



Developing Interactive Simulations for Touch-Enabled Devices

Daniel Loran, PhD *Truckee Meadows Community College, Reno, NV* - dloranz@tmcc.edu

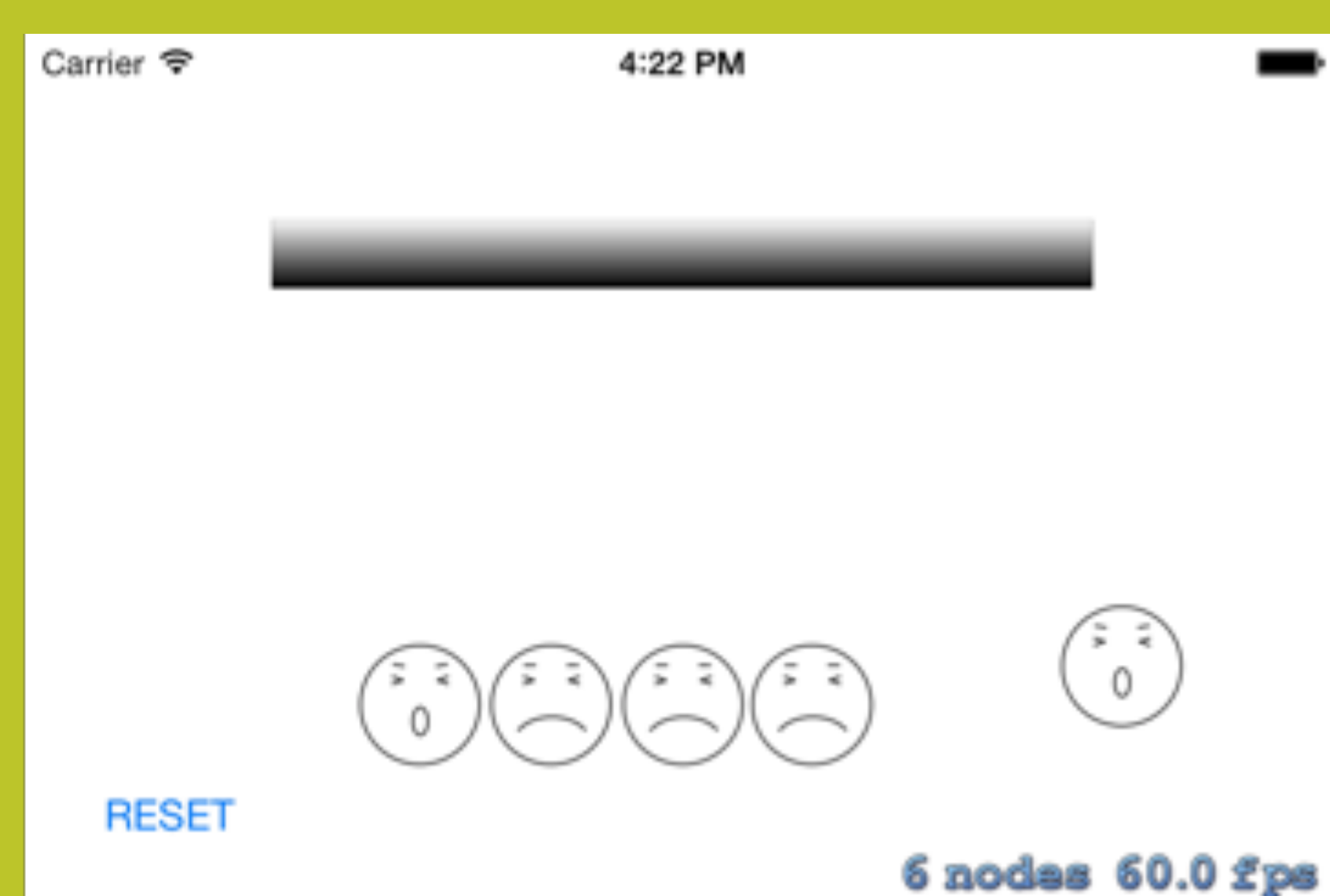
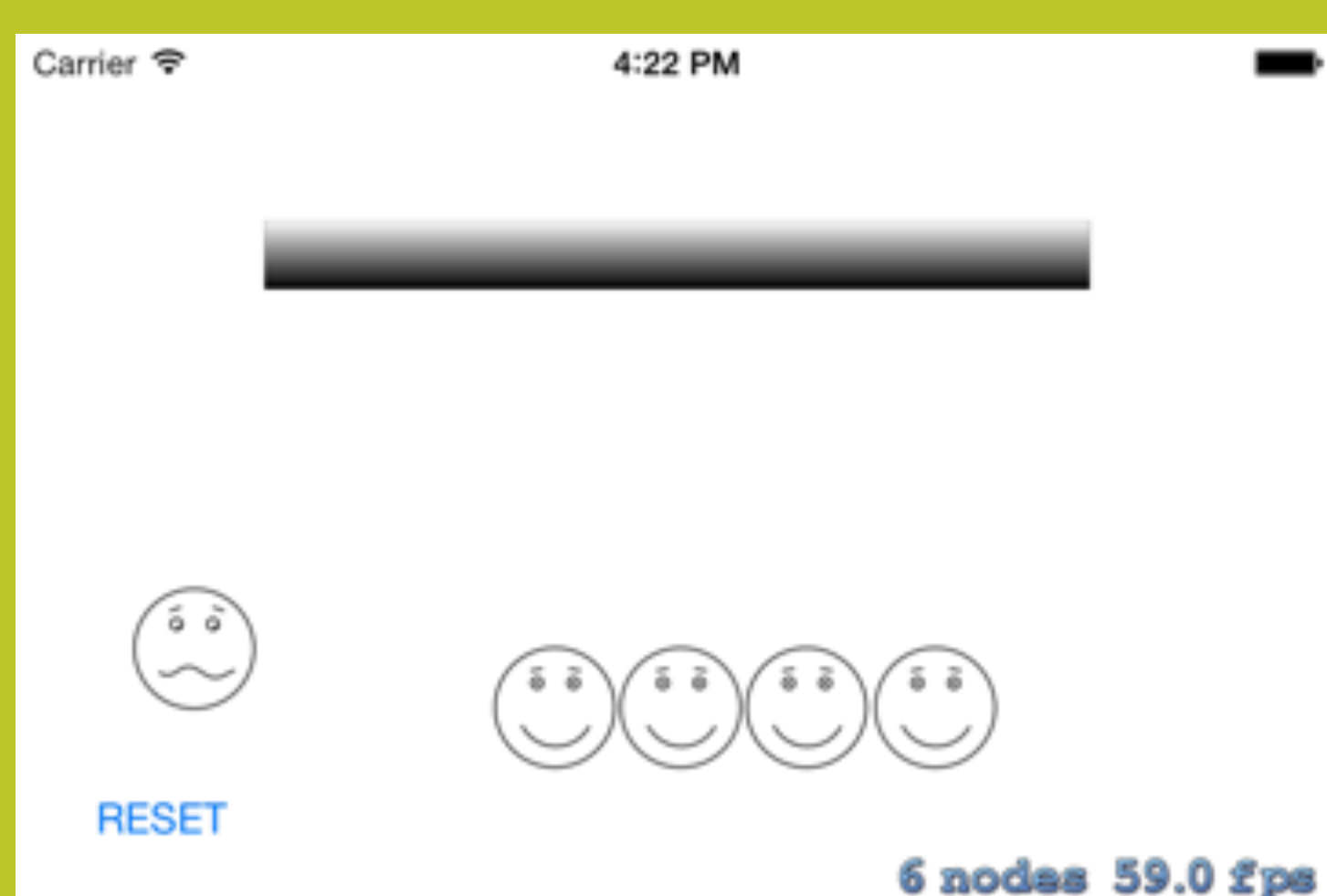
Abstract ⇒ By leveraging the power of modern development environments and accompanying frameworks, an educator with limited programming background can quickly create customized apps for use in his/her classroom. This poster highlight tools and libraries useful for creating interactive simulations, with an emphasis on creating 2D and 3D interactive animations for iOS devices.

