
SIGGRAPH2011

Dynamics in Games:

Toy Story 3 and Cars 2

Phil Knight

Thanks for coming to our talk today. My name is Phil Knight. I am a lead programmer at Avalanche Software in Salt Lake City, and part of Disney Interactive Studios. Today I will be covering the use of Dynamics in our games and tools.

Overview

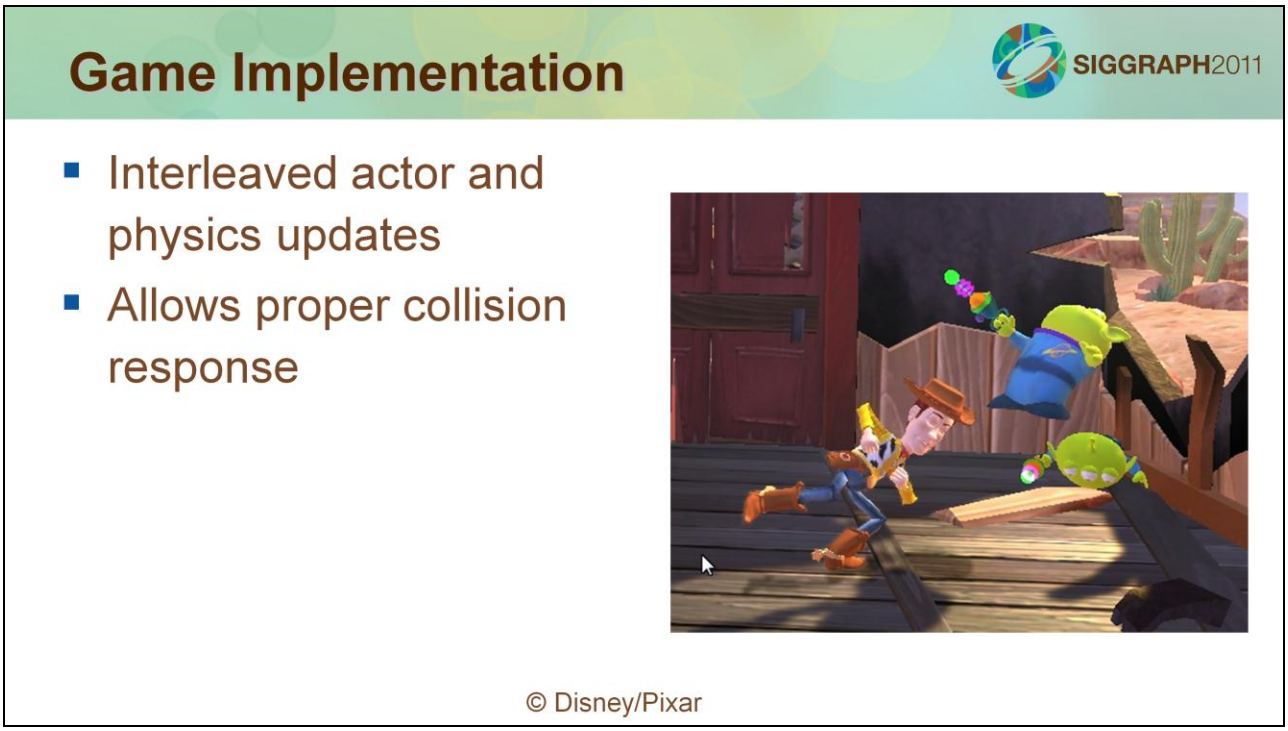
SIGGRAPH2011

- Our games use Bullet:
  - Bolt
  - Toy Story 3
  - Cars 2
- Also some tools:
  - Frag
  - AvaPhysics

We have integrated the Bullet Physics Library into our games, starting with Bolt, which used Bullet's collision detection libraries, and continuing with Toy Story 3 and Cars 2, which additionally utilize Bullet dynamics, including vehicle simulation. Our games are multi-platform, running on PS3, Xbox 360, Wii and on Windows.

Some of our tools use Bullet internally, including Frag, a Maya plug-in that is an automatic model fragment generator, and AvaPhysics, a Maya plug-in that allows real-time simulation and physics based animation key-frame generation.

Game Implementation



Our game actor update and physics updates are interleaved, with two physics updates per 30hz frame. Some actors want per frame updates, and others want an update with each physics update. This interleaving allows proper collision response between game actors and rigid bodies, and the collision response is always executed per physics update.

