

Easy 3D With SceneKit

Justin Miller • Mapbox • @incanus77



Thanks for coming. I'm going to talk about SceneKit, which is one of the things I found most exciting coming out of WWDC this year.

Intro

- Mobile lead at Mapbox
- Build developer tools for custom maps
- Recently: hardware-accelerated map styling



I've been using OpenGL for hardware-accelerated map styling lately, so I can really appreciate better tools to do 2D and 3D hardware acceleration.

SceneKit

- Released with OS X 10.8 (2012)
- Now available in iOS 8!

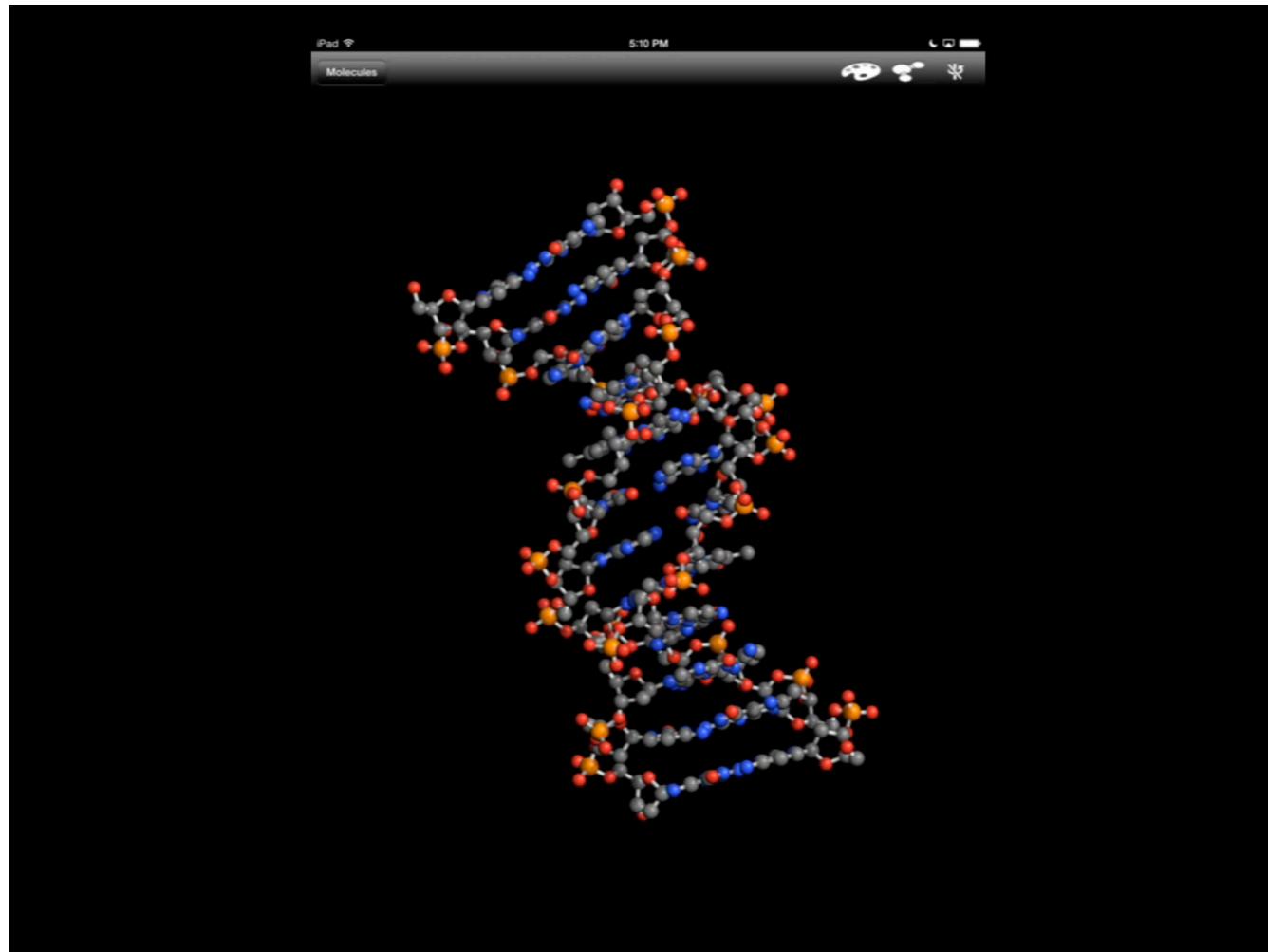


I'm excited about SceneKit because it's now on iOS and your code can be cross-platform!

3D Graphics

- Engaging
 - Interactive models, learning

Why use 3D graphics? Well, they're great for learning tools and interaction.



