Initial inspiration: the demoscene

The demoscene is where groups of coders, artists and musicians create “demos” to show off their skills and engage in friendly competition against other groups by seeing who can create the most impressive demo.

What is a “demo”?

It is a program which when executed, provides an audio-visual experience that is calculated on a computer in real-time. Not dissimilar to a music video, it has eye-catching visual effects which accompanies original composed music. As it is realtime, each frame of the video has to be calculated in just a few milliseconds before it has to be displayed to the screen. Demos are technically challenging to create, and it is even more challenging to make them look good!

What is the point of a “demo”?

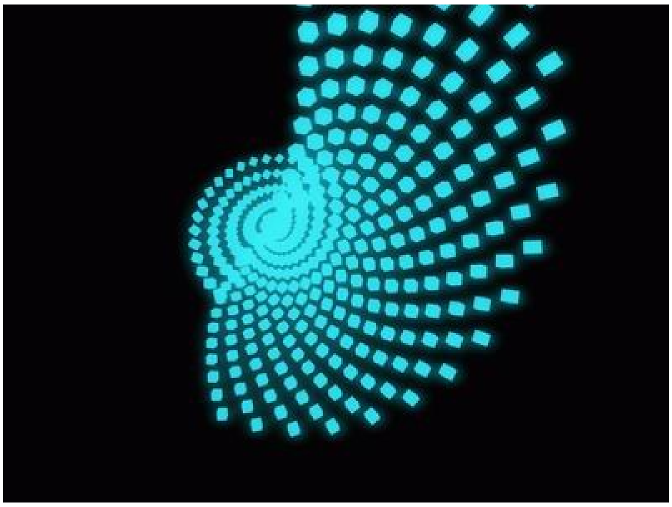
Usually, nothing! A “demo” does not necessarily have an objective nor a storyline. A lot of the demoscene sees demos as an opportunity to amaze others with creativity and technical skill.
In addition, some coders see demos as an opportunity to push computers to their utmost limits. Often, they will cram as many visual effects as possible into a demo, just to show off that they can write effects with enough efficiency that even with the additional calculations imposed, the entire demo can still run smoothly.
There are also artifically imposed limits, like code size. Some demo competitions restrict the size of the executable to be less than a certain size, like 64KB (which is smaller than a typical JPG!). Demo coders will then try to squeeze in as many effects into that space as possible.

On the following pages I have included screenshots of some of the most famous and influential demos. I hope to take inspiration from them, but knowing full well that I cannot hope to match the technical nor artistic expertise of the dedicated groups of demo coders and artists who created the demos!

