Esc Online: A Venue for Believable Agents

Martin C. Martin

Carnegie Mellon University, Robotics Institute
5000 Forbes Ave., Pittsburgh, PA 15213
mm@cmu.edu

Abstract

Esc Online is a free 3D multi-user online nightclub being developed for the PC. A new way for people to socialize online, it combines music, dance, and chat, and allows scripted and live characters and performances.

Esc is a natural forum for creating autonomous or semi autonomous characters that interact with real people. For believable agents researchers, Esc provides an established engine with an existing community for technical support and a ready audience. In addition, our team can work directly with a small number of researchers to help them bring their ideas into Esc.

The project is described, possible models of character interaction are discussed, and an overview of technical details is presented.

Introduction

From chat rooms to massive multiplayer games, multi-user virtual spaces are becoming more and more common. Multiplayer video games have gone from curiosities to an industry mainstay in a little over five years. Networked computers also host social spaces such as text based chat rooms and 3D virtual worlds like Active Worlds.

The computer controlled characters in these spaces are often very simple. For example, infobots on Internet Relay Chat (www.infobot.org) listen for sentences of the form "X is Y." Later, when someone asks "X?" they respond "X is probably Y." They can be programmed to reply simply "Y" or choose one of several answers at random. Beyond recording useful information, they can be secretly taught to respond to someone's pet phrase, or used as the straight man in a joke.

Characters in the massive multiplayer role playing game EverQuest have a traditional branching tree structure to their conversations. In fact, the trigger words in each sentence, the ones they will answer questions about, are displayed in square brackets. Most characters stand in one spot, and for those that do move, their movements are more practical than expressive. This gives them a rather robotic/mechanistic feel.

Obviously, these characters could benefit greatly from research into believable agents. However, the practicalities of the games and software industries
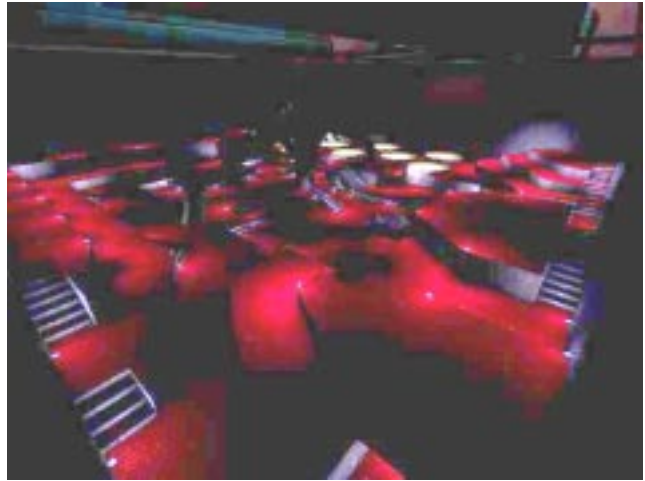
severely limit what is achieved in practice. Most of their budget and time is spent getting the engine and game mechanics up and running. And their product oriented focus is often at odds with research, rendering it virtually impossible to work directly with researchers at a University.

Esc Online

Esc Online (www.esconline.org) is an open source project to be released in beta form by March 2001. Created by volunteers in their spare time, it can naturally accommodate scripted narratives as well as more open-ended interactions. We hope to produce some agents of our own, as well as attract researchers to the environment.

In Esc, users will be able to chat while listening to the DJ spin, then watch the band perform. They can head to the dance floor and bust out their moves, which gets them better clothes, tickets to the back room or power ups. Some power ups will give you new abilities, others will change your appearance, and still others change the way the world looks. Users will be able to create their own dance moves or entire scripted performances complete with special effects. Or just take a few good friends to a secluded booth or overlook and talk the night away.

In Esc users will:

- Choose their name and body, and find different clothes and looks once online.

