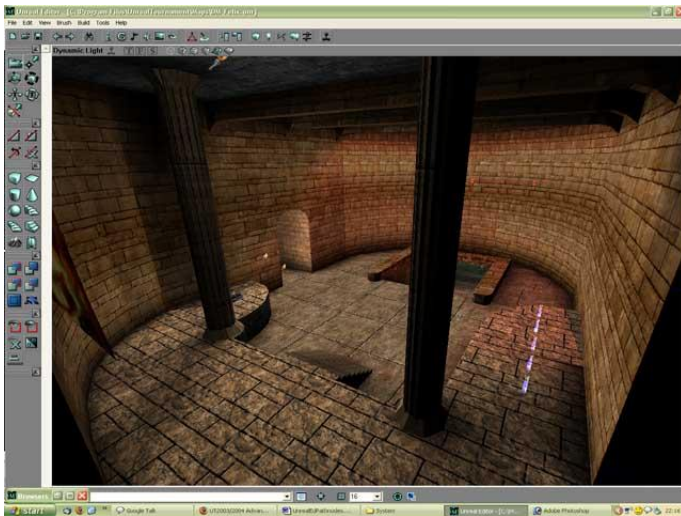


Simple Path Nodes in Unreal

For many adventure, racing and FPS games the various in-game NPC's will need patrol points to know how to navigate through levels, while developing projects we had our own tools which allowed myself as the level designer to set navigation pathways. As these tools are not public domain, however I thought that I would give an example with a freely available level editor.

As with my previous tutorials I prefer to use UnrealEd by Epic, as is the product is widely known and used and I am most comfortable talking about it.

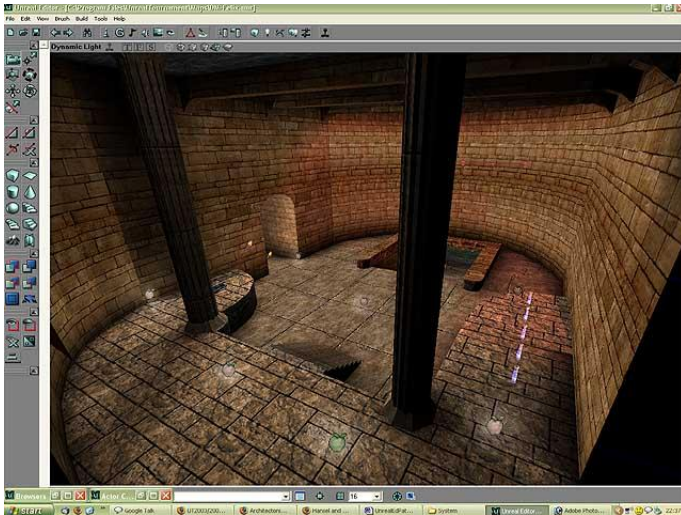
In Unreal, the main thing the NPC's are looking to do is navigate the level, find pick-ups and of course hunt down the player and each other. UnrealEd is really quite useful in that the AI is relatively good at following simple navigation and the actual placement is easy. I will quickly talk about how to set-up the AI for a level you have designed.



Above is a single room I wish to add AI paths to, of course the whole level would need pathing up in a similar way to shown, but this is for example. First, you need to click the actor icon, it is the one that looks like a chess pawn and is in a group with five other icons along the top. Once in the actor editor pop-up window you open the tree selection by clicking on the plus sign next to actor, next click on the plus next to NavigationPoint and finally double click the PathNode icon. This will run an unreal script, you do not have to worry about this for the tutorial so now close any pop up's and go back to the editing screen.

Now the easy bit is setting up the Navigation route, the only thing is to remember that you need to place a new navigation point at least every 1200 UnrealEd units, of course it is acceptable to place them closer (However remember this will increase your level overheads), also they must be in line of sight of one another.

Simple Path Nodes in Unreal



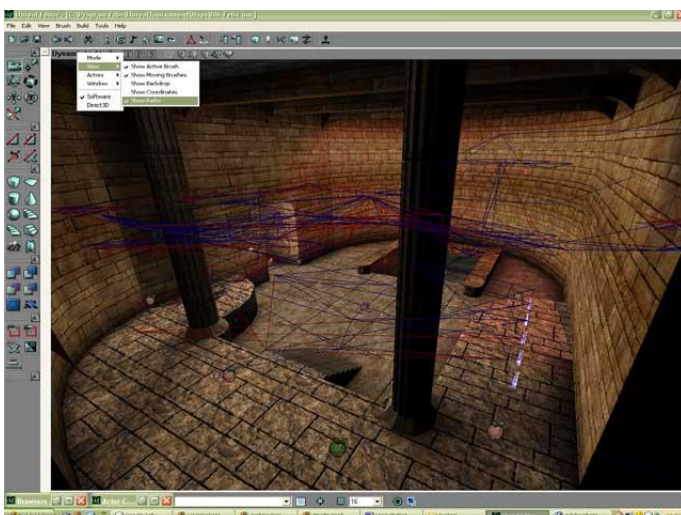
They are relatively simple to add then tweak, right click on a floor area where you need to place a PathNode and select Add PathNode Here. You will notice a little apple icon appear, a bit like Hansel and Gretel the Bots will follow the 'food' until they meet with a sticky end.

You can move the PathNode's around by selecting them and using the same controls as that of moving geometry. Above I have illustrated the PathNode's by highlighting them on the image.

Notice that I have placed one of the nodes at the front of a corridor, not inside it. Of course there is one in the same place at the end of the corridor. As I have already mentioned the nodes need to be in line of sight so placing them just inside the corridor will mean the AI cannot navigate properly. However if you have a long corridor you can always add extra nodes in the middle of the tunnel.

A useful way to check routes are fine is by either selecting the top corner of a display window (where the text says something like Left, Right, Dynamic Light) if you right click there then select View > Show Paths from the pop out window you will be able to visible see the paths.

Another good way is to build the level then simply watch the bots run around in spectator camera.



Simple Path Nodes in Unreal

Finally a little to touch on is how to navigate with small jumps and lifts, in the original UnrealEd you could use the LiftExit pathing that could also be found in the Actors>NavigationPoint selection within Actors.

Personally I think these icons look a little like dragons heads I have no idea why they do they just do, you could say the icon does not make much sense for what it actually does. These are used on the exit from lift mover objects, so the bots would get out at the end of a journey. You can also supplement LiftExit with LiftCenter, which will tell the bots where to stand when on the lift.

In actors expand on LiftCenter in UnrealEd 2.0 or look in the main NavigationPoint listings on higher versions of UnrealEd you will notice something called JumpSpot these are intended to be used when the bots are to navigate small jumps, I tend to use them on the top and bottom of stairwells (as many other UnrealEd users seem to.) They are really best used to navigate small platform-to-platform jumps.



This is the basics of AI navigation using the UT editor. It maybe that for other games you may have to mark in other AI navigation like cover points (for games that use cover for example Gears of War 2)