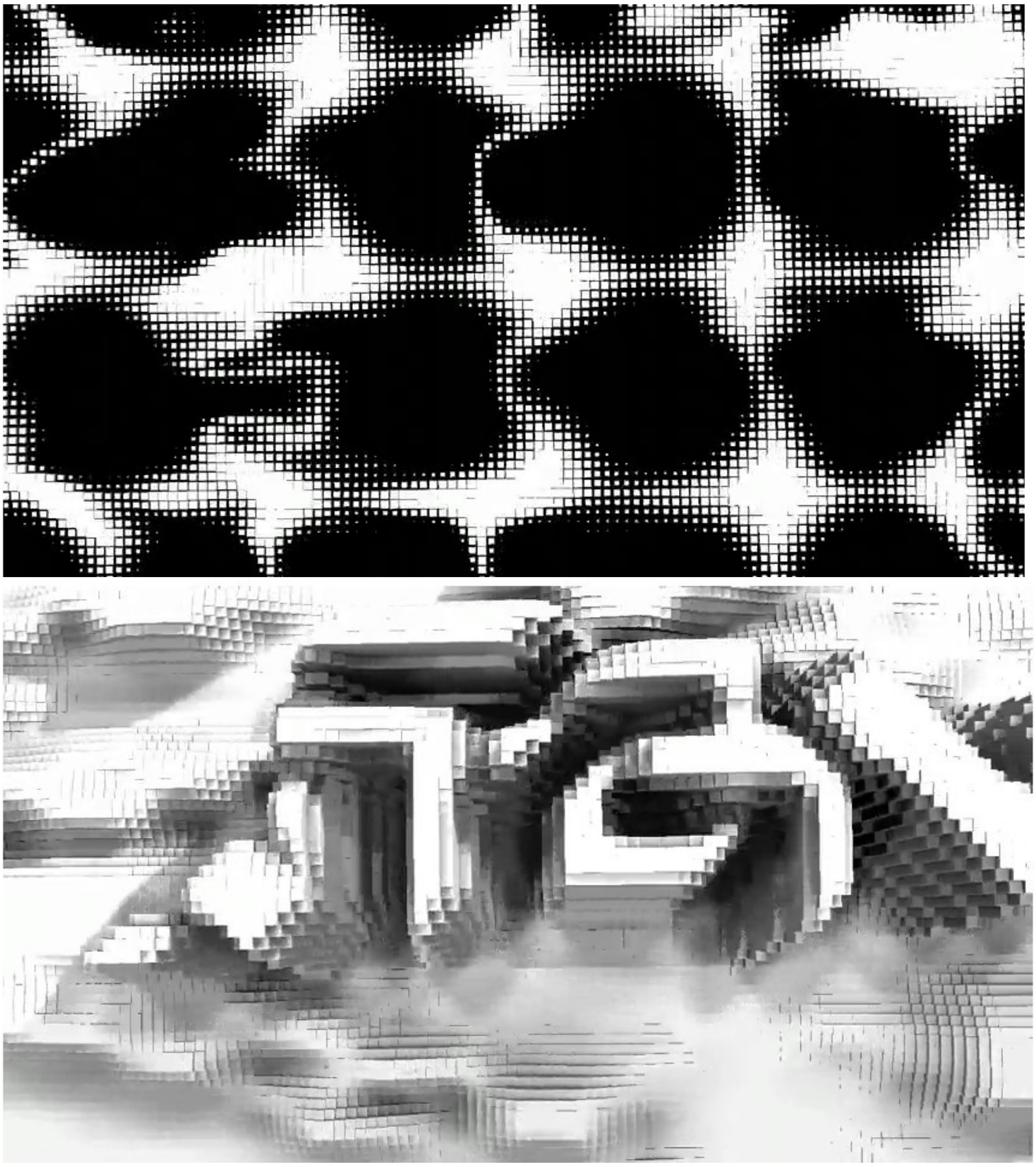
“Oldschool” effects

These are effects that are reminiscent of the demos in the 1990s that ran on much slower computers, hence they are simpler and more abstract. They are usually particles or objects moving around according to a mathematical rule. They use simple elements (such as cubes and grids) in hypnotic patterns.

They involve movements such as rotation, zooms, and effects such as “plasma”, “ribbons”, etc.



