

## **IDS 4687 Games Engines – UnrealEd Tutorial 1**

Author: Erin Hastings phone: 407.926.2979 mail: [hastings@cs.ucf.edu](mailto:hastings@cs.ucf.edu)

This document can be downloaded at: [www.planetunreal.com/squacky/UnrealEd-Tutorial1.doc](http://www.planetunreal.com/squacky/UnrealEd-Tutorial1.doc)

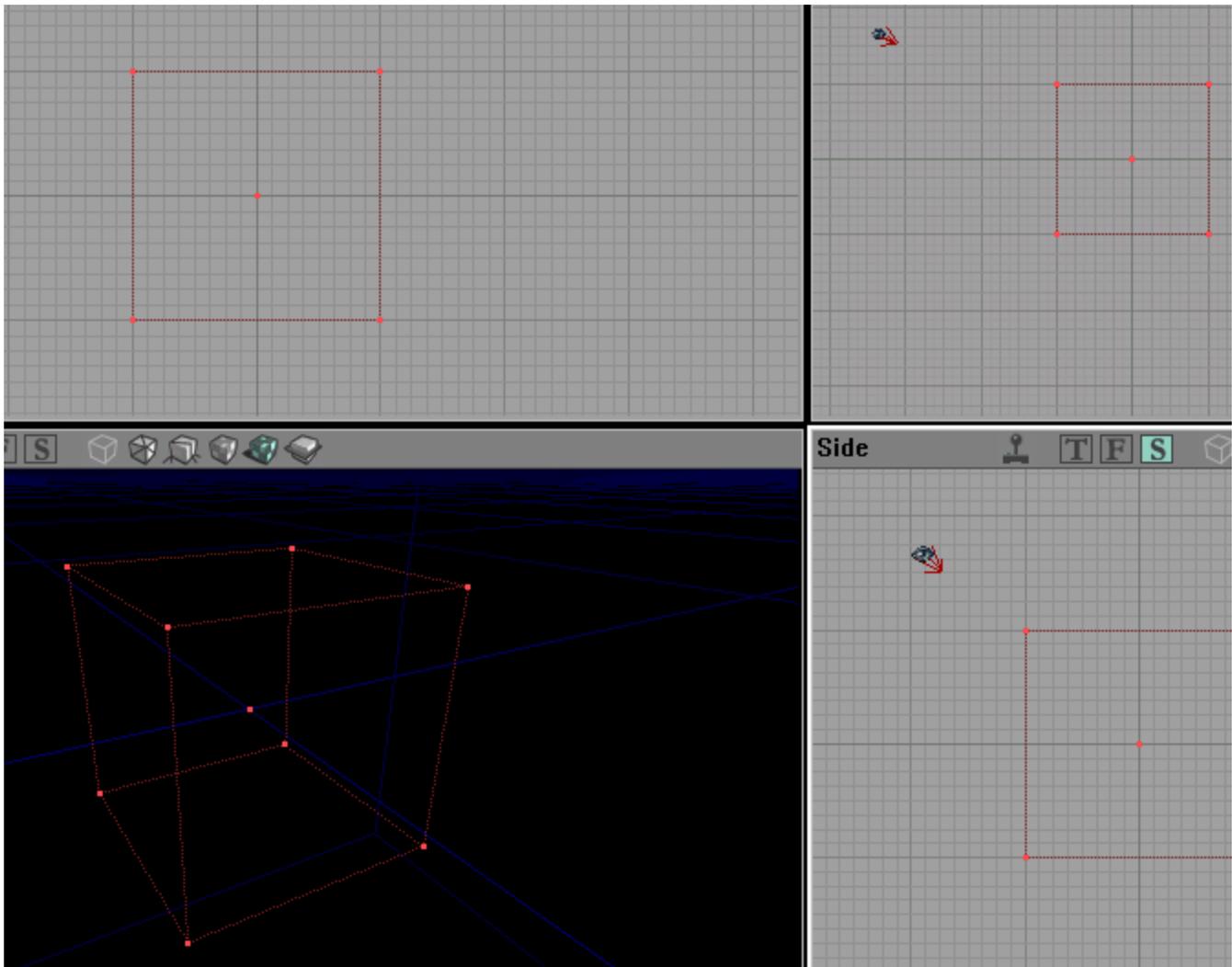
### **Tutorial 1 – Basic Architecture, Lighting, and Texturing**

UnrealEditor is not automatically linked to the start menu when UT is installed. You can find UnrealEd.exe in the UnrealTournament/System directory. Create a shortcut and place it on the desktop if you wish. When UnrealEd starts up you can close the close window called “Browsers”. Notice there are 4 view ports. One gives a 3D view of the map you are working on. The other 3 are 2D – the X,Y, and Z axis views. The first thing we will do is build a CUBE. LeftClick the CUBE BUILDER icon on the left hand menu.



**CUBE BUILDER**

You should see the following. The red wire frame is your cube and the Eyeball icon is the position of the 3D CAMERA.



