Behavioral Scripting:
Method Acting for Synthetic Characters

Athomas Goldberg
NYU Media Research Lab
719 Broadway, 12th Floor
New York, NY 10003
athomas@mrl.nyu.edu
http://www.mrl.nyu.edu

Introduction

"Procedural Animation" is a technique recently developed at the Media Research Lab that allows us to create lifelike responsively animated characters in real time [1][2] Based on the success of this research we have begun exploring the possibilities of applying these techniques to computer games and other scenarios in which human participants can interact with animated characters and each other in a shared virtual environment.

Procedural Animation For Interactive Fiction

The "First Person POV" interface used in games like "Doom" gives the user a direct control over their movement and actions, but limits the user to rotating left and right, moving forward and back and firing. Also, the camera can only view one type of shot, so the number of cinematic possibilities is diminished.

If instead we make use of Avatars, animated characters that represent the user in the simulated environment, then we open up the possibilities enormously. We can then direct the character to do any number of things through high-level control and allow the character to play the scenarios out in a naturalistic and dramatic fashion. Instead of the clunky navigation of directing the movement, step-by-step, (how many times have you run into a wall or missed a door?) which isn't very realistic, you can simply point to a door, for example, and tell the character to open it. The character would then cross to the door automatically, avoiding obstacles in the way, and open it. We've also freed up the constraints on the camera. If we treat the camera as another character in the scene, we can then give it its own set of behaviors. The camera can react to the action in the scene and adjust the shot accordingly, giving us a type of real-time cinematography that isn't possible in POV-type games.

"The Tail That Wags The Dog" Inverse Causality for Animation

Animated characters in a dynamic real-time environment are likely to encounter new situations, especially if we allow for characters that persist from one scenario to the next. A traditional AI approach would suggest that we build mechanisms into the character to allow them to analyze new situations and make judgments as to the appropriate course of action to take (in other words, the right animation to perform.)

This doesn't, however, account for situations where we want the character to appear to know what they're doing, even though they have, in fact, never been in the situation before. We don't want the characters to look like they're learning how to drink a cup of coffee; we just want them to drink. We could try to take into account every possible thing the character might encounter and program him/her to perform

appropriate animations. The problem with this is we end up with enormous character files and eventually we'll want to introduce new props and situations that we hadn't foreseen at the time the character was created. At this point it's too late to go back and rewrite every character to account for the new additions.

But what if we build the animation for using something into the thing being used?

What if, instead of teaching the characters how to drink from a coffee cup, we teach the cup how to be drunk from? Each item the character encountered would have all the instructions for its use built into it, requiring only that the user direct his or her attention to it in order to enable its use. The player points to the coffee cup which sends a command to the cup to display the set of options for its use. (drink, throw, break, etc) The player then chooses one of these options at which point the cup sends a command to the animated character, which contains the appropriate animation, and the character performs it. In a sense, the cup is controlling the character. The cause and effect relationship is reversed, but from the user's point of view the action appears natural. The advantage of this is that we can continually introduce new elements into the environment, because the animations for those elements' uses are included in their design. This method enables us to have an infinitely expandable variety of possible actions/animations available to the character while reducing the character file to a physical description of the character and a set of behavioral modifiers that are used to alter the flavor of the animations giving each character a unique body language and personality. This leads us to the next issue: While there might be millions of actions possible for a single character, we only want to deal with those that are relevant to the given situation. The use of this technique provides an efficient technique for handling this. When the user directs the character into a new environment, that user is only receiving options from the objects in that environment and therefore is limited to those actions that are relevant and feasible. When the user directs the character into the new environment, a new set of options becomes available.

"I Feel Like A Number" Character Definition in Behavioral Scripting

The character's file then consists of a physical description of the character (the 3D Model) and a set of Behavioral Modifiers used to give the impression of a unique personality. Some examples might include: strength, coordination and health, which would govern how a character performed certain physical activities, and intelligence, aggressiveness and amiability, which would affect how the character interacts with other characters. These are examples of relatively static modifiers, which might change slowly over time (one's general strength level isn't likely to vary widely from day to day) but we can also have modifiers that are dynamic, like energy or disposition, which might change continuously depending on the time of day or how long it's been since the character last ate.

The animations a character performs reflect the values of one or more of these modifiers. The current Energy, Coordination, Disposition and possibly other modifiers might influence the way a character walks at any given moment. To be effective, we must carefully determine what influence each of these modifiers has on any given animation so that all the characters actions appear consistent with the personality we've chosen for them. In turn, performing certain actions might have an effect on the values assigned to certain modifiers. For example, drinking beer could adversely affect a character's Coordination, which would then affect actions like walking, which utilize that modifier.

In this way we can have characters that are continuously dynamic, who appear to grow and change over time due to their experiences. These experiences are

reflected in the current state of their behavioral modifiers.

"Sim College" Educating Animated Characters

Other modifiers might represent a character's skill at performing certain activities, like playing darts or driving a motorcycle. When a character encounters a prop for the first time, the file containing the character's modifiers is searched for the relevant skill, in this case the "Drive Motorcycle" skill. If the skill isn't found, it is appended to the character file at some minimal level. This starting level may be modified based on other characteristics. (A character's Intelligence, Coordination and/or skill in driving other types of vehicles may influence how well the character drives a motorcycle her first time on it.) As the character spends time riding the bike, the Motorcycle Driving skill value increases and the animation will reflect the increased proficiency. Next time the character comes across a motorcycle, the character's file is once again searched for the "Drive Motorcycle" skill and this time the value is found. She now appears to drive like an experienced biker. The animations a character performs will reflect the value of one or more of these modifiers. To be effective, we must carefully determine what influence each of these modifiers has on any given animation so that all of the character's actions appear consistent with the personality chosen for them. One part of our research is dedicated to mapping values assigned to these abstract personality traits to the algorithms controlling the animation of various actions. In turn, performing certain actions will have an affect on the values assigned to these modifiers, reflecting the changes resulting from experience.

Conclusion

With Behavioral Scripting we can have characters that demonstrate unique personalities, react to a wide range of continually changing situations and appear to grow and change over time due to their experiences. These are all elements essential to the creation of lifelike believable characters. While this doesn't in itself guarantee brilliant interactive experiences, it does go a long way toward providing a competent set of performers to work with.

References

[1] Ken Perlin, "A Gesture Synthesizer" From the Animation Tricks course notes, SIGGRAPH (July, Orlando, FL) 1994.

[2] Ken Perlin, "A Remarkably Lifelike Implementation of a Synthetic Computer Character" In the proceedings of Lifelike Computer Characters (October, Snowbird, UT) 1994