

Tron Legacy

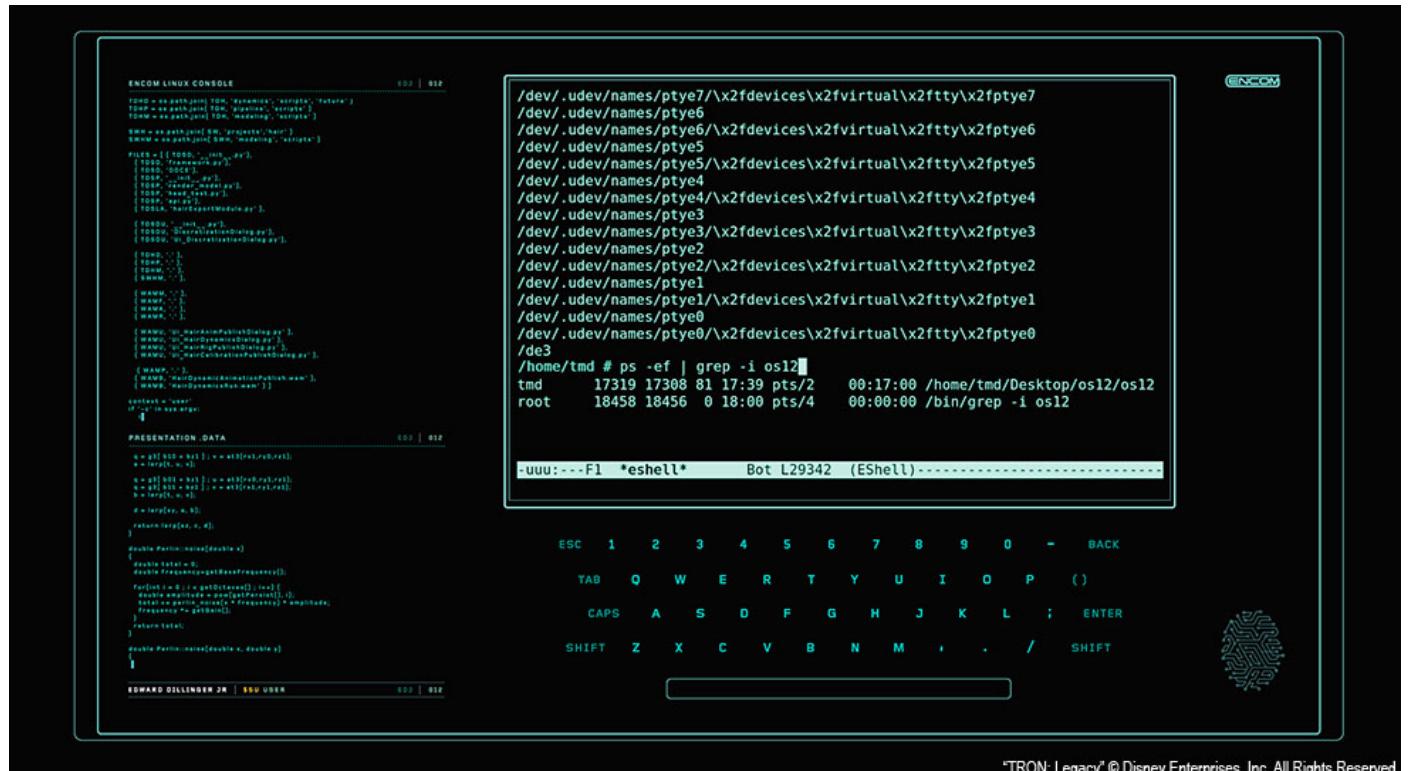
I spent a half year writing software art to generate special effects for Tron Legacy, working at Digital Domain with Bradley "GMUNK" Munkowitz, Jake Sargeant, and David "dlew" Lewandowski. This page has taken a long time to be published because I've had to await clearance. A lot of my team's work was done using Adobe software and Cinema 4D. The rest of it got written in C++ using OpenFrameworks and wxWidgets, the way I've always done it with this team ;) Uniquely however, Digital Domain's CG artists were able to port my apps over to Houdini for further evolution and better rendering than OpenGL could ever provide. Special thanks to Andy King for showing me that what seasoned CG artists do at DD is actually not so far off from what's going on in the Processing community.

Interview with GMUNK about the team's process



In addition to visual effects, I was asked to record myself using a unix terminal doing technologically feasible things. I took extra care in babysitting the elements through to final composite to ensure that the content would not be artistically altered beyond that feasibility. I take representing digital culture in film very seriously in lieu of having grown up in a world of very badly researched user

interface greeble. I cringed during the part in Hackers (1995) when a screen saver with extruded "equations" is used to signify that the hacker has reached some sort of neural flow or ambiguous destination. I cringed for Swordfish and Jurassic Park as well. I cheered when Trinity in The Matrix used nmap and ssh (and so did you). Then I cringed again when I saw that inevitably, Hollywood had decided that nmap was the thing to use for all its hacker scenes (see Bourne Ultimatum, Die Hard 4, Girl with Dragon Tattoo, The Listening, 13: Game of Death, Battle Royale, Broken Saints, and on and on). In Tron, the hacker was not supposed to be snooping around on a network; he was supposed to kill a process. So we went with posix kill and also had him pipe ps into grep. I also ended up using emacs eshell to make the terminal more l33t. The team was delighted to see my emacs performance -- splitting the editor into nested panes and running different modes. I was tickled that I got emacs into a block buster movie. I actually do use emacs irl, and although I do not subscribe to alt.religion.emacs, I think that's all incredibly relevant to the world of Tron.

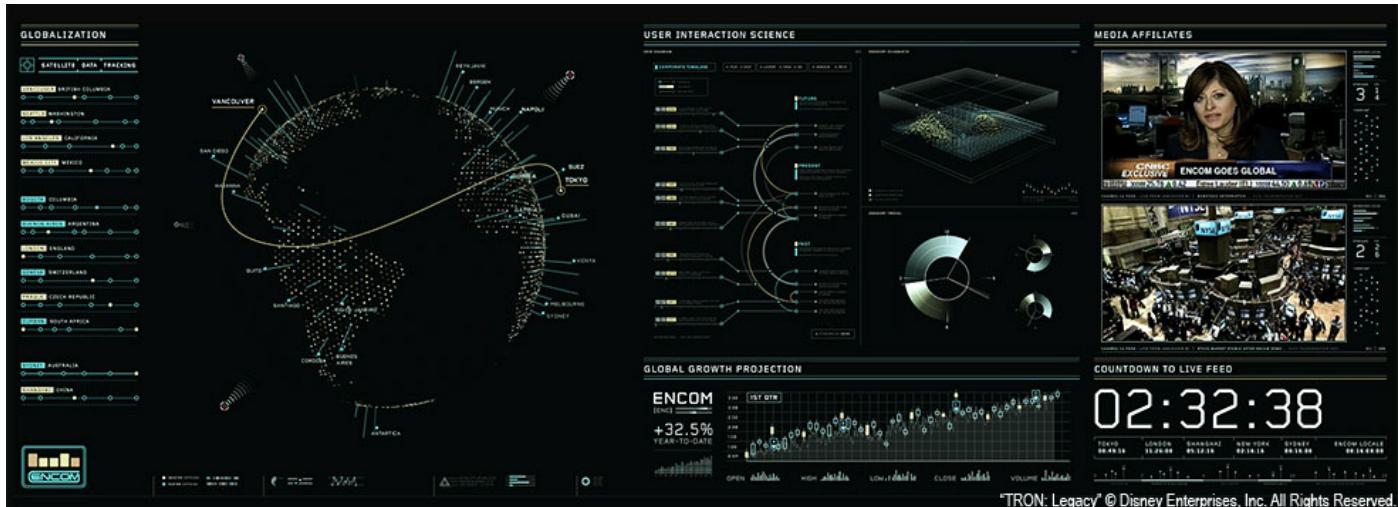


'TRON: Legacy' © Disney Enterprises, Inc. All Rights Reserved.



"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.

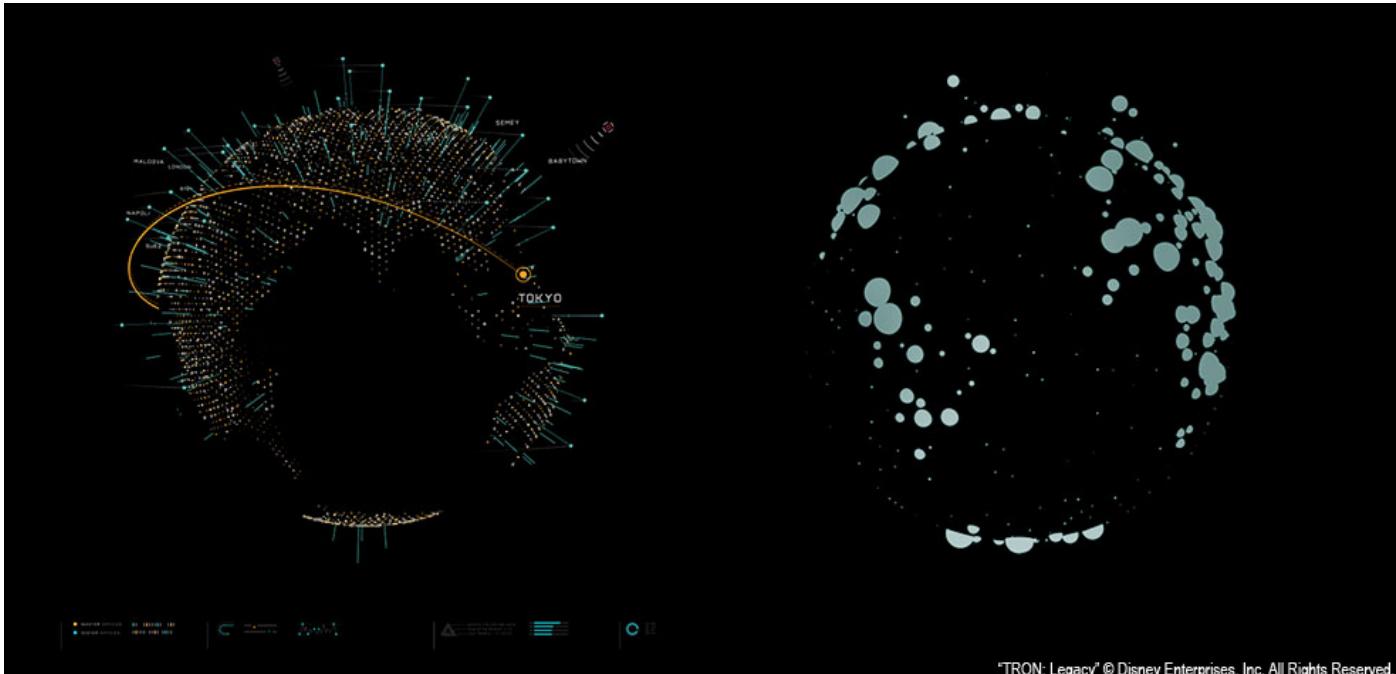
HexVirus is a spherical map of the globe that features vector outlines of the continents. These continent vectors are slowly eaten away by a more hexagonal representation. Algorithmically, this is a path stepping function which looks ahead for the closest matching 60-degree turns. The HexVirus globe was used in the executive board meeting scene, and also inside the grid as a visual aid in CLU's maniacal plan presentation. In the board room interface, the globe element is surrounded by the lovely work of my team.



"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.

