

## 3D GAME STUDIO – TUTORIAL EXERCISE #4 WEAPONS AND ENEMIES

In this exercise we will give the player a weapon and insert an enemy. This exercise includes use of the following scripts:

- weapons00.wdl – defines each of 12 different weapons and 2 kinds of launcher for use by the player and the enemy AI (artificial intelligence).
- weaponsfx00.wdl – defines weapons effects (the script is loaded automatically with weapons00.wdl).
- plBipedWeap00.wdl – defines attributes of the weapons used by the player, including: starting weapon, crosshair image (tga), position (offsets) relative to the player, 1<sup>st</sup> and 3<sup>rd</sup> person model files for each weapon.
- fxGore01.wdl – defines blood splatter and blood pool effects.
- aifps01.wdl – implements simple enemy artificial intelligence, including telling the enemy where to move and to shoot at the player.
- plBipedWeapHUD01.wdl – implements a very basic “Heads Up Display” showing remaining ammo.

By the end of this exercise you will have a basic first person shooter game.

### GIVE THE PLAYER A WEAPON

1. Go to the C:\Program Files\Gstudio6\template\_6\models folder. Open the folder select the all the model files in it (Ctrl A) and copy (Ctrl C). The paste (Ctrl V) into your working directory. Most of these models will be required explicitly or implicitly in this tutorial.



2. [Note: we could have achieved the same objective by changing the “path” in the in your main .wdl file to the location of the model files on the C:\ drive. However, what we did in 1. above is safer since the location of the \models folder may vary from machine to machine.]

3. Open WED. Go to File > Project Manager and select Class “PLAYER” the Script “BiPed Player Weapons”. Several scripts are added including the first three listed above.

4. Go to File > Project Manager and select the Customize tab.

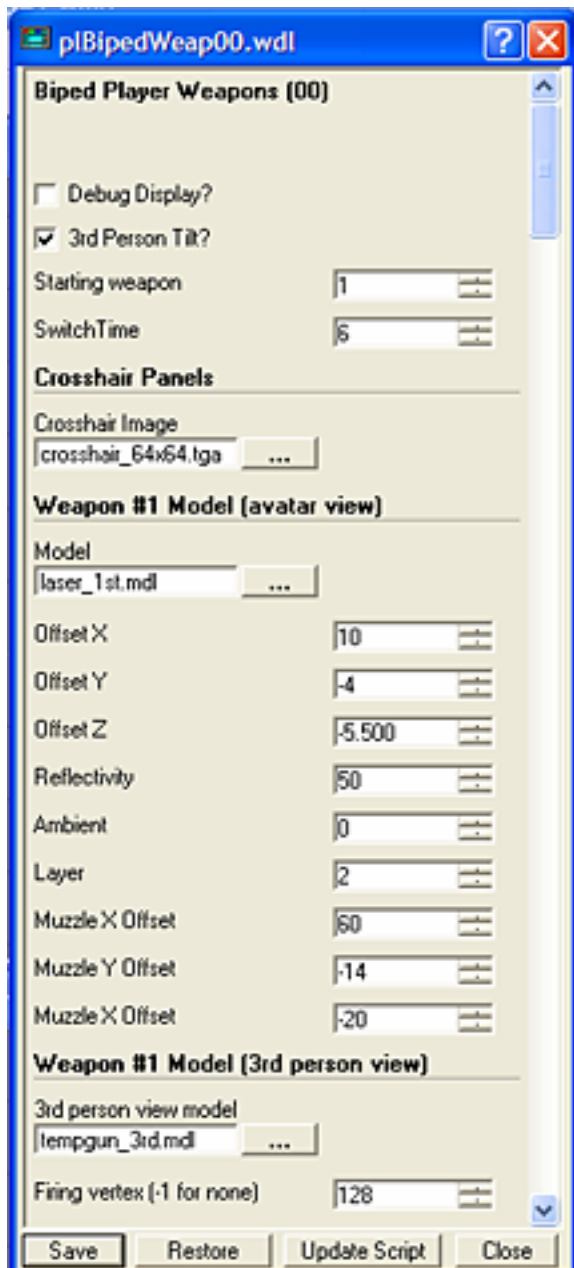
Select Biped Player Weapons (00) and click on ‘Customize.

5. Change the “Starting weapon” to 1. Click Save and close this window.

6. Build  and run  the level. NOTE – to speed build time, click on the “Update Entities” option and move the “Visibility” slider to “low quality”. Some of the lighting may look poor quality, but you can do a full build later to fix this. When the level is run, you should see the muzzle of a cheesy laser weapon. When you fire you should see a laser beam fire and sear the whatever it hits.

NOTE: While in the game, press “Q” to cycle through the player’s available weapons.

7. Experiment. Open the customization window for plBipedWeap00.wdl and change the starting weapon number, build and run and the level. This will show you all the basic weapons available in 3DGS. See how bad they look when you use the 3<sup>rd</sup> person camera (press F7). This is because they all use the rather poor “tempgun” models supplied with 3DGS. To make this look good you either have to make a 3<sup>rd</sup> person view version yourself, or find a matching pair of 1<sup>st</sup> and 3<sup>rd</sup> view weapons on the internet.



