Twig: A Simple, AI-friendly, Character World for Believable Agents

Ian Horswill

Departments of EECS and Radio/Television/Film, Northwestern University
2133 Sheridan Road
Evanston, IL 60208
ian@northwestern.edu

Abstract

There are a number of pieces of core infrastructure that are missing if we want to build intelligent narrative systems. One is the relative lack of availability of intelligent bodies that support wide behavioral repertoires and dynamic worlds. In this paper, I describe *Twig*, a simple physics and motion control system optimized for believable agents. Twig is open-source, reasonably fast, and runs under XNA Game Studio 3.0, making the importation of sets, props, and so on (comparatively) easy. It is being used for a semi-regular "webcomic".

Motivation[1]

Immersive, interactive narrative involves a large number of capabilities ranging from drama management to motion planning, facial animation, natural language generation, physics, etc. Although great progress has been made on individual capabilities, complete, AI-based, interactive narrative systems, such as (Mateas and Stern 2005), are still relatively rare. This is due at least in part to two basic problems. The first is that building a complete system from scratch is prohibitively expensive for all but the best-resourced labs. One is then forced to use a commercial game engine, which is not necessarily convenient for interactive narrative. A second problem is that they tend to have highly variable levels of realism. For example, they may have highly realistic, professionally designed character models such as those from Half Life 2, but then have dialog generated from a speech synthesizer, or motions that are clumsily sequenced, resulting in an uncanny appearance (Mori 1970).

In this demonstration, I will show work in progress on *Twig* (Horswill 2008), a simple, extensible, engine that provides basic physics, characters, and reactive behaviors, suitable for many interactive narrative applications. In essence, it provides a stage and a set of actors that can be directed to move about and interact with props and with one another. Although better systems are available for individual capabilities, such as gait generation (Kuriyama, Kurihara et al. 2002; Hase, Miyashita et al. 2003; Natural Motion Inc. 2006), reaching (Kallmann 2008), physics (Müller, Heidelberger et al. 2007), etc., *Twig* has the advantage of providing a relatively broad range of functionality in a relatively small, fast system. It is open source, supports the XNA Content Pipeline (Microsoft 2008), making it relatively easy to import props and other art assets into the system, and should be straightforward to port to the Xbox 360, although this has not yet been tried.

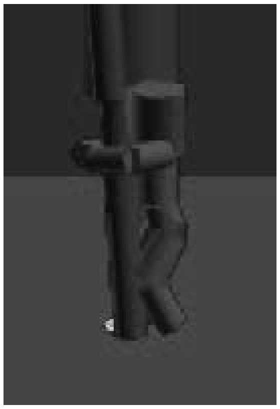

Figure 1: A Twig character hugs a parent

Twig is designed to support believable, cartoon-like animation, and so is not appropriate for all interactive narrative systems. Its character models are simple, and its physics and behavior controllers are intended to produce believable, emotive behavior, not cinematic realism. However for those looking for a simple character-and-world engine designed to play well with AI systems, it should prove useful.

Twig is presently being used in two main systems, one a simulation of parent/child attachment behavior, is relatively AI-heavy, although is focused on behavior-based AI rather than planning and problem solving. The other main application is a semi-regular "webcomic" (Horswill 2008). The webcomic is not AI-heavy, and current episodes aren't even interactive. However, it's a useful way to drive feature development in the engine.

System Overview

Twig is implemented as a library that extends Microsoft's XNA 3.0 Framework. It provides a simple mass-aggregate physics engine that supports constraint satisfaction. Objects are represented as collections of point-particles connected by rigid rods, together with a set of collision volumes and meshes for rendering.

From a user's standpoint, one uses Twig by linking in the library, creating a subclass of `TwigGame` (in turn, a subclass of XNA's `Game` class), and populating it with one or more instances of the `Person` class (the base class for humanoid characters) and various props.

Character Control

The characters can then be directed, either by adding behavioral code directly, or through a scripting interface. Scripting mode can either read the script from an XML file in the Content Pipeline, or the system can be run as a server, allowing non-XNA code, or code on other machines, to control the world. An example of the script code is show in figure 2. Scripts simply consist of sequences of method calls into the objects, along with some limited timing logic. In particular, methods can return information to the script interpreter for testing when an action has completed.

```
titles: "Episode 0: Alpha test"
pause 4
titles: fadefromblack 1
Michael: say "Script ..." Bryan
Bryan: holdforuse script
Bryan: say "Check!" Michael
Michael: say "Physics ..." Bryan
Bryan: lookat script
Bryan: drop script
pause 0.5
Bryan: say "Check!" Michael
```

Figure 2: Example script fragment

Behaviors

Twig is a work in progress. It also supports a relatively small number of character behaviors:

- Locomotion: Approach, PainWithdrawlReflex (triggered automatically when character is hit), TeleportTo, PoseAt
- Prop manipulation: Hold, HoldForUse, Drop, SitOn, WriteOn
- Social: Say, LookAt, Fight, Attach (run to and hug), Gesture

Other behaviors can be added, either as reactive (self-triggering) behaviors, or as discrete actions.