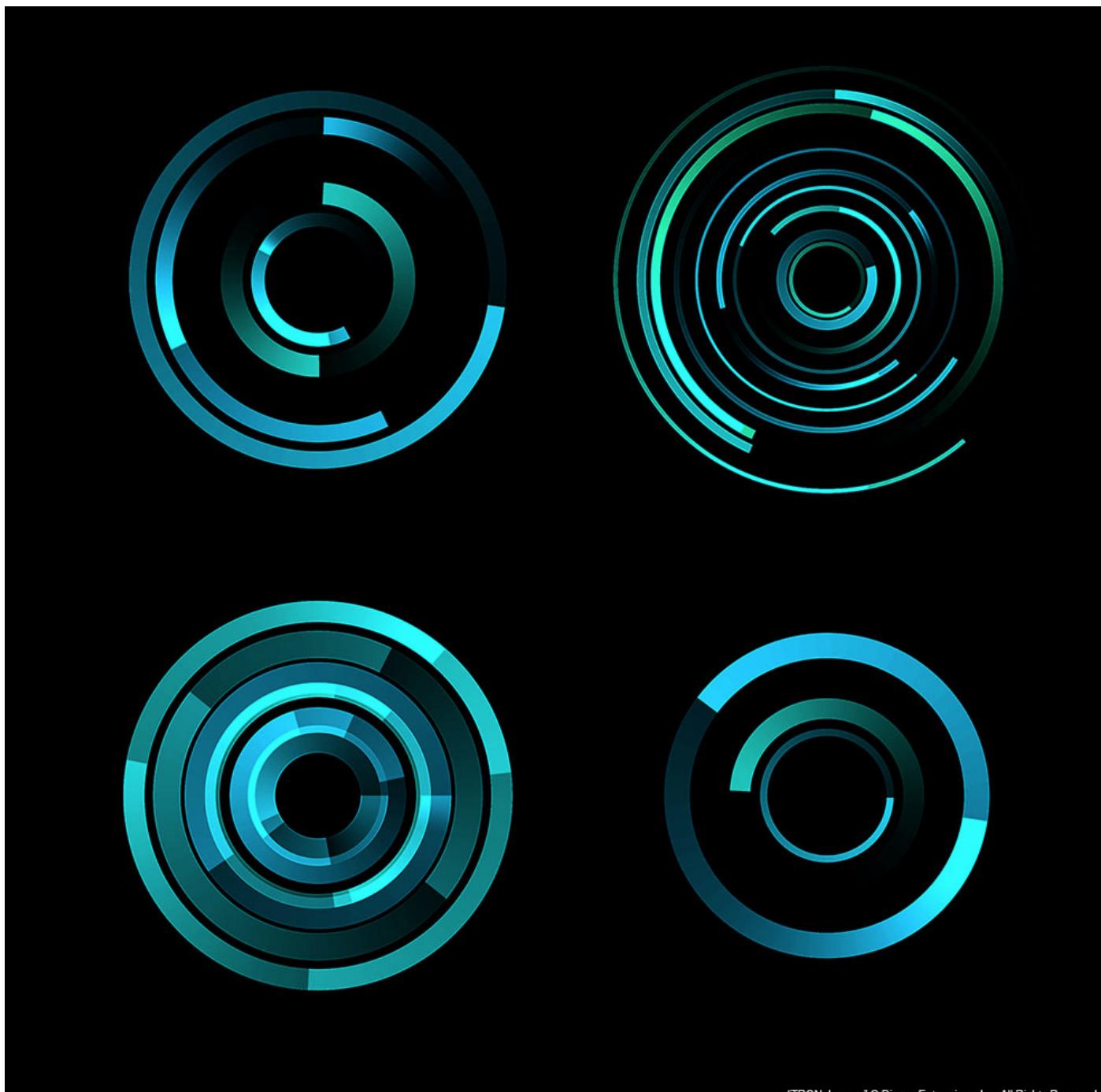


"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.

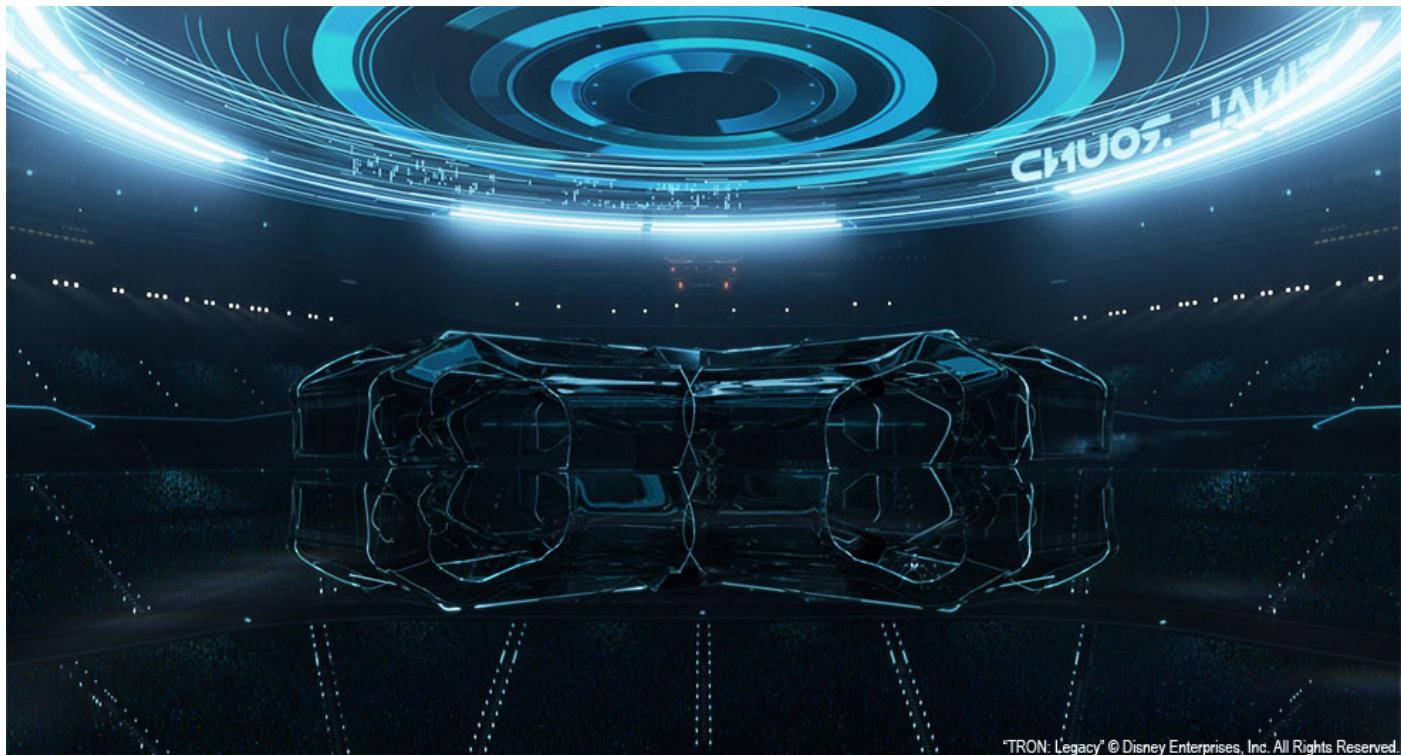


"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.

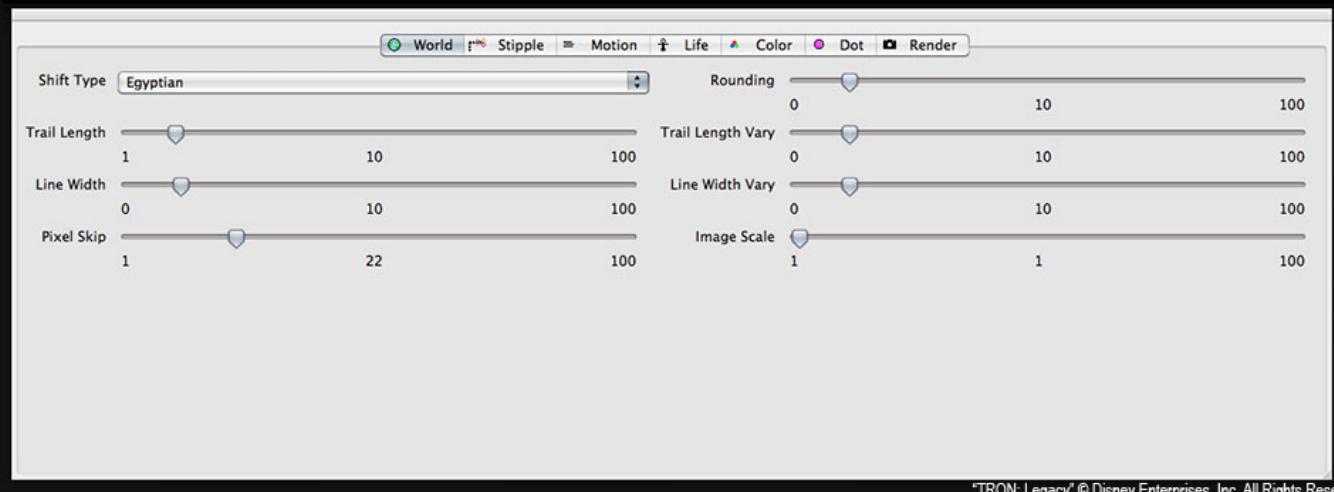
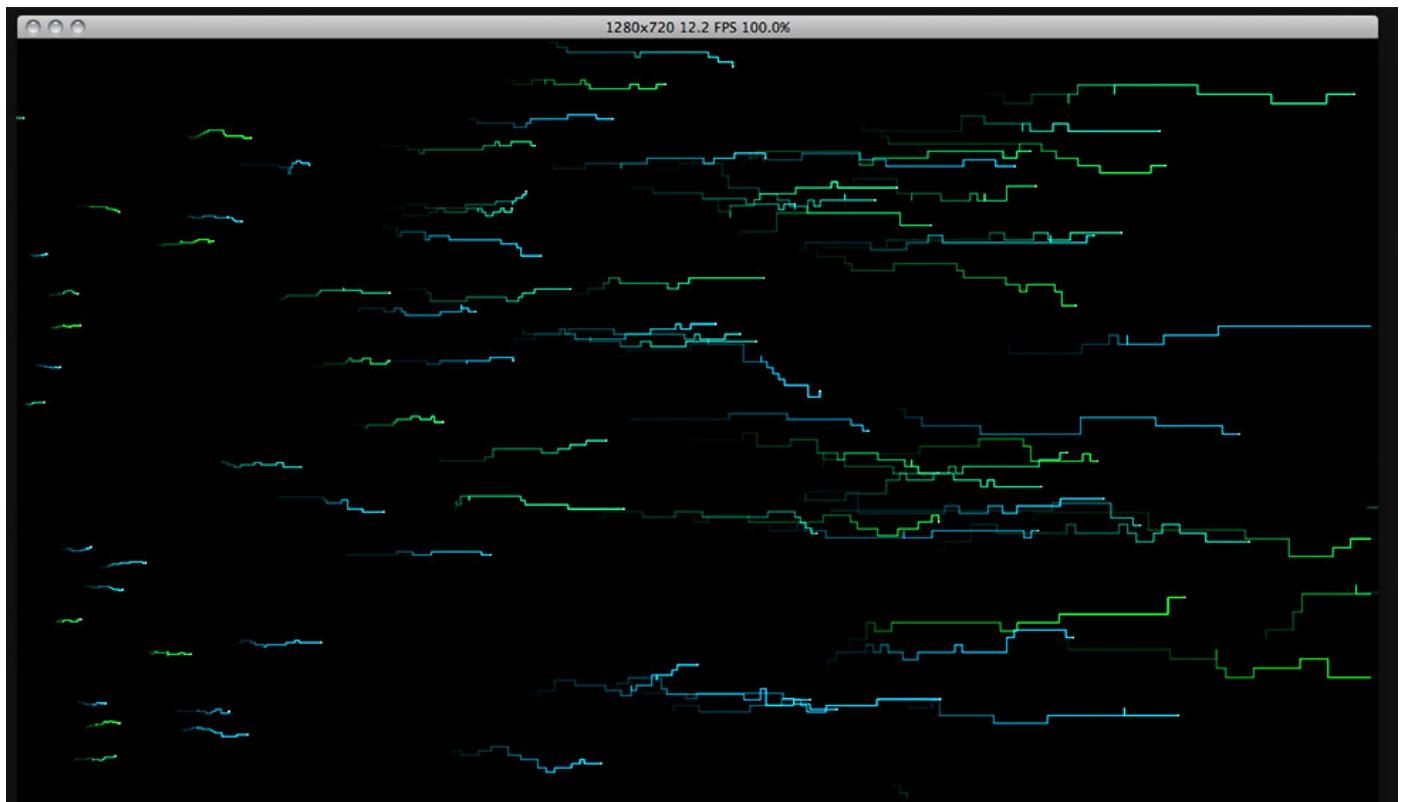
The scoreboard was the first element I worked on. I created a line-generator that produced bursts of lines which turned at adjustable angles. The line generator had "radial mode" which arranged the geometry in concentric circle form. This line generator was used to generate generic elements and layers of style in different things, and is a GMUNK favorite. At this point, I found myself moving to multisampled FBOs because the non-antialiased polygons were just too ugly to work with, and we needed to make film-resolution renders. In fact, this is the highest res I've ever seen my apps render.