Now practice navigating the 2D and 3D windows in relation to the cube. Below is a list of the commands used for navigating in UnrealEditor.

### Navigating the 2d windows:

#### Command:

Hold LeftMouse  
 Hold RightMouse  
 Hold LeftMouse and RightMouse  
 Left Click an object  
 CTRL and LeftClick objects  
 Hold CTRL and LeftMouse  
 Hold CTRL and RightMouse  
 RightClick an object  
 RightClick grid

#### Action:

Moves view SLOWLY left/right or up/down.  
 Moves view QUICKLY left/right or up/down.  
 Push mouse forward/back to zoom in/out.  
 Select an object.  
 Select multiple objects.  
 Move an object.  
 Rotate an object.  
 Bring up options menu for that object.  
 Change grid size. This is usually not necessary.

### Navigating the 3d window:

#### Command:

Hold LeftMouse and move  
  
 Hold RightMouse and move  
 Hold LeftMouse and RightMouse

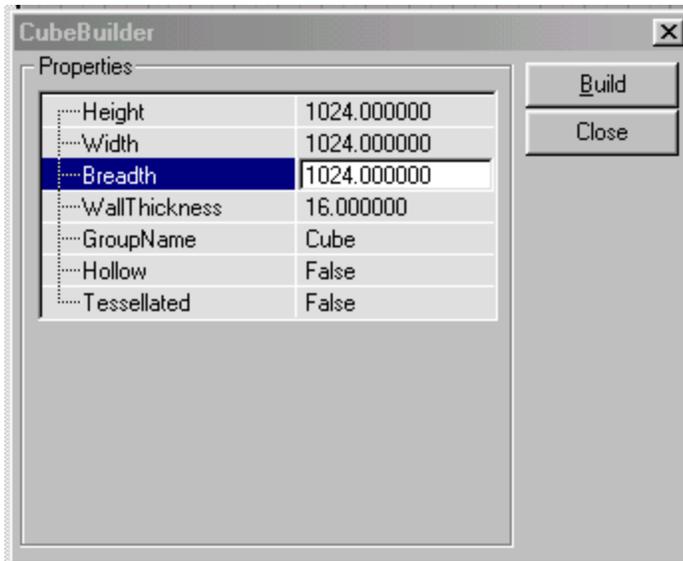
#### Action:

Horizontal plane.  
 Move mouse forward/back to move forward/back.  
 Move mouse left/right to look left/right.  
 Locks view to a point and allows panning in all directions.  
 Move mouse forward/back to move up/down on vertical axis.  
 Move mouse left/right to move right left on horizontal axis.

LeftClick a Surface  
 CTRL and LeftClick surfaces  
 Shift and LeftClick a surface  
 RightClick an object

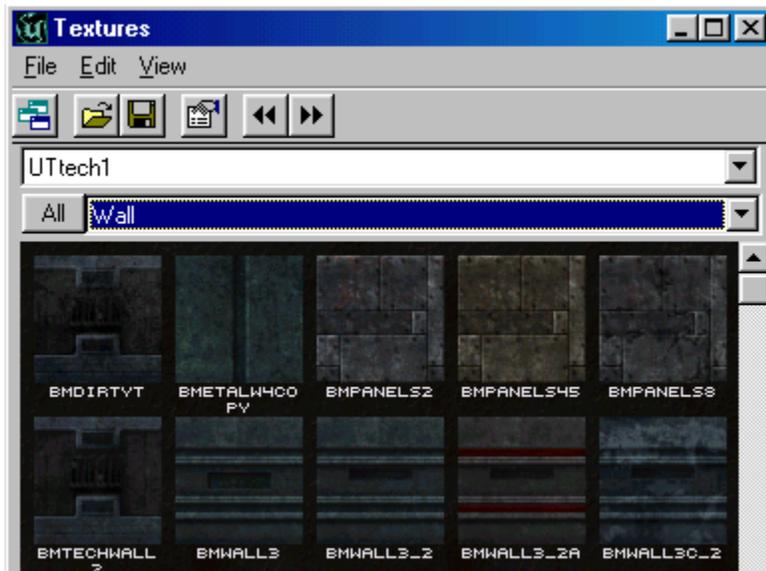
Select a poly surface.  
 Select multiple surfaces.  
 Select entire poly attached to that surface. This is very useful.  
 Bring up options menu for that object.

All of these tutorials will be done using the same map basic map made in this tutorial. First of all let's build a 1024\*1024\*1024 cube. Right click CUBE BUILDER icon and enter the following properties.



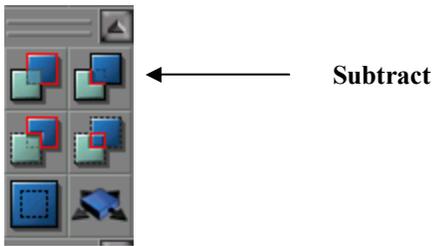
After you hit “Build” on the CUBE BUILDER you should see your wire frame cube in all the windows.

