



Tutorial 5 : 3D

In this tutorial, I'll introduce you to the initial concepts necessary to start programming 3D games. If you haven't read the first tutorials yet, I highly recommend you go through them first.

In DarkBASIC, there is an imaginary 3D World that stretches out infinitely in the X, Y and Z directions. What are X, Y, and Z? Imagine putting your finger in the middle of your computer screen. Assume this is coordinate (0,0,0) – coordinates are listed in the format (X,Y,Z). Now move your finger to the right. As you do this, the X-dimension increases. So if you moved it all the way to the right edge, depending on what our units were, we could say you're finger was at (5,0,0). Now suppose you move it to the left. As you do, the X-dimension will decrease. Once you go past the middle of the screen towards the left side, it becomes negative. Similarly, if you put your finger in the middle of the screen and go up, that means you're increasing your finger's Y-dimension and all the way at the top, your finger's coordinates might be (0,7,0). Likewise, moving it down decreases the Y-dimension.

To visualize the Z-dimension, put your finger in the middle of the screen and bring it back, off the screen, towards your face. In doing so, you're increasing the Z-dimension. All the way to your nose might be (0,0,10). If you could move it through the screen, it would be decreasing in the Z-Dimension. Using these three dimensions, we can tell the computer where to put things in our huge 3D World. I'll start by describing some basic setup commands. Here's a basic 3D program:

```
Sync on
```

```
Sync rate 30
```

```
Autocam off
```

```
Position camera 0,0,0
```

```
Point camera 0,0,5
```

```
Make object cube 1,3
```

```
Position object 1,0,0,5
```

```
do
    sync
loop
```

The first command (Sync on) tells the computer to let you manually decide when it draws the image to the screen. What happens is that the computer will setup everything in the 3D World just like you tell it to but then it has to sync that information to the monitor or you won't see it. When we set manual sync on, it will only refresh when we call the "sync" command which is better than automatic refresh for reasons which I might cover later.

The second line tells it how many frames to show in a second. The reason for this is to define how fast we want the game to run. If we didn't do this, the game would run very fast on new computers and very slow on old computers. This sets a limit to how fast it can go. Of course, some older computers may not be able to run that fast and so it won't have an effect on them. 30 is a reasonable rate for most games that will have fairly smooth action without running too slowly on older computers, in general.

The Autocam off command tells the computer not to automatically position the camera to where it thinks it should be to look at the latest object. It's often better to control the camera ourselves. This brings up the question of what the camera is. Basically, the camera is your view in the imaginary 3D world. Whatever the camera sees is what will be displayed on the screen.

The first command puts the camera at 0,0,0 (the middle of the world) and then points it at 0,0,5. That would be like putting a digital camera on your paper at 0,0,0 and pointing it to look at a space 5 units in front of it. Now that the camera is looking at that space, it would be nice to make something to put there.

You can probably see even before running the code that that 8th line makes a cube object. The first number listed after the command is an id given to that cube. All objects are given ids so you can specify which object you'd like to move, change, etc. later on. The second number specifies the size of the cube. Since it is 5 units away from the camera, 3 is a reasonable size to make a cube that would fill a large part of the screen.

I then positioned the cube on the next line using the position object command. The first number is the id of the object I want to position (in this case, 1 since that's the id number for the cube). The second number is the X coordinate. Since I want it to

be in the middle of the screen, I set it's horizontal position to the same as that of the camera. The next number is for the Y coordinate and I also set that to 0 because likewise, I want it's vertical position to be the same as that of the camera. Finally, I set the Z position to 5 so that it would be a little ways in front of the camera.

Then at the bottom, I told it to just continually refresh the screen. You can go ahead and run the code if you'd like.

If you just ran it, you're probably a little surprised. You didn't see any cube on the screen – just a big square. Here's why that is. Imagine you have a camera and you put a block in front of the very middle of the lens. What would the camera see? It would only see the side of the cube facing it. There are a number of ways to fix this to see the 3D aspect of it.

One thing would be to move the camera up a little bit. Another thing we could do if we wanted to see the sides of the cube would be to rotate it a bit. Here, I do both of those things.

```
Sync on
Sync rate 30
Autocam off

Position camera 0,3,0
Point camera 0,0,5

Make object cube 1,3
Position object 1,0,0,5
Yrotate object 1,45

do
    sync
loop
```

Up at the top, you can see how I raised the camera. I just increased it's y-dimension to 3 instead of 0. You should also be able to see where I rotated the cube. The reason I used Yrotate instead of just rotate was to specify the axis which it should rotate on. Imagine three sticks coming out of the block. One is sticking straight up (that's the Y axis), one's sticking out the side (that's the X axis), and one is sticking out in front (that's the Z axis). To figure out which way it'll rotate the object, imagine turning one of those sticks. Since I wanted to turn the sides to face different angles,

I spin the one sticking out the top (the Y axis). You can also try rotating it on some of the other axis to see what it does.

There is much more to 3D Graphics which will be covered in future tutorials but this introduces the basic idea of what 3D is and how to use it. Here are some exercises you can try to get familiar with the 3D world, the camera, and objects.

- Try making a program that slowly spins a cube (Use a do-loop and a variable that is incremented by 2 every time the computer goes through the loop. Then each loop, be sure to YRotate the cube by that variable.)
- Make a cone instead of a cube. You can also try a sphere and a cylinder as well.
- Make a program where the camera pans by a cube, then a cone, then a sphere.
- Try making a snowman by stacking three spheres of decreasing size (3,2,1 for example) on top of each other. (Hint: Try slowly adjusting the y coordinates to get it looking just right).