"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.



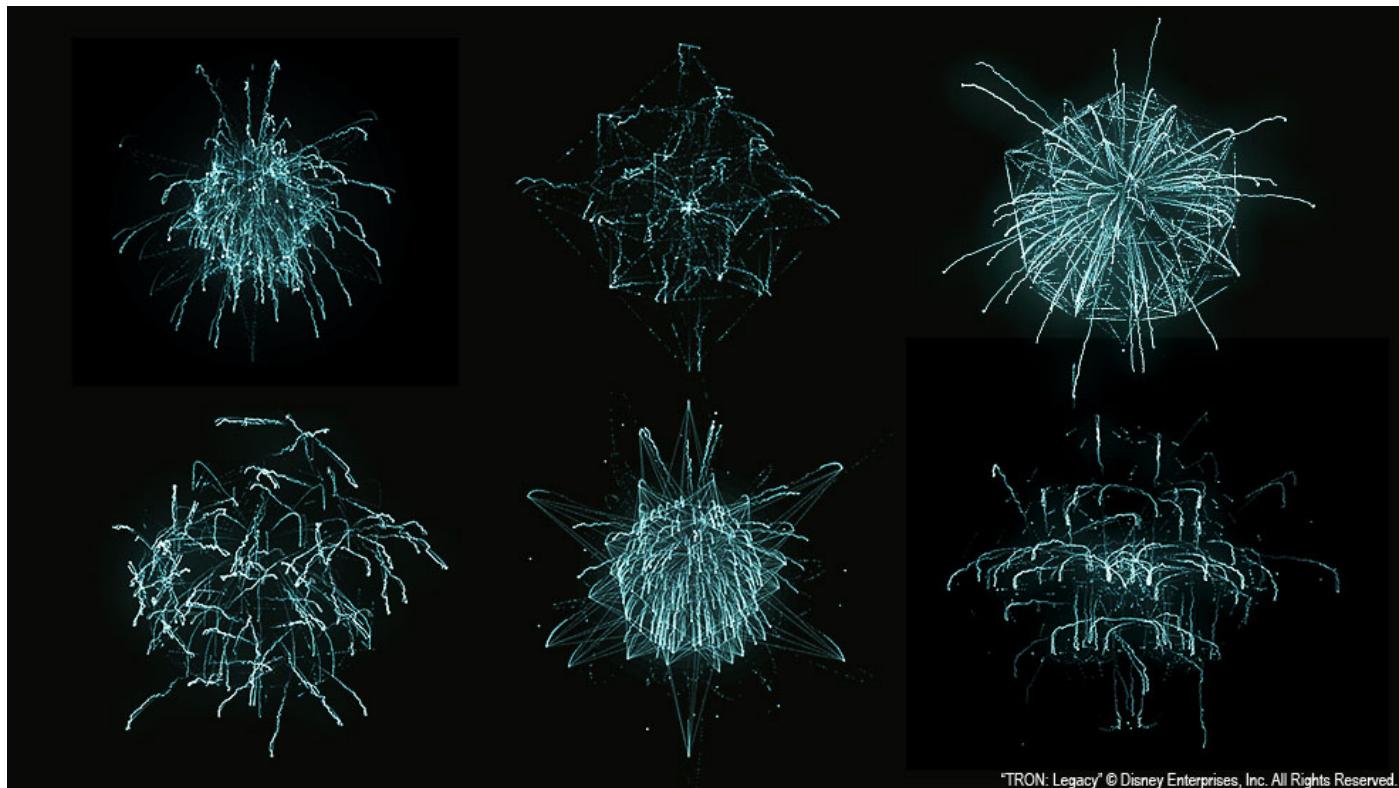
"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.



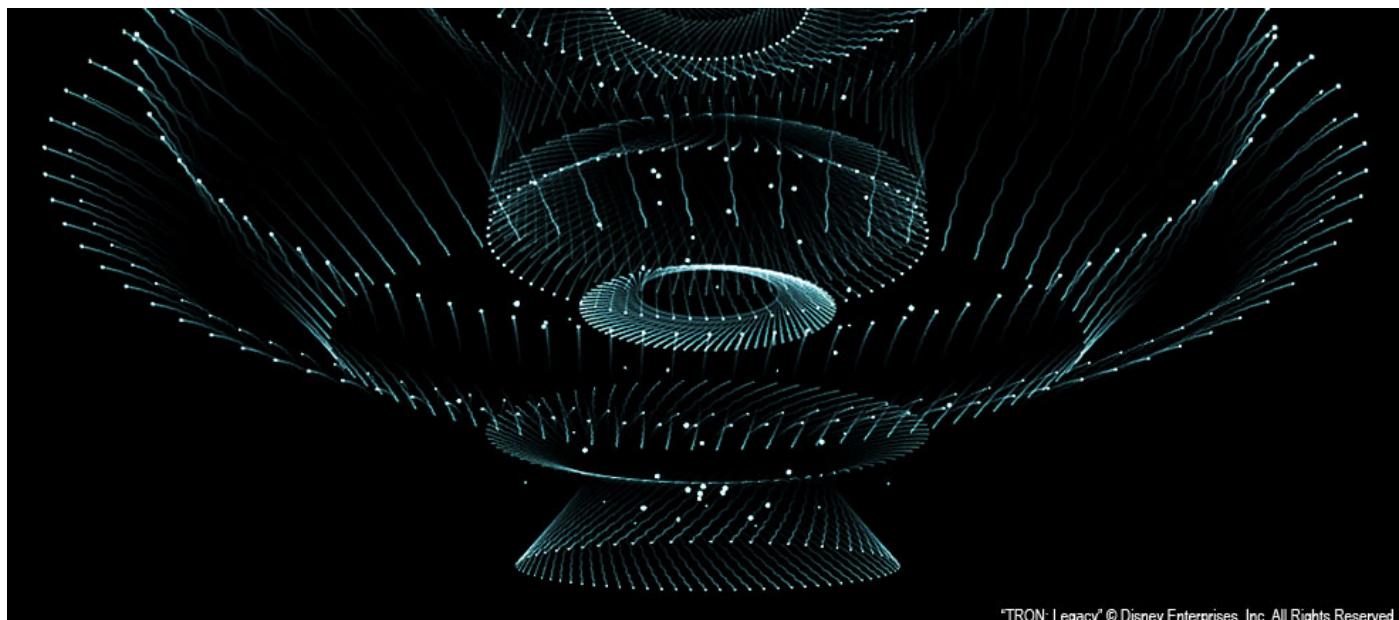
"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.

Fireworks, mmmm. I started with a regular physics simulation where a particle has an upward force applied at birth, sending it upward while gravity pulls it back down resulting in a parabola. I then added particle-children, followed by various artistic styles, including what our team has called "egyptian" across several jobs -- which is a side-stepping behavior. We were trying to create fireworks that looked enough like real fireworks but had interesting techno-aesthetic. As a homage to the original Tron character Bit, we used icosahedrons, dodecahedrons, and similar. I was disappointed that Bit isn't in this one. After doing this simulation, I've grown more aware of how often fireworks are used in movies.

