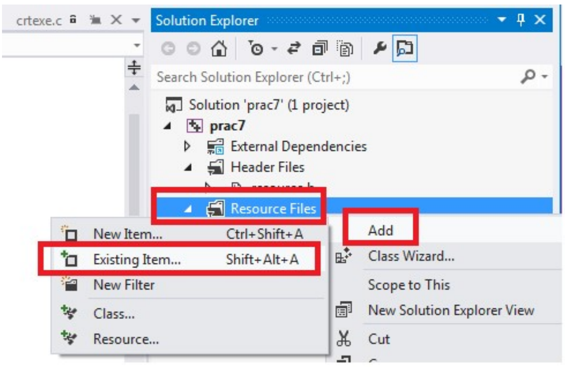
a)
```
GLuint texture=0;
```
- Define a texture name we are going to use for texture mapping. In OpenGL, any non-zero unsigned integer may be used as a texture name.

b)
```
BITMAP BMP;
```
- The BITMAP structure defines the properties of a bitmap texture.

c)
```
HBITMAP hBMP = NULL;
```
- A handle to a bitmap.

2. Step 2: Add picture resource in project

a. Right click a resource file, add-> Existing item



b.Choose the image file which require in project.

3. Step 3: Initialize texture info

a)
```
glPixelStorei(GL_UNPACK_ALIGNMENT, 4);
```
- This function is used to change how pixel data is stored and retrieved.
- GL_UNPACK_ALIGNMENT specifies how OpenGL unpacks image data from data buffers.
- A bitmap file format uses 4-byte alignment for its pixel data.

b)
```
HBITMAP hBMP = (HBITMAP)LoadImage(GetModuleHandle(NULL),
"fileName", IMAGE_BITMAP, 0, 0, LR_CREATEDIBSECTION |
LR_LOADFROMFILE);
```
- This function is going to load a bitmap file to a WINAPI `BITMAP` handle.

c)
```
GetObject(hBMP, sizeof(BMP), &BMP);
```
- GetObject will load bitmap data from the handle into the BITMAP structure.

4. Step 4: Assign texture to polygon.

a) `glEnable(GL_TEXTURE_2D);`

- This function will enable 2D texture mapping for OpenGL.

b) `glGenTextures(1, &texture);`

- This function will return 1 currently unused names for the given texture object.

c) `glBindTexture(GL_TEXTURE_2D, texture);`

- When the texture name is used for the first time, this function will create a new texture object and assigned with the given name.
- This function is also called when we want to bind a previously created texture to an object.

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

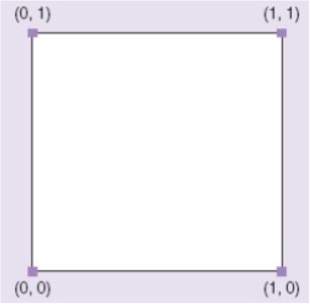
- This function is used to specify the magnification and minification filtering methods.

e)
```
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, BMP.bmWidth,
BMP.bmHeight, 0, GL_BGR_EXT, GL_UNSIGNED_BYTE, BMP.bmBits);
```

- This function is used to define a two dimensional texture.

f) `glTexCoord2f(0.0f, 0.0f);`

- This function specifies the texture coordinates for a vertex. You are to specify the texture coordinates before specifying the vertices. Unlike Direct3D, OpenGL use the following texture coordinate system,



- You are to specify the texture coordinates for each of your quad vertices.

5. Step5: Remove texture info.

a) `glDisable(GL_TEXTURE_2D);`

- This function will disable 2D texture mapping for OpenGL.

b) `DeleteObject(hBMP);`

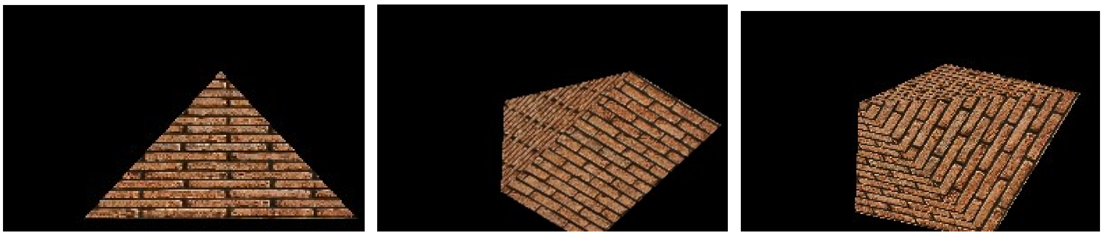
- Delete the bitmap handle once you create the texture object.

c) `glDeleteTextures(1, &texture);`

- This function is called to free the texture name.

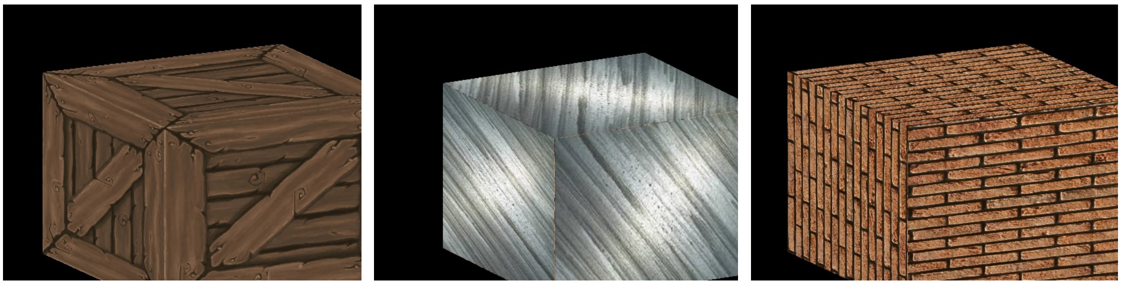
Practical Exercise 7

Q1. Create a 3D pyramid with brick texture. Application should provide interactive features which allow user to rotate at all axes. (4 marks)



Criteria	Marks
Pyramid model	1
Completeness	1
Brick texture	1
Rotation (all axes)	1
TOTAL	**4**

Q2. Create a cube which allow user to siwtch the texture to wood, metal or brick. Application should provide interactive features which allow user to rotate at all axes. (5 marks)



Criteria	Marks
Switch texture	3
Completeness	1
Rotation (all axes)	1
TOTAL	**5**