

OpenGL Course Summer 2013

IT ENGINEERING

4TH YEAR

3D Models

2

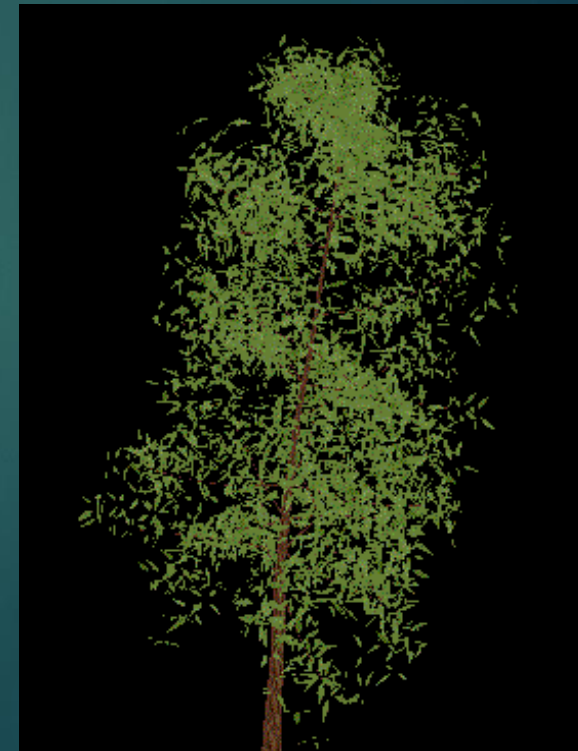
3D Models

3D Models

- ▶ As we know that OpenGL is a 3d graphics library and it can draw 3d objects using primitives as we've studied before.
- ▶ But sometimes we want to use a ready 3d model that's created on some 3d models' editing program (like autodesk 3ds max or autodesk maya).
- ▶ In this course we'll study how to load a **3ds** model, which is one of the most common extensions of 3d objects. It can be edited using 3ds max program or any other editor.
- ▶ **The loader we'll use is called 'Model_3DS.h'**

3D Models

- ▶ In order to perform the operation we need to prepare or bring a ready 3ds file for the object we want to use.
- ▶ Let's try to load the following 3ds shape:



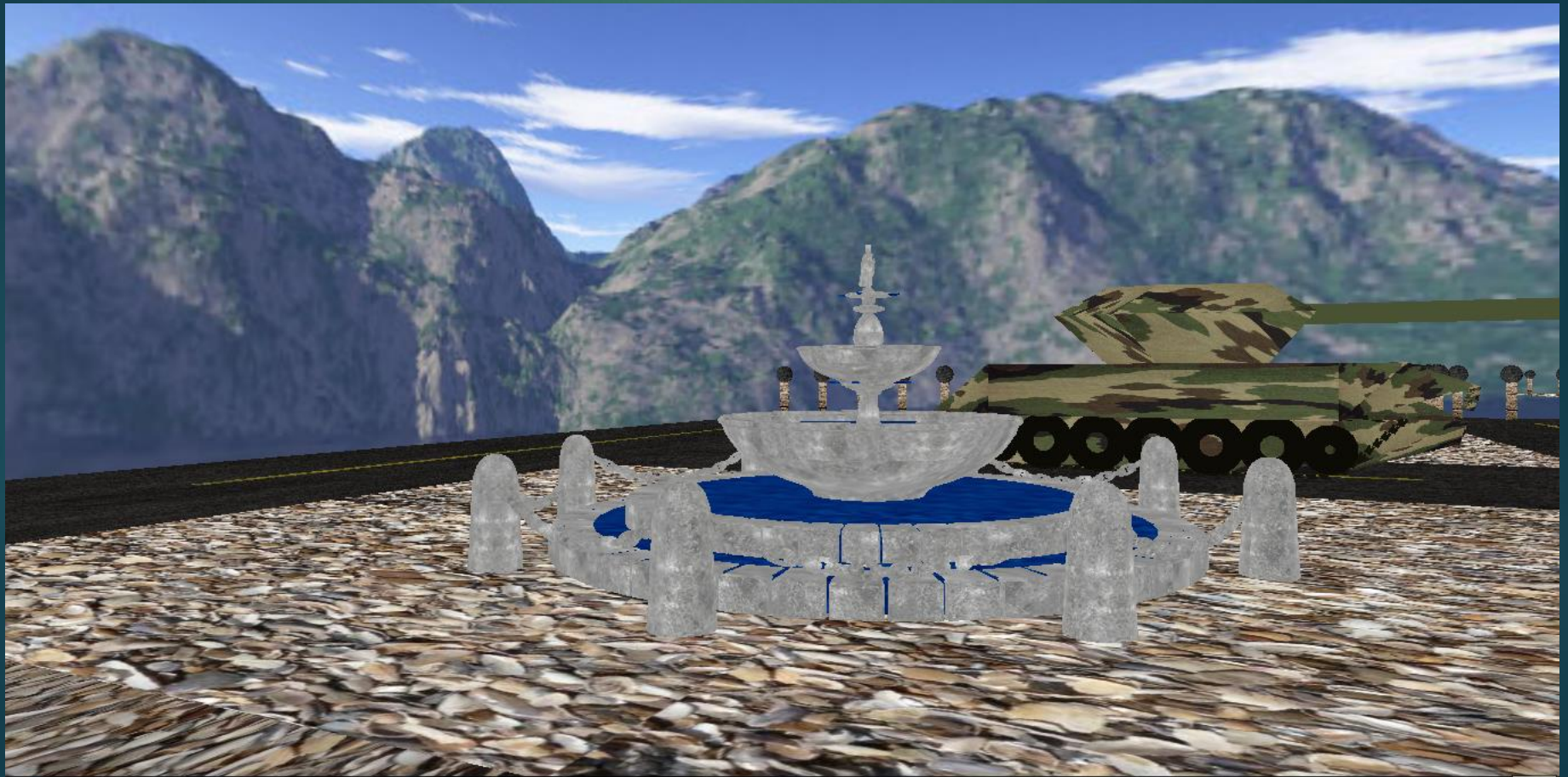
3D Models

5



3D Models

6



3D Models

7

The Steps

3D Models

8

- ▶ We put the texture image in an integer variable as we've seen last lesson.
- ▶ But 3ds object will be stored in a special object from the type 'Model_3DS*', which is a pointer to a 3ds object. So before the InitGL function we'll declare for example:

```
Model_3DS *tree;
```

- ▶ Now we must load the 3ds object from the directory of the project (inside InitGL function):

```
tree=new Model_3DS();  
tree->Load("M_TREE5.3DS");
```

- ▶ Where "M_TREE5.3DS" is a 3ds object inside the main directory of the project

3D Models

9

- ▶ After loading the object we must draw it somewhere in the scene.

```
tree->pos.x=0;  
tree->pos.y=0;  
tree->pos.z=0;  
tree->scale=0.1;  
tree->Draw();
```

- ▶ The previous variables are data members and methods inside the class Model_3DS ☺
- ▶ Now if you try to run the program you will see that the object is white! With no colors nor texture. We must bind texture to it using another texture loader which can deal with this model 3ds loader.

3D Models

10

- ▶ To bind texture on the 3ds object we must include the loader '**gltexture.h**'. This loader is not the same one that we'd used in the lesson of texture 😊
- ▶ Then inside InitGL function we'll declare:

```
GLTexture BARK,Leaf;  
Leaf.LoadBMP("Leaf.bmp");  
BARK.LoadBMP("BARK.bmp");
```

3D Models

11

- ▶ Where every one of the previous textures is a bmp image in the main directory of the project.
- ▶ Then we'll add the following lines to bind the textures on the model **after the initialization of the model** in the InitGL:

```
tree->Materials[0].tex=BARK;  
tree->Materials[1].tex=Leaf;
```

3D Models

12

- ▶ The previous lines bind the textures we've loaded on the 3ds model. But we must know that not all the 3ds models can contain materials. It must be prepared on the 3ds max program or maybe the object is already prepared from someone before.
- ▶ To know the order of materials in the materials array, we must try that by the hand. No rule can be used to specify that.
- ▶ Now the object is ready to be drawn.

3D Models

13