Props

The system currently supports a small number of props needed for testing purposes or for the webcomic. These include balls, trees, a (functional) merry-go-round, tree stumps, pen and paper, and boxes.

Passive props can be easily designed using standard modeling tools such as Maya, SoftImage ModTool, or Google SketchUp, and then imported as models through the XNA Content Pipeline. If a passive prop can be approximated using a box for its collision volume, then it can be imported by making an instance of the `BoxModel` class and specifying its rendering mesh. Importing props that have articulations, more complicated collision volumes, or behavior, requires writing C# code.

Implementation

Core physics

Twig provides a mass-aggregate physics system that supposes geometric constraint satisfaction. The system is based on Jakobsen's work on the Hitman engine (2001). Objects are represented as point-masses (particles) connected by rigid rods (distance constraints). These basic structures are then fleshed out with meshes for rendering, and collision volumes.

Collision handling Collision volumes of objects are represented as sets of primitive collision volumes (spheres, capsules, and oriented bounding-boxes). The engine handles basic collision detection and response, as well as generating percepts (touch, pain) for higher level AI systems.

It should be noted that while the physics is accurate enough to (usually) look real, no attempt is made to be truly accurate, and there are a number of situations in which the system wildly violates conservation laws.

Behavior system

Motor control is implemented primarily through a behavior-based system (Arkin 1998), i.e. a set of simple feedback controllers and state machines, gated by "activation" levels that are continuously updated; when behaviors are in competition for a given limb or other resource, an arbitration system selects the current most-active behavior.

Discrete action system

Operations which are most naturally thought of as discrete actions, such as sitting down, saying a phrase, or writing on a sheet of paper, are implemented through a discrete action system that tracks which actions are currently executing and can distinguish successful completion of an action from action failure. The discrete action system is intended primarily for use in conjunction with a reactive planning system, such as ABL (Mateas and Stern 2002)

Attention system

Characters also implement a short-term memory system used for directing behaviors and visual attention when the characters are running autonomously (rather than being scripted or slaved to an external planner). The attention system periodically scans all objects in view and in short-term memory and evaluates their salience (importance), valence (goodness/badness), and monitoring priority (how much to keep an eye on the object). The highest salience object becomes the new focus of attention, which is then monitored by the motor behaviors, and the highest monitoring priority object (if different) becomes the gaze target. This allows characters to move around and shift their attention between objects relatively naturally.

Obtaining Twig

Twig is distributed as free, open-source software under the Lesser GNU Public Licence (LGPL v.3). The source distribution, example programs, and further documentation are available from http://twigblog.wordpress.com.

Conclusion

Twig is very much a work in progress. It's missing a large swath of features one would eventually want to have, such as facial animation, support for arbitrary character meshes, and reach and grasp planning. Most of these features will be added in time. Part of the idea of making episodic content such as a webcomic is to use the production of the comic as a way of focusing the development of new features.

In the mean time, *Twig* provides a relatively lifelike set of character motions that match the cartoon quality of the rendering fairly well, making the overall effect believable, even if it is not realistic per se. I hope that between it being open source and being designed specifically to interface with AI systems that it will prove to be a congenial environment for intelligent narrative researchers.

References

Arkin, R. (1998). *Behavior-Based Robotics*. Cambridge, MIT Press.

Hase, K., K. Miyashita, et al. (2003). "Human gait simulation with a neuromusculoskeletal model and evolutionary computation." *The Journal of Visualization and Computer Animation* **14**(2): 73-92.

Horswill, I. (2008). Lightweight Procedural Animation with Believable Physical Interactions. *Artificial Intelligence and Interactive Digital Entertainment*. Stanford, CA.

Horswill, I. (2008). "Twig (webcomic)." from http://twigblog.wordpress.com

Jakobsen, T. (2001). Advanced Character Physics. *Game Developer's Conference*. San Jose, CMP Inc.

Kallmann, M. (2008). "Analytical inverse kinematics with body posture control." *Computer Animation and Virtual Worlds* **19**(2): 79-91.

Kuriyama, S., Y. Kurihara, et al. (2002). "Physiological gait controls with a neural pattern generator." *The Journal of Visualization and Computer Animation* **13**(2): 107-119.

Mateas, M. and A. Stern (2002). "A Behavior Language for Story-Based Agents." *IEEE Intelligent Systems* **17**(4): 39-47.

Mateas, M. and A. Stern (2005). Façade.

Microsoft (2008). XNA Game Studio 3.0, Microsoft Corporation.

Mori, M. (1970). "Bukimi no tani." *Energy* **7**(4): 33-35.

Müller, M., B. Heidelberger, et al. (2007). "Position based dynamics." *J. Vis. Comun. Image Represent.* **18**(2): 109-118.

Natural Motion Inc. (2006). Euphoria:core motion synthesis library, Natural Motion, Inc.