Open Source 3D animation

VIRTUAL WORLDS

You need good software and plenty of CPU power to create virtual worlds. Luckily for today's animators, powerful PCs are inexpensive, and some excellent animation tools are absolutely free.

BY JOE CRAWFORD

Not so long ago, professional quality 3D animation required expensive workstations. The software they ran, for example Maya (Unlimited Version), cost more than US$ 15,000 early in 2002, and the software technology was hidden behind restrictive licenses. As hardware prices started to plummet, the software prices stayed too high for amateurs at first, and the tough licensing conditions looked like they were there to stay. For most people, 3D animation on home computers was just a distant dream. More recently, with the appearance of the first Open Source programs, this situation has changed dramatically, bringing affordable 3D graphics to the public, and opening up the underlying technologies to let developers produce a whole new collection of applications. The expression "7 degrees of freedom" was coined by the K3D project. It stands for the 6 degrees in 3D space and the additional freedom to distribute and modify source code. This article describes some of the best Open Source tools for 3D animation.

THE AUTHOR

Joe Crawford is the proprietor of Joetainment Enterprises/Celestine Studios, a 3D graphics studio that uses Open Source software. Joe also teaches various courses in computer animation.

3D Animation

The 3D animation process is shown in Figure 1. The process starts with modeling the shape of the 3D object. The next stage is texturing, that is, adding a kind of skin. This step gives the model a natural looking surface with light reflexes and a surface structure. UV mapping projects 2D bitmaps (such as wood grains) onto the 3D surface; in its simplest form, this is like posting a bill on an advertising column. The 3D model can then be animated. In the rigging process, joints are added to the figure. The joints allow the artist to move a complex object like a puppeteer moves a puppet. After completing the basic animation scene, the animator adds special effects, such as water, fire, or smoke. The scene is then lit by positioning light sources, just like in conventional movie making. The post production stage is concerned with optimizing the nearly-finished product, combining it with the movie or video scenes, and adding a soundtrack.

Normally, many different programs are used to create a 3D animation. First of all, you need a program to create the 3D models. In most cases, animation and rendering (conversion of finished models to photo-realistic images) are handled by separate programs. Rendering is so vastly complex that it typically requires a specialized tool.

Wings3D – Simplifying Complex Shapes

Whereas geometric shapes are fairly easy to construct, irregular or "organic" shapes, such as the shape of a hand or a tree trunk, are a big challenge. Wings3D [1], which borrows from the commercial program "Nendo," has established itself as a specialist tool in this field. Its particular strength becomes apparent when

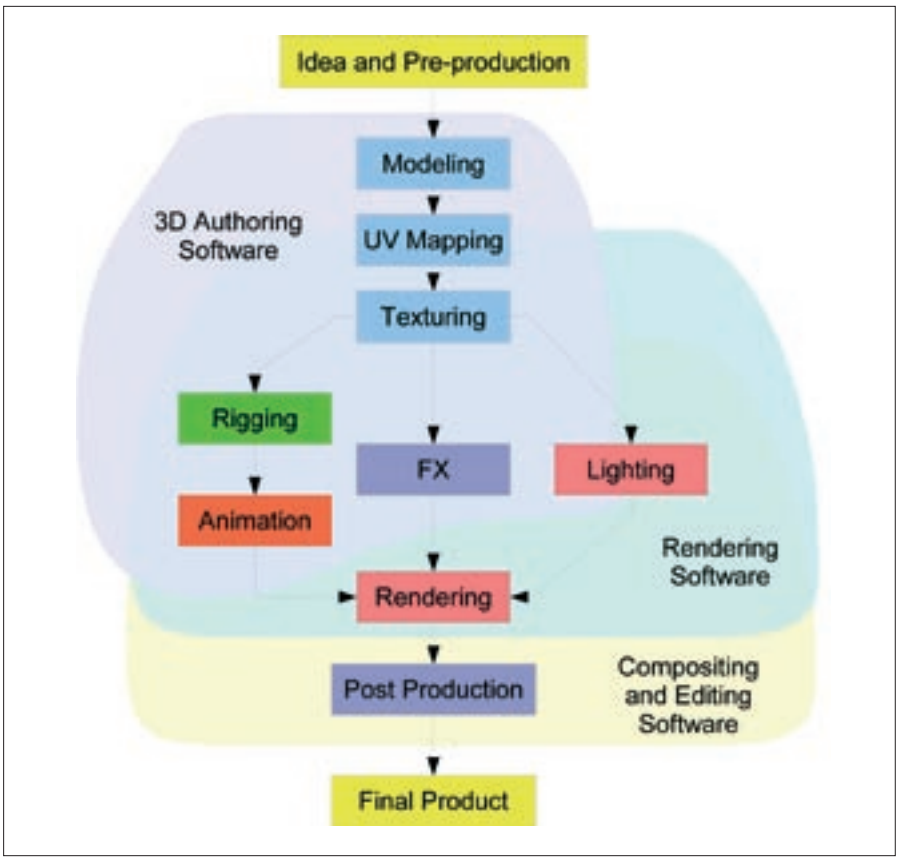

Figure 1: The production process from the idea to the finished product.

you need to model human beings or animals. Many years of development have made Wings3D a hugely powerful but intuitive tool. A learning curve of just a few hours means that users can soon get started with the creative process.

