Torque Game Builder – Fish Demo Tutorial - Part 5

5. Making our Fish Swim Back and Forth

5.1 Setting up our fish's world limit

Now that our fish is swimming in one direction, we need it to swim from one side of the level to the other. We can accomplish this easily using our fish's *world limit*. The world limit is a bounding box that we can define visually in our *Level Builder*. This is the world bounds for our fish object. We can also make it send a *script callback* to our fish class when it reaches the edge of its world limit. A script callback says to the engine, "call this script whenever this event happens". (This is different from a script execution, which is done to tell TGB where scripts we need are located.) By attaching a script callback to the world limit, we tell our fish to swim back and forth.

First we must define the fish's world limit in the *Level Builder*. Open up *TGB,* and you should be presented with your level. Select the fish and hover your mouse over it, and you should see a set of icons appear above it. Click on the icon second from the right. This is the *World Limit Tool (as shown in Figure 5.1.1)*.

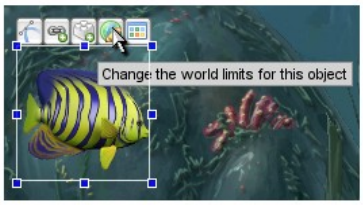

Figure 5.1.1

Once you select it your view should be zoomed far out, and you should have a dragable box appear *(as shown in Figure 5.1.2)*. This box represents the world limit of your fish. Right now the limit is far beyond where we want our fish to move. We want to bring the left and right sides just beyond our level. We want to leave the limits out just enough, so our fish can leave view before we change its direction. If you need to, scroll-zoom in to get a better view. Bring the top and bottom world limits closer to the level, as well *(as shown in Figure 5.1.3)*.

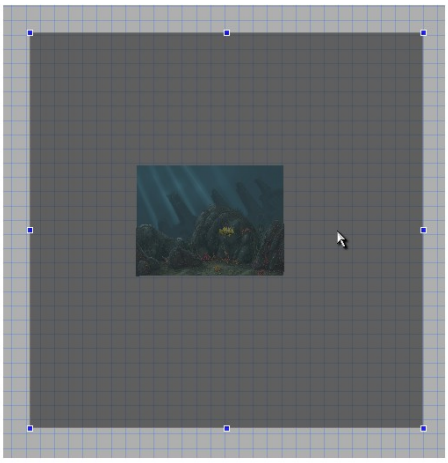

Figure 5.1.2

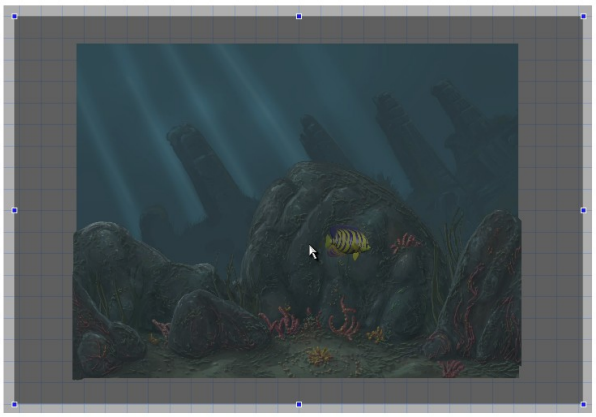

Figure 5.1.3

Now that the world limit boundary is set, we need to configure the settings for it. To exit out of the *World Limit Boundary Tool*, click on the *Selection Tool* in the top toolbar *(as shown in Figure 5.1.4)*.

Figure 5.1.4

Torque Game Builder – Fish Demo Tutorial - Part 5

Now that we are out, click the *Edit* tab to look at our fish's settings again. This time we are going to expand the world limit settings by clicking the *World Limits* label *(as shown in Figure 5.1.5)*. The default world limit is set to "OFF". Click the dropdown and change it to "NULL" *(as shown in Figure 5.1.6)*. As you can see, there are a few other options such as BOUNCE, CLAMP, etc. In our case we don't want any physics response, and we want to control the response by script, so NULL works perfectly. We also need to check the *Callback* box or no script will be called when the fish reaches the world limit. (*See Figure 5.1.7.*)

Figure 5.1.5

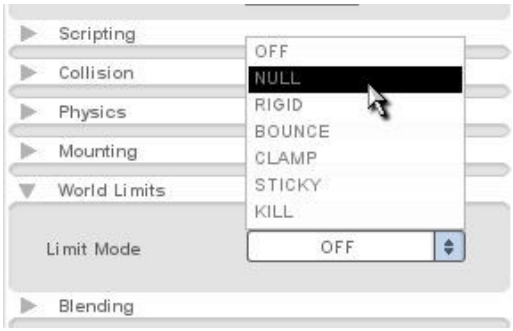

Figure 5.1.6

Figure 5.1.7

5.2 Scripting our world limit response

Now that our world limit is properly set up in the *Level Builder*, we can add the script to set the proper response. Open your *game.cs* script file in your *MyFishDemo/gameScripts* folder.
Add the following function.

```
function FishClass::onWorldLimit(%this, %mode, %limit)
{
   switch$ (%limit)
   {
      case "left":
         %this.setLinearVelocityX(20);
         %this.setFlipX(false);

      case "right":
         %this.setLinearVelocityX(-20);
         %this.setFlipX(true);
   }
}
```

Code Sample 5.2.1

Since we checked the *Callback* option in the *Level Builder*, this function will get called when our fish has reached its world limit. Two values are passed to this function. *%mode* represents the

mode setting for the world limit. In this case, the NULL mode. The *%limit* value represents which world limit it has reached, either *right*, *left*, *top*, or *bottom*. We start our function out with a "switch$ (%limit)" statement. This takes the value of *%limit* and then lets you compare it to different situations, or *cases*. It allows you to have different things happen based on those cases. As you can see, we continue with the statement and compare it to "left" and "right". The "$" after "switch" means that we are going to be comparing string values, such as "left" and "right" instead of numerical values like "1" and "5". In the case where our fish has reached its "left" world limit, we do two things. First, we set its linear velocity along the X axis (horizontal) to 20. This should send our fish to the right. We also call a function *setFlipX()* passing it a "false" value. This ensures our fish is facing the default direction (which, if you remember, is to the right). If our fish hits the right world limit, it calls the linear velocity function along the X axis, this time setting it to negative 20. This causes our fish to go left. We also call the same *setFlipX()* function, though we pass it "true" this time. This should flip our fish animation to face the left side, so it will appear to be swimming in the proper direction (we don't want moonwalking fish).

Save your *game.cs* script file and now you can reload *TGB*. You should be presented with your same level, no visual differences. Click the *Play Level* button. You should see your fish swim back and forth! *(as shown in Figure 5.2.1)*

Figure 5.2.1