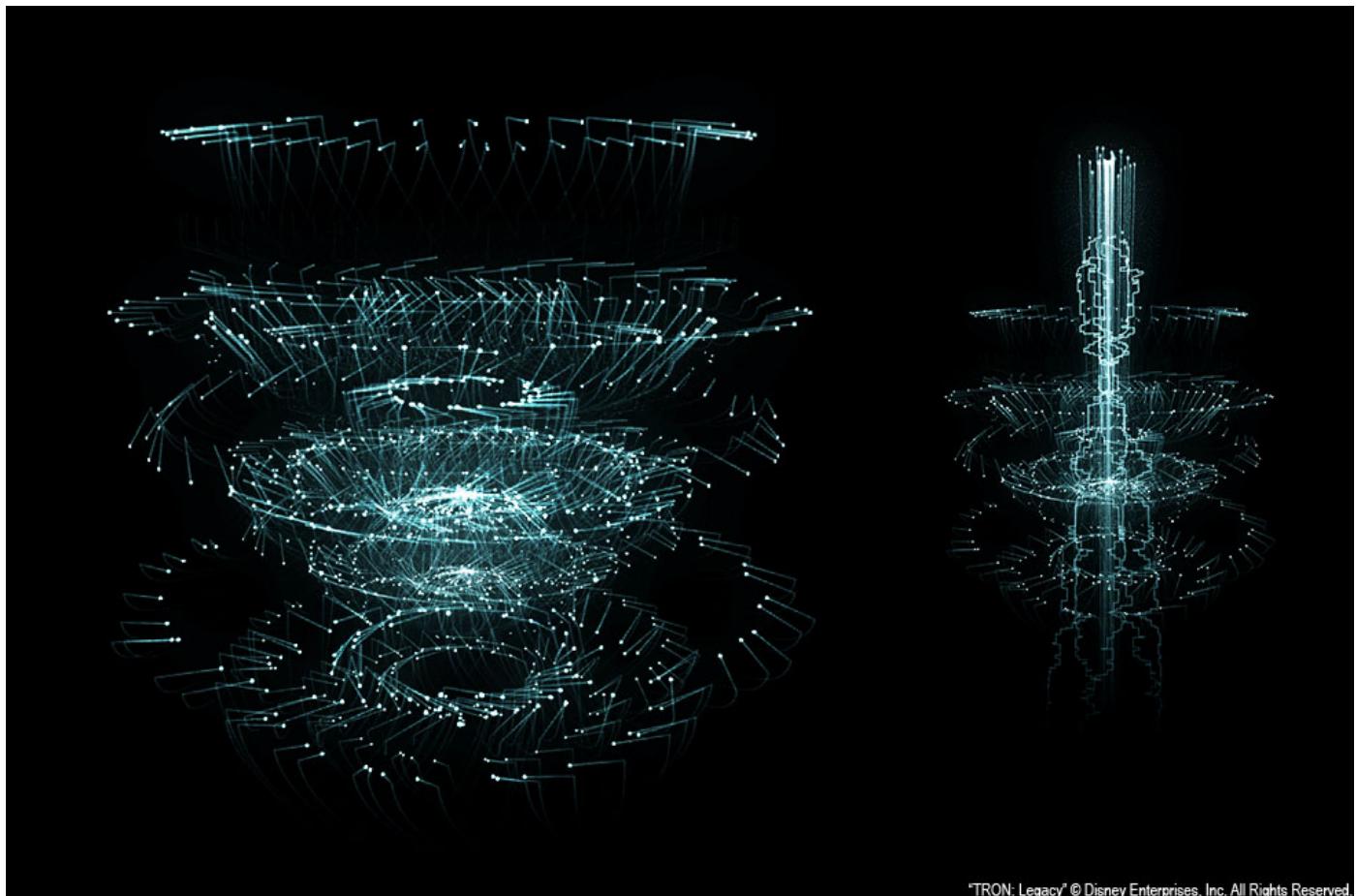




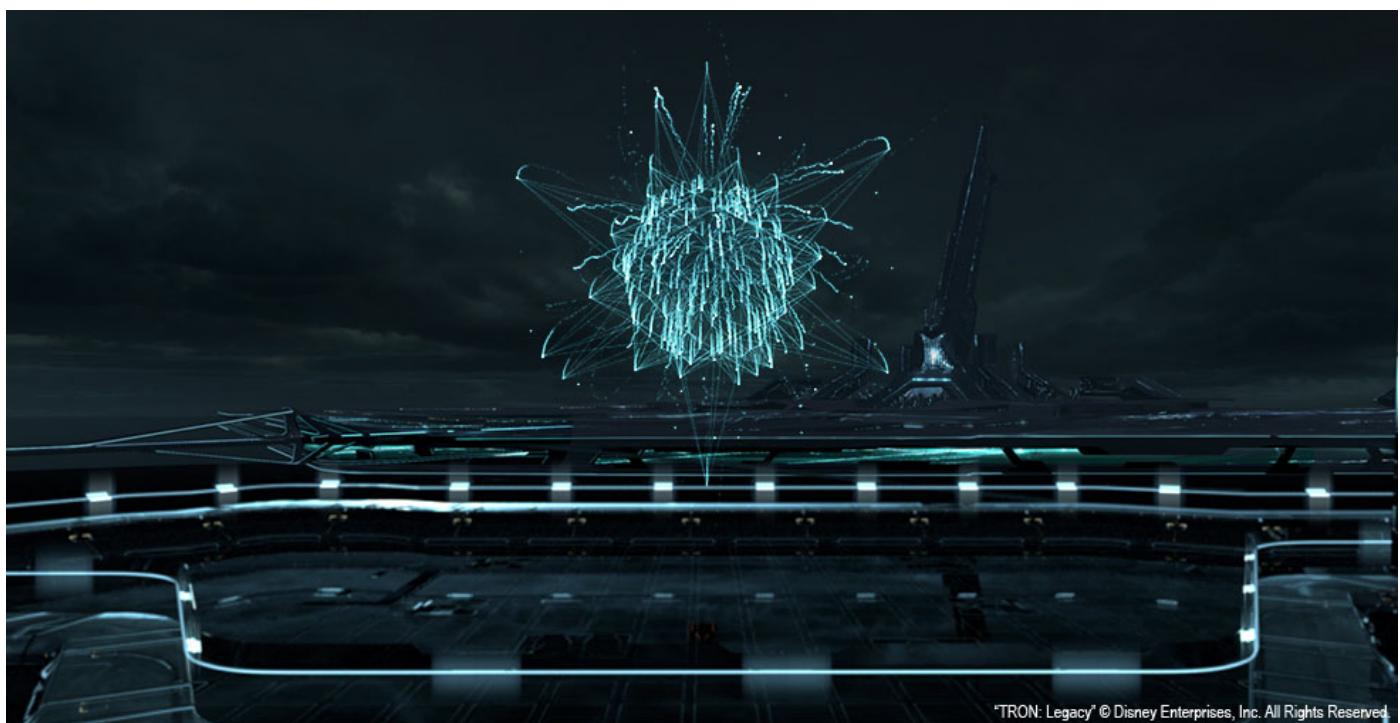
"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.



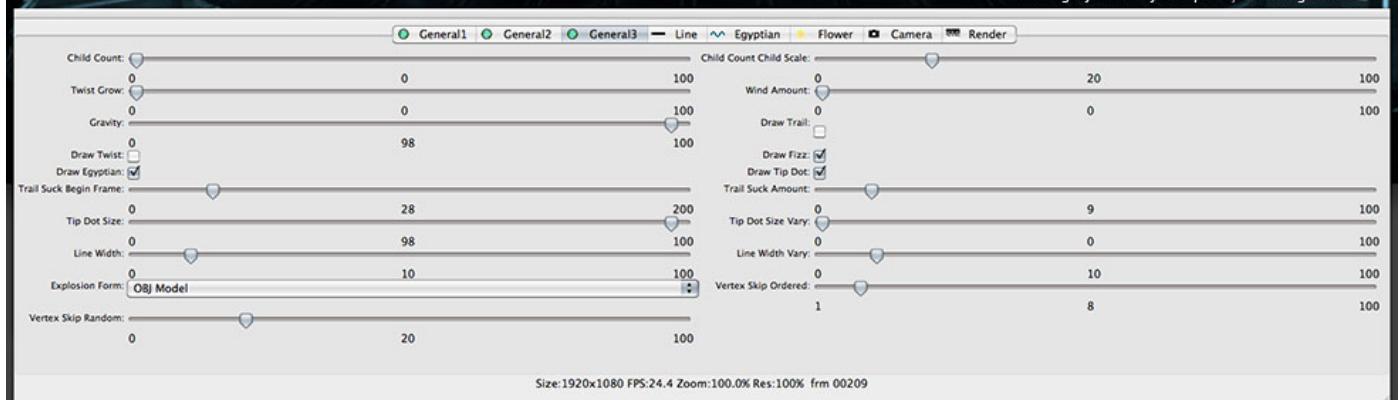
"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.



'TRON: Legacy' © Disney Enterprises, Inc. All Rights Reserved.



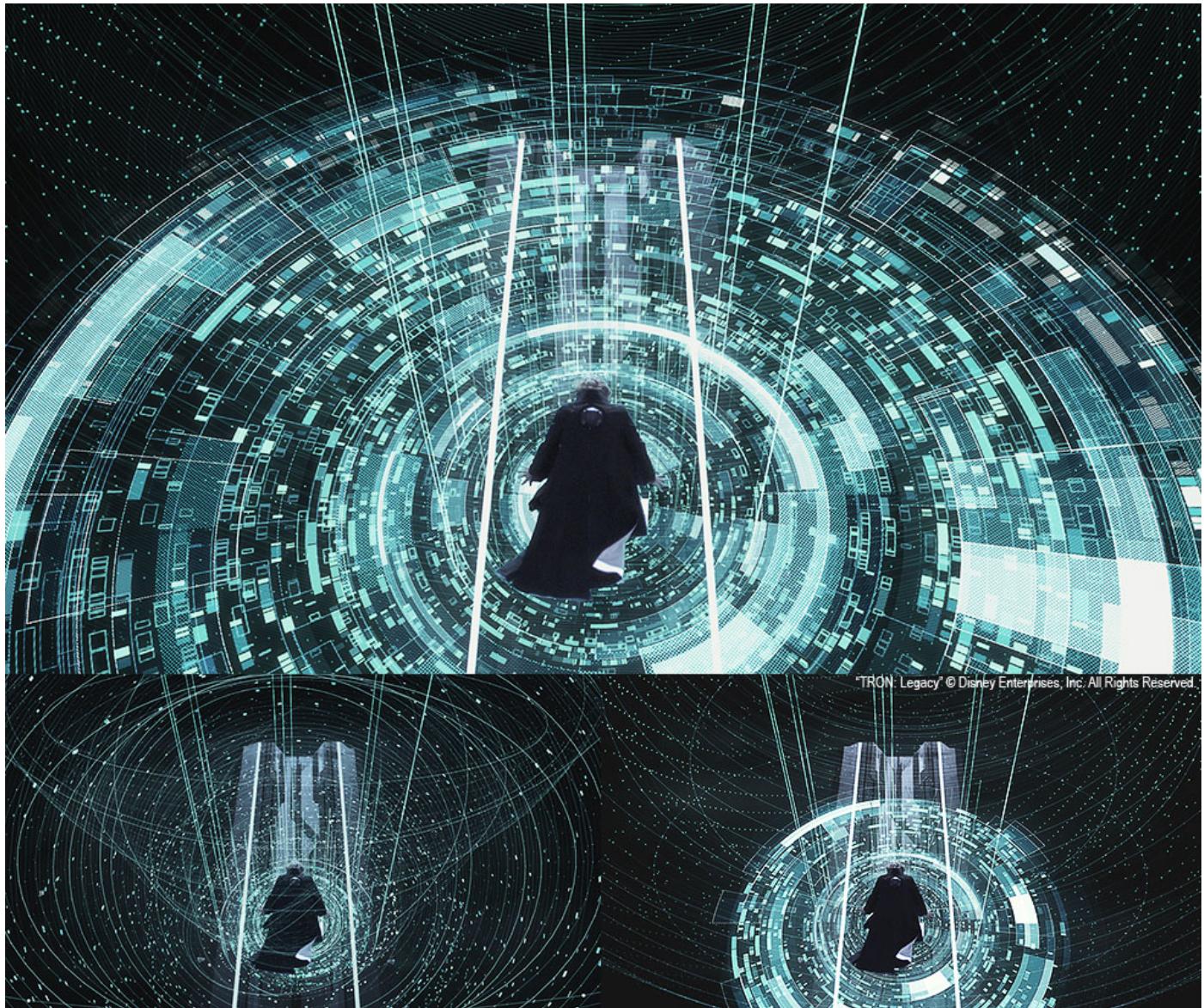
'TRON: Legacy' © Disney Enterprises, Inc. All Rights Reserved.