Although Wings3D is based on polygons, it gives the user the feeling of modeling in clay. The "Tweak" tool allows users to move control points in a 3D space to gradually form the required shape. Also, Wings3D has many functions for dividing simple cuboids to create natural looking shapes. To do this, you simply hover the mouse over the menu item. Pressing [Del] and the required hotkey then defines the link.

Wings3D is basically restricted to modeling and UV mapping. There are no plans to introduce animation features in future versions. This said, Wings3D supports a number of export file formats to simplify the exchange of data with other programs. If you are interested in modeling organic forms, Wings3D can hold its own against commercial programs.

Blender Power

As of this writing, Blender [2] is the only Open Source program package that covers all the steps in the 3D production process: modeling, UV mapping, animation, and rendering of the finished scene. It also has some post production options and a special game engine, the only one of its kind in the 3D sector.

Another of Blender's strengths is its speed, which is superior to commercial products. Blender's Catmull-Clark subdivision surfaces, which create smooth curves from simple block shapes, are particularly noteworthy in this context (Figure 4), and Blender's animation features are very quick.

Blender uses OpenGL for the whole screen display, even for the user interface, and it utilizes the hardware acceleration that today's graphics adapters provide. This gives users the ability to scale images or use transparency without the display becoming too slow. However, the interface is overloaded and designed in a very unorthodox way. Instead of menus and dialogs, Blender mainly uses subwindows that are full of buttons and slide controls. The interface design lacks logical consistency. Even such simple functions as cut and paste take some getting used to in Blender, as they use [Alt] + [C] or [Alt] + [V], rather than the more typical [Ctrl] keys. This is due to the fact that Blender was designed as an in-house solution for the NeoGeo animation studio.

In 1998, one of the initiators of NeoGeo, Ton Roosendaal, founded Not a Number with the idea of continuing the development of the in-house tool now known as Blender. His goal was to release the tool as Open Source. Unfortunately, Not a Number 2001 folded due to financial difficulty, but donations made it possible to purchase the Blender source code and give it to the Open Source community.

Blender's only disadvantage from its proprietary history is the fact that the interface is designed for professionals who use the tool every day. This makes it confusing and even a little daunting for normal users. You need a lot of patience to really leverage the power of this excellent program, which has to be one of the most powerful Open Source programs ever created. Due to the long development roadmap, Blender is extremely stable and has a large user

Figure 2: Wings3D is hugely powerful despite the simple interface.

community that provides art, comprehensive documentation, and good support to help users combat any problems.

K3D – the Future?

K3D [3] is one of the more recent developments in the 3D graphics field. K3D's major strength is that it is a targeted new development based on current experience. Professional 3D developers were consulted to modify the workflow to reflect professional requirements. K3D has a hierarchical Undo/Redo list, for example. What this basically means is that after restoring a previous state and modifying this state, you can reapply the changes that you made after the state you restored. This would be a desirable feature for most applications, but it is particularly useful within the creative design process.

K3D puts particular emphasis on defining the relationship between the individual objects in a scene. If you are animating the human body, the parts of the model need to be governed by multiple relationships to create a natural impression: when the figure starts walking, the thigh moves at the hip joint first. Additionally, the lower leg moves at the knee joint. Both of these movements overlap. In K3D, one object can control another object. Users can save these interdependencies, thus creating customized program features to suit their own requirements.

Development work on K3D is by no means complete. For example, the 3D object modeling tools are not as mature as in Wings3D. However, K3D takes a different approach, which again pays particular attention to object dependencies. This approach, which is known as parametric or procedural modeling, means that changes to the basic frame of a model are applied intelligently to the component parts. The developers have also designed a mature plugin architecture to provide a solid basis for future extensions.

At this time of writing, the program is at version 0.5 (development) or 0.4 (stable), and some critical features are still missing. Work is still in progress on basic tools such as a scaling or rotating tool. The animation functions are hard to use at present. And as the developers are concentrating on quality rather than speed, it might take a while for version 1.0 of K3D to appear.

The models that K3D creates are compatible with the RenderMan specification, the industrial standard for animation. K3D models can be processed using professional rendering tools such as Pixar's Photorealistic Renderman. However, K3D is not restricted to commercial products, as it also supports the excellent, and free, Yafray raytracer, which can be used to convert models to finished images. As the convincing design includes a clear-cut GTK2-based user interface (Figure 5), it is well worth keeping track of K3D's future developments.