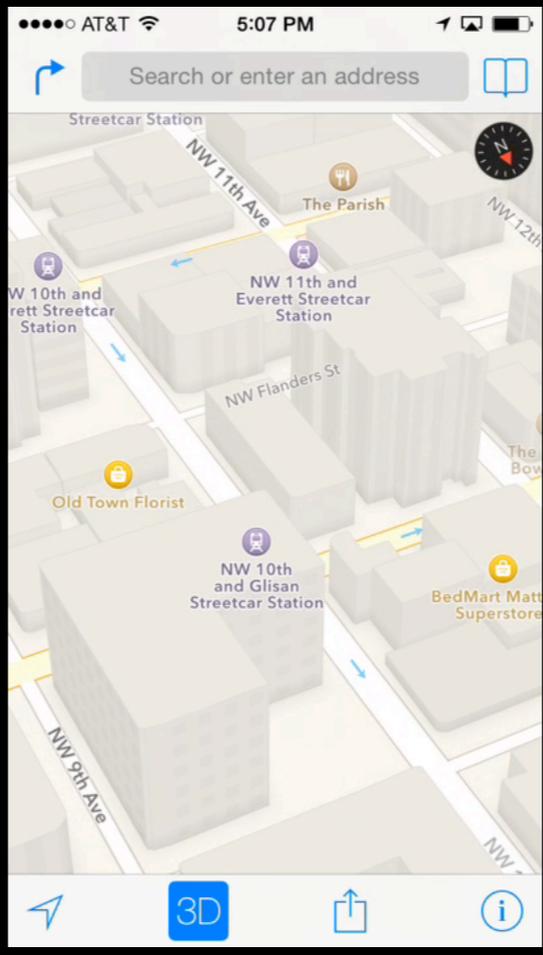
This app by a friend of mine (which is open source) is a great example of an interactive 3D learning tool. What better way to show these concepts?

<http://www.sunsetlakesoftware.com/molecules>

3D Graphics

- Engaging
 - Interactive models, learning
- Mirror the real world
 - e.g. Buildings on maps

3D graphics can help your interface mirror the real world. Apple's maps do a great job of this with 3D buildings.



3D Graphics

- Engaging
 - Interactive models, learning
- Mirror the real world
 - e.g. Buildings on maps
- Games

And of course 3D graphics are great for games. Apple recommends SceneKit for “casual gaming”, so if you’re going to build your own World of Warcraft, you probably want a lower-level tool.



SceneKit would be great for a game like this.



I've heard 3D is scary. I'm a mere mortal. This is probably a lot harder than it even looks. What's the deal?



Imagine you're building a house. SceneKit is like going to the little hardware store run by this adorable couple and buying raw materials. You pick tools, wood, metal, screws, and paint colors. Things aren't trivial, but they are pretty much as expected.



OpenGL, on the other hand, is like planting a tree. Then you grow the tree. Then you fell the tree. Then you build a sawmill and mill the lumber. Then you... well, you get the idea. It's not good or bad, it's just a lower level of working.

The other people in the photo are the folks you'll hit up on Stack Overflow to answer your many OpenGL questions.

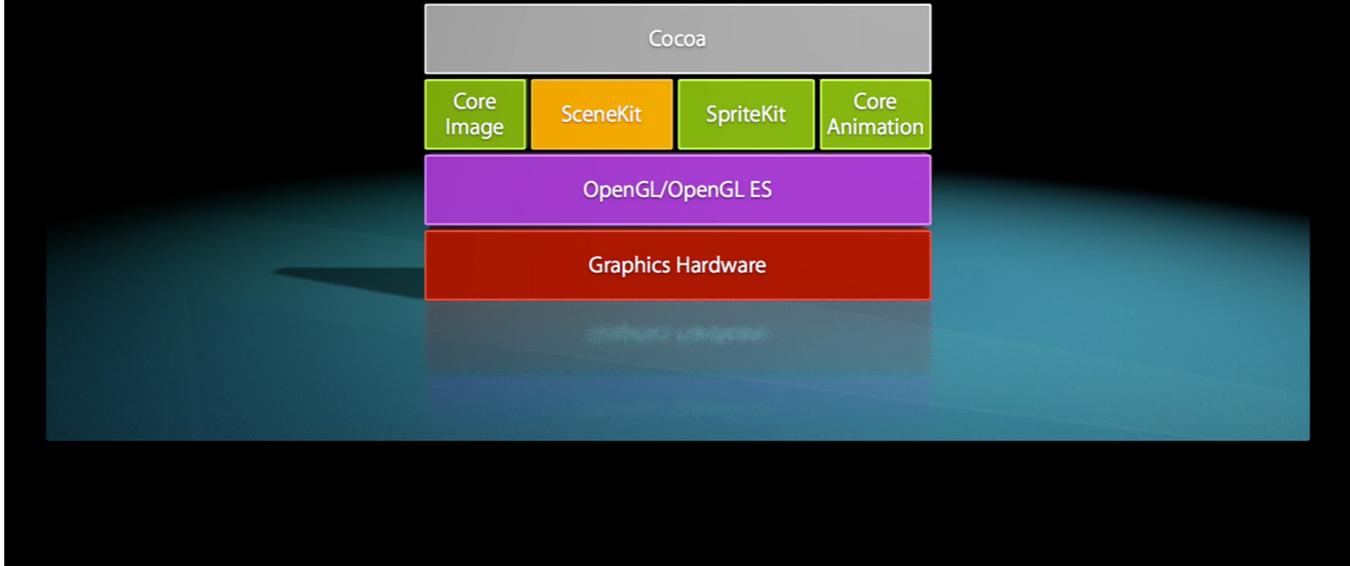
Contrast

- *“Lower-level APIs (such as OpenGL ES) require you to implement the rendering algorithms that display a scene in precise detail.”*
- *“By contrast, Scene Kit lets you describe your scene in terms of its content—geometry, materials, lights, and cameras—then animate it by describing changes to those objects.”*

