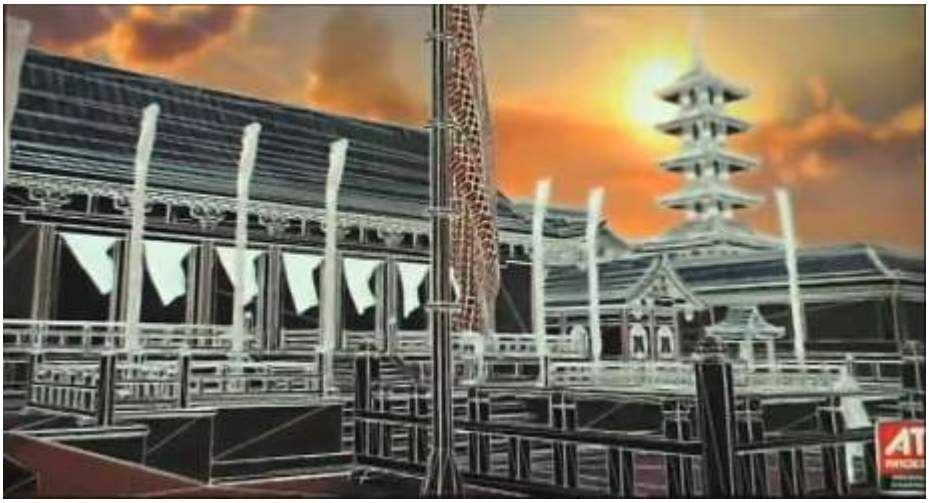
AMD DEVELOPER INSIDE TRACK

GDC 2010 DEMOS: CLOTH, DESTRUCTION, TESSELATION AND MORE



This episode of the AMD Developer Inside Track Video Series brings the AMD GDC demos to you. Software Engineer, Saif Ali, walks through three examples of how OpenCL can be used to create realistic cloth, destruction and fluid particle simulation using the updated Pixelux and Bullet Physics offerings. And our partners from Studio|GPU™ show us how tessellation can be used to change the texture of a character in real time!

TRANSCRIPTS

Saif Ali, Software Engineer, Advanced Technology Group in the Office of the CTO

Sharon: Hi! We're here with another episode of the Inside Developer video Track series. I'm here with Saif at GDC covering what we are doing at the AMD booth for the open physics ecosystem. So, Saif, can you introduce yourself and tell us a little bit about the demo's you'll be showing us?

Saif: My name is Saif Ali. I'm a software engineer in the Advanced Technology Initiatives in the office of the CTO. And today we are showing off ATI Stream Technology that simulates various kind of natural phenomenon (like) destruction and cloth. We're here at Game Developer's Conference 2010. We have three demos that we worked very hard on and hopefully you will like them.

So I'm going to start by showing the cloth demo. That simulates cloth in real time. It's written using DX11, working exclusively on ATI Radeon cards, the 5870.

So we have the samurai in the middle. The samurai is not cloth it is an animated character, but all the banners and curtains, as you will see them soon, are all simulating cloth in real time.

So you can see that all these curtains simulate in real time, written as I said, in DX11. I think it's quite pretty.

Sharon: That's amazing. Can we see the wireframe mode?

Saif: That shows off the triangle count. The reason those curtains look so pretty is that we have very very high polygon count. And the detail is amazing.

Saif: The City Demo demonstrates, along with Pixelux DMM, Bullet and Trinity Engine, destruction that has such desirable characteristics in games. You can blow up stuff in real time. So, this is still a work in progress but we can make this look really awesome. This is what we come up with for GDC. So, we have the Mega robot running around this little town. He can walk through stuff and destroy it. We have DMM objects embedded in this city. All the DMM objects are the ones that are capable of being destroyed. So for example this construction structure right here, you can walk right through it. And I particularly love showing this one because you can systematically walk through all the supporting turrets and it keels over on its side. It's very satisfying to watch. Hopefully I got it by that time, aha there it goes.

He thinks about what he'll do next, maybe he'll bump into this jello building which I think is kind of fun and it kind of swings around. Oh it keels over.

Saif: The next one isa a fluid simulation demo and this simulates fluid as particles. This is written using OpenCL and the rendering is done using DX10.1. This, we built from the ground up, from scratch, from absolutely nothing. So, also a work in progress. Obviously we can make it look way more awesome but this does 64,000 particles in real time on the ATI Radeon 5870. And the rendering is pretty fast. As you can see it goes in real time over 30 frames per second.

