PHYSICALLY SIMULATED CLOTHING BY CCP (EVE Online) USING NVIDIA APEX

Vigfus Omarsson CCP Games
Snorri Sturluson CCP Games
Monier Maher NVIDIA

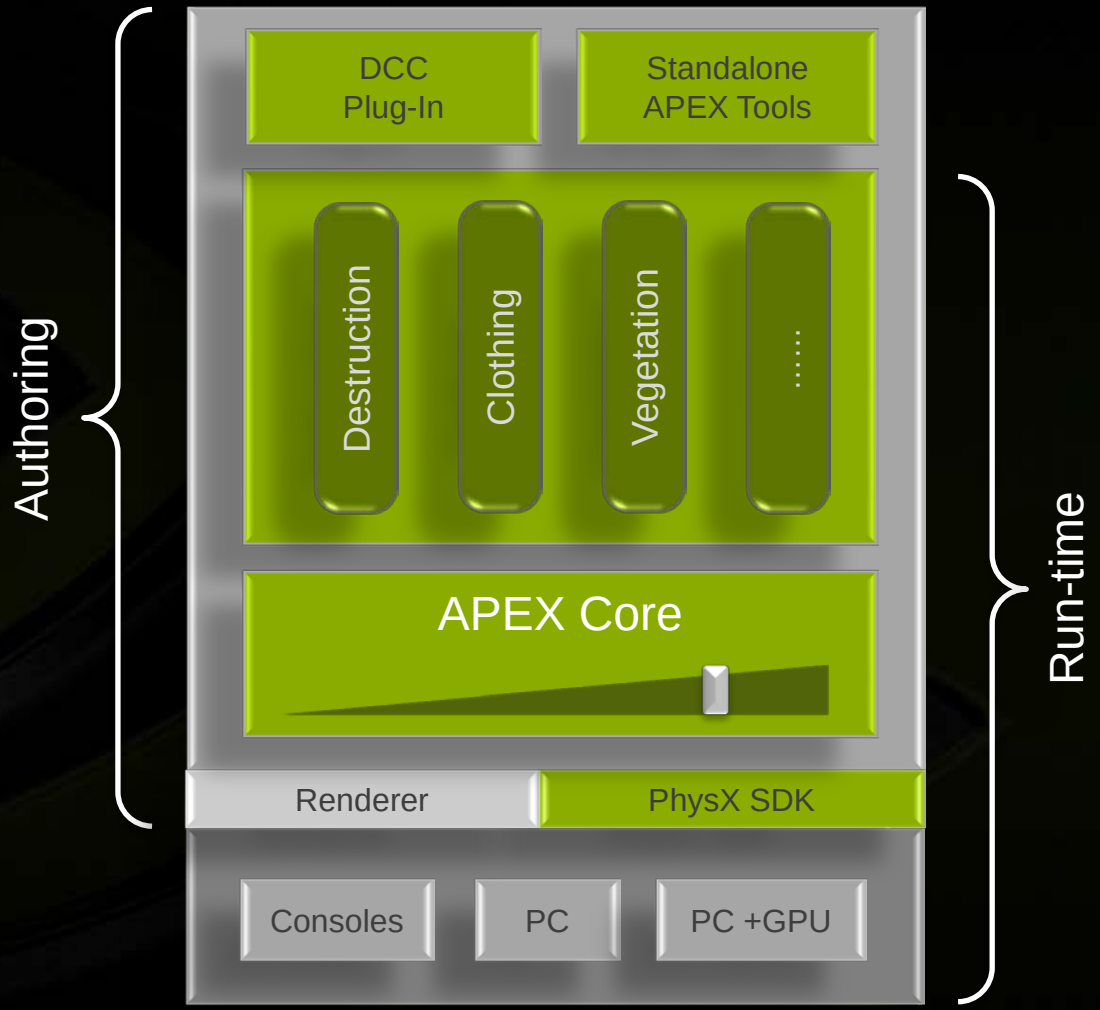
Outline

- APEX Overview
- APEX Clothing used by CCP
  - Clothing considerations
  - Maya DCC plug-in overview
  - APEX Integration
  - Demos
- APEX Modules
  - APEX Destruction
  - APEX Particles
  - APEX Vegetation
  - APEX Turbulence

What is APEX?

