Hand-Drawn looking volumetric effects in the Peanuts Movie

Ilan Gabai Alen Lai
Blue Sky Studios *

Abstract

For the first time in the franchise's history, the Peanuts gang is being brought to life in a CG feature film. The biggest challenge we faced was how to successfully maintain the iconic style established over the past 60 years. Some of the FX requirements for this film included the need to maintain a 'hand drawn' look, to obey cartoon physics, and to animate on 2's (matching character animation). To achieve this we departed from our usual simulation techniques, and developed a mix of hand-animated and procedural techniques.

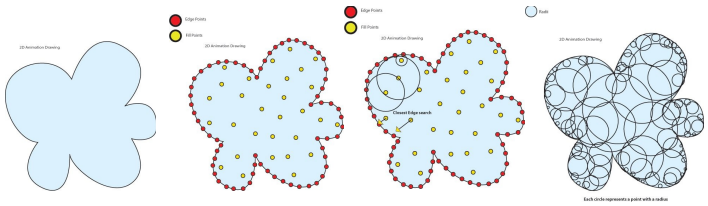
1 Pigpen's Dirt

Pigpen's dirt is a good example for the use of procedural techniques to create a 3d version of Pigpen's graphic dirt cloud. We began by studying the Peanut's strips, and broke down Pigpen's dirt in to 4 simplified elements: pen-stroke arcs, highlights, filler dust, and specks. Rather than using more common simulation techniques, we constructed the dirt cloud by procedurally building all of the elements from a single underlying particle system, and integrating them at the end. This assisted in maintaining uniform motion between all elements. Furthermore, due to the nature of the comic strip look, we needed our FX to be camera dependent. This ensured that the FX elements would look consistent and front-facing regardless of the camera angle in the shot. To establish a hand-drawn look of the effect, we identified Schulz's drawing characteristics that we wanted to integrate in to our effect. These included varying line quality, asymmetry, and controlled arc overlap. Using Houdini's procedural work-flow, we were able to develop tools that mimicked those characteristics in an automated fashion. We began by instancing semi-circle lines on to the particles. We converted the arc-lines to volumes and applied a volumetric noise to the collection of arcs. This gives them an asymmetric "hand drawn" feel. Though the arcs were placed in different Z space, we needed them to appear intersected in camera view. So using the camera vector, we extruded out the arcs away from camera and converted them to volumes. Using the inverse of the extruded volume, we multiplied that by the volumetric arcs. This resulted in a much cleaner, aesthetically pleasing result. In order to maintain a consistent look in all scenarios, we "baked" lighting in to the dirt-cloud by creating volumetric elements to represent highlights. The specks were original Schulz drawing scanned in as a library of 3d geometry, and instanced on to the particles.

* Email: {igabai,alen}@blueskystudios.com

2 From 2D Pen lines to 3D Poofs

For damage smoke FX notable in the "Red Baron" scenes, we initially tried to create volumetric FX that resembles Schulz's drawings. While they had an appealing look, we had a hard time hitting notes from the art dept to precisely match Schulz's poof shapes. Animation provided us with draw overs on top of their animation frames to assist us in matching, but it would be too time consuming to modify every element to match. So instead of struggling to match the hand drawn guides, we took a different approach that would source the hand drawn graphics and build a poof cloud from it. Our technique leverages a sequence of 2D hand-drawn images to generate two sets of points: Fill and Edge. The 2D points are transformed into true 3D. This transformation maintains the same X & Y relationship to frame that the drawing had, but allows artists to control the depth. Each Fill point is given a radius based off the distance to it's nearest edge point. The result is a 3D volume that matches the exact silhouette of the 2D drawing.



3 Running FX on 1's, 2's & 3's

The animation style for this film was very different from our previous films. Characters in this movie were animated in a limited fashion in order to stay true to the original Peanuts specials. Our FX work shared screen space with these characters, and therefore needed to be treated similarly. There was no easy way for us to determine how a character was animated, especially when working with geometry caches. In some instances characters were animated on a combination of 1's, 2's and 3's. So we developed a tool that would analyze the incoming caches, and based off of the geometry's delta from frame to frame, it would drive our FX animation to match. If the tool detected that the cached geometry has not changed from the previous frame, it would lock the FX to the previous frame as well. This allowed us to maintain consistency between FX and animation.

SIGGRAPH 2015 Talks, August 09 – 13, 2015, Los Angeles, CA.
ACM 978-1-4503-3636-9/15/08.
http://dx.doi.org/10.1145/2775280.2792536