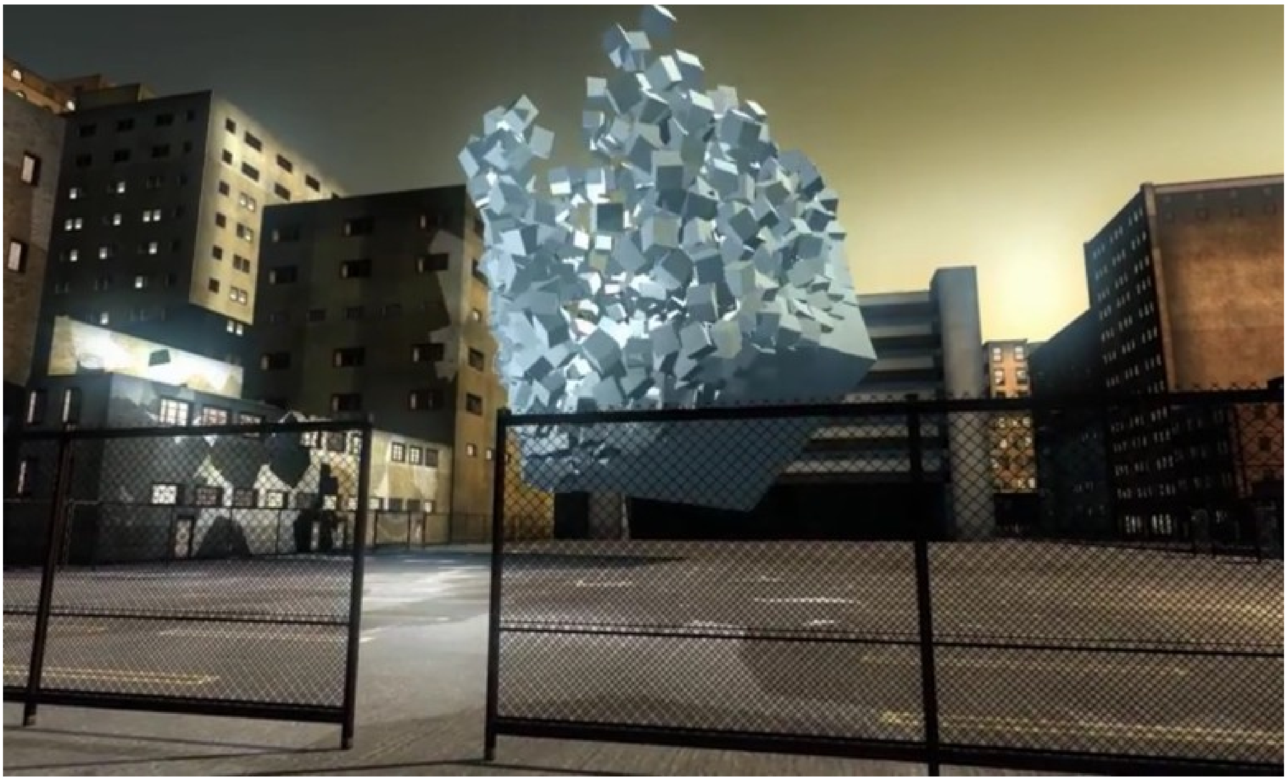
“iterate” by imbusy & xerxes (2006)



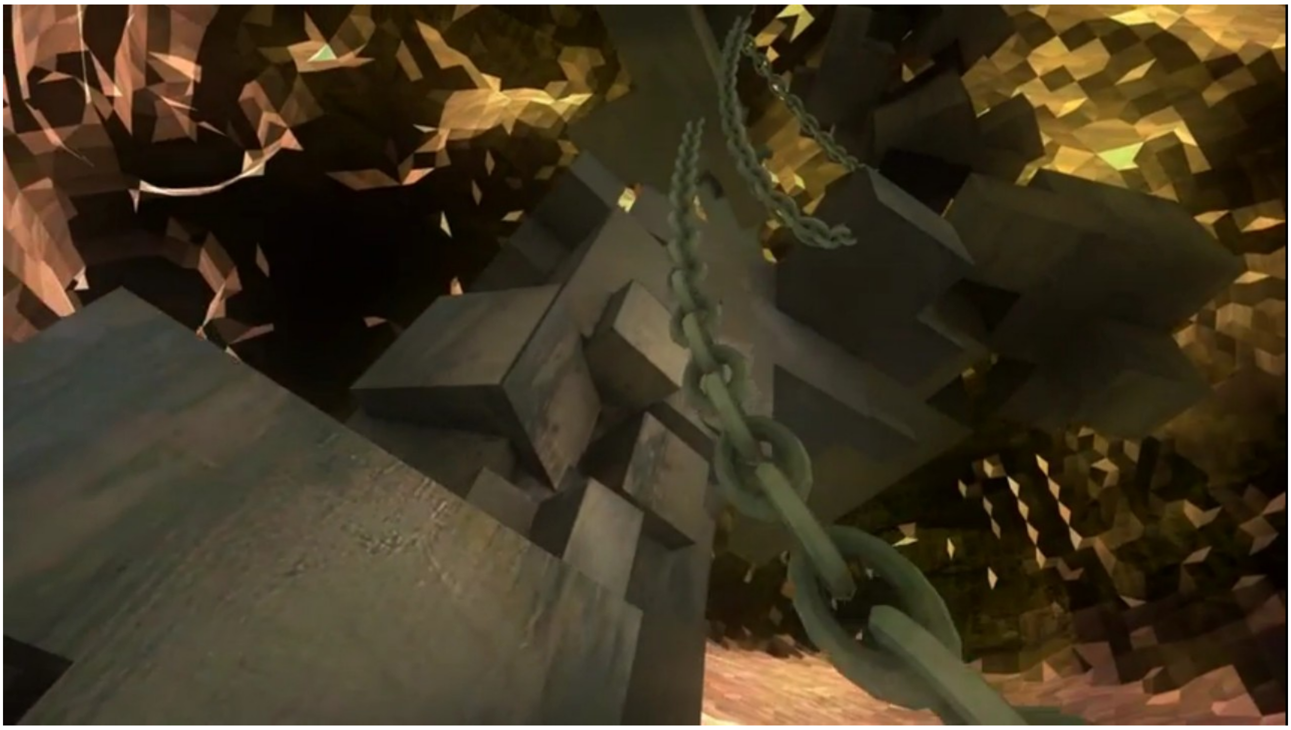
Two scenes from "Frameranger" by Fairlight & orange & CNCD. 1st prize at Assembly demoparty 2009.