- Dance by pushing buttons in rhythm with the music, similar to the Playstation games Bust-A-Move and Parappa the Rapper.
- Chat using text tagged with emotions such as sarcastic, whispered, etc.
- Communicate using simple body language or animations such as waving.
- Acquire items that give new abilities, change their appearance, change the way the world looks, etc.
- Create their own dance moves or entire performances, complete with special effects. This could become a new medium and venue for machinima (movies made with 3D games.)
- Start their own club with music from their own CDs or MP3s.
- Leave messages for other people with the bartender.

Esc Online is built on the commercial video game Unreal Tournament. We replace almost every aspect of the game with our own, so that each multiplayer server becomes its own nightclub. Users will need to own a copy of Unreal Tournament to join.

Possible Applications of Believable Agents

Issues ripe for Artificial Intelligence abound. Many traditional problems, such as path planning, are already solved, leaving room for more expressive and open ended tasks such as the dynamic generation of camera moves. However, this paper will focus on the possible roles for believable agents.

One role is the personality of the bouncer, bartender and other computer controlled agents. The bartender can crack jokes, the bouncer can talk back to angry patrons, and the dj can chat with the crowd.

Another role is patrons at the bar. In real life clubs it is often socially acceptable to talk to strangers who are seated at the bar, but not at tables or booths. This is especially true for people sitting alone. Hearing other people's life stories or discussing current events can be an enjoyable way of passing the time. Creating a believable agent that has an interesting or funny life story is an interesting challenge. What makes conversations with random people interesting? How can we capture that in a program?

Having two or more agents interact with the user can enrich the situation. Beavis and Butthead, Laurel and Hardy, Jay and Silent Bob, and the cast of *Friends* all provide interesting dynamics. One possible model for this style of interaction is the robot improv of Bruce, et al.

Bruce et al. use real robots to improvise a simple scene where a man wants to leave a room and a woman doesn't want him to. They use simple models of appropriate emotions, namely frustration, fear, empathy and boredom. Emotions are affected by the character's actions and how well they are achieving their goal. In addition, all emotions have an equilibrium level that they decay toward at a controllable rate. By setting the equilibrium values and decay rates, they set the character's personality. The emotions and current goal determine the set of possible behaviors.

Each behavior has an associated goal, e.g. move to the door. Once a behavior is chosen, its associated goal becomes the character's current goal. The set of all actions is simulated to find the one which best satisfies the goal. A line of dialog is chosen and spoken while the action is performed. A message is then sent to the other character, which then takes its turn.

Patrons needn't be entirely passive either. For example, a computer controlled patron could start picking fights with other patrons, both computer controlled and human. The bartender could warn them, or eject them if they get too rowdy.

The band's show could be dynamically generated, using information about the song that is already available to the program. To judge the user's dancing, the program already knows where each beat falls and which parts are chorus or verse. That information and feedback from the audience can be used to create the band's stage show in real time. An M.C. could introduce each band. After the show, the band members could get a drink and chat with the audience.

More narrative elements are also possible. For example, in a user's conversation with the band, they could let slip that a drug ring is being run out of the club, and a large reward is being offered to whoever exposes them. By talking to people, watching them, following them, etc. the user can eventually discover who's behind the operation -- and decide whether or not to join them. If the drug ring is exposed, those involved could be arrested in front of the entire club. New characters that look remarkably similar could replace them, and the story could start over with the new guilty party/parties chosen randomly.

Two or more users could collaborate and experience the story together. In fact, the story could be arranged so that such collaboration is required to solve the mystery, which would encourage social interaction.

Action/romance movies such as The Matrix and Casablanca are another source of inspiration. For our purpose here, the key element of these movies is two people thrown together in dangerous times who fall in love. A user could experience it "single player," as one of the two lead roles, or two users could experience it together. This could make for an interesting way to meet and get to know someone. In real life, challenging or dangerous situations are thought to bring people together. How well would that work in a fictionalized, interactive setting? How would other genres adapt to this medium?

Designing characters for this medium is different than designing characters for a single player experience, much the way that writing for TV is different than for movies. In TV, characters develop over many episodes. It takes several viewings of a TV show to get a feel for the depth of a character, whereas movies have only an hour and half to establish characters and tell a story.

