

DEMO REEL BREAKDOWN

- 1. Smoke** – C++, OpenGL March 2012
Semi-Lagrangian simulation system written in C++. Implementation includes: Advection of velocity, temperature and density; Poisson pressure projection; Cubic interpolation; RK2 integration; Modified incomplete Cholesky preconditioner for the conjugate gradient; Buoyancy forces; Vorticity confinement. Rendered with OpenGL.
- 2. Explosions** – C++, OpenGL, Maya, Mental Ray May 2012
Simulation system written in C++. Modified Smoke by adding particle system for fuel and soot. Combusting fuel particles introduce divergence into velocity field, producing rapid expansion. Particles are advected by fluid velocity field, and used to produce a new density field for rendering. Ported to Maya using fluid emitters. Rendered with OpenGL and Mental Ray.
- 3. Cooking Kinect** – C#, Unity, Maya, Microsoft Kinect May 2012
Group project to design and build a cooking simulation game controlled by the Kinect system using the Unity game engine. Includes unique simulation and Kinect detection for seven different cooking motions: Chop, Distribute, Flip, Grate, Pour, Whisk and Wobble. Personal contributions included: Integration of all code, art, UI, sound into the final game; Setup and maintenance of code framework and class structure for all game objects and data; Half of Kinect motion sensing code; 3D assets for all food and ingredients; GUI and HUD; Menu screens; Core gameplay loop.
- 4. “Robot”** – Maya, Mental Ray October 2011
Short animation project including all storyboarding, modeling, shading and lighting of characters and environment. Animated using keyframes. Robot spring base and antenna use bend deformers. Eyes and antenna use glow effect.
- 5. FLUD – Faster Liquids Using Divisions** – C++, OpenGL, MEL, Maya, Mental Ray May 2012
A Maya plug-in that subdivides SPH fluid particles within a user-specified geometric boundary area to create higher resolution simulation while saving the cost of universal higher level of detail. Based on a SIGGRAPH 2011 paper “Two Scale Particle Simulation” by Barbara Solenthaler and Markus Gross. UI written in MEL. Utilizes a modified OpenGL SPH framework, FLUIDS v2, as a DLL to set up the fluid in Maya as an nParticle system. Rendered with Mental Ray.
- 6. “One Glance”** – MEL, Maya, Mental Ray, Photoshop December 2011
60-second animation project including all storyboarding, modeling, shading, rigging, animating, texturing and lighting of characters and environment. Facial animations use blend shapes. Used MEL scripting to automate the walk cycle. Forest environment created by applying a transparent texture of a duplicated tree model to multiple planes. Rendered with Mental Ray.
- 7. Space Invaders** – C#, Unity, Maya, Photoshop November 2011
Designed and implemented all components of the game using the Unity game engine. Scripted using C#. Modeled and shaded the aliens, ship, and missiles in Maya. Textures for game menus and buttons created in Photoshop.
- 8. Cerberus** – Maya, Photoshop May 2011
Modeled in Maya, rendered with Maya Software. Cave environment textured using Photoshop. Torches and back flames created with Maya Particles. Lava created with a modified Ocean shader.
- 9. Crab** – Maya, Photoshop March 2011
Modeled in Maya, rendered with Maya Software. Textured using Photoshop.
- 10. Pirate Attack** – C++, OpenGL July 2006
Group project to build a simple OpenGL game. Personal contributions included: Modeling the ships; Collision logic for ships and bullets; Camera; Lighting; Keyboard controls; Bullet explosion animation. Rendered with OpenGL.
- 11. Raytracer** – C++, OpenGL October 2010
Uses the Blinn-Phong lighting model. Accounts for specular highlights, various materials, reflections, shadows, and multiple light sources. Bounding spheres are used around each object for rendering speedup. Rendered with OpenGL.