You must now select a texture for your room. Open the TEXTURE BROWSER. The icon looks like this . In the TEXTURE BROWSER open the “UTTech1” texture package. Then use the drop down menu to bring up the “Wall” textures. Your browser should look like this.

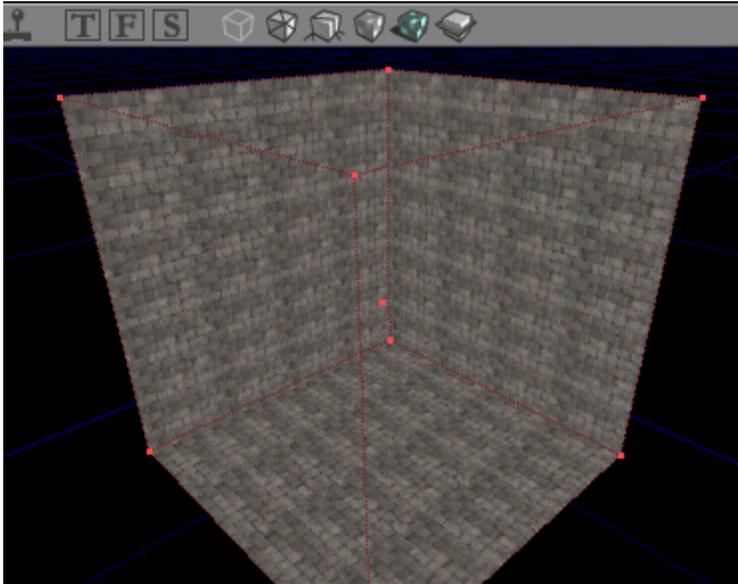


In the TEXTURE BROWSER click on a texture of your choice. This is the texture that will be applied to your cube when SUBTRACT it. Close the TEXTURE BROWSER.

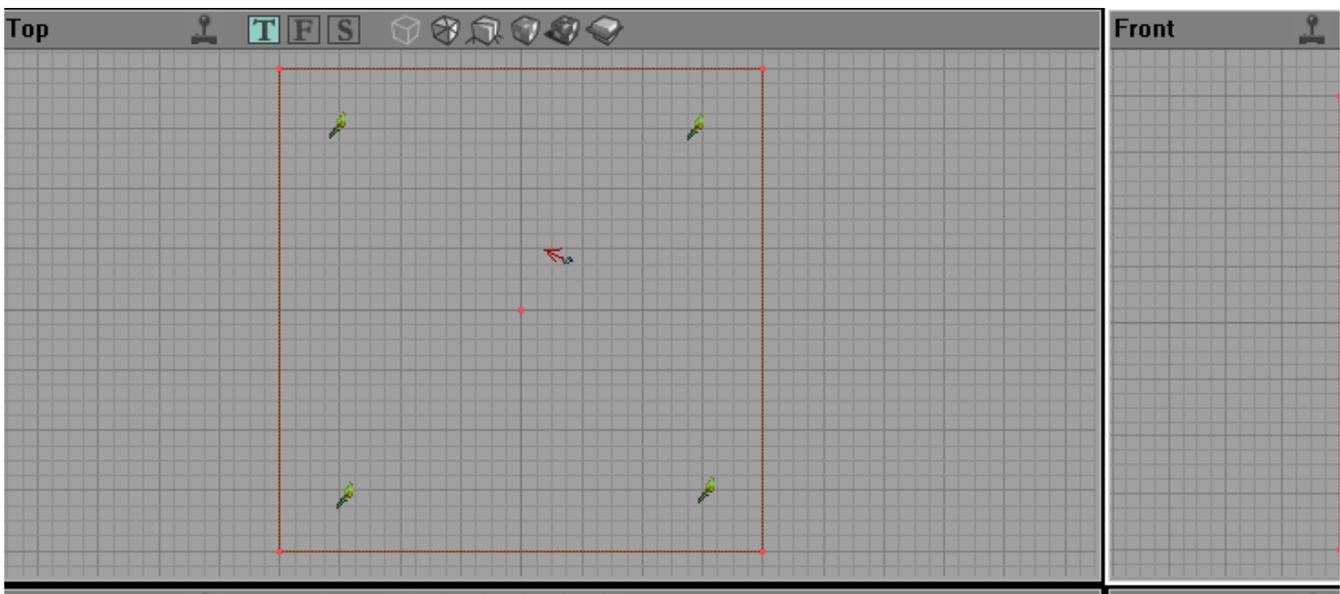
Now we will SUBTRACT the cube from the 3d geometry. Hit the SUBTRACT button on the left menu bar.



Your map should look like this now:



Now, using the 3D WINDOW, place LIGHTS in all 4 corners of the cube. To place a LIGHT simply right click inside the cube and select “Add Light Here”. The LIGHT icon looks like a little torch. Using the 2D WINDOWS, try to line up your LIGHTS up as pictured below. The LIGHTS shown below are 128 units from each corner and centered vertically within the box. This will light your room more brightly.