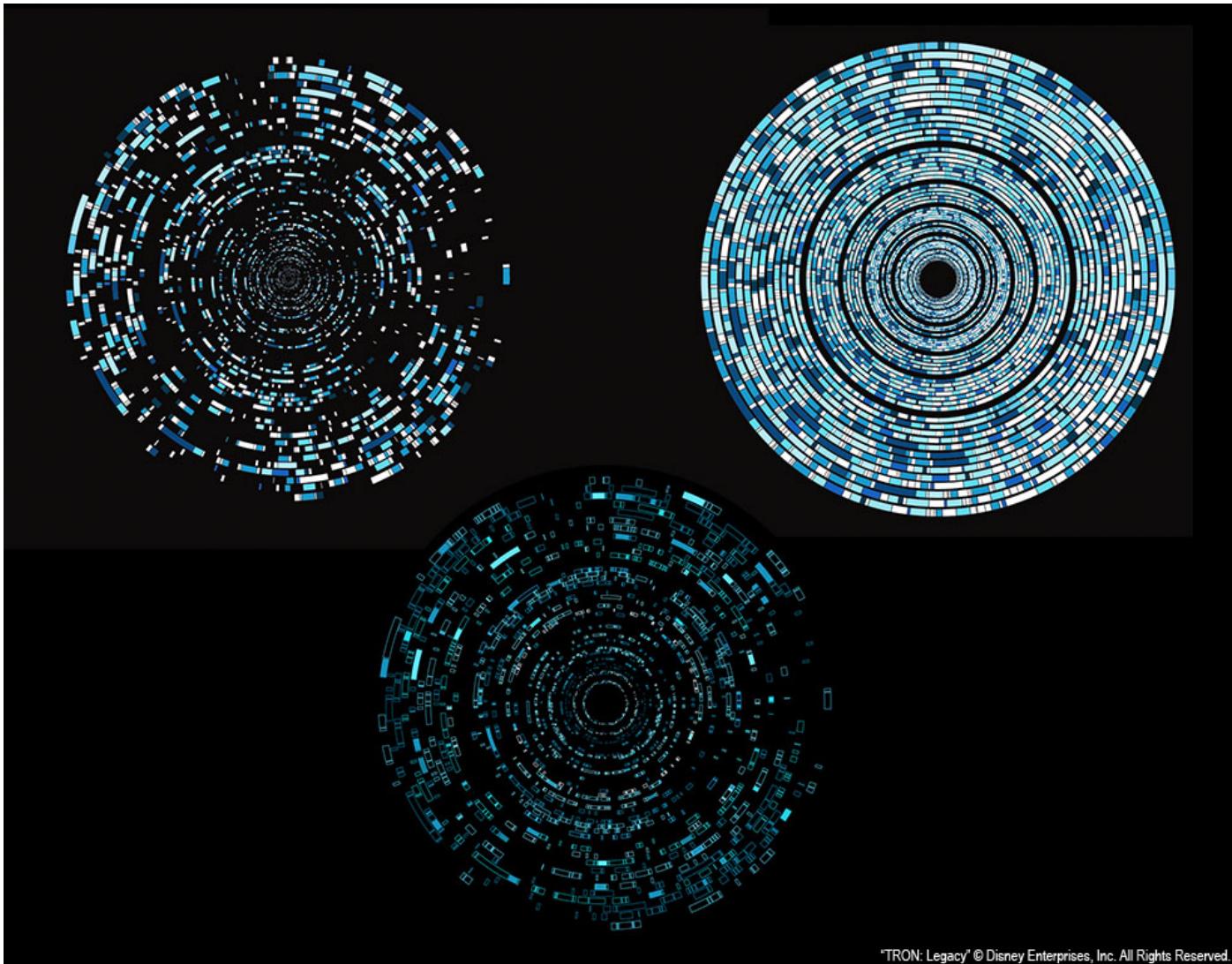




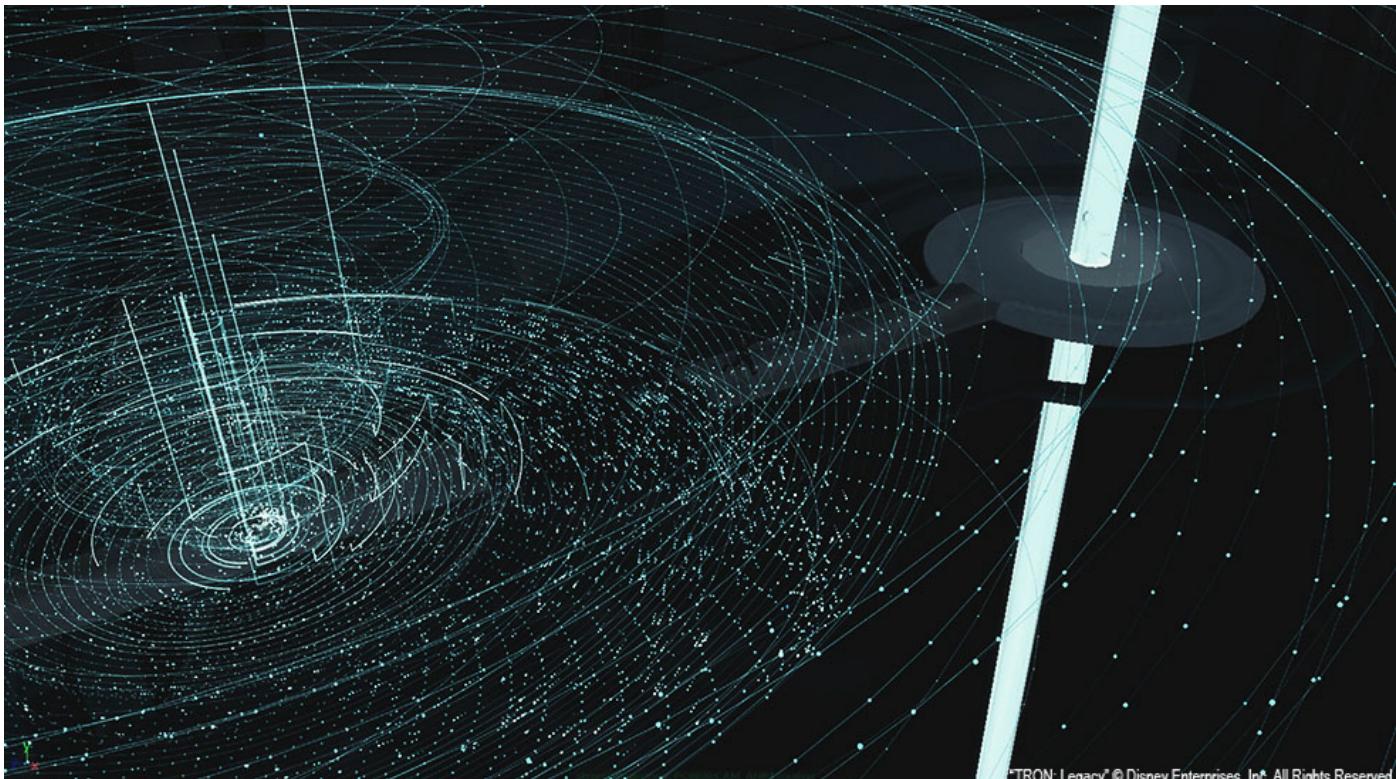
"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.

For the portal climax, the TronLines app was used, but also apps like "Twist" from our team's previous jobs. Once the look was mocked up by gmunk, a houdini artist recreated the rig for deeper control.

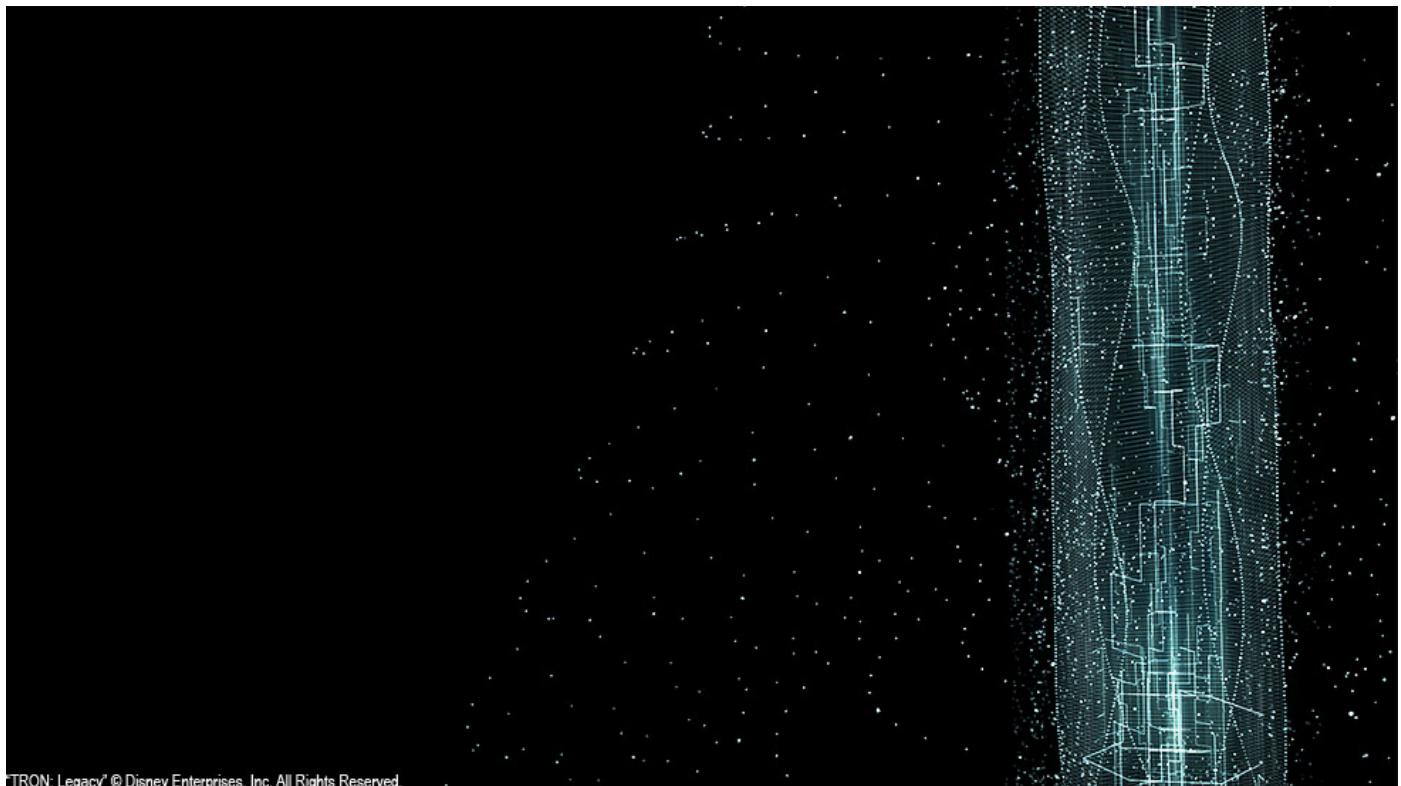




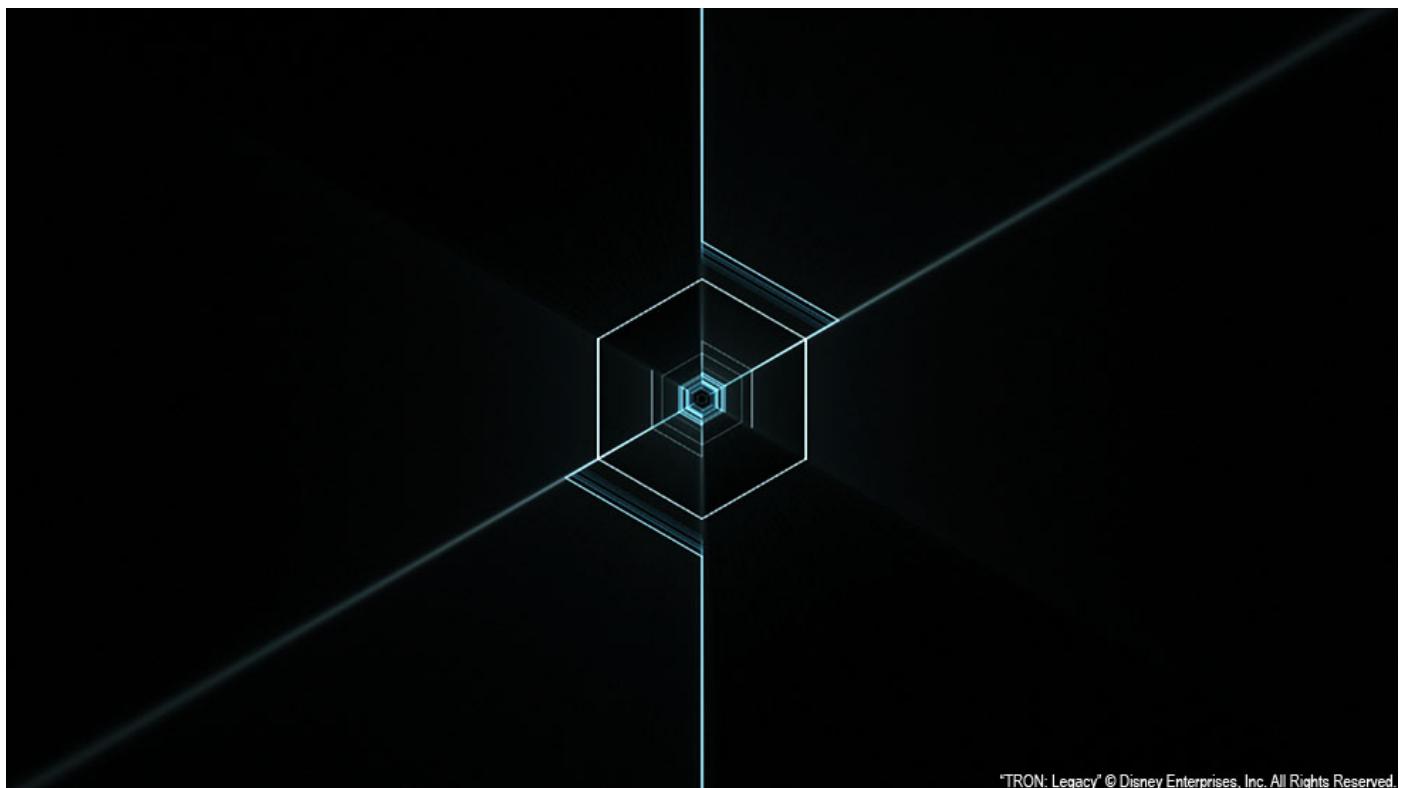
"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.