Complex geometry and lighting effects

Usually the largest demogroups (those with most coders and artists on a team) try to compete to see who can make the most photorealistic and complex demo, whilst still maintaining a smooth frame rate on a modern computer. These demos are all rendered in realtime, adding to the challenge.



"Debris" by farbrausch. 1st prize at Breakpoint demoparty 2007. The executable for the entire demo is only 177KB -- smaller than the saved image of this frame alone. Note the lighting inside the cube, and the shadows created around it because of this.

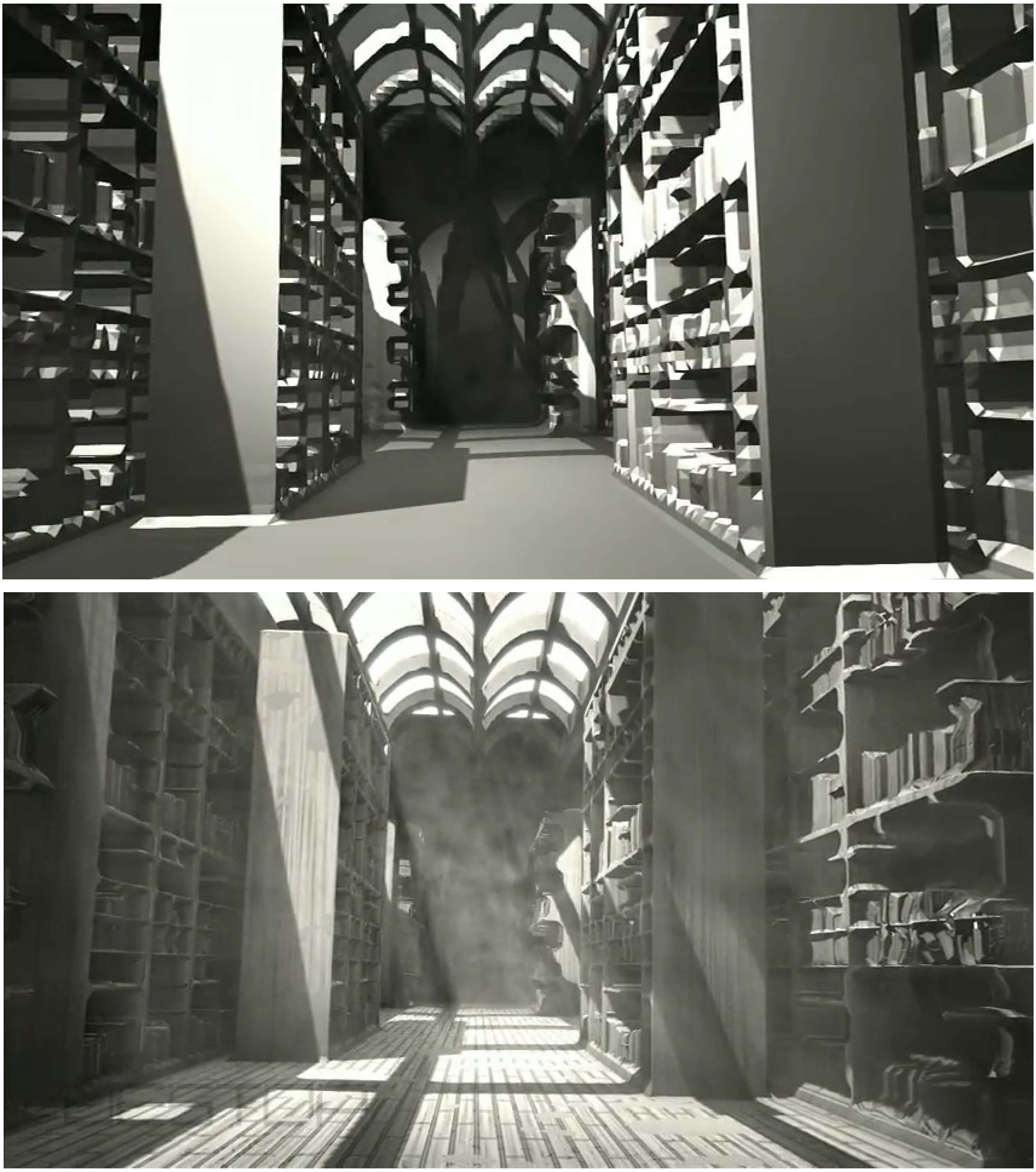


"Lifeforce" by ASD. 1st prize at Assembly demoparty 2007. The entire demo is over 7 minutes long. The metallic, well-defined foreground is superimposed on an abstract, morphing background.