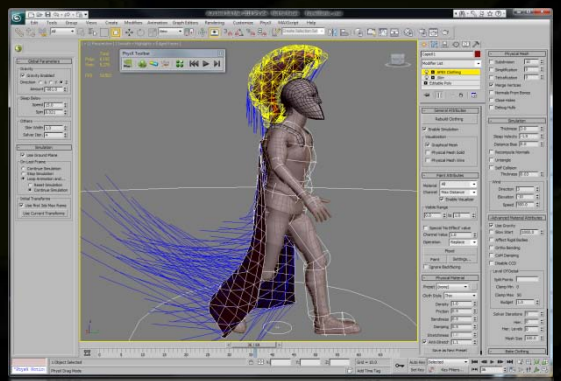
- APEX is a “Scalable Dynamics Framework”
  - *Scalable*: Content adapts to different hardware capabilities
  - *Dynamics*: The way things move and interact
  - *Framework*: A structured environment
- APEX consists of two major components:
  - Authoring:
    - High-level authoring of dynamic systems
    - DCC plugins, standalone tools, and game engine plugins
  - Runtime:
    - A modular SDK – minimal integration into game engine
    - Leverages PhysX for simulations

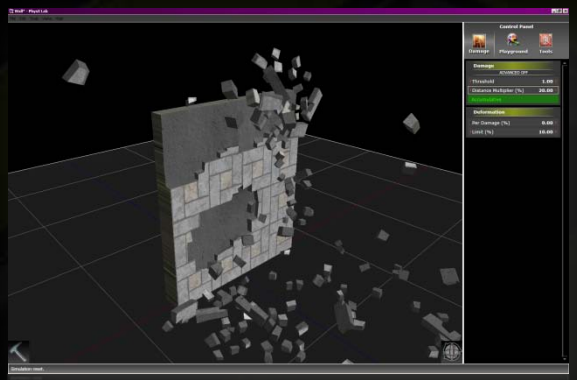
APEX Architecture

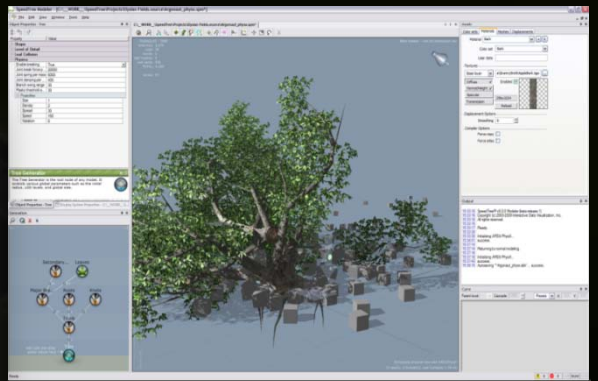


APEX is Artist Focused

- Artist level abstractions of dynamic systems
  - “Destructible bunker” vs. “collection of bricks”
- Intuitive and easy to use







Why Cloth Simulation?

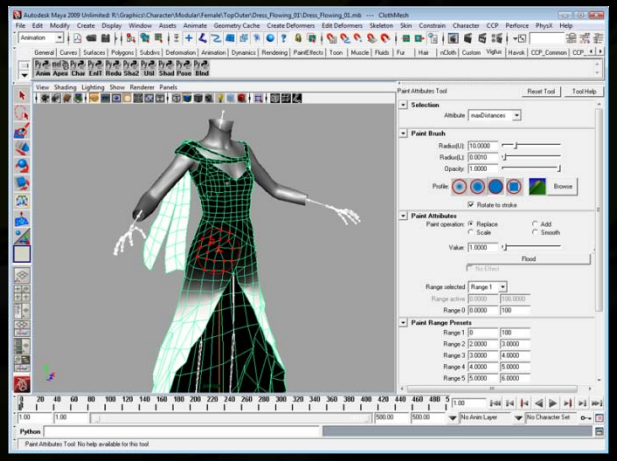
- Adds variety and secondary motion to animation
- Can increase plausibility in realistic scenes a lot
- The same toolset can also be used to create soft body simulation

Things to be aware of

- Like any other simulation, cloth requires boundaries and “taming”
- Sometimes non-realistic cloth simulation looks better
- Elaborate cloth assets can be quite complex to keep nice at all times
- Iteration and testing are the key

