A 64KB GRAPHIC INTRO

Pavel Gunia
Master Degree Programme, FIT VUT
E-mail: xgunia01@stud.fit.vutbr.cz

Supervised by: Adam Herout
E-mail: herout@fit.vutbr.cz

ABSTRACT

This work deals with the phenomenon of a graphic intro, a "digital grafitti" of the modern age. The focus is put on size restricted animation of size of the executable file lower than 64 kilobytes. It reveals the main techniques used.

1 INTRODUCTION - A DEMO AND AN INTRO

A demo and intro basically refers to the same thing. It is a non-interactive multimedia presentation. The key difference from a classical computer animation or video is that the display of a demo or an intro is computed in real time, providing challenge such as computing power or size restriction.

Intro, is a shortcut for *introduction*. First intros were created in late 1970's to be distributed along with pirated software as cracker's signatures. They were often nothing but plain text screens crediting the cracker or his group. Gradually, these static screens evolved into increasingly impressive-looking introductions containing animated effects and music. Eventually, many cracker groups started to release intro-like programs separately, without being attached to pirated software. There have been well organized parties and meetings since the early 1990's, organized as serious categorized competitions with attractive prices. A category of restriction on the size of the executable file has been introduced to provide additional challenge for programmers, typically 4, 64 or 256 kilobytes. Typically, a demo refers to an animation without any restriction while an intro is restricted by size.

2 MAKING IT NICE, MAKING IT SMALL

Simply said, as many programmers, as many different approaches. Yet, there is a general pattern. The core of the program is the rendering subsystem, which typically takes advantage of the power of today's accelerated graphic hardware. In this case, the graphic routines are built on *OpenGL* graphic library *(www.opengl.org)*. A variety of attractive effects can be achieved through its system of extensions. Graphic intros typically use a wide range of high-end techniques and post-processing filters to give it an interesting visual style and to get rid of the fa-

miliar look of realtime rendered computer graphics. Yet, there is also tendency to use software realtime raytracing to keep a specific style.

2.1 MATHEMATICAL APPROACH

Attention is put on storing data in a compact way. In order to squeeze in as much as possible, this intro involves quite an amount of mathematics. Bézier curves and surfaces are indeed very good approaches for representing paths, shapes and surfaces. However, we go further to procedurally generate geometry and textures. Procedural generation works through a chain of operators (e.g. each operator provides input for the next one). Most of the operators were implemented to use the computational power of the graphic device, significantly speeding-up the animation loading time. For an example see *fig. 1*.

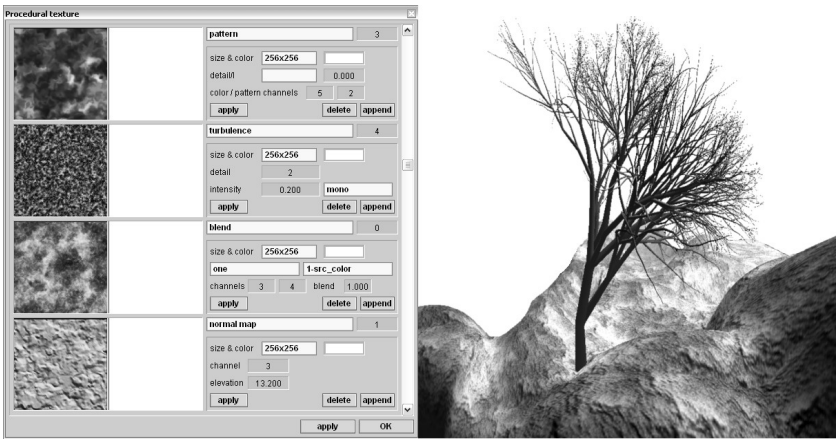

Figure 1: A generated terrain texture. A normal map is also produced for per-pixel lighting.

2.2 VISUAL

The tendency in the development of modern graphic hardware is towards fully programmable processor on the graphic device. We indeed take use of programmable vertex and pixel shaders. For the sake of simplicity of the program, ARB (Architecture Revision Board) approved vertex and fragment program extensions are used. From the techniques used it is definitely worth mentioning *per-pixel lighting*, an approach providing significantly better quality of the output as the lighting calculations are done for every pixel rendered, in contrary to the standard per-vertex lighting. Combined with the technique of *bump mapping* provides an attractive result *(see fig. 2)*. In order to make the animation run on older hardware also, these advanced techniques are enabled when modern graphic devices are present.

3 AT THE END

There is a wide variety of short computer animations already, either the ones made just for fun of it or the ones winning prices at competitions. Some of them carry a message, other rather compete in the programming excellence. This animation is wished to carry a message about problems of the environment we created for ourself. Music playing in the background makes

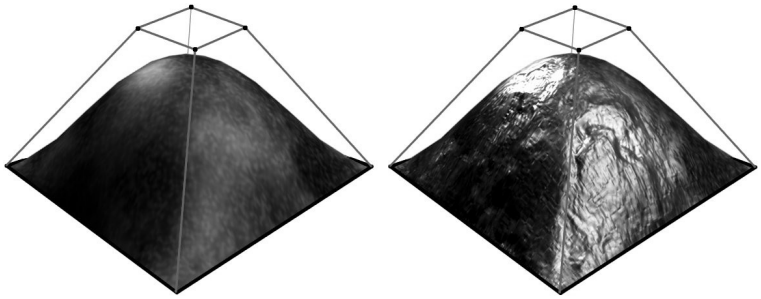

Figure 2: A Bézier surface rendered using standard lighting*(left)* and custom per-pixel lighting*(right)*.

the animation feel complete. MIDI synthetised music is used, however, a properly tailored audio decompressing might allow for digital samples. That is left as a possible future work though.

Personally, i see the beauty of this short animations in the fact that they are so short. People were always attracted to the idea of finding the hidden capacity of things or devices, and using it for purposes they were not intended to. Not very common, yet animations for calculators or even for sophisticated copy machines exist. In the most common category, on personal computers, this animations show how much is possible to squeeze into such small file size, in contrary to today's trend of huge solutions, hundreds of megabytes of data and a whole lot of compact disks. Quite probably, the price of coffee consumed during the making would raise the expenses value the most.

Figure 3: Screenshots from the animation.

REFERENCES

[1] The International Scene Organization (scene.org), *http://www.scene.org*

[2] Addison-Wesley Publishing Company: OpenGL Reference Manual, *http://www.opengl.org*

[3] Silicon Graphics, Inc.:OpenGL Extension Registry, *http://oss.sgi.com/projects/ogl-sample/registry/*