Signed distance field meshes

There are many ways of representing “meshes”, or the complex geometry that gets shown to screen. One of the newer techniques is the concept of “signed distance fields”, commonly used when the demo needs to be constrained to a size limit.

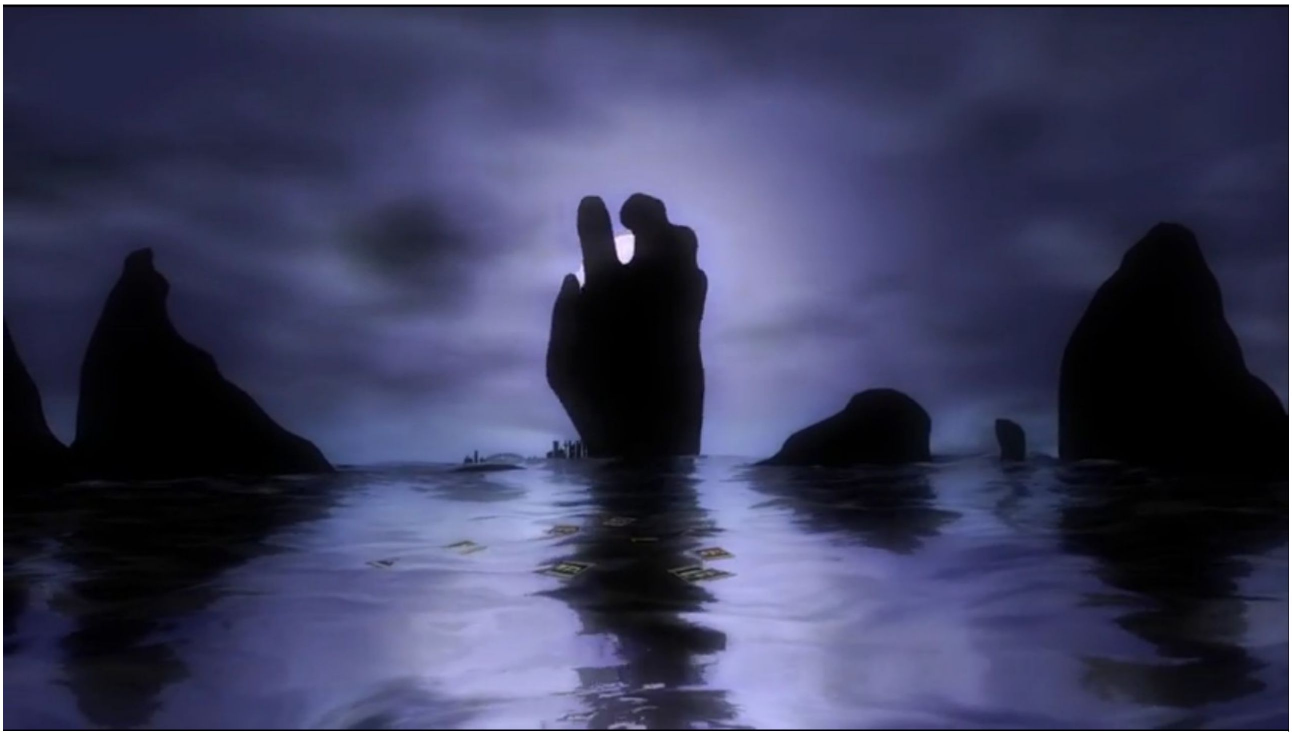
The two screenshots below also show the importance of texturing. Without textures (above), the scene looks artificial and undetailed. With textures (below) the scene comes to life, and feels fleshed out. Demos usually have both 2D and 3D artists working on a team, where the former provide the textures and the latter provide meshes.