Pipeline Integration

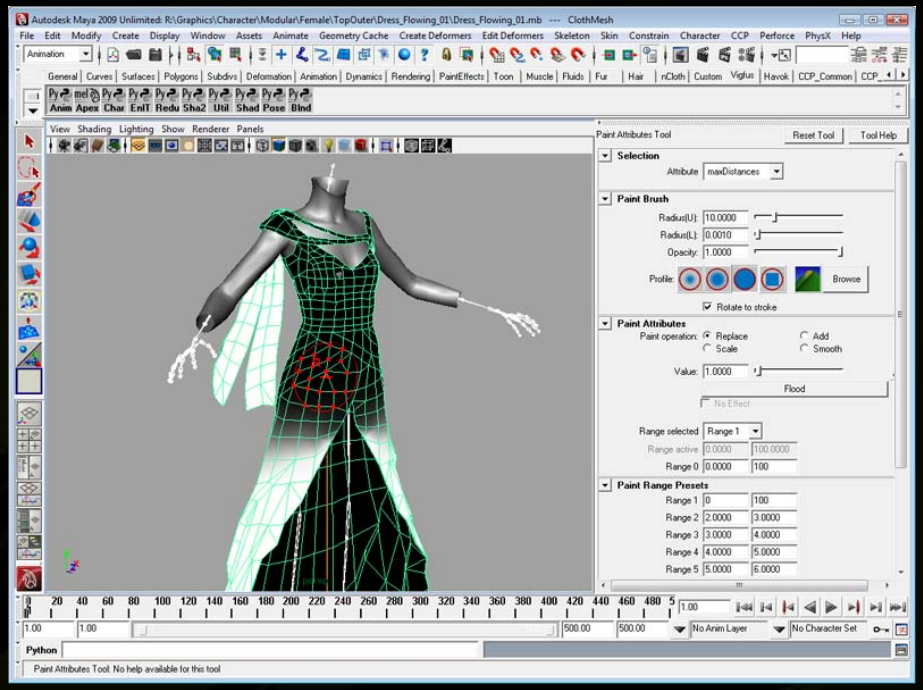
- APEX export was easily integrated into CCP export pipeline
- Use of cloth templates speeds up clothing creation
  - Create once, use often



