Computer Engineering Student Project Proposal

Procedural Generation of 3D Art for Games

Modern game artists spend a great deal of their time meticulously crafting the art for a game. Even small details that are part of the background, or are simply viewed for a few seconds are often worked on for many hours by a talented artist. Not only do these art assets take an artist's time, they also increase the size of the finished product. Most modern games contain thousands of times more art data than algorithmic data. In 2004 a first person shooter game called ".kkreiger" was released as a 96kb executable (See Figure 1). By comparison, most games at that time with comparable graphic capabilities would typically occupy 2 to 4Gb of space. This feat was accomplished using procedural generation.

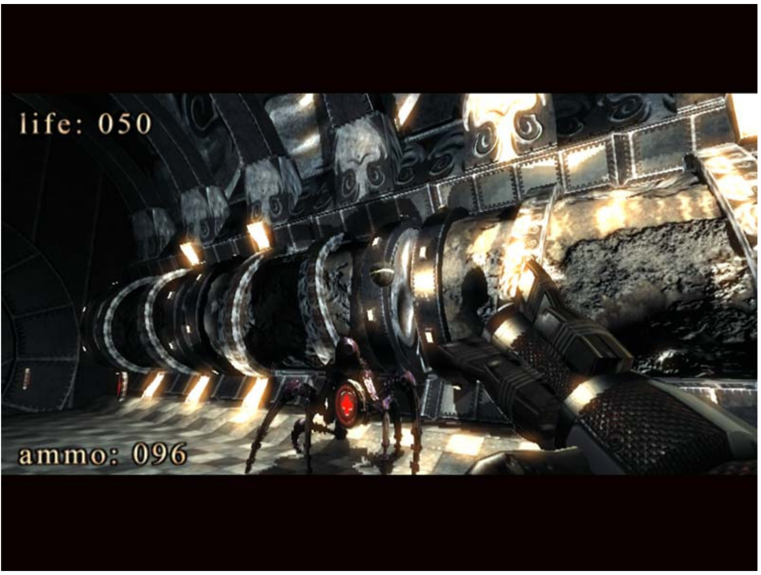

Figure 1: .kkreiger

Procedural Generation refers to the media production technique of creating art assets on the fly rather than crafting everything beforehand. Early games had very strict size constraints and often turned to procedural techniques to save space. Modern games are not as limited by size, but generally require a much higher number of unique objects. In many cases, seeing the same object appear multiple times in a scene can be jarring for a user playing the game. By using procedural techniques, unique objects can be generated for every instance required without increasing the size of the game at all.

The goal of this project is to create a library for procedural generation of 3D art assets for games. The first step in the project is to create a set of functions that will generate the basic building blocks such as cubes, spheres, cylinders, and plain or patterned textures. These functions will help students get used to working with 3D objects and image processing techniques. The project can then expand to more complex objects and textures to fit the interests of the students. For example, terrain, trees, clouds, planets, and buildings are some examples of things that can be generated. Generating the more complex objects will generally require the student to apply math equations developed in research in a variety of fields. For example, clouds are sometimes generated or simulated using the math from fluid dynamics, and terrain is often generated using techniques from image processing such as erosion.

As a starting point for this project, the students will be given the Horizon game engine developed by Source Studio. The engine will allow the students to easily create a test environment so they can focus on the algorithms for their project. In addition to the engine, students accepting the project will be provided with support from the development team, a project outline to help guide students through the early stages of design and development, a list of potential areas of research in procedural generation, and a collection of research articles that may be useful in developing the project.