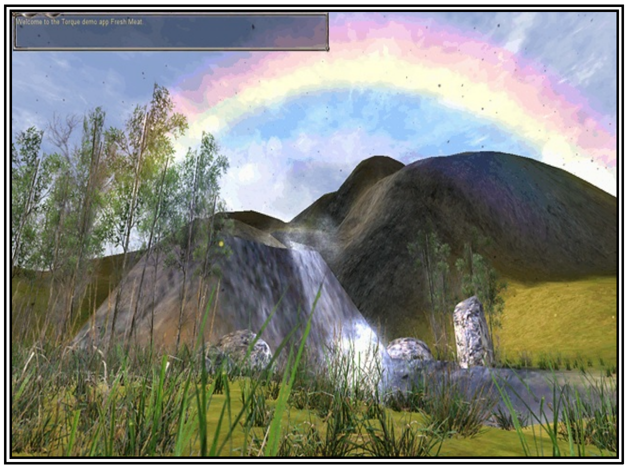
A 3D Game Development Course Using Torque

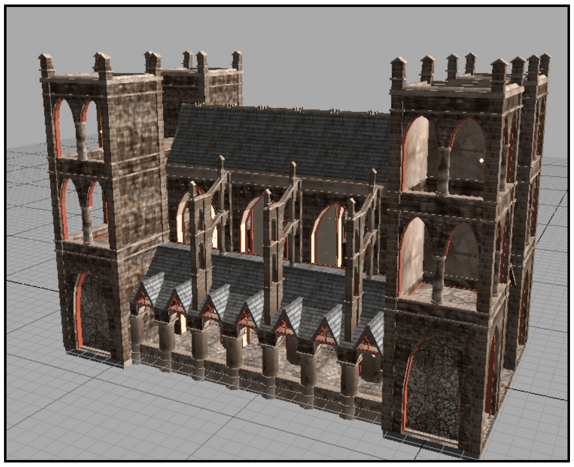
Dave Anderson – February 21, 2010


Welcome to the Torque demo app Fresh Meat.

A Torque game landscape, with atmospheric effects and abundant foliage.

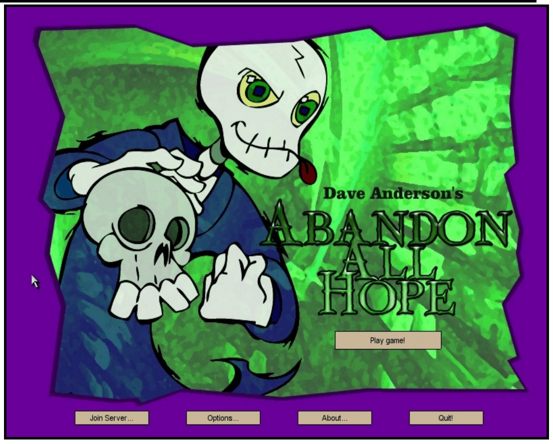
For the past few years, I've been offering a course at Holyoke Community College in developing 3D computer games using a commercial game system. The Torque Game Engine is a high-quality system that's widely used in college computer programs, and is even used in some high schools. Torque is fairly easy to work with, allowing students to go through a fairly complete introduction to the field in a single semester.

A large part of the appeal of Torque is that Garage Games, the publishers, give away the basic system for free, as an incentive to purchase their more advanced products. There are several texts written for Torque, all of which come with Torque on a CD, along with various tools and graphic resources. A good thing, because the budget I've had for these classes is zero. It can be done for this price, but it requires a lot of scrounging on the Internet for free stuff.


A CSG cathedral, built with Torque Constructor

Torque is amazingly flexible. There are several variants of the basic system, depending on what you want to construct, and what platform you want to publish it on. There are 2D and 3D versions of Torque, and a new and improved HD Torque has just been released. Torque can construct games on Windows, Mac and Linux, and can publish on any of those platforms plus the iPhone, the Xbox, the Wii, and online gaming through Internet web browsers. Torque will build first-person shooter games, role-playing strategy games, 2D and 3D platform games, and avatar spaces like Second Life, among other styles. Up to 256 players can occupy the game world at one time.

The pictures on this and the following pages give some examples of the high-quality imagery Torque is capable of creating. In many ways, game engines like Torque resemble model trains. You can build everything yourself, you can swap components with other enthusiasts, or you can purchase kits of components to use as a basis for more sophisticated projects. Garage Games' web site has a great many components for sale.