Apex Cloth Data

Other Model Data

Maya Demo

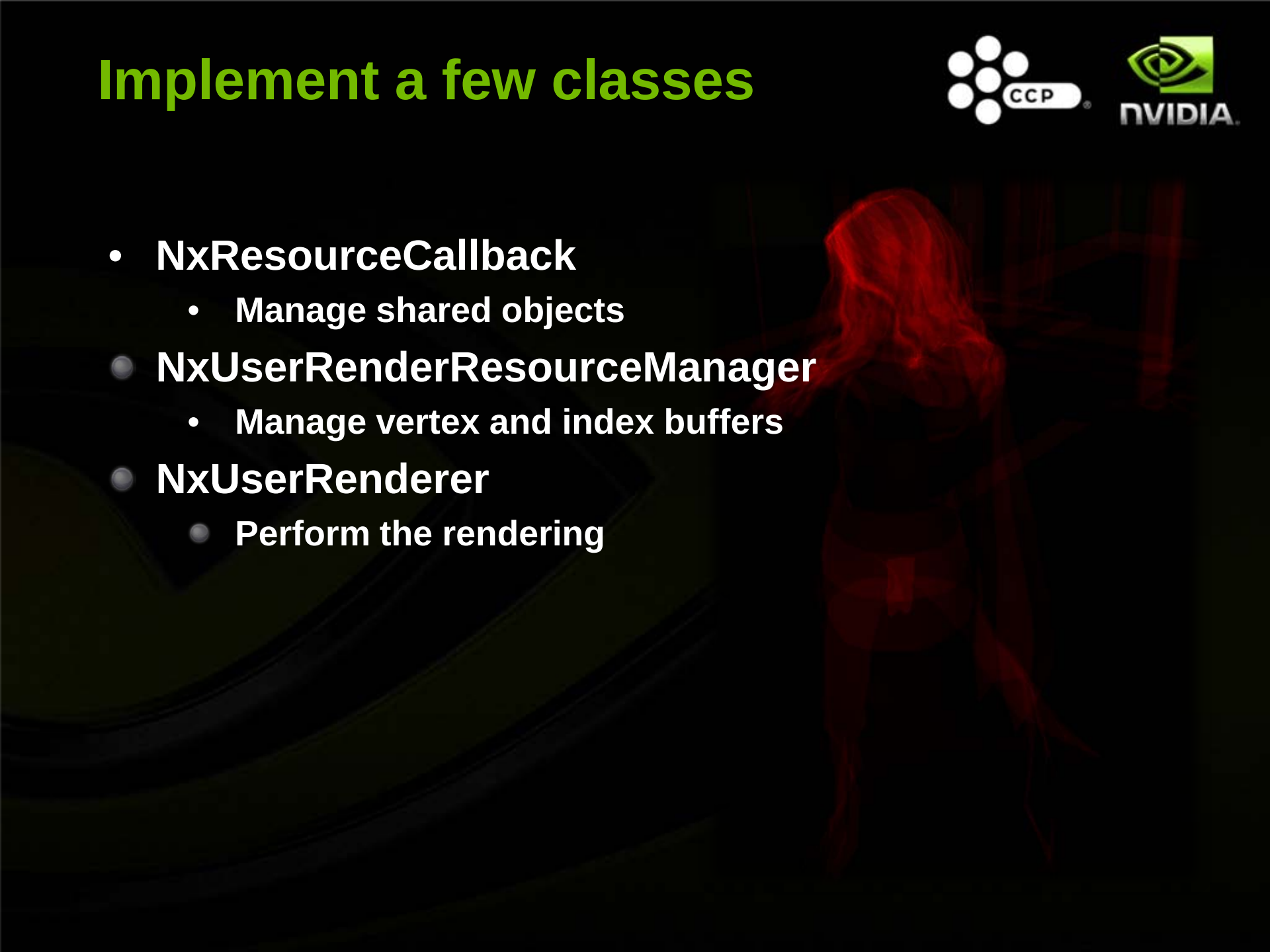

Autodesk Maya 2009 Unlimited: R:\Graphics\Character\Modular\Female\TopOuter\Dress_Flowing_01\Dress_Flowing_01.mb --- ClothMesh
File Edit Modify Create Display Window Assets Animate Geometry Cache Create Deformers Edit Deformers Skeleton Skin Constrain Character CCP Perforce PhysX Help
View Shading Lighting Show Renderer Panels
Paint Attributes Tool
Reset Tool
Tool Help
Selection
Attribute maxDistances
Paint Brush
Radius[U]: 10.0000
Radius[L]: 0.0010
Opacity: 1.0000
Profile:
Browse
Rotate to stroke
Paint Attributes
Paint operation: Replace Add Scale Smooth
Value: 1.0000
Flood
No Effect
Range selected Range 1
Range active 0.0000 100.0000
Range 0 0.0000 100
Paint Range Presets
Range 1 0 100
Range 2 2.0000 3.0000
Range 3 3.0000 4.0000
Range 4 4.0000 5.0000
Range 5 5.0000 6.0000
No Anim Layer
No Character Set
Python
Paint Attributes Tool: No help available for this tool

Challenges

- Numbers of characters on-screen can’t be controlled in an MMO
  - LODs are a must
- High visual fidelity requires lots of consideration for collision meshes
  - Sitting on furniture – how do you handle that?
  - Characters must affect each other too

APEX integration

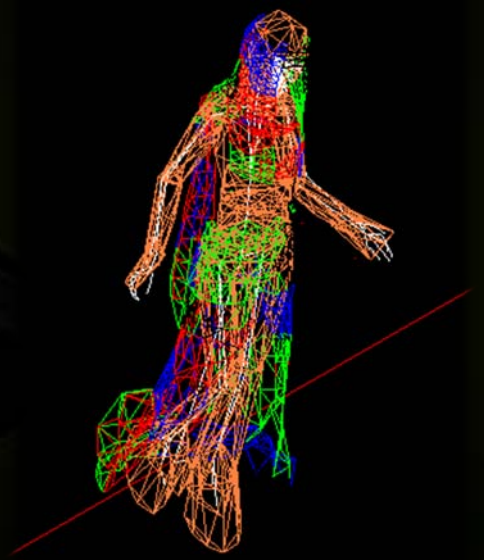
Implement a few classes



- NxResourceCallback
  - Manage shared objects
- NxUserRenderResourceManager
  - Manage vertex and index buffers
- NxUserRenderer
  - Perform the rendering

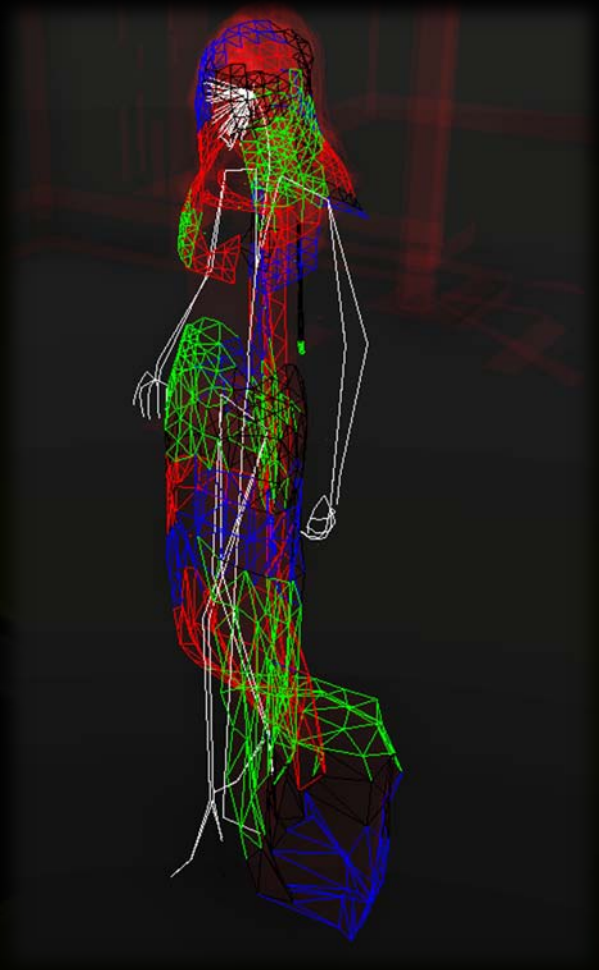
Actors

- Load clothing assets from .aca files
- Create clothing actor from asset
- APEX renders actors through your engine



