

Travis Gorkin

Demo Reel 2009 Breakdown

travis.gorkin@gmail.com
www.gorkin.com

Fire and Smoke Effects

Smoke Simulation: March 2009

Fire Simulation: April 2009, 2 person Group Project

These fire and smoke effects demonstrate two variations on physically based fluid simulations that I implemented from scratch in C++. The fluid in these animations is physically simulated using the Navier-Stokes equations on a 3D MAC grid, following the methods outlined by Bridson, Fedkiw, and Muller-Fischer in their Fluid Simulation course notes from SIGGRAPH 2007. The conjugate gradient solver, the main bottleneck for simulation speed, was accelerated using NVIDIA's CUDA programming language to take advantage of GPU based parallel processing, resulting in roughly a 10x improvement in simulation speed. The fire animations were rendered by slice-based volume rendering using OpenGL, and also as shaded blobbies and particles using PR Renderman. For the fire simulation, I worked in a group of two, and was mainly responsible for simulating the evolving flame front using the level set method, implementing the Conjugate Gradient algorithm using CUDA, and programmatically generating RIB files for rendering with Renderman.

Maya City Generator

Maya Plugin Project: Spring 2009

I developed this procedural building and city generator plugin for Autodesk Maya using the Maya C++ API. The plugin is designed so that an artist can quickly and interactively generate detailed buildings and urban skylines within the Maya environment. The user can interactively modify building generation parameters, such as floor divisions and mass model dimensions, through user interface elements developed in MEL. Artists can import their own custom geometry as OBJ files for mass model, floor, and facade features. For this project, in addition to developing the plugin, I also pitched the concept and created a detailed design document outlining the features, software design, and development tasks.

Cloth Simulation

February 2009

I implemented this cloth simulation from scratch using C++ and OpenGL. The cloth was physically modeled as a lattice of particles connected together through structural, bending, and shearing springs and dampers. External forces for gravity and collisions with the character were added to realistically simulate the cloth in a 3D environment. The simulating was integrated forward in time using the 4th order Runge-Kutta method. The cloth is attached to a character model driven by motion capture data and rendered using OpenGL.

Rigid Body Dynamics

April 2009

In this interactive rigid body simulation, implemented using C++ and OpenGL, the user can shoot projectiles at stacks of rigid bodies. Physical interactions between the moving rigid bodies are simulated in real-time, using the methods outlined by Guendelman, Bridson, and Fedkiw in their SIGGRAPH 2003 paper "Nonconvex Rigid Bodies with Stacking." In addition to the basic simulation of the rigid body physics, I implemented an optimized broad-phase collision detection using the sweep-and-prune algorithm in order to achieve interactive rates.

3D Physics Games

Blockbreaker: June 2009, 2 person group project

Bowling: July 2009, 2 person group project

I developed 3D versions of these classic arcade-style games in C++ using the Gamebryo game engine and NVIDIA PhysX. I worked in a team of two to create these games over a three week time period. I was responsible for creating, importing, and animating all 3D models within the games, as well as large portions of the game logic such as handling user input, collision detection/response, and physics simulation.

3D Modeling

Robotic Spider: November 2008

I modeled and textured this robotic spider in Maya 2008 and rendered using MentalRay.