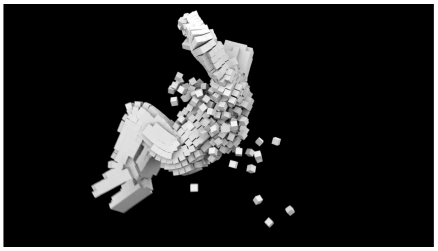
End of Line: Character Destruction in "Tron: Legacy"

Atsushi Ikarashi, Digital Domain
Eddie Smith, Digital Domain
Ryo Sakaguchi, Digital Domain
Brian Gazdik, Digital Domain
Nafees Bin Zafar, Dreamworks Animation


1 Introduction

In the world of Tron: buildings, vehicles, and humanoid programs are constructed of tiny, glass cubes. When a program is destroyed it is derezzed into as many as 500,000 cubes. We chose to use our proprietary RBD toolset DROP, which was developed for 2012 [Bin Zafar et al., 2010]. However, this effect was quite different from destruction effects for ordinary rigid objects such as buildings and vehicles, and the tools from 2012 were not adequate for the character derez effect in Tron. These characters were soft body objects that were being deformed by key-framed animation. While some parts of the body were derezzing, the rest of the body continued its animation. The destruction was art directed instead of being driven by physical factors such as force or collision. This made it very difficult to to create an effect that looked natural and believable. There were 10 different types of characters that needed to be derezzed in 30 shots. We had to develop robust tools that allowed for much more artistic control while also addressing the unique technical difficulties of this effect. We developed a flexible user interface to handle all the challenges of derezzing human characters.

2 Preparation for RBD simulation

The FX team was given water-tight, T-poses for each character model that needed to be derezzed. These were then diced into cubes. When there needed to be a higher level of detail, the cubes would be split a second or even a third time, so that they could continue to break down into finer pieces. In order to maintain an interactive workflow, and preserve scalability each cube was represented as a single, centered control point with RBD attributes. The control points were then deformed by a barycentric coordinate based deformer, so that they animated the same way as the original key-framed character animation. Once a character began derezzing, their cubes independently switched to a dynamic state. Their constraints with other cubes was removed and they individually separated. In order to look organic, the body first had to break into random shaped chunks before dissolving into cubes, so we added a number of constraints between chunks of cubes. In some shots a character had to tumble in a rag-doll like motion while it was derezzing, so we added constraints around its joints to simulate this motion. We developed an interface where artists only had to set the position of the joints and the number of chunks, then all bonds were set automatically. The derez propagation was quite different between shots, so we had to control it in a way that could be as art directed as possible. Initially, only the start position of the derez was needed, and the resulting propagation was automatically calculated. However, this made it difficult to art direct the propagation, so we chose to implement a bounding geometry to activate when certain parts of the body would derez. In some cases a few intact (not yet derezzed) parts of the body, such as arms, had to lose support and fall off as a result of the torso derezzing. So we used another bounding geometry to determine when these parts could actively detach from the body. Regions dynamically activated by the derez bounding geometry entered the simulation first as chunks. Artists could then set the rate at which each chunk gradually broke into cubes from their edges. Finally, the time at which each cube switches from key-framed animation to dynamic simulation and the times at which constraints between cubes was cut were calculated and stored as attributes in the control points.

3 RBD simulation

In order to optimize simulation runtime all pieces were simulated as implicit cubes, and we used a multi staged approach. First, the body control points with the derez timing attributes were input to run the initial simulation. These control points represented the largest sized cubes, so the geometric complexity was relatively small and the simulation ran fast. After these cubes released their constraints and entered a dynamic state, they waited a predetermined amount of time before switching to the second simulation. The half sized cubes followed their parent cube's animation until this switch was initiated. Each of the original large sized cubes appeared as though they had randomly broke into 8 octants. For shots requiring a higher level of detail this process was done a third time. This meant that the final number of cubes was 64 times as many as the original large cubes, but since constraints were not necessary by the second stage in the process it ran very fast. Artists could quickly review the result of the first sim, and make fast adjustments until they created a satisfactory result. Afterwards, the second and third sims could be ran automatically, thus saving the artist a lot of time. The final output points of the simulation were then replaced with the high res diced geometries. Any intact body parts that had yet to derez were then rebuilt as seamless geometry.

4 Conclusion

Our goal was to create an efficient set of tools to derez a character with a simple interface that was art directable. The point based data representation of DROP created a scalable foundation to build these tools. Cube IDs, physical parameters, and derez timings were stored in control points that made them easy to manipulate. This also permitted us to modify or build the derez tools quickly. With the simple interface of the derez tools, artists could focus on propagation of the derez, and general physical parameters for the RBD simulation without having to take care of complicated creation and configuration of constraints. These tools provided interactive workflow for the artists to iterate the simulations quickly, and get the best results possible.

References

Bin Zafar, N., Stephens, D., Larsson, M., Sakaguchi, R., Clive, M., Sampath, R., Museth, K., Blakey, D., Gazdik, B., Thomas, R. 2010. Destroying LA for "2012". In *ACM SIGGRAPH 2010: Sketches and Applications*.