

RenderMonkey: An Effective Environment for Shader Prototyping and Development

Natalya Tatarchuk*
ATI Research, Inc

Introduction

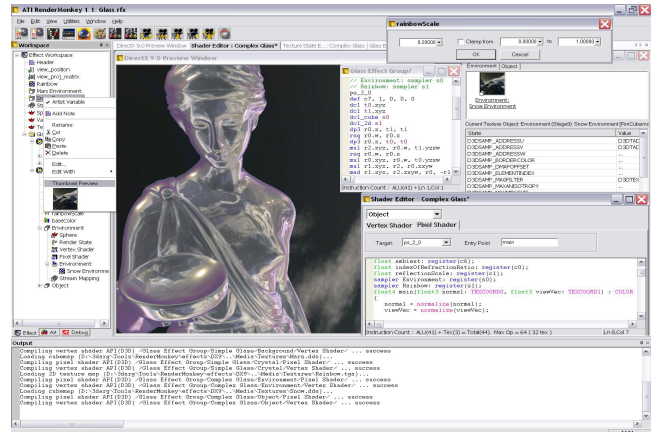
We present a new process for real-time shader development using an improved environment for shader content creation—the RenderMonkey IDE. This environment enables anyone with an interest in real-time graphics to quickly embrace programmable shaders for projects from game engines to scientific visualization or anything in between. The RenderMonkey IDE is a general tool for effective shader development, allowing collaboration between artists and programmers in a single, unified environment. We will present the design philosophy behind the RenderMonkey IDE, the process for shader development, and new methodologies for integrating RenderMonkey into pre-existing production pipelines.

Application Design and Philosophy

Real-time high performance shaders are at the heart of all new visual effects and they will continue to be the foundation of the amazing graphical experience for the future. Although real-time shaders have been available for a few years, their adoption into existing projects has been slow due to a lack of convenient tools. The RenderMonkey IDE has been created to fill the need for shader content creation in a coherent environment—flexible and powerful for programmers, yet familiar and intuitive for artists. With the introduction of the DirectX and OpenGL high level shading languages, the complexity of real-time shaders has increased. With the rapid improvement in shading capabilities of current graphics hardware, the amount of shader content necessary for real-time graphics projects has exploded, fueling the need for a powerful real-time shader IDE. The RenderMonkey environment not only allows easy shader prototyping and development, but also provides a mechanism for managing the shaders and all of the associated visual resources in a single environment.

Standard Graphics APIs The RenderMonkey IDE enables shader production on any hardware supporting standard graphics APIs—the latest release of the RenderMonkey IDE provides integrated support for both DirectX and OpenGL shading languages.

Programmers and Artists RenderMonkey has been designed with the needs of programmers and artists in mind, allowing both to collaborate on shader effects. Separate interfaces exist in RenderMonkey for setting up familiar controls for technical shader developers as well as for purely visual modification by non-technical artists. RenderMonkey allows anyone interested in creating shaders to bypass the tedious setup steps and dive straight into the shader creation process. The program allows visualization of the a visual effect’s inputs and assets in a tree structure as well as the visualization of the body of the shader itself. Users can view the gradual build-up of an effect by examining individual rendering passes in the application’s rendering windows or by dynamically examining the contents of all renderable textures. RenderMonkey enables quick debugging of shaders via interactive output of intermediate results computed by the shader. We have found this to be particularly useful in teaching environments



where visualizing an intermediate result is fundamental to understanding. RenderMonkey has been chosen by several universities to teach shader technology in their graphics curriculum.

The application includes a variety of convenient features enabling programmers control over their development environment. To enable development of fast, efficient shaders, RenderMonkey provides developers with optimization hints for high level shader development for better analysis of shader performance in real-time. Both artists and programmers can use intuitive GUI widgets for interactively editing visual parameters for shaders—modify a color or the dissipation parameter for fluid flow and see the results instantly in the rendering window.

Integrating Into Existing Production Pipelines

Another key feature of RenderMonkey is its flexible, extensible framework that supports easy integration of custom components. RenderMonkey can be easily customized and integrated into any developer’s regular workflow by using its standard SDK for component development. The application ships with a set of plug-in examples and a plug-in wizard, enabling easy creation of custom components and thus allowing fast integration into an existing production workflow. The SDK gives developers full access to the entire run-time database of shader data. The SDK is written in standard C++ with all of the necessary components for plug-in development. It also provides libraries for user interface widget creation to allow developers to maintain the look and feel of the application without investing significant effort into UI development. A variety of game companies are integrating RenderMonkey into their shader production pipelines for the upcoming projects.

RenderMonkey is designed to make adoption of shader technology easier for both developers and artists, providing easy integration into existing workflows and the ability to customize desired components to suit the needs for any production. The interactivity of the program provides the advantage of reducing time for shader content development, enabling quicker iteration on each design, and thus improving the final look of the product for any production.

*e-mail: natasha@ati.com