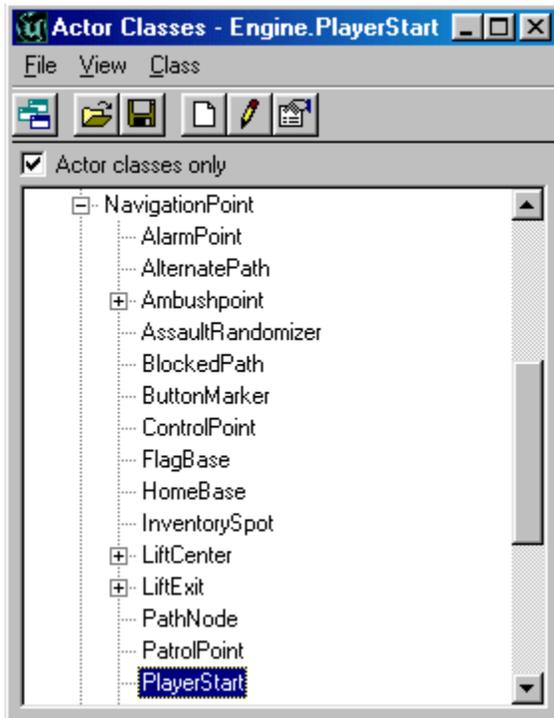
Now that our room is built and lit, let's add a PLAYERSTART. This is where players will “spawn” in the game. Open the

ACTOR CLASS BROWSER. It looks like a chess pawn.

