

demo reel breakdown

01) “Photlcon” (stop-motion animation)

Stop-motion animation of a Polaroid land camera opening with Flash. Shot frame-by-frame using a DV Camera and FrameThief. A curved showcard painted Ultamate green was used as a mini-cyclorama and the background was then keyed out in After Effects using the Keylight Plug-in. The flash of the bulb was created using the Flare filter on an adjustment layer.

02) Night Sky

The sky was created using Fluid Effects in Maya. A dynamic 2D fluid container has several emitters positioned at the outer edges of the container. The general motion of the fluid is controlled using various attributes of the fluid shape such as turbulence, viscosity, buoyancy etc. Volume axis fields (one per emitter) were used to push the fluid outwards into the container. The color attribute of the fluid shape is a ramp with various shades of blue and the input is set to speed. Small “invisible” cylinders were used as collision objects so the fluid would swirl around the stars. The collision also reduces the speed of the fluid so that the color changes to a pale blue. The stars are a particle system that was created by drawing on a plane with the particle tool. A volume axis field was used to animate them moving towards the camera. The simulation was baked and the animation curve was reversed so they fall from behind the camera and land on the “invisible” plane. The particles were instanced to diamond-shape geometry with a Blinn material. The specular color of the shader has a “super-white” value of 2 to enhance the glow.

03) Animated QTVR object

Animated QuickTime VR object movie. The object is a sphere with a water/glasslike shader and a twist deformer animated to morph the shape. The animation is 6 frames long – it bounces up and down because it is an oscillating forward/backward loop. The camera has a “pan” and “tilt” node, animated to rotate around the object in 20-degree increments per frame - it then tilts 20 degrees and does another 360 degree pan so the total number of frames to “virtually” photograph a 360-degree-view of this object at is 324 (18x18). Since the object has six frames of animation, it had to be photographed it six times - once for each position of the animated object - so the total number of frames required was 1944 (324 x 6). The QTVR object movie, assembled in VR Worx, has 6 layers - one for each frame or “state”. An HDR image mapped to a mental ray environment sphere was used for image-based lighting and virtual environment.

04) Neptune

The planet and rings were created and animated in Maya. The planet orb is a NURBS sphere with an animated texture for the swirling gasses. The texture map was created in After Effects using the “Turbulent Displacement” distortion filter. The offset attribute moves the distortion mesh across the surface and wraps on the edges so that it wraps “seamlessly” when applied to the sphere. The implosion of the sphere was animated using the After Effects Shatter Plug-in. The blobby particles for the rings were emitted from a ring-shaped NURBS surface. A “U Ramp” with interpolation set to “none” was used to control the emission. The color of the Blinn shader applied to the particle shape is a linear projection of a multi-colored striped circular ramp. A similar grayscale ramp is mapped to the shader transparency to create the delineation between the ring segments. A vortex field was used to spin the particles around the planet orb and the conserve was set to .8 (to reduce momentum and maintain the circular shape). To make the rings spin outward, the magnitude of the vortex force was increased and the conserve was set to one.

05) MEL Scripting – Tornadomaker

Demonstration of a Mel Script that creates a tornado, debris cloud and interface. The interface window has four tabs containing controls for the funnel cloud, funnel cloud sprites, debris cloud and debris cloud sprites. The interface always the user to adjust min/max values for emitter rate, lifespan, vertical speed, spin rate, cloud radius and position jitter using sliders as well as a ramp to control the shape. It also allows the user to control the sprite opacity, twist, scale and color.

06) Tornado

MatchMover was used to track camera movement, which was then imported into Maya. The funnel and debris clouds were both created by animating geometry with a lattice deformer, which then was used as a surface emitter. The distance from the center was calculated using the linstep function and a pointOnCurveInfo node so that the particles closer to the center would rotate faster. The vortex of clouds in the sky at the top of the tornado was created using volume axis field. All elements were composited onto the background plate using After Effects.

07) **Light Streaks**

Abstract animation created using After Effects and the Trapcode Particular Plug-in. Expressions were used to animate the particles based on the amplitude of the sound.

08) **MEL Scripting – Particle Controller and Random Instancer.**

Demonstration of a Mel Script written to emit particles from a NURBS surface, create a controller to interactively position the particles into rows and columns on the surface, and then instance geometry to the particles using a random Object Index number. The user is prompted to select the geometry that will be used as a surface emitter and then prompted again to select the geometry to be randomly instanced to each particle. The script creates a surface emitter and names the particles based on the name of the surface from which the particles are emitted. The surface used to emit the particles is connected to the particle shape as a goal object and a runtime expression is added to the particles. The runtime expression generates integer particle Ids that are used place the particles into evenly spaced rows and columns on the surface. A controller is created so the user can adjust the amount of rows and columns interactively from the channel box during playback (the max count of the particle shape is adjusted automatically).

09) **House of Cards**

The house itself was constructed entirely with particle systems and referenced geometry. The cards (instanced geometry) are polygon cubes textured with a building material on one side and images of playing cards on the other. The MEL script (described in item No. 08 of the breakdown) was authored specifically for this project to align particles into rows and columns on a NURBS surface as well as adding randomization to the scale and position.. The particles are goal to the surface and have an expression to release the goal based on the magnitude the input force.

10) **Solar Flares**

Animated Perlin noise shader was applied to a NURBS sphere for color/texture and displacement. The flares are comprised of 6 particle systems (2 for the surface flames, inner flares layer and outer flares layer). The NURBS sphere (for the orb) is a surface emitter of points that emit tube (sw) particles (with a particle cloud shader) for the actual flares. Radial fields push the particles back in toward the orb creating the arcing loop shapes. Turbulence fields with expressions animating the z-phase were used for randomness and secondary animation.