The main technique behind this is SPH that stands for Smooth Particle Hydrodynamics

Sharon: Are any of these demos available online or is this …

Saif: Right now this is GDC only. I don't want to say anything about what the plans are, but with most of our demos we aim to get them online as soon as we can.

Christopher Jess, studio|GPU™, Technical Director

Sharon: Ok, we're here with Chris from Studio GPU. Chris, why don't you introduce yourself and give us a walkthrough of your demo?

Chris: Alright, so I'm here with Studio GPPU. My name is Chris. I'm here to demo our software, Mach Studio Pro. Mach Studio Pro is a real time rendering and compositing engine that runs live off the

graphics card. In this case we are demonstrating with the ATI Fire Pro V 8750 which is a 2 Gig top of the line workstation graphics card.

So this is Mach Studio Pro and what we are showing today is a small scene from a 75 minute movie that was brought out of Maya. The scene itself was modeled and animated in Maya, UV mapped in Maya, and we bring it into our software and do all the final lighting and texture work. We hit the magic button and we have a final rendering. Because we are running off of the graphics card, everything is being rendered live on the fly. So, we're not writing out any temporary files, we're not writing out any cache. Everything is coming straight off the graphics card. Only in the end when we write out our final files do we send anything out to the harddrive.

So what we are actually looking at now is a composite image of multiple passes. See down here? All these passes are running live by the graphics card and composited together. So what this allows me to do is, I'm actually seeing a depth of field diffused specular shadows, transparancies, reflections refractions, ambient inclusion and everything is being rendered individually and being composited together by our software.

The other thing we are showing here is hardware tessellation and hardware mapping which is exclusive to ATI cards currently. If I switch back here I'm going to focus on one of objects here really quick and switch my camera here so that we can focus on what is going on. So, I'm just going to come into this character here really quick and what we are looking here is a wire frame. This is the base mesh that came out of Maya. We have a volumetric light in the background. That's what all those white rays are coming from, but we have a built in cadmiral clark sub D system, we turn that on, that's where the basic sub division is coming from.

What the ATI hardware allows us to do is turn on the hardware tessellation on this and I can just turn up the tessellation amount on the object. So what is actually happening now is that the wire frame is taking on the color of the texture that is on the wire frame at that point. So the wireframe, the mesh, is getting so dense now that the details show in the texture map despite the fact that we are still in wire frame mode. So we are getting a sub pixel tessellation which means that we are getting single triangles that are getting smaller than a single pixel on screen. So we are still in wire frame mode and I'm probably running at twenty to twenty-five million polygons just on this one character currently and this is all being generated live by the graphics card.

So this is a little overkill. I'm going to dial this down; I'm probably running at seven to eight million polygons now. What this allows me to do is, on top of normal maps we support anyway, I can bring in a displacement map which is a standard black and white displacement map that we created in Photoshop. And if we bring this in, I'm going to switch back out of wire frame mode, what this allows me to do now is create an actual physical geometry displacement based on our black and white map off the graphics card. So, this alone would take another couple of minutes or more than that on any kind of software package. So, this is just another pass for the graphics card. Because it's coming out of the graphics card, I get a full rendering and lighting pass off of the displacement map. So you can see with a full lighting pass, I can still adjust this. I get a full ambient inclusion pass out of my displacement mapping here so I can turn this up and down too with the ambient inclusion.

So, I'm just going to switch back to this real quick and you'll see here that I get a really nice displacement map off the whole character's head. It's definitely a 3D displacement. So, if I switch back here to my full view and bring back the other objects really quick, we get all these passes with the hardware tessellation, displacement mapping and depth of field straight off the graphics card in a split second of any kind of software rendering that is out there. We've done some test renderings and come to the conclusion that in the higher end settings we are running up to 500 times faster than current standard software packages. So currently, with this information, in here in the viewport we are probably running anywhere from two to three frames per second. With the final output, with all the settings turned up all the way to production quality, we'd probably be running, I'm guessing no more than five seconds per frame for full HD and final quality. So that's kind of what we are looking at. And this is Mach Studio Pro.

Sharon: Thanks so much Chris that was amazing, really amazing.

Chris: Thank you, I'm glad you liked it.

Visit developer.amd.com for more information.