8. Also experiment changing other customizable numbers. Note: your window may look slightly different to the one shown on the left.

The offset x (ahead), offset y (left) and offset z (up) changes the location of the gun relative to the player. 0,0,0 being at the center of the model.

Muzzle offset changes the location of the muzzle flash relative to the player. If you change the weapon offset you will need to change the muzzle offset.

Reflectivity and Ambient do not have large effects on the appearance of the gun model.

The default cross hairs are bad. Create your own by creating a 64 x 64 pixel image with an alpha channel for transparency.

You can restore the script defaults at any time by clicking on “Restore” at the bottom of the plBipedWeap00.wdl customization screen and reloading the script when prompted.

9. Experiment with changing attributes of the actual weapons. In Project Manager >

Customize click on weapons00.wdl and select “Customize”. For each weapon you can select items such as accuracy, amount of starting ammo, and effects such as tracer, eject bullets, muzzle flash, creating holes in the wall, etc. Again, you can restore defaults, by clicking on “Restore” and the bottom of the window.

You can have all sorts of fun changing weapon sounds. Why not a mooing machine gun? To change weapon sounds, right click on weaponsfx00.wdl and change the file name for the sound to another you want to replace it with. Make sure any sounds you use are in your working directory. Use .wav files.

## INTRODUCE AN ENEMY

NOTE: Enemy models must include a weapon which is part of the model. If an enemy has no weapon, bullets, flashes, etc will fly from it's body (which can be good for monsters, etc). 3DGS includes no models with weapons built in. Download badguy\_gun.mdl from [http://www.twcdc.com/games/2006/11/download\\_the\\_bad\\_guy\\_model.html](http://www.twcdc.com/games/2006/11/download_the_bad_guy_model.html) and save it to your working directory.

1. Go to Object > Add Model and navigate to your working directory and select badguy\_gun.mdl.
2. Scale and move the model as needed, so he's about the same size as the player and standing on the floor.
3. Build and run the level. The bare-chested bad guy will be cycling through his animation sequence (quite silly). Nothing will happen if you shoot at him, because we have not given him an action.
4. Go to File > Project Manager and select Class AI and Script Simple FPS AIs. (AI = ARTIFICIAL INTELLIGENCE).
5. Right click on the bad guy, select properties.

Click on the behavior tab, click on the folder icon and select the AIFPS01\_Guard.wdl behavior.

6. Build and run the level. When you come into the bad guy's view he will start blasting at you. If you shoot him enough times he will die and fall to the floor.



7. You can change the behavior of the bad guy by right clicking on him in the level and selecting behavior. Here's an explanation of each property that can be changed:

Health – number of times you can shoot him before he dies (default 25)

Armor – amount of protection (default 0)

Explode – check the box and check it out

#### Movement

ForceX – controls AI forward speed

ForceY – side-step speed

Pan – turning speed

#### Limits

MinMoveZ – controls upward sliding

Jump – jump height

#### Animation

All of the following are used to control of the speed of the animation cycle (not the speed of movement). For example, if it looks like his legs are moving too fast for the speed he his covering, you can adjust animation speed here.

WalkSwimDist

RunCrawlDist

StandJumpTime

DeathDamageTime

RunThreshold

### Weapon

ID – weapon used by the enemy. Note that the AI's weapon takes on the attributes of the corresponding weapon ID in the weapons00.wdl file. Since the enemy model already has a weapon there is no need to specify a model file like we do for the player.  
Vertex – the height from which the weapon fire appears to come from the entity. The default is -1 which is the center of the entity. If your model is not holding its weapon at its center you can adjust this value, until the fire lines up with the muzzle of its weapon.  
AccReload – never used it. Check the manual

### Senses

FOV – range of hearing and field of view in degrees. Number before the decimal point is the range, the number after is the angle in degrees. For example, 2000.090 (the default) is a range of 2000 quants and an angle of 90 degrees. This controls how easy it is for the enemy to see the player in an open space with no obstructions.  
Hearing – hearing range (in quants?). This controls the enemy's detection of the player when the player is not visible to the enemy. Increase this number if you want the enemy to be able to start chasing you from a faraway distance

NOTE: definitions for any item can be accessed by right clicking on the attribute, e.g. ForceX.

8. You can change some global attributes for all enemy AI by customizing "Simple FPS AIs through the project manager

Display AI health – displays what the AI doing and its health on the screen. Good for developing a game but a bit clunky.

Minimum close range – the distance in quants AI will maintain from the player.  
Alert range - The max distance between enemies that they can alert each other to the presence of the player.

Blood and sparks speak for themselves.

9. Add a few more bad guys and customize each one by right clicking on it and selecting behavior.

## **HUD**

This simple heads up display (HUD) shows player's remaining ammunition.

File > Project Manager > Class > PLAYER and Script > Biped Player Weapon 'HUD'.

Not a great panel, and not much customization available. The code, however, is short and could easily be changed.