"MEGAMIND" : FIRE, SMOKE AND DATA

Krzysztof Rost, Greg Hart, Scott Peterson
DreamWorks Animation

Introduction

Exploding a super hero in DreamWorks Animation's "MegaMind" was a large scale task. The detail and scale of the explosion was to be massive and the collaboration between effects artists unprecedented. This created several technical hurdles and eventually several innovations such as: a fluid clamping plugin, a volume splatting tool adjustable by noise, and a collaborative caching tool.

Dependency

At the start we identified the list of required elements: fire, dust, smoke, debris trails, shock wave, splashes, water interaction and a light beam. The production schedule required us to create a smooth work-flow between seven effects artists working at the same time on shared data. Our goal was to design a clear plan for communicating responsibility, not only between effects artists but also between the lighter and compositor.

Fire and smoke

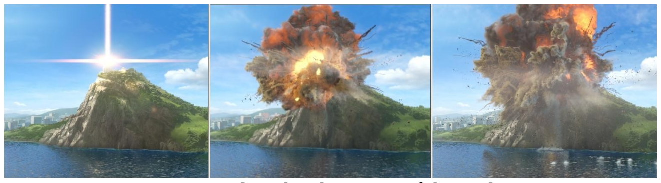
Figure 1: Final rendered images of the explosion

We developed several tools to help us achieve greater detail in our Maya fluid simulations and greater control over our iterations. The first notable development was a Maya plugin which allowed for manipulations of the fluid. The plugin's main feature was to clamp user-specified fluid fields such as density, temperature, fuel, or velocity at user-specified rate. Our implementation used a noise field to mask out areas for clamping to be applied. The noise field could be animated over time to add a variant as to how values are affected. Another function that was implemented was temperature erosion. The result was that a significant amount of perceived detail could be added. Also, more control was attained over what would have otherwise been an unstable simulation.

The second development was another Maya plugin which allowed stamping fluid attributes into a fluid container using particles by simple splatting at the simulation time. By coupling a noise function with the plugin , we were able to transfer information from each particle using noise as a mask into the fluid grid, and as a result achieve a better distribution. Fig(2).

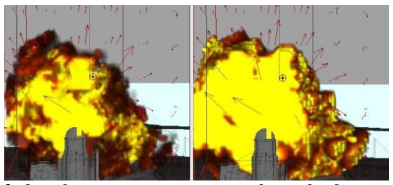
Figure 2: on left distribution using noise, on the right direct particle transfer

We also designed a particle clustering technique to further control the shape and motion of the explosion. In a nut shell, the clustered particles were staggered with an offset along the parent velocity vector and at the same time rotating around the same axis. This method helped to create a sense of a rolling fire ball. Fig(3-left).

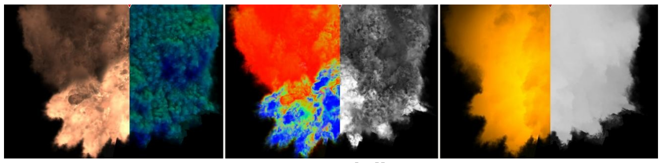
Figure 3: Fireball pass

For the fire ball pass, density, temperature and fuel were simulated simultaneously. At the rendering stage smoke and fire were rendered separately Fig(3), and later composited together. Fig(4-center).

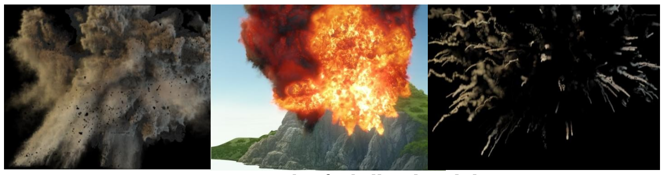
Figure 4: smoke, fireball and tendrils

For the smoke and ash layer, the simulation conditions were kept the same, only subtly modifing density, temperature, and fuel fields. Only density was exported and rendered. Fig(4-left). Lastly, we added layers such as debris, water splashes, ripples, volume lighting and used extensive compositing to further increase visual richness. Fig(5).

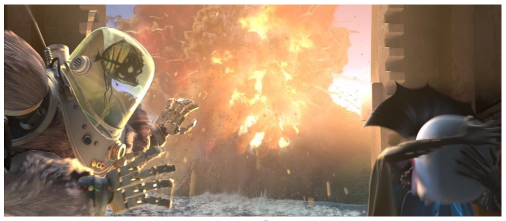
Figure 5: Final composite

Data

A new level of collaboration tools were necessary to coordinate between the seven FX artists involved. The observatory explosion required generating massive amounts of particles, volumes, curves, geometry, and deep image data. Our goal was to work as efficiently as possible by generating heavy data only once. To achieve this level of collaboration, we designed a pipeline tool which uses another artist's cached data when it exists and generates it when it doesn't. A publishing paradigm allowed artists to decide when and what to share with others. By systematically sharing cached data, artists could stay in sync, save on disk space and reduce the number of times the same data is created.

Credits

Krzysztof Rost, Greg Hart, David Lipton, Scott Peterson, Devon Penny, Kyle Maxwell, Tobin Jones, Shaun Graham, John Allwine, Markus Burki, Gianni Aliotti, Abhik Pramanik, Eli Bocek-Rivele

email: greg.hart@dreamworks.com
email: scott.peterson@dreamworks.com