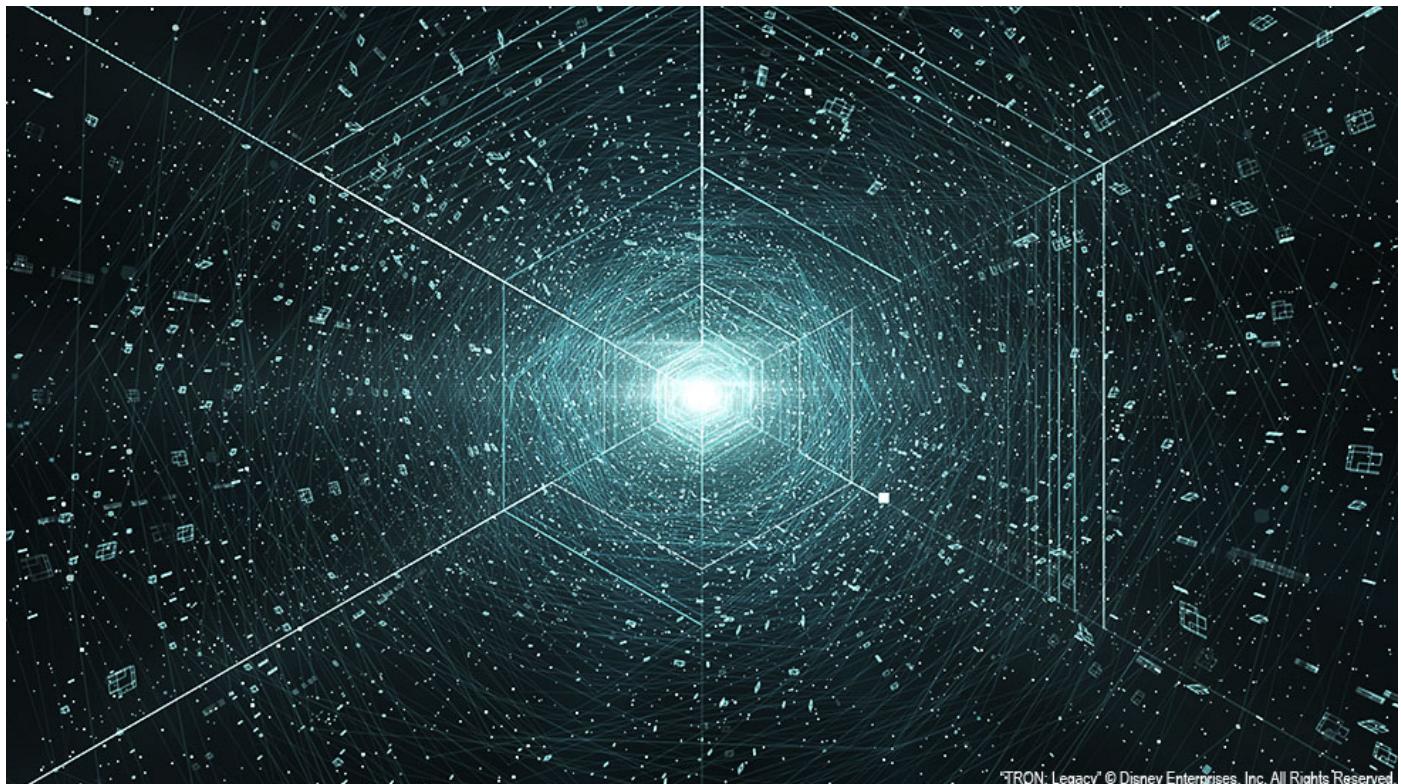
"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.



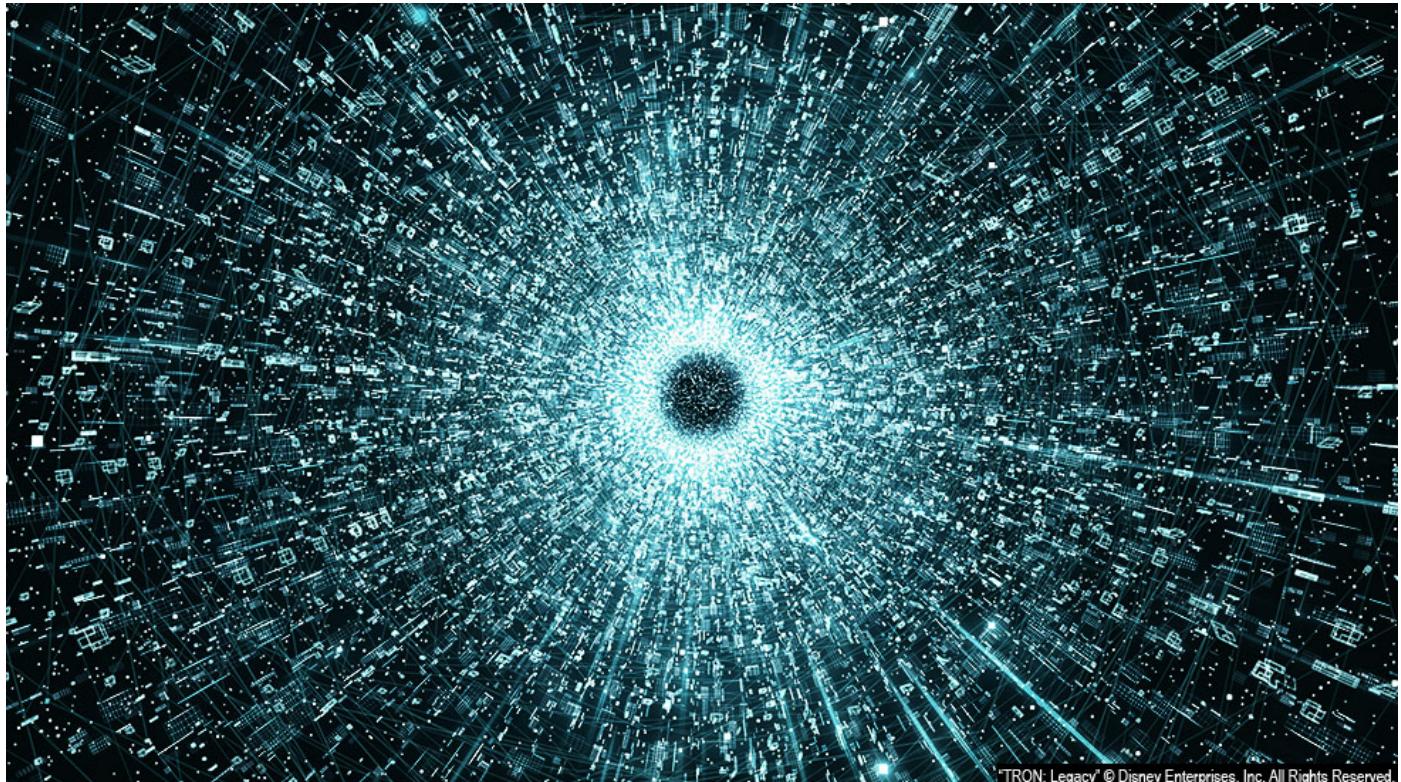
"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.



"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.



"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.

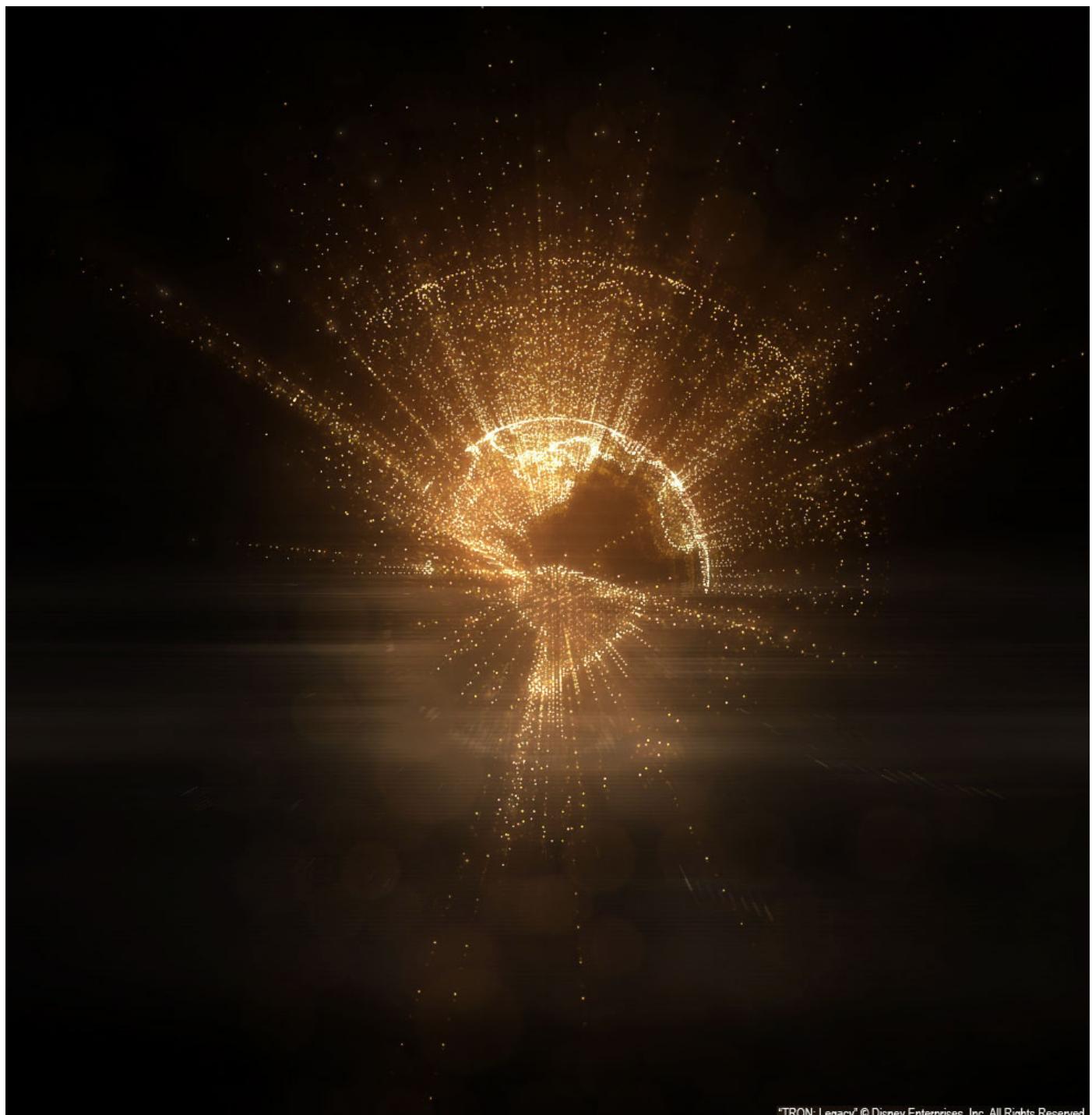


"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.

I wrote a particle renderer that could make the head holograms slurp in and out of the data discs. Special thanks to Keith Pasko for CLUing me in about using exponential functions to create a sliding-gooey sort of delay.