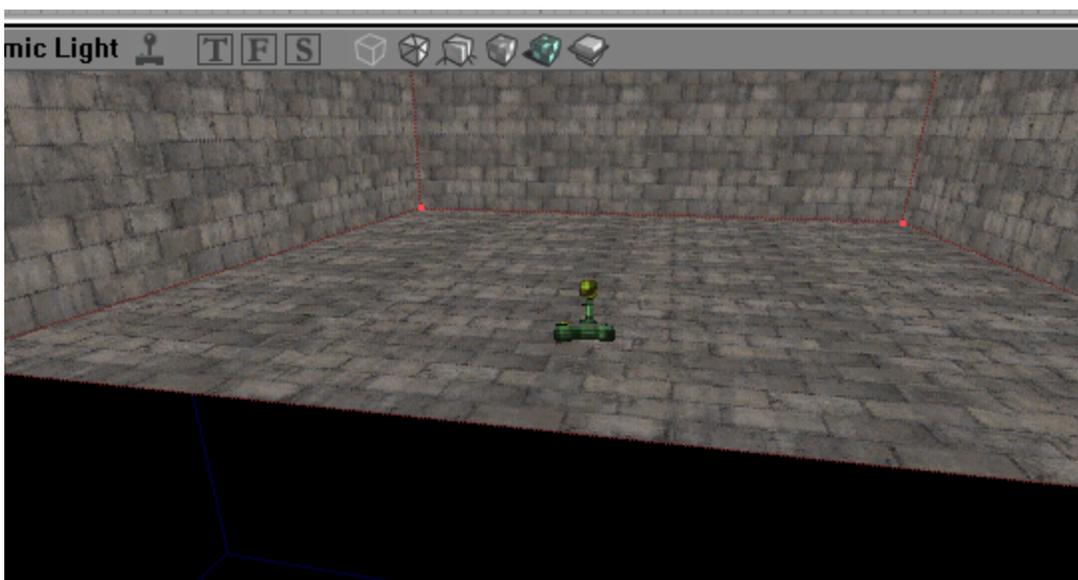
Actor Class Browser



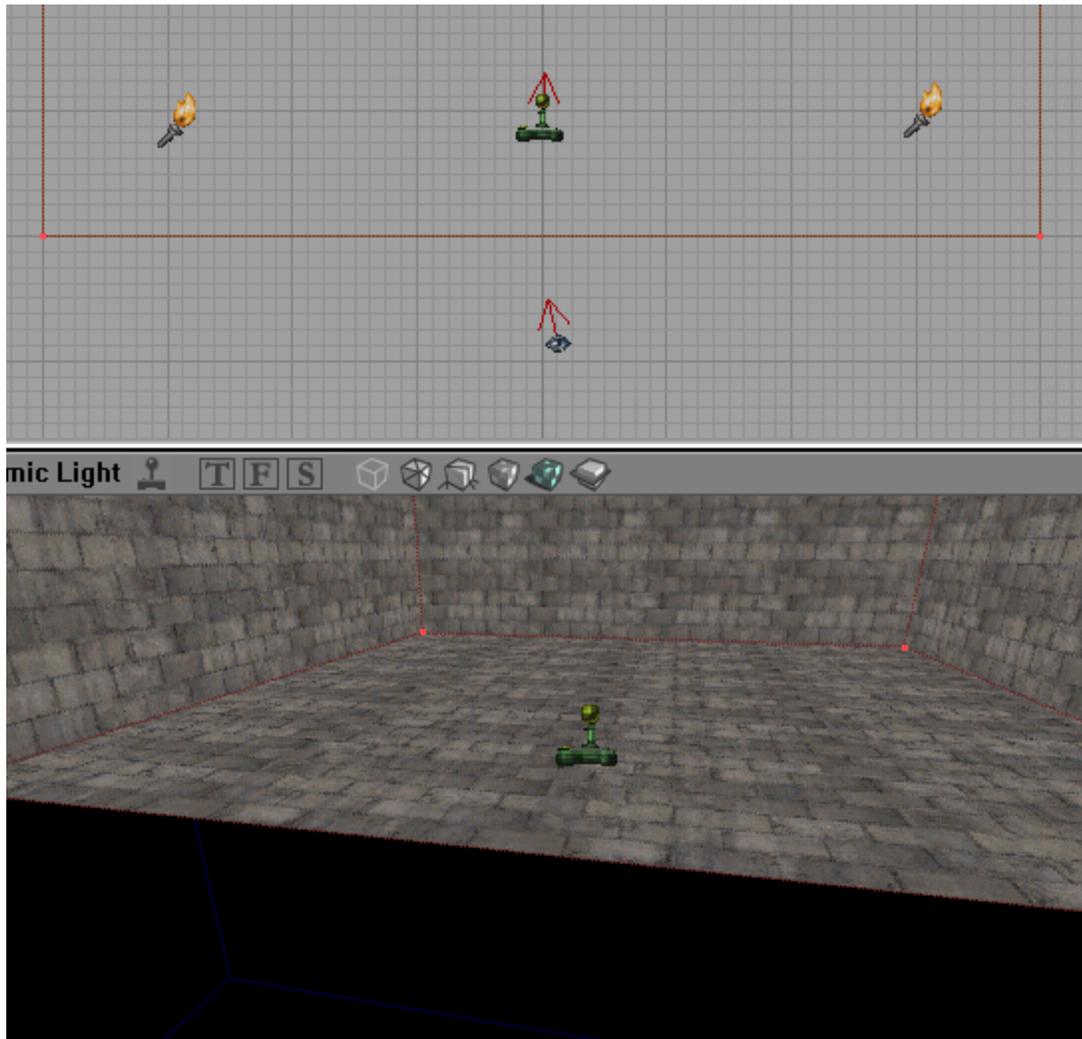
In the ACTOR CLASS BROWSER select NAVIGATION POINT >> PLAYER START.



Now right click on the floor of your room in the 3D WINDOW and select “Add PlayerStart Here”. As a general rule, you should always insert actors using the 3D WINDOW, not the 2D WINDOWS. Why? In the 2D windows you cannot be sure of where you are putting them. You should see something like this.

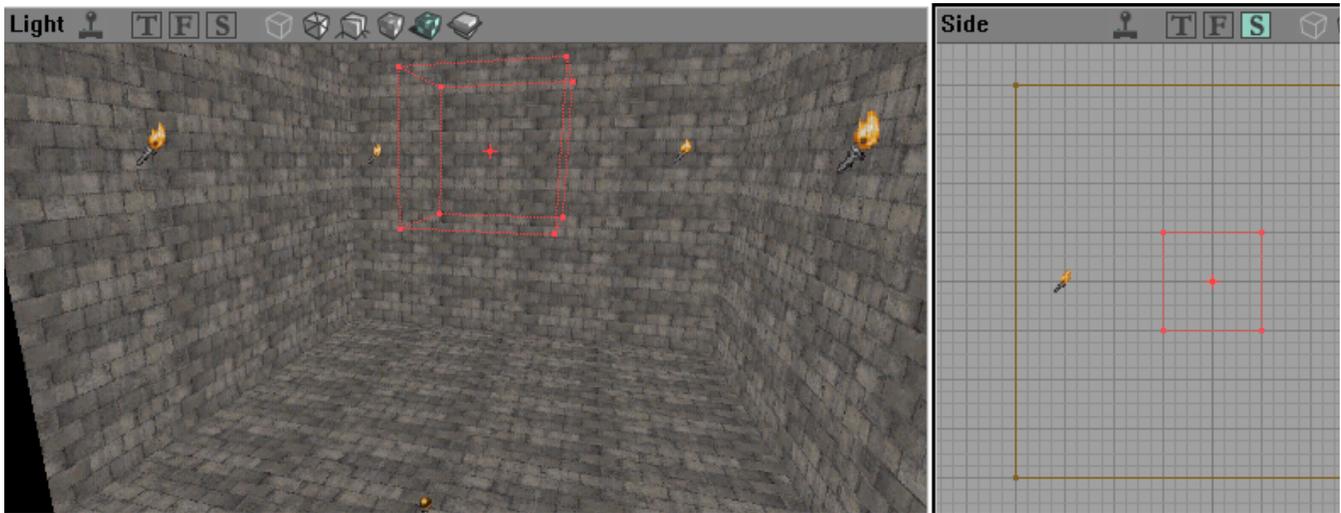


Using the 2DWINDOWS, line up your PLAYERSTART on the south wall and make sure it is facing north as in the following picture.