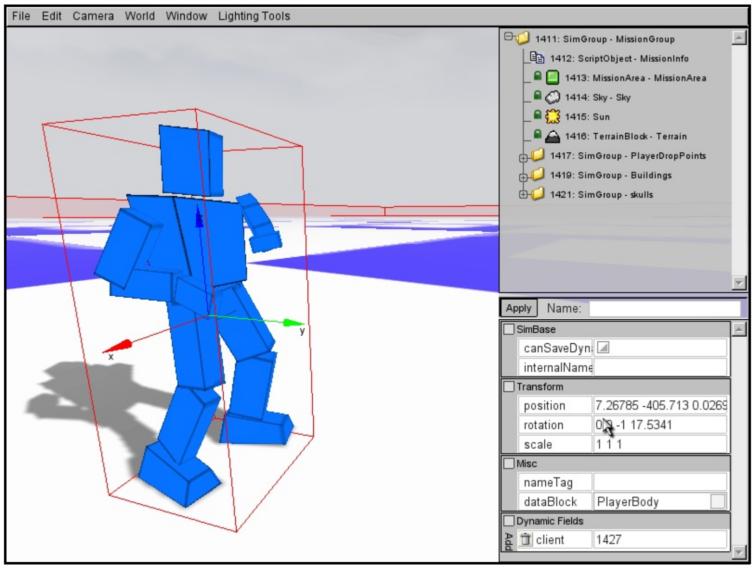


A Torque GUI screen, with a static image as a background, and various buttons which trigger game control functions.

A Sample Curriculum

My course attempts to introduce students to most of the major components typically encountered in a professional game engine, without getting too deeply involved with any one of them. Here's a summary of the main topics I cover:

- Programmer's Editors
- Terrain Models
- 3D Coordinate Systems
- 3D Objects and Scenes
- Color, Light and Shadow
- UV Mapping
- Rendering and Ray Tracing
- Event-Driven Systems
- Coding and Debugging
- Constructive Solid Geometry
- Constructing Bodies
- Animating Bodies
- Particle Animation
- GUI Screens and Interactivity
- Client-Server Systems
- Networking



```
1411: SimGroup - MissionGroup
  1412: ScriptObject - MissionInfo
  1413: MissionArea - MissionArea
  1414: Sky - Sky
  1415: Sun
  1416: TerrainBlock - Terrain
  1417: SimGroup - PlayerDropPoints
  1419: SimGroup - Buildings
  1421: SimGroup - skulls
```

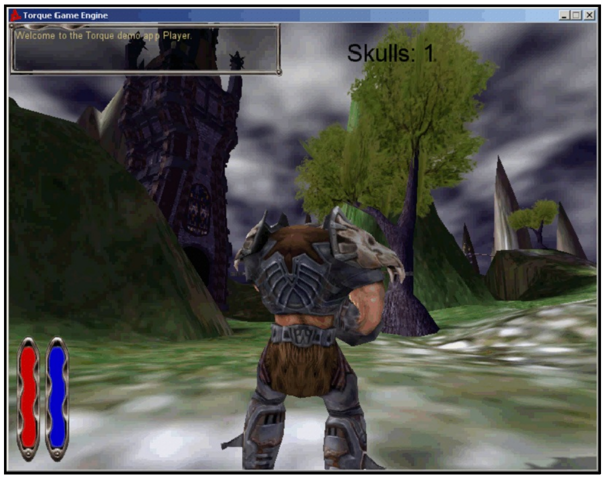
Name:

SimBase	
canSaveDyn:	
internalName	
Transform	
position	7.26785 -405.713 0.0269
rotation	0 0 -1 17.5341
scale	1 1 1
Misc	
nameTag	
dataBlock	PlayerBody
Dynamic Fields	
client	1427

The "blue man," one of two standard Torque bodies. The red box around the body is its collision mesh. The arrows can be manipulated to position and scale the body. At the right are a list of objects in the game world, and parameters of the currently-selected object.

In addition, there are other topics, such as AI, that would fit well with Torque. I have code that implements searching and flocking behavior, which would be a good addition for more advanced students. Time is a constraining factor, though. We don't have time in one semester to get into any of these topics too deeply. As it is, students shouldn't expect to single-handedly design and build a complete game in one semester. This course merely introduces them to the concepts and basic mechanics.

The Tool Chain


Welcome to the Torque demo app Player.

Another student game. The red and blue indicators represent health and ammo. The rectangle in the upper left displays messages from the system.

In addition to Torque, there are numerous other applications and utilities needed to pull this all together. It's a challenge to make all of this work. Nevertheless, my students get a good introduction to the various kinds of work done in a real game development house, in a manner that allows plenty of room for creative work.

