

UDK3

AI with Kismet

An Introduction to Computer Game AI Using the UnrealKismet Scripting System

Introduction

AI or Artificial Intelligence is used to add realism to your computer games. AI can take an otherwise unconvincing, boring game and turn it into a blockbuster best seller.

AI for computer games is not the same as the AI in the field of Computer Science. The AI in a computer game only has to convince the player that things are acting intelligently, where in the Computer Science AI, things have to be really intelligent.

What this means is that the approach to AI for the game developer is somewhat different from that of the Computer Scientist.

An Example

A simple example of AI in a computer game would be the following:

- A locked door can only be opened during a specific game time.
- When the door is opened you may walk in and optionally join a guild.
- If you join the guild, you may then open the door any time you wish.
- If you do not join the guild, the door will never unlock for you again.

Doing an AI analysis of the above example produces the following:

- ***The game environment is making decisions*** – When will the door be unlocked.
- ***The player is making decisions*** – To enter and join the guild or not.
- ***The player's decisions change the game environment*** – The door may or may not stay locked.

This simple example illustrates the three basic requirements of computer game AI:

- The game environment must be able to make internal decisions independent of the player.
- The actions of the player must be monitored by the game environment.
- Player actions are able to change the internal decisions made by the game environment.
- Changes in the game environment can affect future actions available to the player.

Essentially, these game AI requirements reflect what happens to us in the real world; how we behave now can influence our future options in the real world.

Using Unreal Kismet

The UDK3 development kit comes with Kismet. According to Epic Games:

“UnrealKismet is a very flexible and powerful tool that allows non-programmers to script complex gameplay flow in level (design). It works by allowing you to connect simple functional Sequence Objects to form complex sequences.”

Purpose of These Documents

The documents used in this series of **AI with Kismet** will illustrate the techniques and methods used to create complex and believable AI environments in UDK level design.

These documents will illustrate the following using concrete Kismet examples in real levels:

- Using Kismet to create internal game decisions independent of the player.
- Using Kismet to monitor player decisions.
- Using Kismet to modify game behavior based on player decisions.
- Using Kismet to influence future player decisions.

What all this means is that you do not have to be a Rocket Scientist programmer to create games with a real professional and polished AI built into them. By using Kismet's simple functional sequence objects you can script powerful AI actions into all of your level designs. Doing this can help you produce computer games that are convincing, addictive, and just plain fun to play.

Summary

The documents that will follow in this series will allow you to use Kismet to create a rich and believable game environment by creating levels that act intelligently. Using the techniques presented in these papers can turn a dull and unbelievable game into a blockbuster success.