Apple sets up a nice contrast in their docs. SceneKit is like making a movie — concepts are relatively high-level and easy to grasp.

SceneKit

Graphic Frameworks



Apple likes stack diagrams. Note how SceneKit is on the same level as Core Animation. It works very similarly in terms of placement, animated properties & transactions, and the amount of code it takes to do things. You don't really have to go down to the purple level, but you can.

Scene Graph

- Geometric objects in space
- How they are connected
- The materials they are made of
- Where camera(s) is/are
- Lighting
- Constraints (light always looks at node X)

SceneKit is built on the concept of scene graphs. They're basically just arrangements of nodes in 3D space (with triplet coordinates) that you attach things to — geometric objects, lights, or cameras. Then you move your nodes around.

Scene Graph

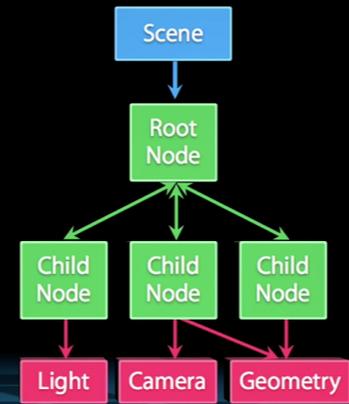
Node attributes

Geometry

Camera

Light

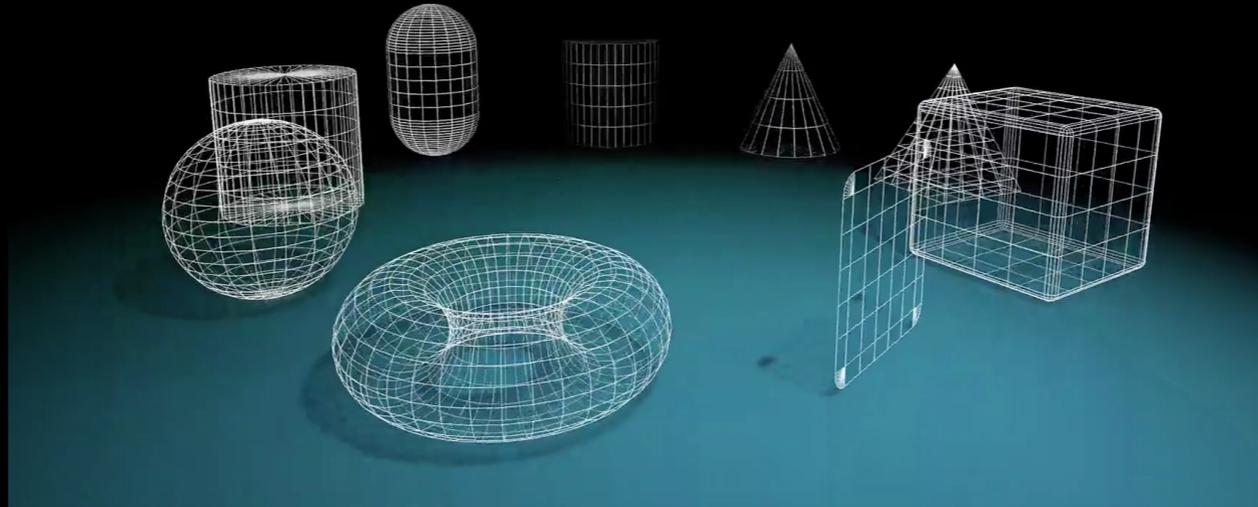
Can be shared



Here's a slide from WWDC showing scene graphs another way.

Creating Geometry

Built-in parametric primitives



In OpenGL, to make geometric shapes, you'd need to describe all the vertices of the shapes, how they relate, and otherwise build the little polygons to make shapes like these. With SceneKit, you just say, "give me a cube with dimensions X".

Demo Code

<http://tinyurl.com/scenokit>

Check out the sample project on GitHub, which uses a SCNText geometry with gesture controls.

There's a lot more to SceneKit, some of it new, like physics modeling, integration with SpriteKit, and particle systems like water, fire, and fog. Really cool stuff.

I highly recommend the WWDC sessions from this & past years. In one, they even built the whole presentation *in* SceneKit so it's animated and in 3D with embedded demos. Plus you can check out that code.

Have fun with SceneKit!