Rendering

The process of converting 3D models to finished graphics or animations is referred to as rendering. As the rendered result is the only stage of the creation process that the audience gets to see, it is a very important phase. And as rendering is, at the same time, a very complex process, it is typically performed by specialized tools. Of course, you do need to ensure a

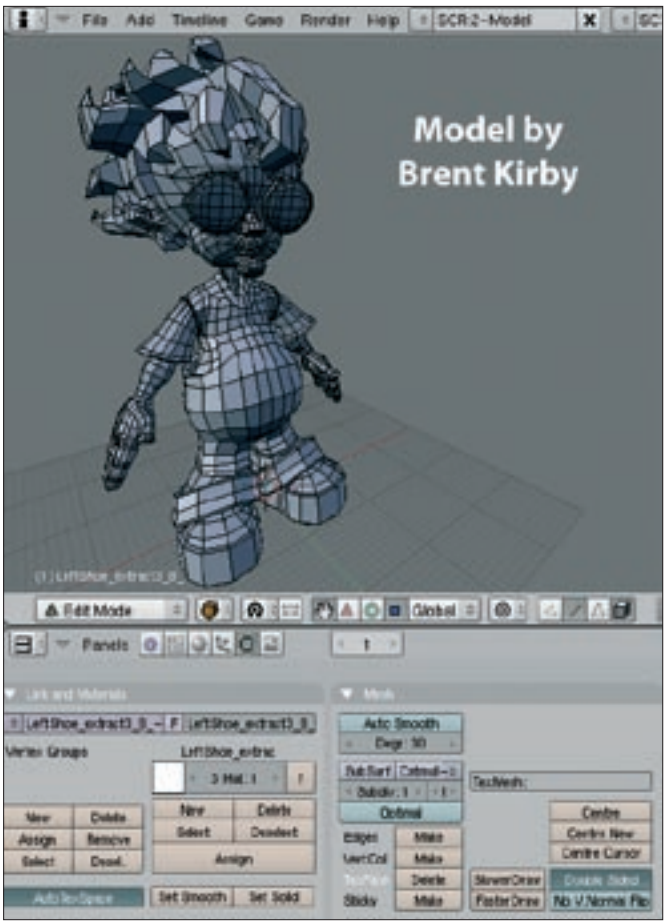

Figure 3: The Blender user interface with a finished model.

seamless exchange of data with the other production steps.

Yafray

Yafray [4] is currently the most powerful free rendering tool. Yafray can generate both extremely realistic and highly stylized images. The way that light and shadow are applied has a huge effect on how realistic rendering appears to the viewer. Yafray calculates the effect of light on the model's surface far more precisely than other programs. It calculates soft shading, light reflections, and refractions based on the Fresnel algorithms. It also traces the path taken by scattered rays of light, or of light refracted by transparent objects. This is known as "global lighting" or as "caustics."

Normally, the Yafray rendering tool is run as an external program, however, it does have a plugin API that other programs can call; and Blender can actually do this to give users a more convenient and conclusive preview.

Although Yafray is currently at version 0.0.8, the features the tool implements are sufficient to make it interesting for production work. The rendering quality is comparable with that of professional tools that cost many thousands of dollars. And this makes Yafray, with the exception of the far more basic internal rendering tool provided by Blender, the only Open Source rendering software suitable for production use at the current time.

Rendering requires a large number of highly complex calculations; this means that rendering times of several hours, or even several days, are conceivable. Right now, the developers are working on a fundamental redesign that will additionally harness the power of the graphics card processor (GPU) for the rendering process. This should drastically reduce computation time in the not-too-distant future.

Toxic

Toxic [5] is a rendering tool based on physically exact computations. It uses the "Bidirectional Reflection Distribution Function", among others, to produce excellent and extremely realistic image quality. In many aspects of the rendering process, Toxic adopts an approach similar to Yafray, however, it places more emphasis on physical correctness. The aim is to achieve results that not only appear realistic to the human eye but also emulate reality in a mathematically correct way. Toxic uses its own, well-documented XML file format. However, it can't import scenes from other 3D packages, and that makes Toxic unsuitable for integration with the production workflow.

Scripting

To help automate recurring tasks, graphics applications typically integrate a scripting language. Proprietary programs often use proprietary languages, which users then need to learn (for example Maxscipt in 3D Studio Max).

Open source applications are characterized by the fact that they use widespread open standards, which give users without C or C++ programming skills easier access. Blender, for example, supports Python scripting.

Aqsis

Aqsis [6] is characterized by the fact that it provides the most complete support for the RenderMan standard syntax for 3D model definition. This means that it is easier to create complex shapes with Aqsis than with other modeling tools. Unfortunately, Aqsis does not support raytracing, that is, realistic tracing of light paths, at this time of writing. This detracts considerably from the realistic appearance of the results. Aqsis is currently more interesting for developers who need a free tool that supports the

Figure 4: Blender with a model before and after applying Catmull-Clark subdivision surfaces.