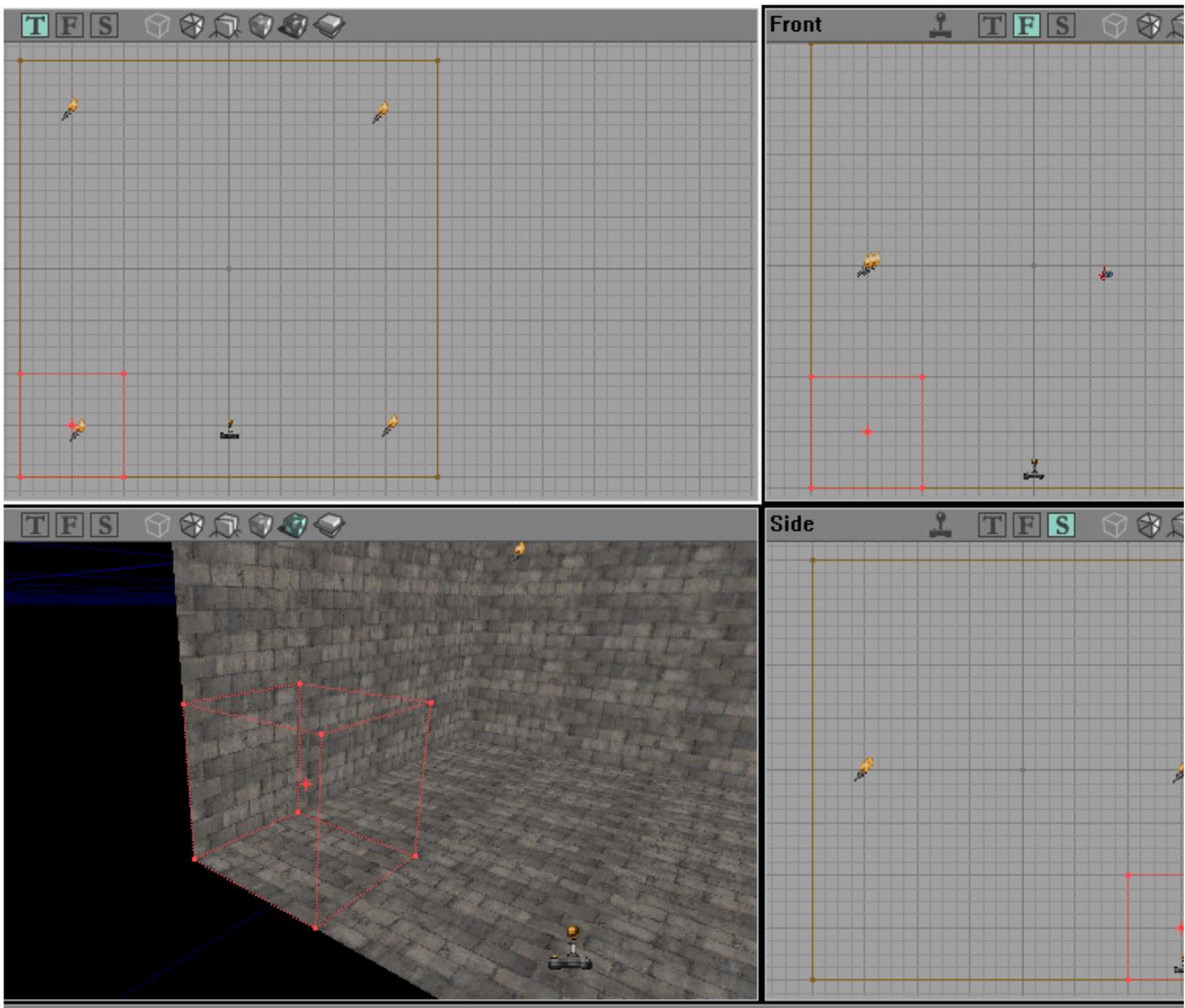


There are two helpful things to note from the picture above. First, the PLAYERSTART has an arrow pointing out of it. This is the way a player will face when he/she/it spawns. Make sure it is not facing a wall. Second, the eyeball shaped icon in the 2D WINDOW is the position of the camera in the 3D WINDOW.

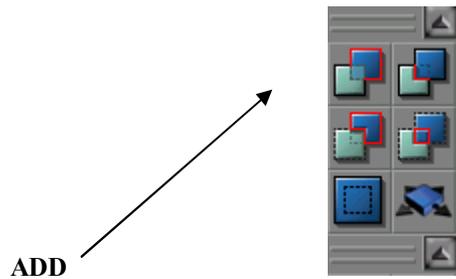
Now we want to make our cube more interesting by adding things inside of it. Using the CUBE BUILDER again, build a 256\*256\*256 cube. You should see the following.