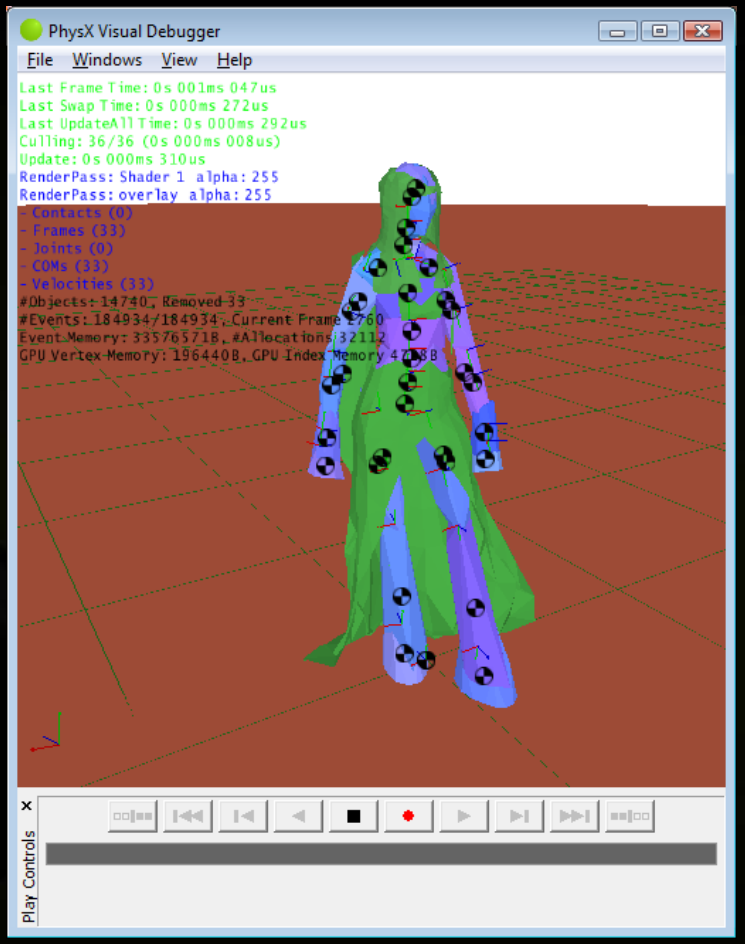
Debugging aids

- Lots of debugging info to be rendered
  - Useful for the programmers
  - But even more so for cloth authoring
  - Worth spending time to support it all



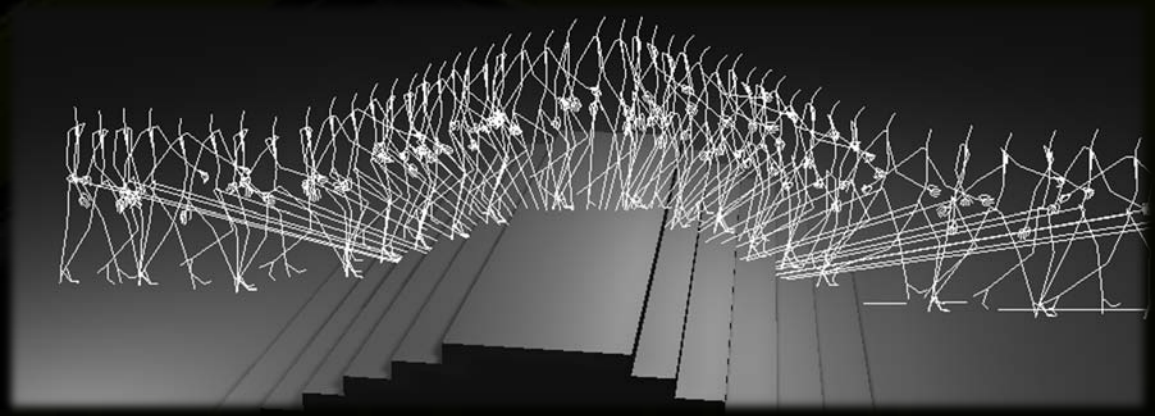
More debugging aids

- Visual debugger
  - Allows recording of data
  - Analyze simulation without game engine