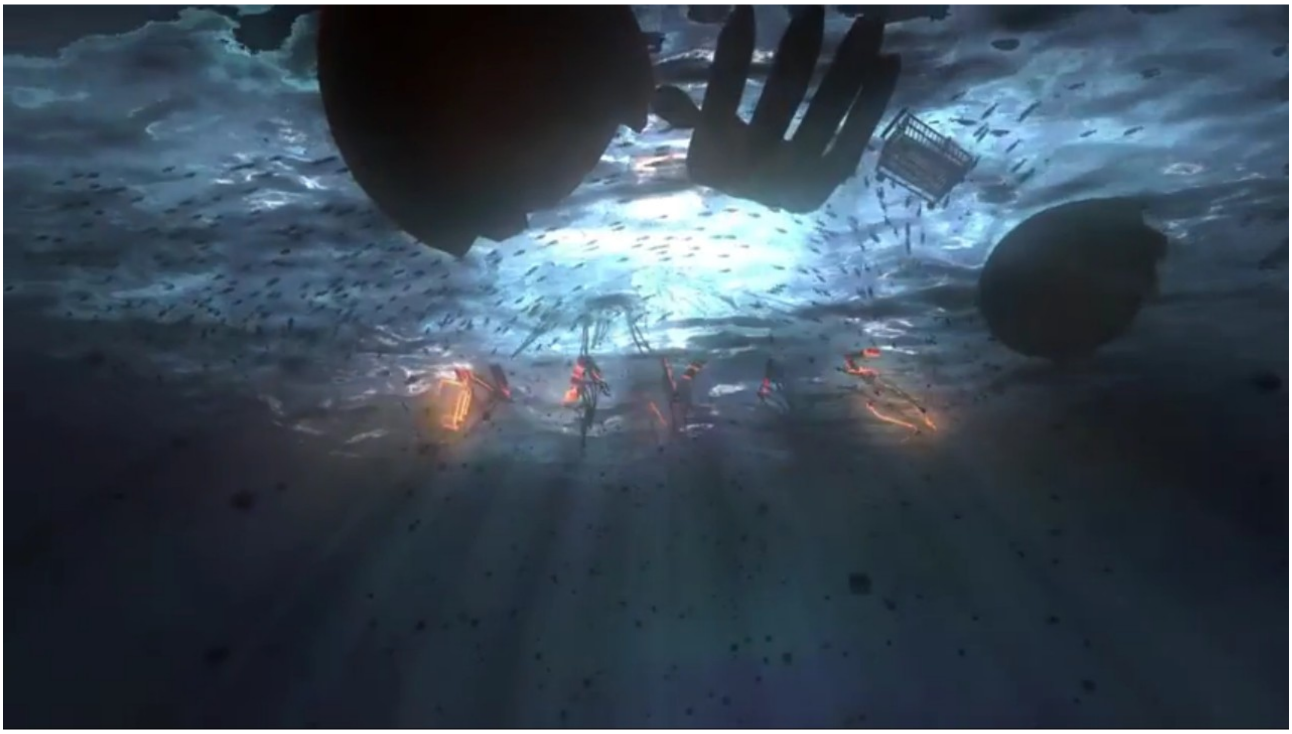


Both screenshots from "uncovering static" 64KB demo by Fairlight & Alcatraz. 2nd prize at Assembly demoparty 2011.

