Rapid Development of Characters in FPS/3PS Games Using Visually-Specified Behavior-Based Control

G. Michael Youngblood, Frederick W. P. Heckel, D. Hunter Hale and Arthur Carroll

University of North Carolina at Charlotte
9201 University City Blvd.
Charlotte, NC 28223
{youngbld, fheckel, dhhale, artcar12}@uncc.edu

Abstract

First/third-person simulations in virtual environments have become increasingly used in training; however, creating intelligent, interactive characters to populate these environments presents a large authorial burden. Our work focuses on building tools to enable rapid creation of intelligent characters for first/third-person game-like environments with no programming knowledge required by the user. This is made possible using behavior-based control combined with a user interface employing natural language-like character specification in the form of English sentences and interactive testing during development.

Introduction

Simulation training (i.e., *Serious Games* or *Games with a Purpose*) utilizing first and third-person interactions have played an increasingly important role in facilitating mission rehearsal, environment familiarization, and cultural awareness. The two most important parts of any immersive game are the environment and the characters in the environment. Typically, developing any new scenario requires building a new environment and new characters. This process is almost always time critical, as the earlier the scenario is developed, the more it can contribute to effective training. It is important to develop intuitive tools that facilitate the rapid development of such scenarios. While there has been significant progress in developing tools to model environments, the tools to model interactive characters are lacking. Often the design of interactive characters requires programming, which considerably slows down the process of building scenarios. This demo demonstrates how behavior-based control can facilitate the rapid development of intelligent, interactive characters without traditional programming.

Our work focuses on the use of behavior-based control in first/third-person training simulations, initially reported in (Heckel et al., 2009) with the results of a small pilot study and a more extensive usability study in (Alexander et al., 2010), which was conducted with an updated version of our DASSIEs (Dynamic Adaptable Super-Scalable Intelligent Entities) research tool. Participants of the study (the majority not from a computer science background) were provided a text description of characters in a scenario and were asked to design one of those characters using our BehaviorShop tool. The study involved a total of 13 different baseline character descriptions, divided into five scenarios, and over one hundred participants. Results indicate that at least 80% of the participants were able to build characters with at least 80% behavioral accuracy (based on compatibility with text descriptions of each scenario).

Behavior-based Control Using BehaviorShop and BEHAVEngine

DASSIEs is our primary research framework, comprised of three main components that together implement behavior-based control of characters in a first-/third-person game environment. These key components are BehaviorShop, the character design tool, BEHAVEngine, the AI controller that instantiates characters from BehaviorShop-produced specifications, and FI3RST (FIrst- and 3rd-person Realtime Simulation Testbed), which is a wrapper around game engines to provide a standard interface for BEHAVEngine—FI3RST currently supports the Panda3d (www.panda3d.org) and Irrlicht (irrlicht.sourceforge.net) game engines, but is easily adapted to others.

BehaviorShop, the component that allows users to build intelligent, interactive characters using behavior-based control, has an intuitive user interface based around using sentence-like constructions to define character behaviors. Screenshots of the behavior layers window (where the user defines the layers) and the trigger-action editor are presented in Figure 1 and Figure 2, respectively.

Figure 1. Behavior layers are constructed individually with the lowest priority layers on the bottom. Layers can be reordered by dragging them to a new location with respect to the other layers.

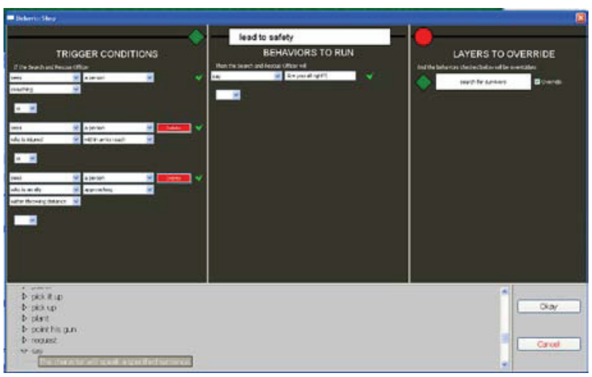

Figure 2. Each behavior layer is defined by selecting options to fill out an if-then sentence structure, possibly containing multiple trigger conditions (disjunctions or conjunctions) and multiple resulting actions (to occur sequentially or in parallel).

The behavior layers window is used to add, delete, and prioritize layers. The trigger-action editing is used to select the trigger(s) and action(s) defining a single behavior layer and to define the layer's subsumption policy over lower layers (i.e., which layers it can/cannot run in parallel with). Often, special actions performed by the characters require extra configuration that is handled by data-specific popups (e.g., map positions). At any, the user can test and debug the character by watching the simulation in the output window as depicted in Figure 3.

Each behavior layer in BehaviorShop is defined by selecting choices to fill out an if-then sentence, possibly with multiple triggers and/or actions. For this reason, the

Figure 3. Users can interactively test characters in the simulated environment during the creation process.

vocabulary presented to the user is very important. The language in our early prototype was based on developer opinion, a practice commonly referred to as armchair design. Of course, the HCI community has long been aware of the difficulties with this approach (Furnas et al., 1987). To bring our interface vocabulary more into line with the vernacular, we conducted a study in which participants were asked to read a brief scenario description and provide free-form text instructions for a selected actor and to watch a short video clip and describe the actions of one of the actors in the scene. From this vocabulary study, we were able to present a more natural syntax in our interface as well as to ensure we included the most commonly used words for describing behaviors.

Characters defined by BehaviorShop are executed by BEHAVEngine in conjunction with FI3RST and the game engine. BEHAVEngine constantly receives percepts for the characters in the simulation, computes the appropriate actions based on the character specification, and passes the action messages on to FI3RST. These action messages are interpreted by FI3RST and appropriate action animations (e.g., walking, jumping, and shooting a target) are chosen from a library of basic actions and played in the game.

References

Alexander, G., Youngblood, M. Heckel, W.P., Hale, D.H., and Ketkar, N. 2010. Rapid Development of Intelligent Characters in First/third-person Training Simulations via Behavior-based Control. *In Proceedings of the 19th Behavior Representation in Modeling & Simulation Conference (BRIMS 2010).*

Heckel, F. W. P., Youngblood, G. M., and Hale, D. H. 2009. BehaviorShop: an intuitive interface for interactive character design. In *Proceedings of the Fifth International Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2009)*.

Furnas, G. W., Landauer, T. K., Gomez, L.M., and Dumais, S.T. 1987. The vocabulary problem in human-system communication. *Commun. ACM*, 30(11), 964-971.