

# A Demonstration of ScriptEase Motivational Ambient and Latent Behaviors for Computer RPGs

Maria Cutumisu, Duane Szafron, Jonathan Schaeffer, Kevin Waugh, Curtis Onuczko, Jeff Siegel, Allan Schumacher

Department of Computing Science, University of Alberta  
Edmonton, Canada

{meric, duane, jonathan, waugh, onuczko, siegel, schumach}@cs.ualberta.ca

## Abstract

This demonstration describes the generation of *ambient* and *latent* NPC behavior scripts using generative *behavior patterns* with ScriptEase. Our behavior model supports behavior roles, a powerful mechanism that allows an NPC to change behavior sets during the story. Our motivation model for selecting ambient behaviors generates more realistic NPC behaviors and our novel collaboration protocol simplifies and extends the NPCs' collaborations with a broader range of NPCs. We demonstrate motivational ambient and latent behaviors for a guard NPC in a custom *Neverwinter Nights* game module. With ScriptEase, game authors can use, adapt, and create behavior patterns that generate complex, versatile, and engaging NPC behaviors.

## Introduction

Non-player character (NPC) behaviors in computer role-playing games (CRPGs) have a significant role in the enjoyment factor of interactive stories, but game authors find them difficult to implement and adapt. The increasing complexity of behaviors, especially for collaborative NPCs, makes manual scripting impractical, especially for non-programmers. Therefore, in games such as *Fable* and *Morrowind* the NPC behaviors are repetitive and characters rarely interact with each other. NPCs walk predefined paths, make random comments about the player character (PC), stand still or perform a simple animation. In *The Sims 2* ambient behaviors are more developed, but they are dependent on the design model integral to the game. Although the *Oblivion* "radiant AI" system provides rich ambient NPC behaviors, there are still cases in which NPCs do not react if the PC steals from them (Spronck 2006), casts spells, or attacks nearby NPCs.

We distinguish NPC behaviors on two axes, *independent* vs. *collaborative* and *ambient* vs. *latent*. A behavior is further classified as *proactive* or *reactive*. An *ambient* behavior is spontaneously initiated by an NPC (*proactive*) or performed in response to a collaborative behavior initiated by another NPC (*reactive*). A game event may interrupt the NPC during an ambient behavior, triggering a

*latent* behavior. When the latent behavior completes, the NPC resumes the interrupted ambient behavior.

We highlight our behavior model using a custom *Neverwinter Nights* (NWN) module that features three guard NPCs with ambient and latent behavior scripts generated from ScriptEase behavior patterns (ScriptEase 2007). A guard NPC patrols, rests, checks that the guarded item remains in its chest, and converses with other guards. At the end of the shift, the guard goes to a tavern, talks to other patrons and orders drinks. If the guarded item is stolen, the guard flees the area. To create similar complex behaviors in thousands of background NPCs, we need to eliminate manual scripting. We must create reusable behaviors that can be quickly and reliably adapted to a wide range of NPCs. We developed a mechanism that generates engaging NPC behaviors without writing code.

## Ambient Behavior Patterns

A *behavior pattern* encapsulates *independent* (NPC acts alone) and *collaborative* (NPCs interact) behaviors.

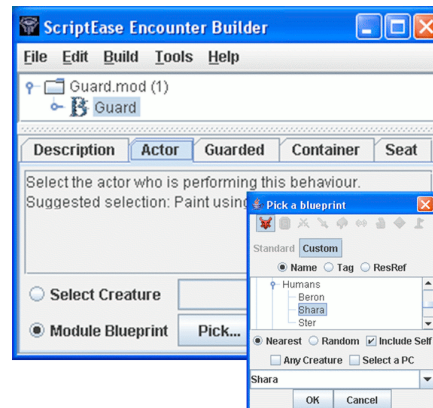


Figure 1. Using a **Guard** behavior pattern in ScriptEase.

The Guard pattern instantiated in Figure 1 has four *proactive* ambient behaviors: **Patrol** near the guarded object (e.g., chest) in a non-deterministic manner, **Rest** on a bench, **Check** the guarded chest, and **Converse-Talk** on a topic (e.g., "weather") with a friendly NPC. The guard also has a reactive ambient behavior, **Converse-Listen** (topic "weather") to respond to any NPC with a proactive **Converse-Talk** on this topic, as shown in Figure 1. The collaborative model looks for any NPC with a reactive

behavior on that topic who makes *eye-contact* and who satisfies any conditions included in the initiator's collaborative behavior. Collaborative behaviors are rare in CRPGs because their existence complicates event synchronization. We developed a concurrency control mechanism that solves the synchronization problems (Cutumisu 2006). An *eye-contact* mechanism prevents new ambient behaviors from being initiated while an NPC is performing other ambient or latent behaviors.

## Latent Behavior Patterns

A *latent* behavior is an NPC's reaction to an appropriate *cue* (external event). These behaviors are not considered ambient, since the NPC does not choose to perform them.