What sort of AI is needed to bring these to life? How well do current believable agent methodologies work in

this setting? How would we build computer controlled characters that the user interacts with night after night? Would they be like the cast of *Cheers*, or something different?

Technical Details

Esc Online is a total conversion of the commercial game Unreal Tournament. We replace all of the graphics, sounds and game logic with our own, using the original game as a graphics, physics and input engine. Esc only runs under Windows, although a port to Linux or MacOS would be straightforward. However, the development environment runs only under Windows and would be significantly more difficult to port. Unreal Tournament has its own scripting language, UnrealScript, based on Java and C++. Agent code is typically written in UnrealScript, although C++ is possible with some effort. Other languages that are binary compatible with C++, or can be interfaced to C++, are also possible.

UnrealScript has most of the features of Java. Like Java, it's byte compiled to a platform independent byte code for a virtual machine that is simulated at run time. UnrealScript simulates threads by calling each object's code in turn. This drastically simplifies interthread communication and eliminates the overhead of context switching, allowing thousands of independent threads. C++ can be used for computationally intensive tasks, or to extend UnrealScript's capabilities. Unreal Tournament's navigation and path planning, central components of First Person Shooters, are accessible through functions such as MoveTo(vector) and MoveToward(object).

Microsoft Visual C++ can be used as a development environment, although many programmers prefer the shareware tools UClasses or WOTgreal. They have syntax highlighting, a class browser, package and class management and many other useful features. The downside to UnrealScript is the lack of a debugger or profiler and lack of documentation of advanced features. The developer community largely offsets the last problem, more on that below.

Objects that correspond to characters or items in the world get callbacks for various events. For example, when two object touch in the virtual world, their Touch() functions are called. Other such events include AnimEnd(), ChatMessage(), HearSound(), SeePlayer(), Spawned(), Destroyed(), BeginState(), EndState(), PreRender(), PostRender(), and Tick(), which is called every frame. Other objects correspond to information displays, the windowing system, dynamic lights, the camera, the rules currently in effect, and the effects of the environment.

Every object is in one and only one user-defined state. An UnrealScript state is simply a list of events to ignore, a set of state-specific functions, and the top level code for that state. State functions can override global functions or be entirely new. Also, states can execute latent functions, i.e. functions that suspend the thread until certain conditions are met.

The input interface in UnrealScript is very convenient as well. In the default interaction mode, any key press (or release) can be set to call an arbitrary function while mouse movement rotates the character. The default mapping of keyboard to function is set by the code and customizable by the user. Before being dispatched to the function, all key presses pass through a function where they can be caught. In a separate mode, the mouse controls the traditional pointer to interact with a fairly full featured windowing system. In this mode the keys no longer map to functions, but are used for typing in text boxes and the like.

UnrealScript also provides integrated networking facilities for coordination between client and server. Variables on the server can track those on the client, or function calls on one can be executed on the other. Access is also provided to UDP and TCP sockets. In fact, Unreal Tournament comes with a simple IRC client, a simple web browser and a simple web server, all implemented in UnrealScript.

Every model has a finite set of animations created with the model. For example, the minimum for Esc is one for walking, one for running, and four dancing animations. These animations are created along with the model and can't be changed at run time. However, the code can play back subsequences or individual frames. Also, any two frames can be linearly interpolated at run time, which helps make transitions smooth.

The Community and The Team

There is a sizable community of programmers, virtual architects and model builders for Unreal Tournament who openly share information and assistance. The UnrealScript mailing list has over 200 members, including a professional game programmer who works full time with the source. They're typically good at describing how to implement something, and often provide sample code. There are literally hundreds of character models for Quake and Half-Life that can be converted to Esc.

On top of all that, we want to work with you. We can help you with technical problems, and may be able to provide animations, models and virtual spaces. We're planning our first public release by the time of the symposium.

References

"Robot Improv: Using drama to create believable agents," Allison Bruce, Jonathan Knight, Sam Listopad, Brian Magerko and Illah Nourbakhsh. In *Proceedings of ICRA 2000*.
http://www.cs.cmu.edu/afs/andrew/scs/ri/robotimprov/www/robotimprov.html