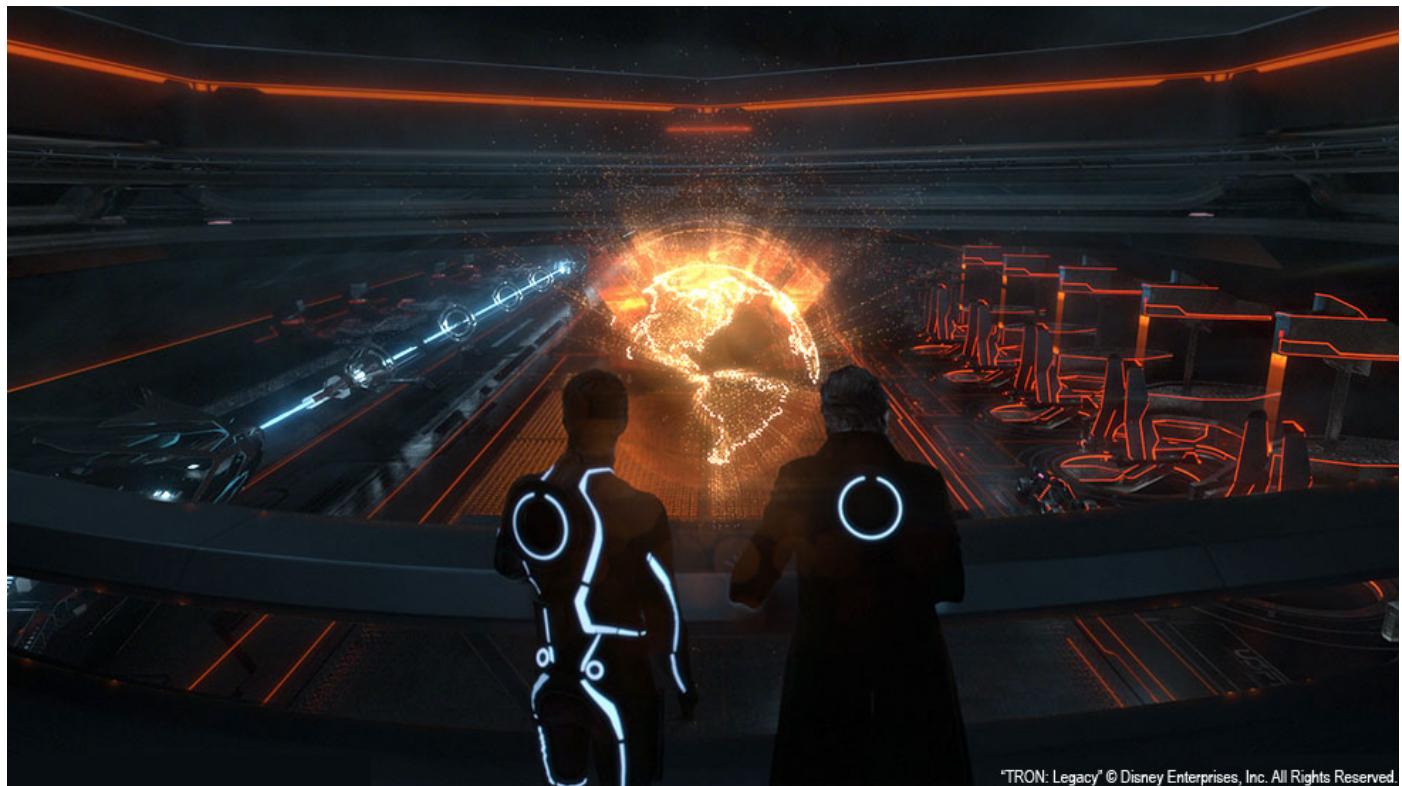
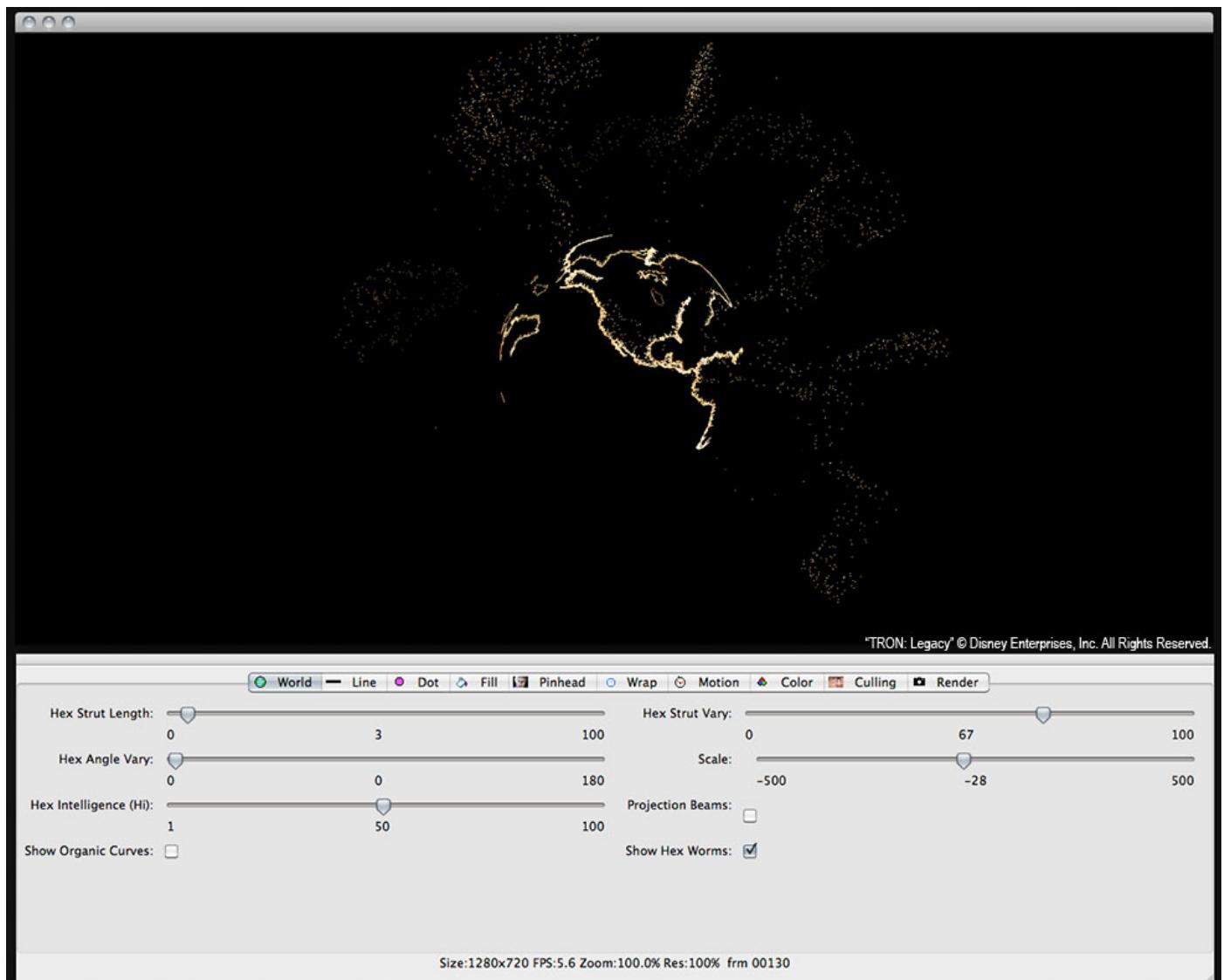
"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.



"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.



"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.



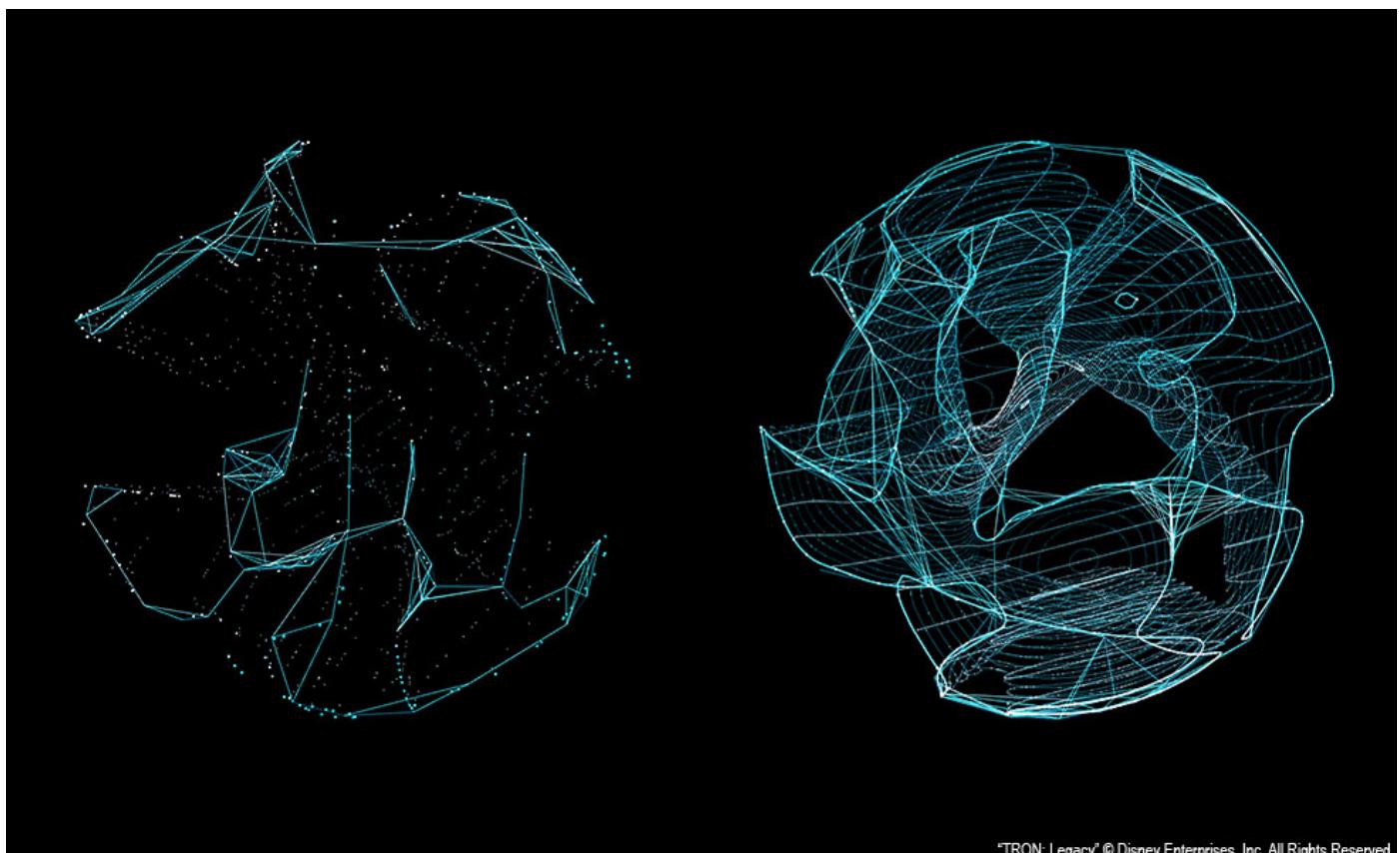
"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.