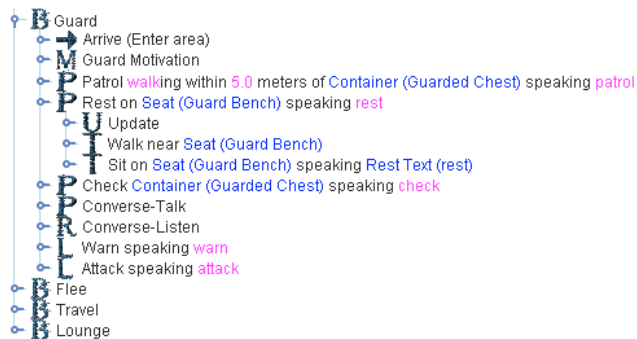


Figure 2. The structure of a behavior pattern.

The **Guard** behavior pattern illustrated in Figure 2 shows two *latent behaviors*, **Warn** when the intruder is close to the chest and **Attack** when the guard sees the intruder stealing the item. A latent behavior may be independent (**Warn**) or collaborative. For example, the PC may cue a latent collaborative behavior (**Converse-Talk** with topic “clue”) on the guard that starts a conversation with another NPC to reveal an important story clue to the PC. In our model, an inattentive *Oblivion* merchant (Spronck 2006) can respond to a PC/NPC, behaving in a more realistic manner, by using appropriate latent behaviors.

## Demonstration Overview

This demonstration shows the process of generating complex *ambient* and *latent* behaviors with ScriptEase. The generated guard behaviors can be viewed in a series of movies (Guard Behavior Movies 2007) that illustrate motivational ambient and latent behavior patterns for three guard NPCs in a custom module that we developed using *NWN* and ScriptEase. Shara guards an item stored in a chest, Norman initiates a conversation with Shara from time to time and then returns to his original location, and Granan arrives at the scene to replace Shara at the end of her shift. The first movie shows ambient independent behaviors selected on the basis of the **Guard Motivation** (with attributes *duty*, *tiredness*, and *threat*). A motivational model allows the NPC to select a proactive behavior based on motivation attribute values. The second movie shows how an ambient collaborative reactive behavior interrupts an ambient independent behavior, since ambient

collaborative behaviors have a higher priority than ambient independent behaviors. After the ambient collaborative behavior is complete, the ambient independent behavior is resumed. The third, fourth, and fifth movies show how a latent collaborative/independent behavior interrupts an ambient collaborative reactive behavior, since latent behaviors have higher priority than ambient behaviors. Two other movies illustrate the PC interrupting both an ambient collaborative behavior and a latent collaborative behavior. The last two movies illustrate how cues can also be used to change behavior roles when an event-based or a timer-based cue is triggered. Granan changes his role from **Guard** (activated by an **Arrive** cue when the guard enters the area of the container) to **Flee**, leaving the area when he notices that the guarded item is gone. Shara changes her role from **Guard** to **Travel** when her daily shift ends. A game author can create an interactive CRPG story with complex ambient and latent behaviors using ScriptEase behavior patterns. The author opens a game module in ScriptEase and instantiates the desired behavior patterns, as illustrated in Figure 1 for the **Guard** behavior pattern, setting options to the guard (Shara), the item, the container, and the seat. Depending on the author's intent, the pattern can be further adapted. The module is saved and compiled with ScriptEase to automatically generate *NWScript* code for the module. Finally, the module is played in *NWN*.

## Conclusion

This demonstration illustrates how *ambient* and *latent* behaviors can be generated for custom *NWN* modules to create more engaging interactive stories. ScriptEase behavior patterns hide the level of complexity inherent in the motivational ambient and latent behaviors for interacting NPCs. Our generative pattern technique allows the game author to rely on a catalog of pre-defined behavior patterns for diversity and to generate the scripts for these behaviors without manual scripting. These behaviors are general enough to cover the behaviors of NPCs in other settings, such as a tavern, commonly occurring in CRPGs (Cutumisu *et al.* 2006). The ScriptEase behavior patterns generate versatile NPC behaviors, since they allow the selection of possible ambient behaviors dynamically, based on the NPC's current motivations and the world state.

## References

- Cutumisu, M., Szafron, D., Schaeffer, J., McNaughton, M., Roy, T., Onuczko, C., and Carbonaro, M. 2006. Generating Ambient Behaviors in Computer Role-Playing Games. *IEEE Journal of Intelligent Systems* 21(5) 19-27.
- Guard Behavior Movies. 2007. <http://www.cs.ualberta.ca/~script/movies/aiide07>.
- ScriptEase. 2007. <http://www.cs.ualberta.ca/~script>.
- Spronck, P. 2006. Adaptive Entertainment. *AIIDE*, Marina del Rey, USA. <http://www.aiide.org/aiide2006/speakers/spronck.ppt>.