Faked water effects using normal mapping

Sometimes it is necessary to fake effects that would otherwise be too expensive to calculate accurately. The water effects seen below are both faked, using mathematical “noise” functions to make it look like the water is rippling. In fact there is no physics behind it – it just looks cool.

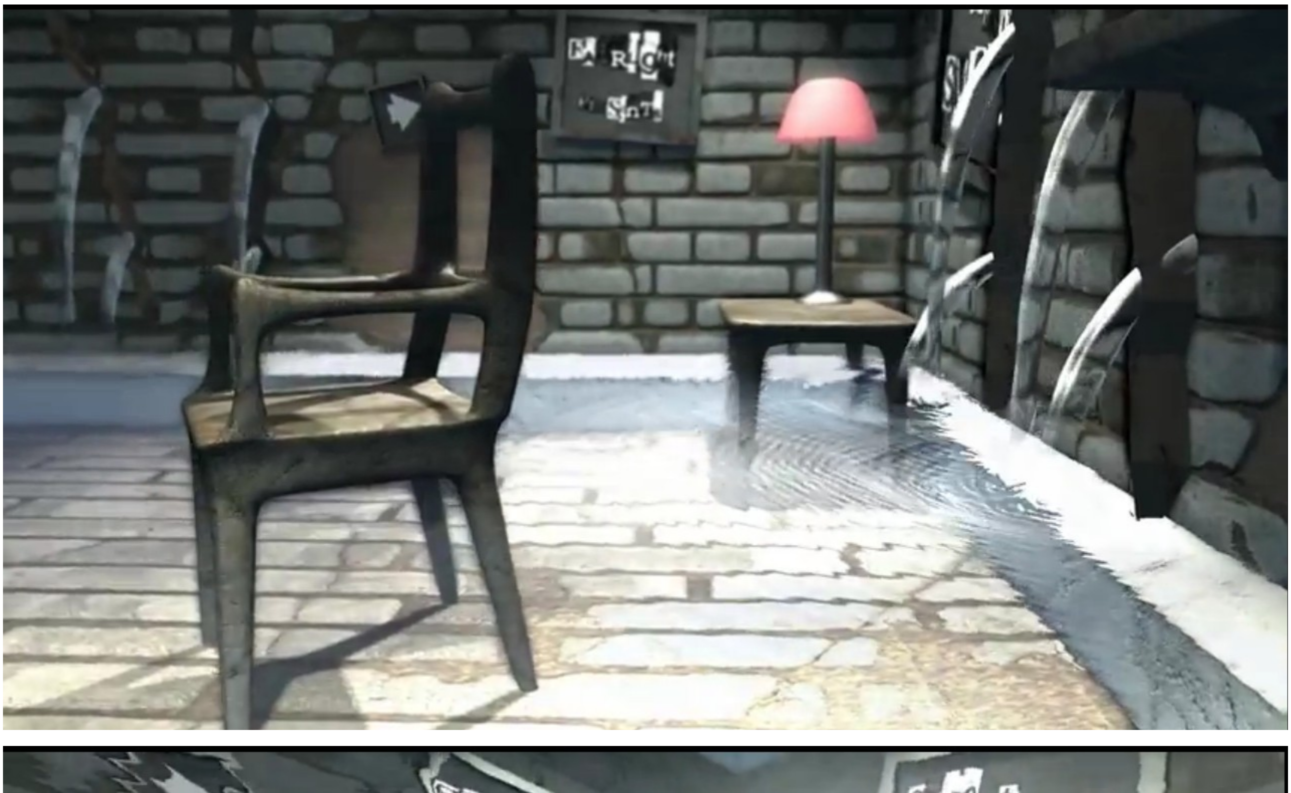




Both screenshots from "Lifeforce" by ASD.