Maximizing performance

- Cloth simulation is heavy
- GPU can do heavy lifting
- Maximizing parallelism maximizes performance
  - Delay skinning to match up with simulation
  - Rendering lags further behind
  - Syncing audio and other effects with animation may become a bigger issue



Benefits of APEX

- Fast, easily iterated authoring
- Artist friendly
- Tweakable data easily exposed in engine for final tweaking
- Debug preview in engine available
- Easily integrated into game engine

Live Demo

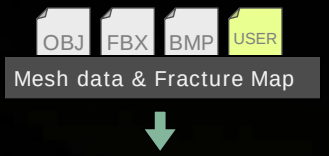
APEX Modules

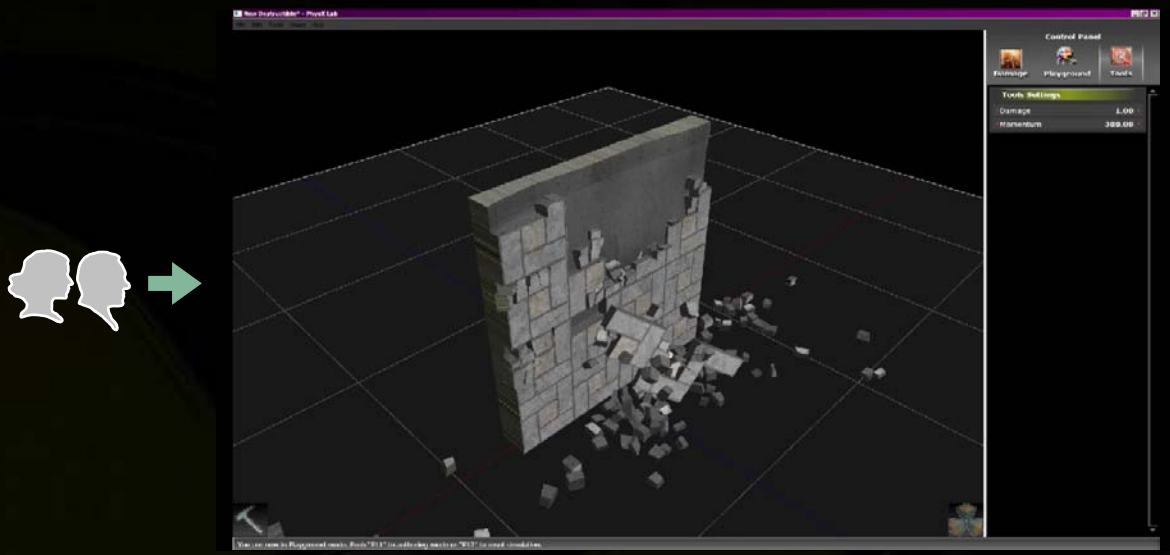
APEX Destruction

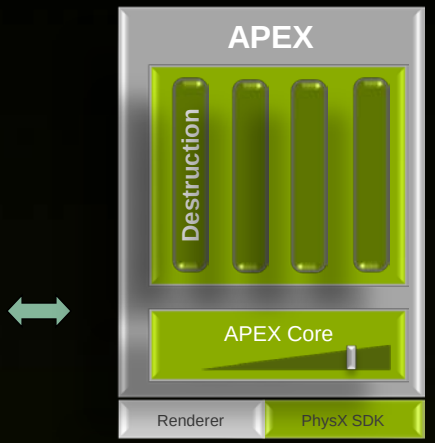
- Fully and partial destructible environments
- PhysXLab tool with preview functionality
- Fully integrated with APEX Particles
- Fracture with noise
- Hierarchical destruction
- Plastic deformation
- Level of Detail
- Scalability