RenderMan standard. On the upside, raytracing is on the developers' roadmap, and development work on Aqsis has progressed quite quickly so far; Aqsis is an interesting proposition, currently at an early stage of development.

Benefits of Open Source

Access to source code is a big advantage for professional studios and freelance artists. Larger studios can modify the standard software to suit their requirements. And even normal users can benefit from minor program hacks.

One characteristic of computer graphics software is that many features associated with the software are actually implemented as plugins. In case of proprietary programs, plugins are typically hard to find, and you need to reinstall them with each installation. In contrast to this, the Open Source developer community tends to quickly adopt the plugin source code into the mainstream program. In Blender's case, many features started life as plugins. And the fact that real users help create documentation and tutorials takes the headache out of learning the plugins that accompany Open Source graphics tools.

Conclusions: Open Source 3D Software

The future of Open Source 3D software is looking good. Many fantastic designs and realistic renderings have been produced with Blender and Yafray. However, a number of issues remain unsolved. For example, there is a lack of convincing standard file formats to improve the exchange of data between programs. And producing high quality 3D animations still means resorting to too many tools without standardized user interfaces. We can hope that the experience of a large and partly professional userbase, as well as the commitment of the developer community, will help to spawn the development of these missing pieces, as it has in many other Open Source software fields.

Finally, there are still a few gaps in the production workflow that leads from the initial 3D modeling phase to the release of a finished movie: professional studios still use proprietary tools for post-production cutting and scoring. Projects such as Jahshaka [8] or Cinelerra [9] might close these gaps in the near future, allowing studios to handle the whole production cycle for computer generated movie material with open source software. ■

Figure 6: This image shows the Aqsis rendering tool's potential for creating complex shapes.

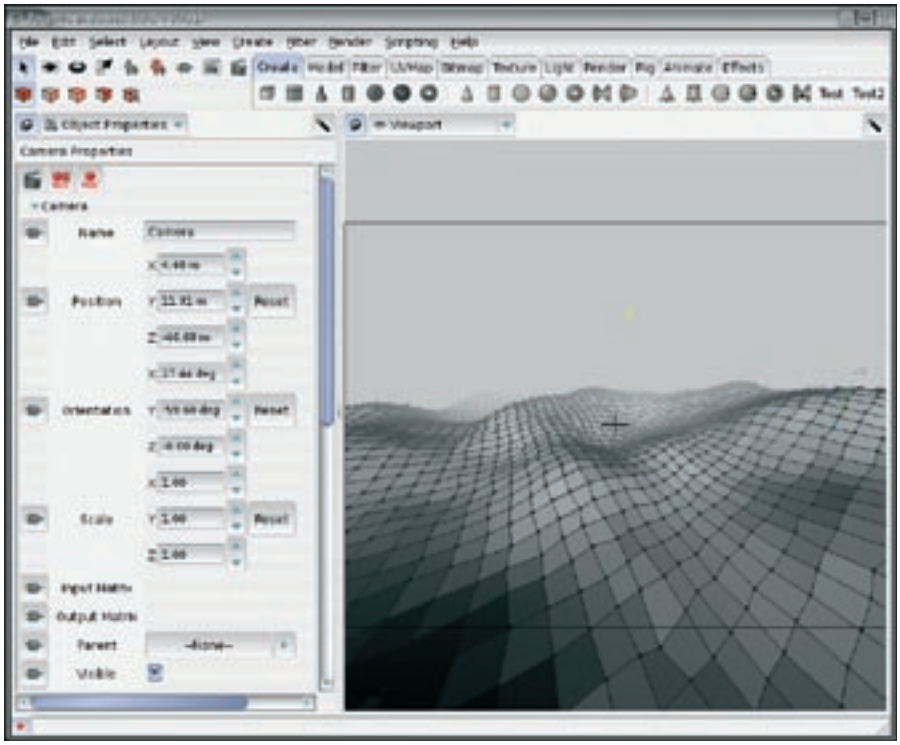

Figure 5: K3D has a clear user interface based on GTK2.

INFO

[1] Wings3D project: *http://www.wings3d.com*

[2] Blender homepage with documentation and various resources: *http://blender.org*

[3] K3D (online documentation available): *http://k3d.sourceforge.net*

[4] Yafray: *http://www.yafray.org*

[5] Toxic renderer: *http://www.toxicengine.org*

[6] Aqsis: *http://www.aqsis.org*

[7] Povray (Documentation, Community): *http://povray.org*

[8] Jahshaka: *http://www.jahshaka.org*

[9] Cinelerra: *http://heroinewarrior.com/cinelerra.php3*