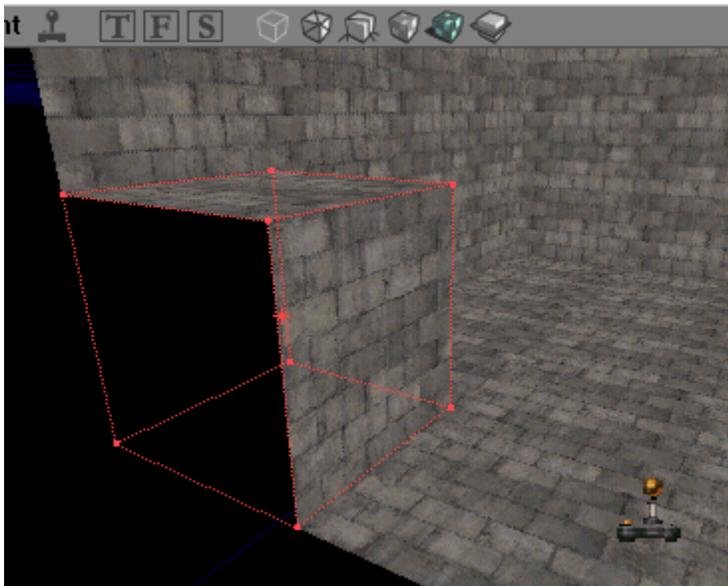
Using the 2D windows, line up the cube in southwest corner of your room and make sure it is on the floor. When correctly lined up it should look like this:



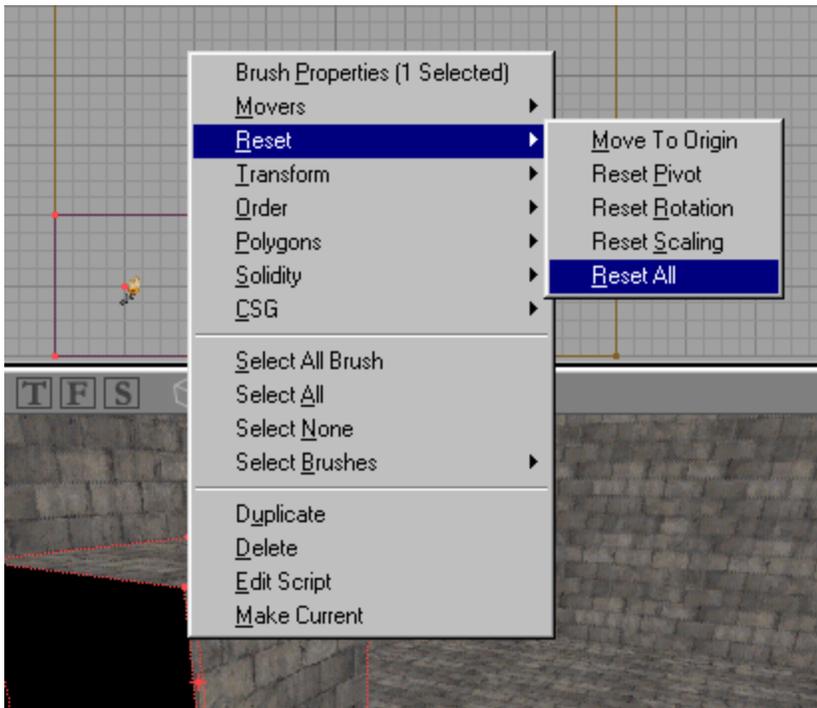
Notice that new shapes are always a red wire frame. This red wire frame is called your BUILDING BRUSH. Now hit the ADD button to ADD your new brush to the map. Note that inside our large SUBTRACTED cube areas we must ADD new brushes. You can think of an Unreal Map as a solid block out of which we must carve, or SUBTRACT, areas for the players. We can also ADD details into these hollowed out areas.



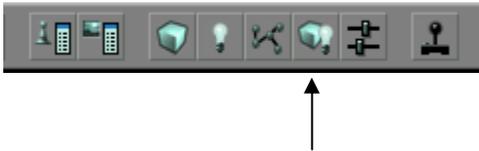
After you hit ADD, your map will look like this:



Notice that our BUILDING BRUSH is still there. It might be in the way, so the easiest way to return it to the center of the map is to right click the BUILDING BRUSH and select RESET>>RESET ALL as shown below:



Now hit “BUILD ALL”. This renders the lighting and computes collision radii etc... to make your map playable in UnrealTournament. Also make sure to save your map as “DM-Mapname”, where map name can be whatever you wish. Your map will be stored in the UnrealTournament/Maps directory.



**Build All**

You just created a working Unreal Tournament level! You can now hit the PLAY MAP icon and test your level.