APEX Destruction

Authoring Pipeline





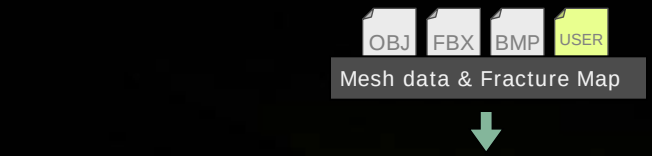


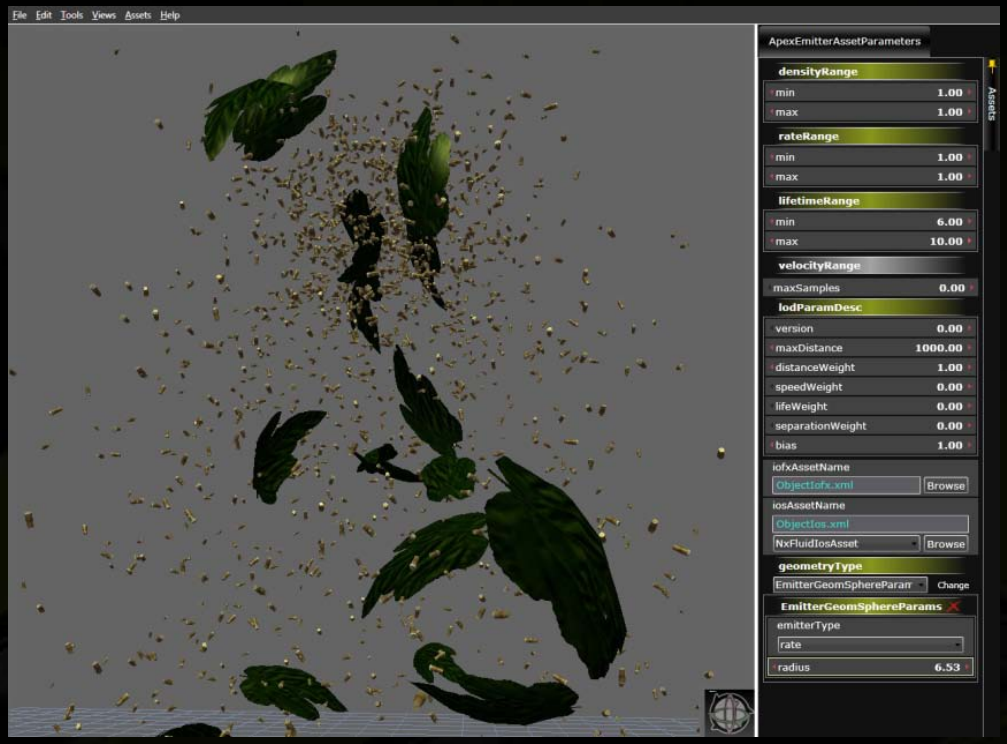
APEX Particles

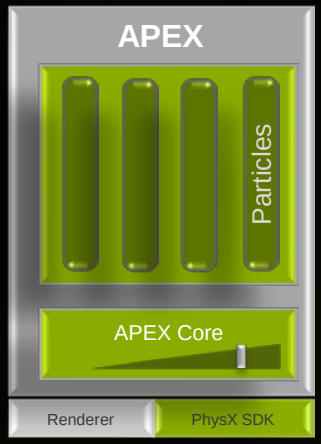
- Full Collision with PhysX environment
- Force fields (wind, explosions)
- Authorable behavior and effect modifiers
- Renderable as sprites or meshes (with orientation)
- Generic emitter
- Special purpose emitters
  - Air/Ground emitter
  - Weapon emitter