Touch-enabled phones and tablets eliminate the abstract model of user interaction required of pointer-based user interfaces and provide a more immersive user experience.

And modern Integrated Development Environments (IDEs) give individual faculty the power to create customized animations and interactive simulations for phones and tablets that are visually sophisticated and feature rich but without needing a dedicated support team of specialist programmers and designers.

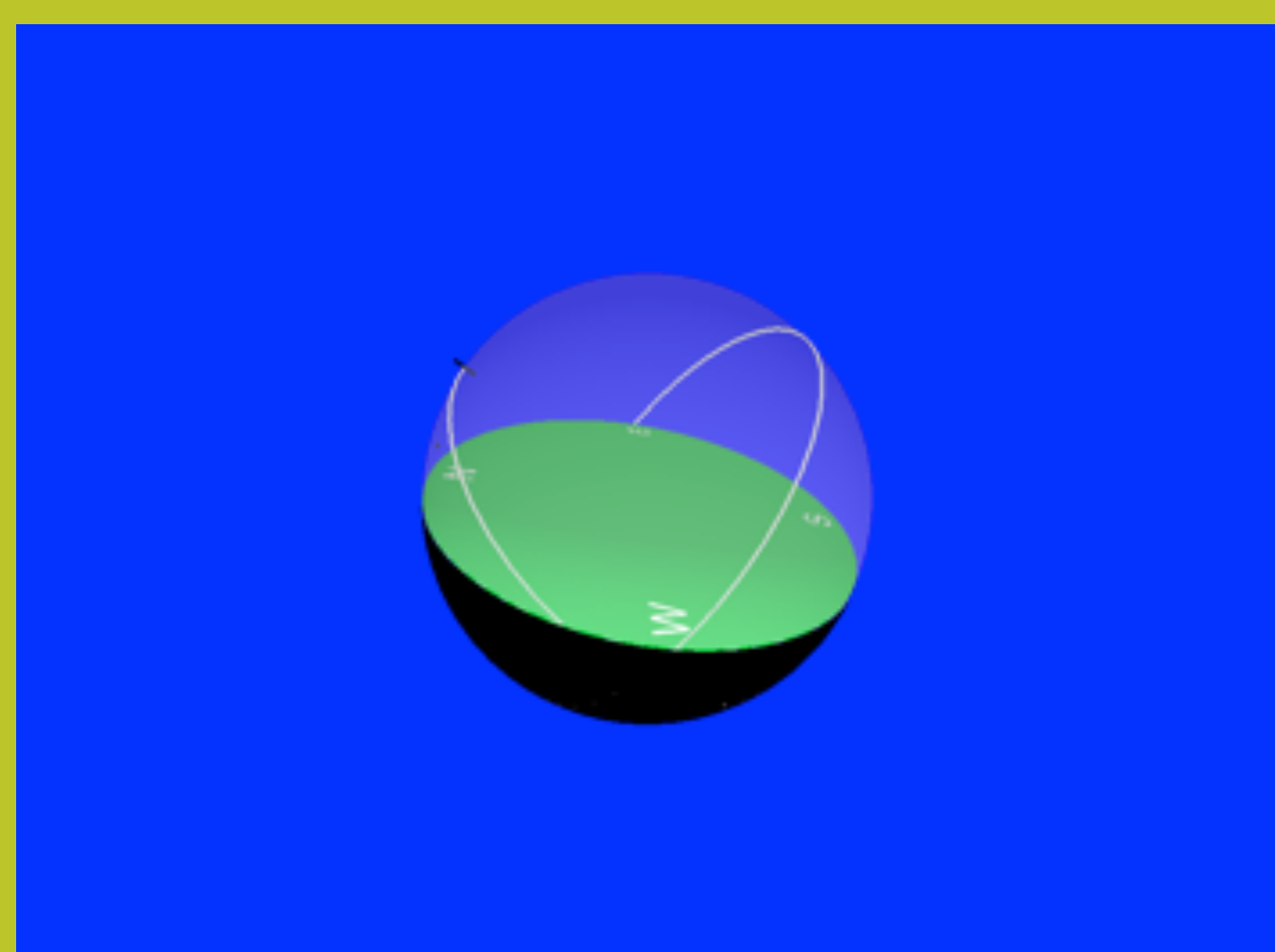
- ✦ Faculty already use word-processors and presentation software to build customized curriculum.
- ✦ Being able to create customized animations and interactive simulations is the next logical step in effectively communicating important ideas to students.

If a picture is worth a thousand words, then an animation is worth a thousand pictures.



A Newton's Cradle simulation created with SpriteKit

The screen shots show different frames where animated smiley faces are colliding with one another. This app example includes realistic physics modeling for the collisions, frame-by-frame animation of the smiley faces to evoke a little personality, and also sound effects to bring realism to the collisions.



An Animated Celestial Sphere Model using OpenGL ES

Sky rotation is animated, and the user can adjust the latitude of observation. The user can also skew around the angle of view by touching and dragging on the model. A simple Arcball implementation is used to convert touch events on a 2D screen to the underlying 3D model, evoking a sense of interacting with a physical object.

Pros and Cons of Building Your Course Apps for Apple Devices

Pro ⇒ unified and consistent deployment across all devices.

But, skills are transferable to other development environments, for example HTML5 / Javascript.

Con ⇒ Not universally adopted.

Development tools for Apple Phones and Tablets

Xcode	Apple's IDE. