“Low Level” Intelligence for “Low Level” Character Animation

(Online ID: sketches_0152)

Damian Isla Bruce Blumberg
MIT Media Lab, 20 Ames St.
Cambridge, MA 02139
{naimad, bruce}@media.mit.edu

Abstract

It is suggested that low-level cognitive modeling can allow for a rich set of low-level behavior to be produced. For example, a character that has the cognitive capability of Object Persistence can intelligently direct its gaze over a scene and even respond emotionally to certain world events. This level of cognitive modeling allows for complete behavioral control by a human controller or a script, if that degree of control is necessary for the application.

CR Categories: I.2.0 (Artificial Intelligence): General – cognitive simulation

Keywords: AI and Behavior, Animation Techniques

1 “Low Level” Animation and Intelligence

We use the term “low level” to designate animation such as eye-motion, gaze control and facial expression – character animation, in other words, that has less to do with what a character is doing and more to do with how the character is feeling, the character’s personality, etc. Part of the irony of the term is that while the motion is physically minimal, its impact can be subtle and very powerful. It is also, in some cases, tedious or impossible to script beforehand, for example in interactive computer games where events must be intelligently recognized and reacted to on the fly.

An AI solution to the above might seem natural, since much of the low-level animation described is significant precisely because it is indicative of some kind of emotional or knowledge-state internal to the character. If a character frowns and continually glances towards a door, we might infer that it is because the character is anxious about someone soon coming through it. Much of the previous work in behavioral modeling has focused on the problem of action selection. This work, however, shows that cognitive models could be used on occasions where complete character autonomy is undesirable (where the activity of the character is scriptable beforehand, or is under a player’s control).

3 Object Persistence

As an example of a cognitive capability that can produce much interesting behavior, we will discuss the phenomenon of object persistence. If a character tracking an object sees the object pass behind an occluding obstacle, an unsophisticated character will simply say “I no longer know where the target is”. A slightly more sophisticated character might say “the target is at the location I last observed it.” A character with a full-blown model of object persistence would say “the target is behind the obstacle (and furthermore, based on the velocity it had when I last observed it, I expect it to come out again in x seconds).”

4 Implementation

A model of object persistence can be implemented as follows: the character’s environment is divided into a grid of discrete locations called a “Probabilistic Occupancy Map” (POM). This map is a two-dimensional discrete probability distribution representing the possible location of a single target object. When an observation of the target object is made, the grid element the target is in is given a probability of 1. For every time-step of the simulation in which the object is not observed, each grid element diffuses some portion of the probability it contains to its immediate neighbors. Thus the longer the target is unobserved, the “fuzzier” the character’s idea of the target’s location. However, in maintaining this distribution, probability is always culled from locations that are currently wholly visible to the character (based on a limited field of view, and the inability to see through occluding obstacles). This ensures that the character’s ideas of where the target is remain “realistic”. Each time-step the distribution is also renormalized. Thus a location unlikely on one time-step might be much more likely the next, if the most likely location has been rejected (perhaps the character, in searching for the target, has gone behind the occluder only to find that the target is not there. In that case, the much less likely-seeming possibility that the target might have reversed directions and hidden in another location should be considered much more seriously).

5 Example Results

The above scheme was implemented in an animated character named Duncan, a virtual sheepdog. Duncan was instructed by a virtual shepherd to keep track of a sheep (both shepherd and sheep being controlled interactively) as it moved about a simple environment with two occluding walls. Duncan could be instructed to “look at the sheep”, “go to the sheep” and “go to the shepherd”. Interesting new behaviors included:

- Emergent look-around: by instructing Duncan to look at the sheep after he had not observed it for some time, Duncan used his POM to intelligently direct his gaze toward the sheep’s most likely location. When the sheep was especially elusive, Duncan would glance back and forth over the scene in a natural and evocative way.
- Emergent search: when instructed to “go to the sheep” (assuming the sheep was hidden behind one of the occluders) Duncan would go check out the most likely-seeming hiding place. Not finding it there, he would intelligently and systematically search the environment until the sheep was found (or until he became so frustrated that he gave up).
- Secondary Emotions: by taking into account the likelihood of the location the sheep was eventually found in, Duncan could be made to display certain secondary emotions such as surprise (to have found the sheep in a place it was not expected to be) and confusion (at NOT finding the sheep in a place it was expected to be).

While only a very preliminary example of “low level” intelligence for “low level” animation, we believe that this level of cognitive modeling could prove extremely fruitful for crafting the next generation of intelligent (and intelligent-LOOKING) characters.

References

D. Isla, B. Blumberg. *Object Persistence for Synthetic Creatures*. In the Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), Bologna, Italy. To Appear July 2002. (http://web.media.mit.edu/~naimad/aa2002-isla.pdf)