Toy Story 3

- Characters not physics based
- Rigid body mini games
- Bullet ray-cast vehicles

In Toy Story 3, our characters are simulated with custom platforming game characteristics, and are not physics based, but many of our mini-games are based on rigid body dynamics and vehicle simulation. Our vehicles are based on the Bullet ray-cast vehicle and are customized to behave more like toy cars.

Minigames

Some examples of the physics based mini-games are toy car soccer, similar to that seen on TV's Top Gear, bowling, cow tossing, vehicle and horse races with destructible barriers, and carnival style ball toss games.

Exploders

SIGGRAPH2011

- Damage states
- Explosion with simulated fragments

We have a type of game actor called Exploder that explodes in response to a weapon impact. Exploders can have several pre-built damage states that replace the initial model with a progressively more damaged model until the final stage, where the model will explode with prebuilt fragments controlled by the physics simulation. These fragments are generated offline, either with our Frag tool or by hand.

Here is a video showing dynamics in Toy Story 3.

Cars 2

- Primarily a racing game
- Characters are vehicles
- Based on TS3 vehicle model
- Tuned to be less toy-like

In Cars 2, which is primarily a racing game with weapon power-ups, the characters are vehicles just like in the movie. We based the Cars 2 vehicle model on our model developed in Toy Story 3, but tuned it to feel less toy-like.

Cars 2 Simulation Style

- In between sims and cart games
- Animated and simulated as characters

The simulation style in Cars 2 sits between hard-core simulations and cart racing games, matching the graphics and animation style of the Cars world. Since our cars are also characters, we give them unrealistic abilities such as the ability to wall ride and jump without ramps.

More Cars as Characters

© Disney/Pixar

Here are some more in-game screenshots showing our cars that are characters.

More Cars as Characters

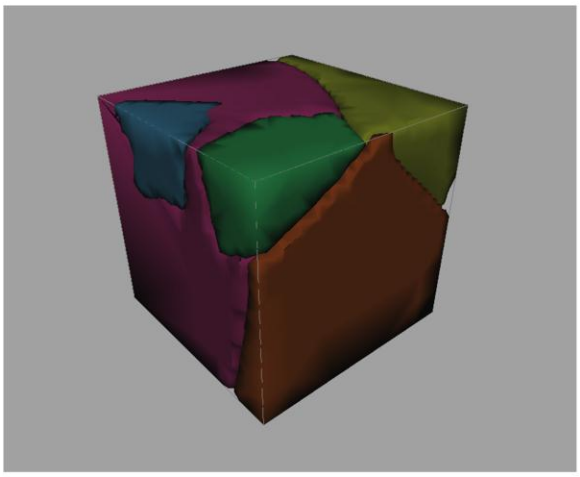
<pause>

More Cars as Characters

<pause>

Here is a video showing dynamics in Cars 2.