Faked water ripples

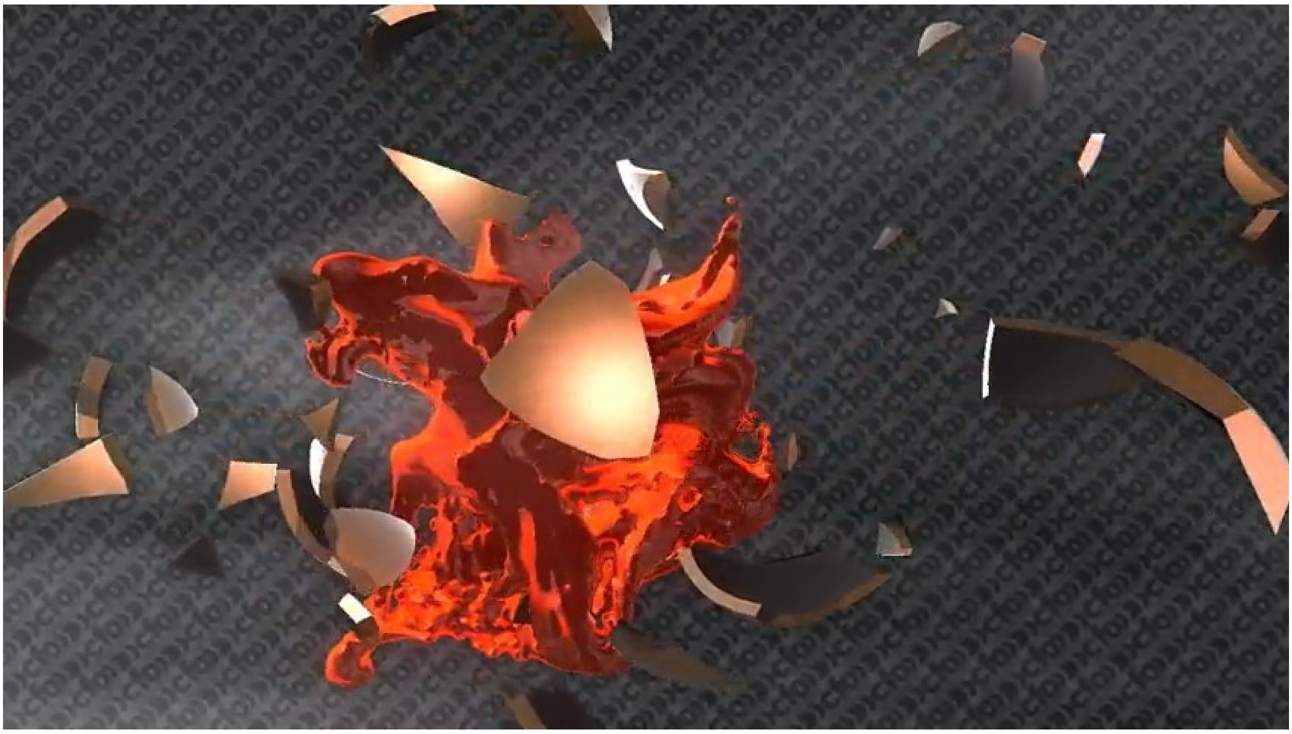
This one looks realistic because of the ripples created. However, the water is actually being modeled as two separate entities: the falling streams are computed separately to the pool of water, which is being modeled using a heightmap to show ripples. There is some physical basis to this simulation, however it is at best an approximation of true water behaviour.



Both from "panic room" 64KB demo by Fairlight. 1st prize at Assembly demoparty 2008.

True fluid simulation using particles

This demo uses hundreds of thousands of particles to model true fluid behaviour using accurate mathematical equations, rather than making approximations. Because of this it only runs on very high-end computers, however the final result is accurate and quite spectacular.



"Frameranger" by Fairlight & orange & CNCD. This demo only runs on very high-end machines.