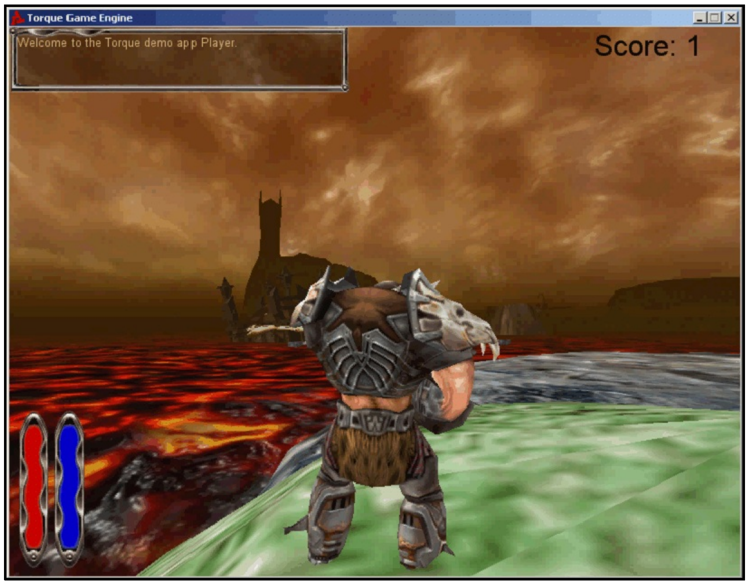
Torque features a good set of environment editors built right into the system. The picture at the left shows one such editor – an object placement and parameter editor. The user can manipulate objects directly, by grabbing them with the mouse and moving them, or by setting object parameters in the right-hand window. The entry-level version of Torque takes the unusual approach of building construction tools right into the game. Students can switch from designing a Torque environment to walking around in it just by pressing a key. For those students who have never used tools such as a compiler or a debugger, this makes it a lot easier to learn the basics.

In addition to the Torque Game Engine itself, there are several other applications and utilities needed:

- a programmer's editor (I've been using jEdit – it's free, it's highly programmable, and runs on any platform);
- a graphics editor, such as Photoshop or GIMP (which is also free);
- a Constructive Solid Geometry (CSG) modeler (Garage Games provides their own, specifically designed to work with Torque);
- a 3D modeling application (Blender is free and fairly respectable, although its user interface presents a steep learning curve);
- various batch files for managing game assets such as scripts and models.

An Introduction to Object-Oriented Programming

Another nice feature of Torque is "TorqueScript," the scripting language it uses. The Torque development concept is that the game builder initially uses a scripting language to create a prototype and work through conceptual issues. When the prototype passes muster, the developer then switches to using C++ (Torque's native language) to implement the TorqueScript routines in the system software. Developers who purchase Torque get all of the source code as part of their purchase. TorqueScript resembles



Another student project, with a sea of molten lava. Environmental effects are all programmable – this student has created an approaching dust storm. Rain and snow are implemented as particle effects, and are fairly realistic.

C++, but with several important exceptions: it's untyped, sparing students from learning the complexities of C types; it's interpreted, so students don't have to use a compiler; and there are no pointers to worry about. TorqueScript routines are typically short; as a good scripting language, its main function is to act as the "glue" that coordinates objects, rather than being a real programming language.

Software "objects" are an important concept in game development. TorqueScript is a useful way of introducing the concept of objects without imposing the complexities of a real programming language on beginning students. And those students who are already familiar with C++ or java will take to it immediately.

Support

There are a number of serious game engines available, including some open source offerings. Educators need to consider how well-supported the system is, and how accessible it is to students – two areas in which Torque excels. Garage Games, the creators of Torque, have made a serious effort to make their system student-friendly. There is a very active user community supported by Garage Games, and answers to software questions are generally available on-line. For tricker questions, they have a support staff for owners of the system (which I am), and will supply answers to just about any reasonable question within a day or so. They host an annual Torque in Education forum, which gives educators a chance to meet each other and share notes.

More than 200 colleges worldwide (and even some high schools) have licensed the full Torque system (including all the source code). An unknown number of other schools use the runtime development system, which is free with the purchase of textbooks (typically in the range of $25 to $50) That number is growing constantly, largely because of the relative ease of use, high quality output, low price, available textbooks, and good support.

A system such as this does present challenges, however. It's not a tidy application which comes with an installation program. The large number of components requires a lot of testing and adjustment before it's ready for the classroom. It's more demanding of a school's IT support staff, particularly in the areas of storage space and networking. And it requires computers intended for serious graphics.

The Torque system that the student ends up with contains over 200 files in about 20 hierarchical folders. I've been using an approach that stores student work on Flash drives. This allows students to transport their homework to and from home fairly easily. But there is a problem if a student forgets to bring his or her Flash drive to class. For that reason, it's desirable to have some sort of large storage facility available.

I use the complexity of the Torque system as an introduction to "real" software engineering projects. The editor I use works with multiple files, and has macros which search for significant lines of code across the entire folder tree. Although it sounds complex, the class is usually working in only one or two files at any given time. I have macros which assist in finding syntax errors, missing files, and other common mistakes.

Torque is inherently a client-server system. Any student can invite other students to explore his or her game world, by selecting an option to make their server open to the rest of the class. This requires no more than basic network functionality.

Conclusion

I've used Torque with novice computer users, with mostly good results. The games worlds they create are quite satisfying – this is the real thing, not clunky little geometric shapes. Many students come to class early, and put a lot of effort into what they create. Torque appeals to a wide cross-section of students, as it allows students to explore technical issues and aesthetic skills.

I'd be happy to discuss the possibilities of working with you to bring a course like this to your college or school. Feel free to contact me by phone or e-mail.

Dave Anderson
267 Locust St.
Florence MA 01062

(413) 586-2579
davidjanderson.net
anderson@crocker.com