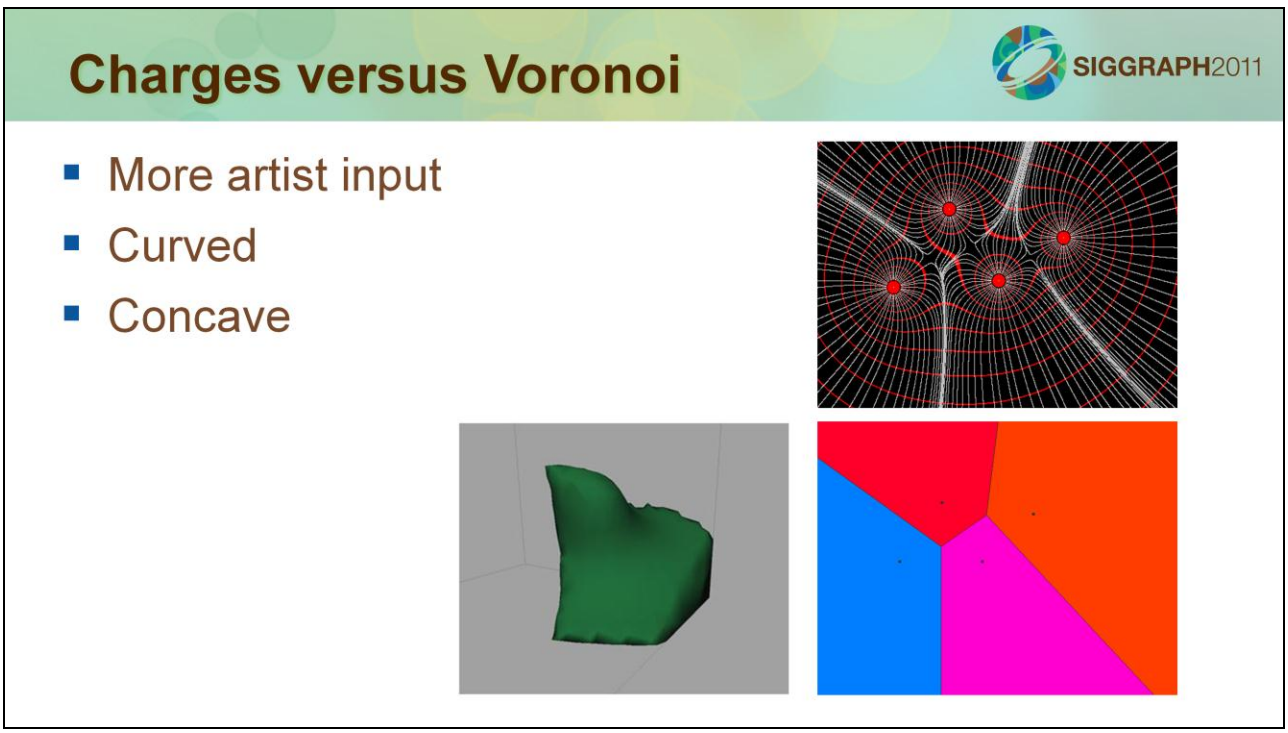
Dynamics in our tools - Frag



- Generates fragments of a mesh
- Based on electrical charges
- Meshed using Dual Contouring

Our Frag tool automatically generates fragments of a mesh based on an electrical charge model. The generated fragments are meshed using Dual Contouring (2), which allows the outline of the fragments to closely match the original model.

Charges versus Voronoi



Comparing electrical charges with Voronoi, the two biggest differences are that the additional charge strength parameter allows more artist input at authoring time, and that electrical charges generate rounded edges and concavities. These concavities are visible in 2d, but are more pronounced in 3d. For more details, read the Game Developer Magazine article, “Fragged”, listed in the references (1).

Frag: Before and After

This image shows a crate model on the right with its fragmented counterpart on the left. Notice the matching outlines of the two shapes.

Dynamics in our tools - AvaPhysics

- Real-time dynamics in Maya
- Physics preview
- Physics generated key frames
- Collision primitive authoring

Image by Annielogue (Own work) [CC-BY-SA-3.0 (www.creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons

Our AvaPhysics Maya plug-in allows real-time physics simulation in Maya, including a playable labyrinth. AvaPhysics uses the Bullet physics library for its dynamics calculations. This plug-in can be used to either preview physics effects that will be used in conjunction with animation using the same dynamics engine as in our game, or can be used to generate key-framed animations with physics based elements. In addition, AvaPhysics allows collision primitive authoring, including automatic fitting of shapes to meshes. Convex hulls are also generated within Maya, allowing authoring-time refinement.

Data Pipeline

- Mesh processing
- Edge normals
- BVH generation
- Load-in-place data

Image by Traroth (Own work) [CC-BY-SA-3.0 (www.creativecommons.org/licenses/by-sa/3.0/)], via Wikimedia Commons

Our data pipeline also includes the Bullet Physics Library to assist in collision data processing. At export time, our meshes are processed to generate edge normal information and to generate a quantized bounding volume hierarchy with one axis aligned bounding box, or AABB, per triangle. This hierarchy is then compressed to one AABB per five triangles, resulting in an 80% memory savings compared to quantization alone. This allows us to use the data on the memory limited Wii platform. Using this data, our pipeline tool generates a load-in-place file that ensures quick loads in the run-time. Our other collision shapes are simple enough that they are just passed through as reflected data to the run-time engine.

Results

SIGGRAPH2011

- Varied game play
- Playful realism
- Property tuned experience
- Fun!

Through the use of dynamics in our games, we achieve the ability to have varied game play, and though our games are meant to have a playful style, the element of realism adds the necessary familiarity that allows a player to feel comfortable in our world. By varying the level of realism, we are able tune the experience to match each property we are developing. Finally, physics based game play is just fun!

References

1. Robert Perry and Peter Wilkins, "Fragged", Game Developer Magazine, December 2010, Salt Lake City, Utah
2. Tao Ju, Frank Losasso, Scott Schaefer, Joe Warren, "Dual contouring of hermite data", Proceedings of the 29th annual conference on Computer graphics and interactive techniques, July 23-26, 2002, San Antonio, Texas

Thanks for attending my presentation. I hope you found it interesting. Are there any questions?


SIGGRAPH2011