back <- up -> next

Academic Resources for the Unreal Development Environment

by: Robert J Martin

- Basics of Unreal
- Map Making
  - UTEditor (coming soon)
- Mod Making
  - Architecture Overview (brief synopsis of UT components)
  - Getting Started (new gametype and intro to tools like WOTgreal)
  - UnrealScript Syntax (Quick Reference)

Basics of Unreal

Unreal Tournament is a FPS multiplayer combat game based on its predecessor, Unreal, which was a single player scenario game. The player's avatar/character is competitor in a gladiator-like competition which takes place in a dismal, Blade Runner style future, where mega-corporations sponsor contained conflict to decrease violence among the populace. As opposed to quests, the thrust of the gameplay is sport-like matches or tasks with human and computer-controlled participants. Good reflexes and coordination are paramount, but strategy also plays a role in gameplay.

You can play a single-player game against Bots (computer controlled opponents) or a multi-player competition between humans and bots over the Internet or a LAN. Once you have configured your avatar by selecting its look and name, you are ready to begin a match. Simply select a new game from the menu and you're off. When you start the game, you will be spawned in a level and immediately have control over your avatar's perspective, movement, and action.

The environment you are in and the rules that govern its behavior and content are found in a map file (xxx.unr). Each map can be considered a world of its own. After you have played for a while, you will start to find you like certain environments and will want to play those maps more.

Each map will have a default game type associated with it. Within UT, there are several different types of games. There is DeathMatch, in which every man is out for himself and everyone you see is your enemy. The goal in this type is to kill, or "frag", as many opponents as possible to increase your ranking. Then, there are team games like Capture The Flag and Team DeathMatch. These are often populated by two teams, red and blue. The object is to cooperate with other players on your team and combat those who oppose. There is also the Assault type game, which should be of special interest to anyone who wants to make a Mod that differs greatly from UT. These types of games have a specific, sometimes complex agenda. The object is to complete some task in as little time as possible. There is the typical combat oriented conflict but one must solve a puzzle of sorts to score highly. The reason that this should be noted by Mod and map builders is because there are special techniques that are often used to make Assault scenarios work and each map can be almost like a different game in itself.

For the sake of simplicity, the composition of UT can be divided into two, almost distinct parts. The first is the engine. If you have not heard this term before, it is basically the low-level software that does things like quick 3D mathematics and network and graphics hardware interface. The idea is that aside from tweaking and special uses for different needs, the engine can be reused and separated from the content, which is the second piece of UT. Though UT comes with its own content out of the box, it is conveniently packaged with tools to help you design your own content without changing the underlying engine, but by simply interfacing with its API.

For those who want to change UT's content, there are three basic options. First, one can make her own maps using the UTEditor, or UTEd, which comes with the UT game itself. This will not change any rules about how the world behaves, it simply takes the pieces of content and building blocks of the vanilla UT, and allows you to build your own environment, placing decorations, landscapes, objects, and characters for others to play. A second way is to use a hook built into Unreal for Mutators. Mutators are variations on UT for things that only slightly change or customize the gameplay of typical UT game types. The third way is to make a Mod (Modification) using the engine's API for content creation and management, UnrealScript. We will focus on the first and third techniques, as Mutators are limited, especially if one wishes to make a completely new game, often called a "full conversion Mod".