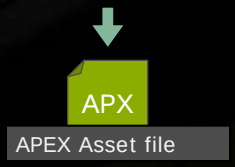
APEX Particles

Authoring Pipeline







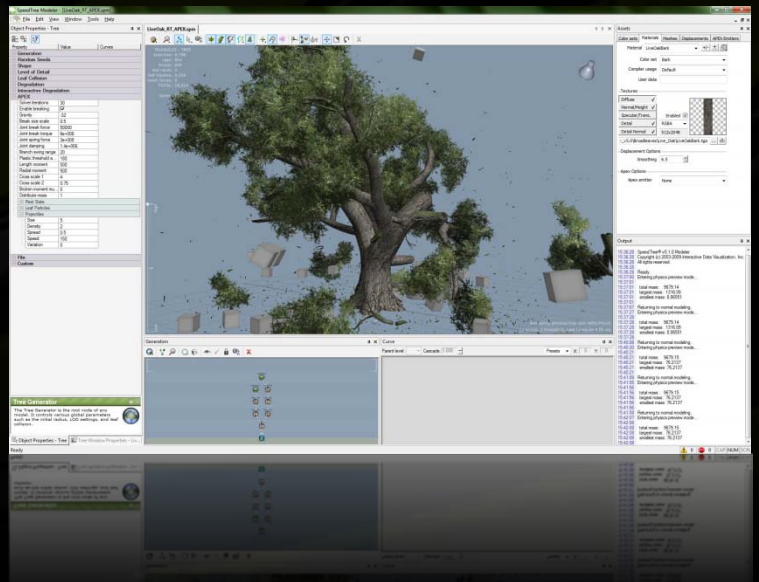


APEX Turbulence

Game Example

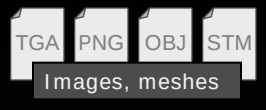
APEX Vegetation / SpeedTree

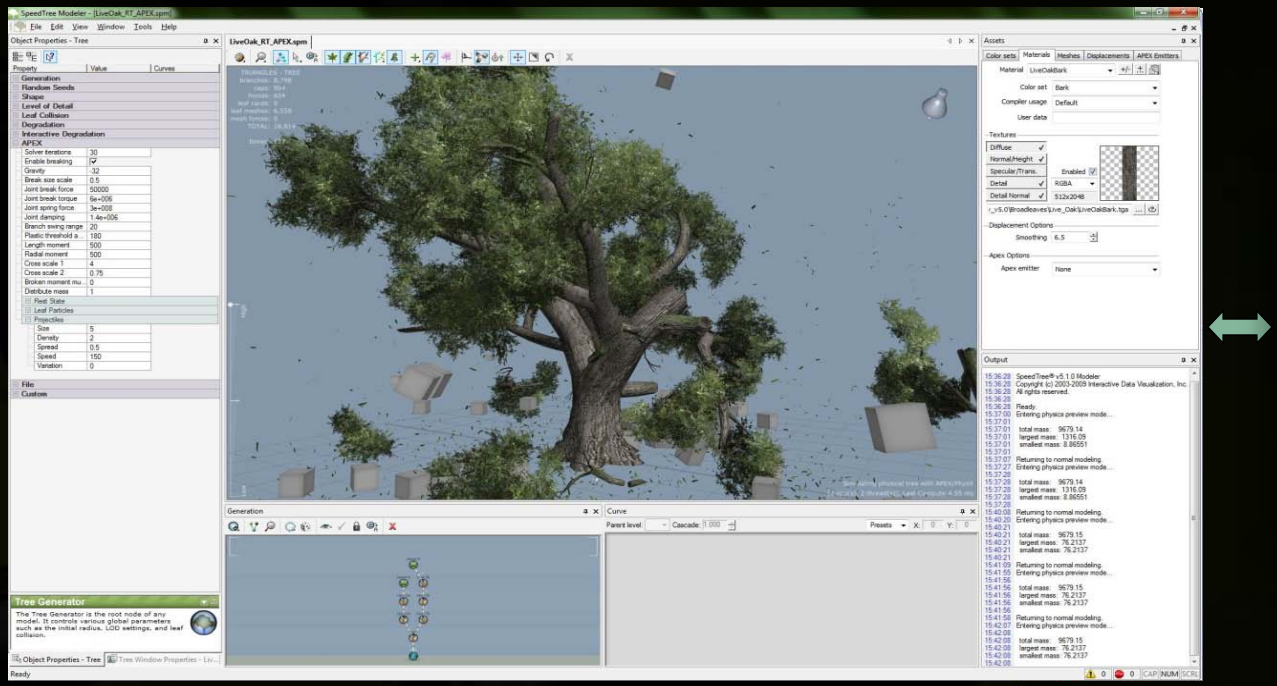
- Full and partial tree destruction/deformation
- State transition between physical and static trees
- Tight integration with APEX Particles
- Level of Detail
- Fully integrated into SpeedTree® Modeler
  - Automatic generation of tree skeleton
  - Configurable bone and joint system
  - Support for multiple APEX Particle Emitters

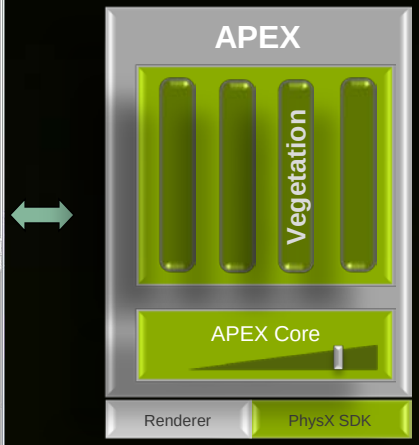


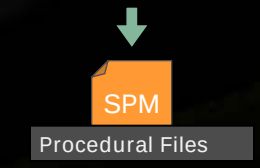
APEX Vegetation / SpeedTree

Authoring Pipeline









Where To Find Us

- Mobile-friendly schedule: http://bit-ly/gdc-nvidia
- NVIDIA - Main Expo Area, Booth 1702
- CCP – Career Pavilion, Booth 2502
- March 11th Sponsored Sessions

0900-1000	Room 310, South Hall	Tegra - Developing Killer Content for Advanced Mobile Platforms
1330-1430	Room 310, South Hall	Physically Simulated Clothing by CCP (EVE Online) Using NVIDIA APEX
1500-1600	Room 310, South Hall	Authoring Physically Simulated Destruction with NVIDIA APEX
1630-1730	Room 310, South Hall	NVIDIA's New Game Development Environment: NVIDIA Parallel Nsight™

- March 12th Presentations

0900-1000	Room 304, South Hall	Taking Fluid Simulation Out of the Box: Particle Effects in Dark Void, Sarah Tariq (NVIDIA), Joe Cruz (VFX)

- Twitter: nvidiadeveloper, Website: http://developer.nvidia.com