**PLAY MAP**

### Miscellaneous Useful Info

**Unreal File System** – File types and where they are stored.

<u>Name:</u>	<u>Description:</u>	<u>Location:</u>
UnrealTournament.exe	Runs UT	UnrealTournament/system
UnrealEd.exe	Runs UnrealEd	UnrealTournament/system
*.unr	Map file	UnrealTournament/maps
*.utx	Texture file	UnrealTournament/textures
*.umx	Music File	UnrealTournament/music
*.uax	Sound file	UnrealTournament/sounds
*.u	Actor file	UnrealTournament/system

**Basic Architecture Tips** – Here are some useful things to remember while designing your maps.

**Grid** – UnrealEd’s default grid is 16 unit blocks. You can change the grid size but remember to ALWAYS line your architecture up on grid lines. Failure to do so may result in poor performance of your map or rendering errors in the game (BSP holes).

**Scale** – Just as in real world architecture, scale is very important in 3D world architecture. Here are some basic scale examples/guidelines for UnrealEd.

Staircases - A step on a staircase is usually 16 units in height. Anything larger may result in the player having to “jump” up the step.

Tunnels/Hallways - The dimensions of a rather spacious tunnel could be around 256 units in height and 384 units wide. A cramped tunnel could be around 128 units in height and 256 units in width. Ceilings lower than 128 may not allow for players to be able to jump.

Item Placement – Usually try to place items about at least 32 units from the nearest wall. Otherwise, players or bots may not be able to pick them up due to collision problems.

**Basic Lighting Tips** – To change a light’s properties right click the light then select “LightProperties”. The most important properties are under “Light Color”. Brightness, Saturation, and Hue are on a scale from 0 to 255. “Lighting>>LightRadius” is also a very important attribute.

**Brightness** 0 = total darkness and 255 = extremely bright. Brightness from 64 to 128 is generally sufficient for most map lighting.

**Saturation** is the amount of white light in a color. Saturation 0 = no white (a pure color) and saturation 255 = all white. No color will show with a saturation of 255. Saturation of 128 to 200 is generally used in most map ambient lighting. Very pure colors (Saturation = 0 to 64) over a large area will make your map look like a “Disco” which is usually not desirable.

**Hue** uses the “ROYGBIV” color scale over the range 0 to 255. Use the following reference table which gives approximate color values.

<u>Hue Value:</u>	<u>Color:</u>
0	Red
32	Orange
64	Yellow
96	Green
128	Aqua
164	Blue
180	Purple
196	Violet

**Radius** is self-explanatory. The default is 64 is and is usually too large. Indoor light radius is generally best set from 16 to 32. Outdoor ambient lights often need a radius of 128 or more. Many lights in the same area with large radii often results in multiple colors blending together to form an ugly, unwanted color.

**Adding Basic Actors** – Use the ACTOR CLASS BROWSER select Actors to add to your maps. The following list details the basic UT classes that will be used in your maps. Make sure you use “Tournament Weapons”. The other weapons listed are from Unreal 1 and may not function in a UT network game.

UT Weapons:	Inventory>>Weapons>>TournamentWeapon
UT Ammo:	Inventory>>Pickup>>Ammo>>TournamentAmmo
UT Inventory:	Inventory>>Pickups>>TournamentHealth
	Inventory>>Pickups>>TournamentPickup
PlayerStart:	NavigationPoint>>PlayerStart

### Other Basic Stuff

**Naming your map** – When you save your file you must save it as “DM-<Mapname>” in order for it to appear on the DM list in the game. Name it “CTF-<Mapname>” in order for it to appear on the CTF list.

**Giving Yourself Credit** – From the “View” drop down menu select LevelProperties>>LevelInfo. Next to “Author” type your

name and email address.

### Console Commands

Command:

**SAY** <message string>

**SETNAME** <newname>

**SUICIDE**

**TEAMSAY** <message string>

**ADDBOTS** <number>

**ALLAMMO**

**FLY**

**GHOST**

**GOD**

**KILLALL** <class name>

**KILLPAWNS**

**PLAYERONLY**

**SLOMO** <number>

**SUMMON** <class name>

**VIEWCLASS** <class name>

**WALK**

Action:

Broadcast a message to other players in the game.

Change your name (works in network play too).

Commit Suicide

Broadcast a message to your teammates.

Adds more bots to the game.

Gives you all the ammo for your current weapons.

Fly around (instead of walking). See WALK.

Fly around and go through walls (see WALK).

Makes you invincible.

Kills all actors of a certain class, for example "KILLALL  
WARHEADLAUNCHER".

Kills all bots

Pauses all non-player actors in the game. Cool for debugging scripts, taking  
screenshots.

Sets the game speed, 1.0 is normal.

Spawns an actor of the specified class (for example, SUMMON  
BOTPACK.WARHEADLAUNCHER) in front of the player.

Cycle through actors of this class, viewing from their location. eg  
VIEWCLASS BOT

Walk around normally (see FLY and GHOST).