When fixing Quorra, there was an element in the DNA interface called the Quorra

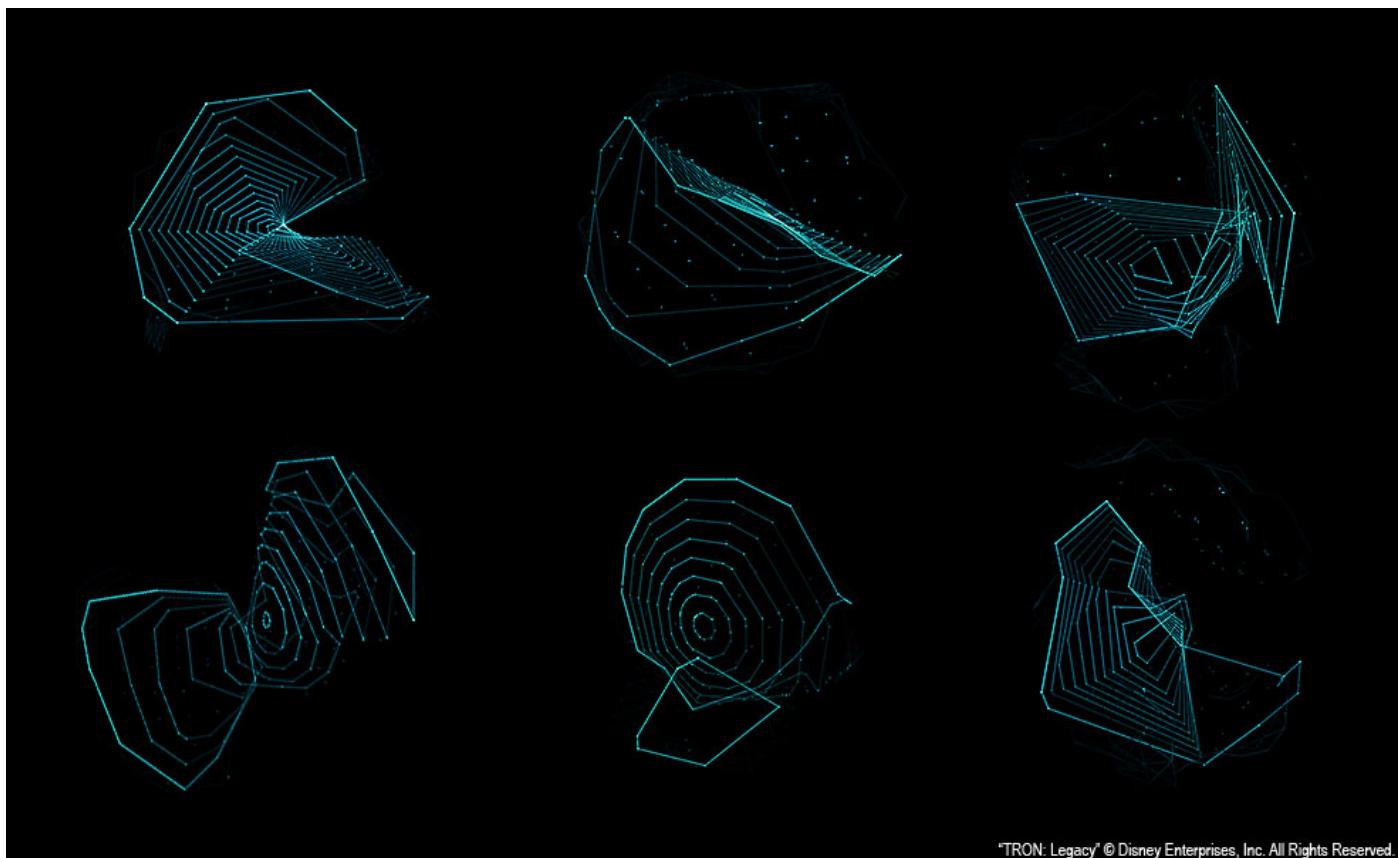
Heart which looked like a lava lamp. I generated an isosurface from a perlin-noise volume, using the marching cubes function found in the Geometric Tools WildMagic API, a truly wonderful lib for coding biigital jazz, among other jazzes. The isosurface was then drawn along different axes, including concentric spheres. The app was mesmerizing to stare at.



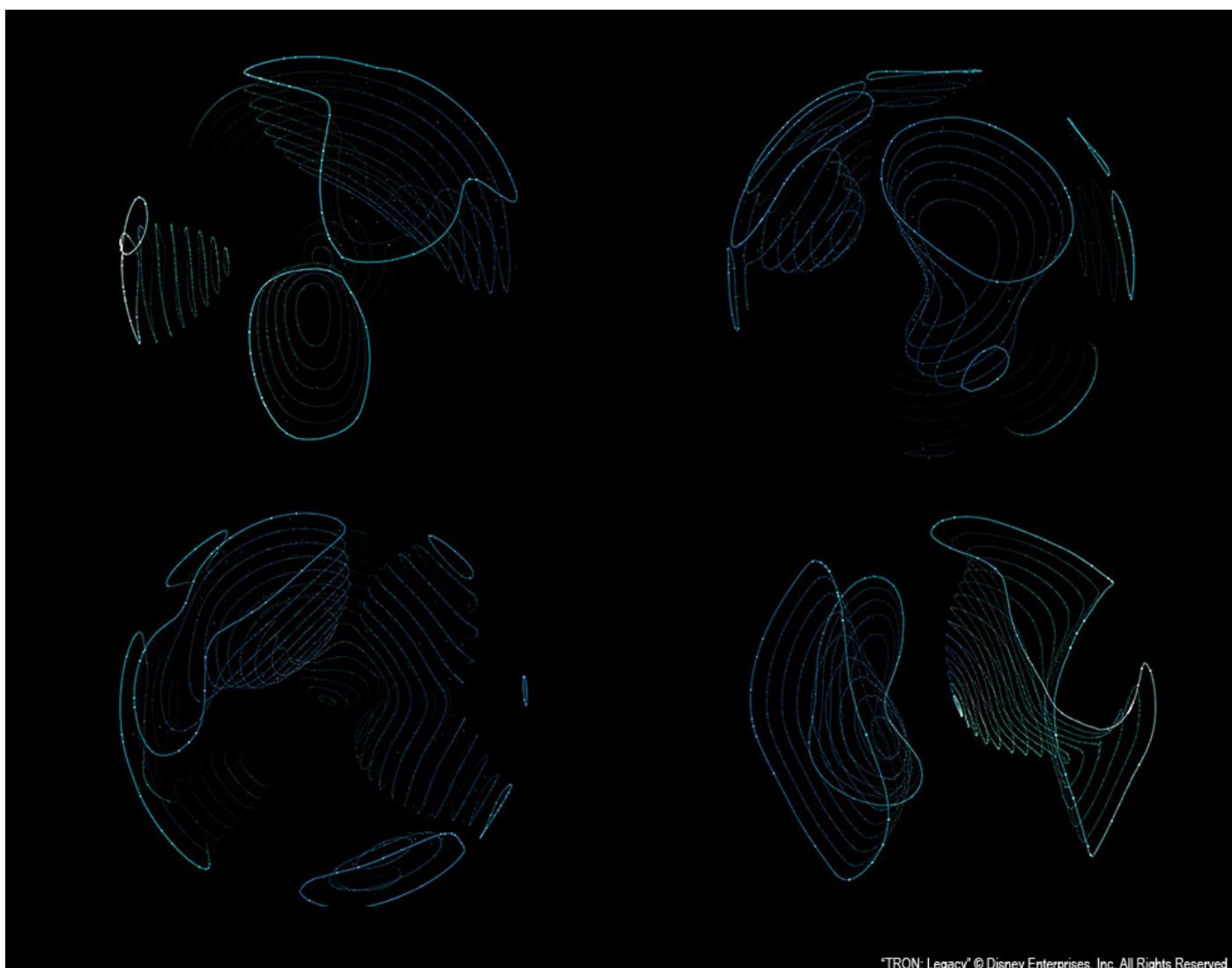
"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.



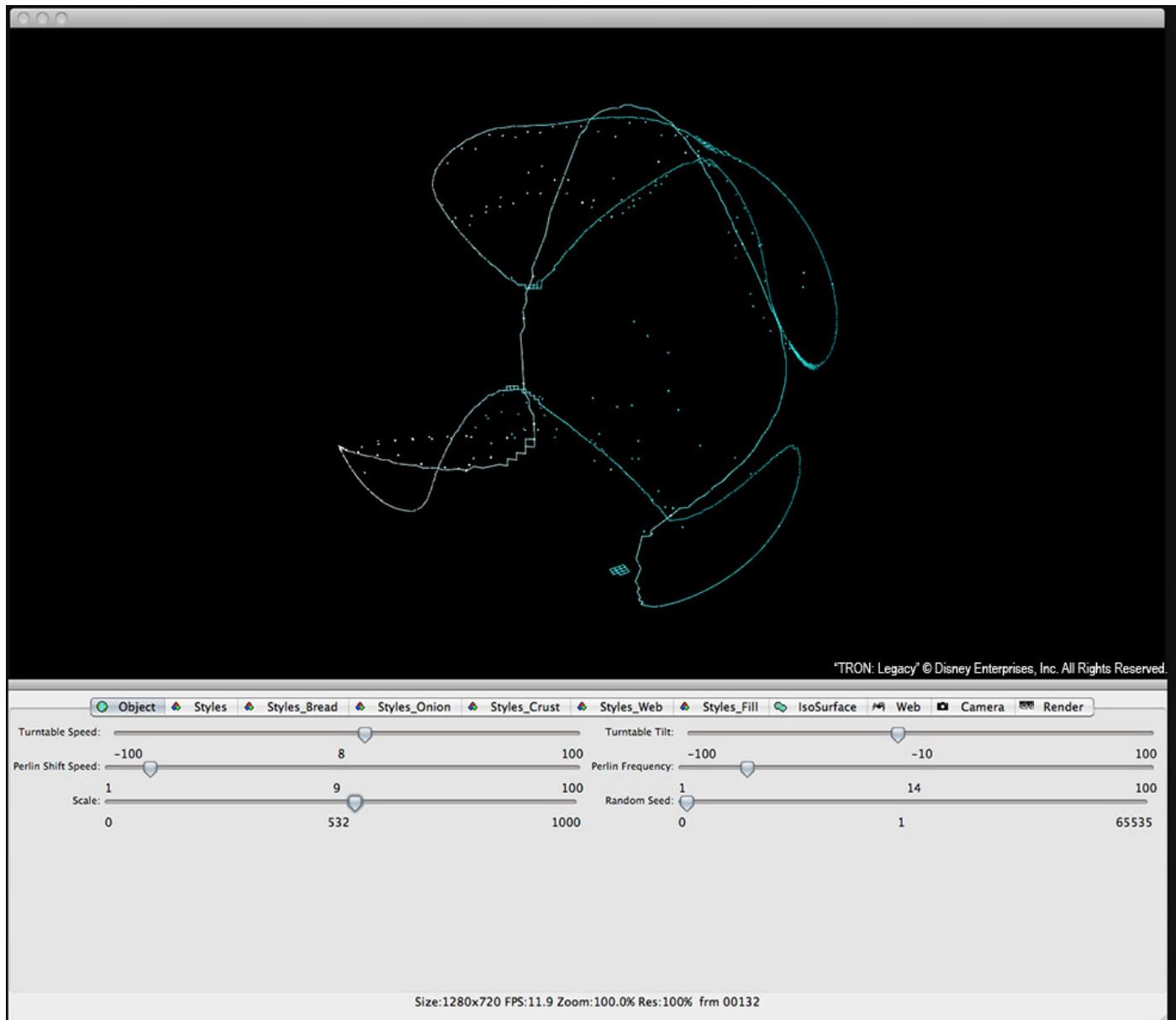
"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.



"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.



"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.



After this project, I was fed up enough with wxWidgets and Carbon that I was ready to author my own OpenGL based UI. The most important thing I could use is a floating-point slider. I also got irritated with the way the Carbon sliders would not slide all the way to the minimum and maximum values. It totally messed with my zen thing. Also, after a job like this, it's clear that a member of the Processing community working within a CG community is greatly restricted by the differences of realtime graphics rendering engines, and that probably messes with an art director's zen thing.

"TRON: Legacy" © Disney Enterprises, Inc. All Rights Reserved.

tags: cplusplus digitaldomain maya movie openframeworks opengl tron wxwidgets