

TREK SOFTWARE

Tutorial 6 : Objects

In the last tutorial, I focused on the fundamentals of 3D. In this one, I'm going to specifically focus on 3D objects – what they are, how you make/get them, and some of the things you can do with them.

To get started, here is a screenshot from my game, Codename Countdown:



Everything you see in the picture is a 3D object except the green text. The spaceship is a 3D object, the boxes are 3D objects, the walls and floor are 3D objects, even the dials and HUD (Heads Up Display) are 3D objects even though they look two dimensional.

So how did I make all of them? All of the items in the above screenshot except the spaceship are simply primitive shapes (cubes, cylinders, cones, plains, or spheres) textured with an image. Basically what that means is that I told DarkBASIC to take a

shape (e.g. one one of those cubes) and paint it with some picture (e.g. a metal graphic). I also made parts of some of those objects transparent but that's a more advanced topic which I won't cover here. Here's an example of how you can apply a graphic to an object:

```
sync on
sync rate 30
autocam off

load image "Metal.jpg",1

make object cube 1,2
position object 1,0,0,5
yrotate object 1,45
texture object 1,1

position camera 0,5,0
point camera 0,0,5

do
    sync
loop
```

Most of the above code I already covered in tutorial 5. The only new commands are "load image" and "texture object". Both do what their natural meaning indicates. Load image loads the picture in quotes and gives it the id number which I assigned after the comma. So in this case, it loads up the picture "Metal.jpg" and gives it an id number of 1. Then, to texture the cube with that image, you can simply use the texture object command followed first by the number of the object you want to texture, then a comma, and then the number of the picture you want to texture it with.

DarkBASIC makes things like this really easy. One important thing to know about this is that the code will look for the picture file starting in the folder the code itself is in. So for this code to work, I would need to put the picture "Metal.jpg" in the same folder as my code. Alternatively, I could put it in a folder called "Graphics" inside my folder and load it by using the following command: `load image "Graphics/Metal.jpg",1` The word "Graphics" followed by / indicates that Metal.jpg is in a sub-folder called Graphics. The .jpg is used to denote the kind of

image it is. You probably won't see that in the name of the file on your computer since those "extensions" are usually hidden. Instead, the way you can find out what the extension is, is by right-clicking on the image file on your computer, then clicking properties, and then looking at the place where it says, "Type of file".

Now I've covered how to texture boxes like in the above screenshot but what about the spaceship? The spaceship in this case was a 3D file. Just like there are music files, picture files, text files, etc. there are also 3D files that contain 3D objects. The major formats that DarkBASIC Professional supports are .X (DirectX) and .3DS (3D Studio). It also supports MDL, MD2, and MD3 formats. To use one of these instead of a primitive object, you can load it, similar to how you loaded the image:

```
sync on
sync rate 30
autocam off

load object "trumpet.x",1
position object 1,0,0,20

position camera 0,0,0
point camera 0,0,10

do
    sync
loop
```

If you go to My Computer → C: → Program Files → DarkBASIC Professional Online → Media → Models you might see a folder called *trumpet*. Copy both files inside that folder into the same folder as your code to get the above example to work. If you run the code, you should see a funny looking funnel kind of shape. This is because the trumpet you loaded is too big. The solution is to scale it down. To do this, add the following line of code after the `position object` command : `scale object 1,5,5,5`

This scales object 1 to 5% of it's original X dimension, 5% of it's original Y dimension, and 5% of it's original Z dimension. Now run it and see what happens.

It still doesn't look much like a trumpet does it? That's because it's being viewed head on. To fix this, rotate it 90 degrees on its Y-axis. Remember how to do this from the last tutorial : `yrotate object 1,90`

Whoops... now it's over on the right side of the screen. To fix that, just change the X position to -10. Finally, it's all set. This was quite a bit of work going back and forth tweaking it a little bit each time. There is a way you can speed this up somewhat. When you hit the "Esc" key while the program is running, you'll probably notice a little window comes up with a place for you to type. This is the debugger. It let's you type in code and see what happens as you enter it. Go ahead and run the above code sample and then once it comes up, press Esc. Then type the following command in the debugger window, "Rotate object 1,0,0,0" and hit enter. You should see the trumpet go back to the rotation it originally was set to. This is how you can tweak your object's position and rotation without having to continually exit out and recompile it. As you find the correct values, you might want to write them down so that when you are done, you can actually change the code with the new values (because any changes you make in the debugger are lost when you exit).

One thing you might run into is a situation where after running your code, and then exiting by clicking on the DarkBASIC Professional editor and clicking the "Cancel" button, you can no longer run your code without an error. The way to fix this is by clicking on the debugger window still left on the taskbar, and closing it out.

To get some experience with objects, textures, the scale and rotate commands, and the debugger, I recommend getting some of the objects and textures that come with DarkBASIC (or the DarkMatter collection if you have it) and use them to make a 3D room. It may take awhile, but it'll help you familiarize yourself with those commands and using the debugger effectively.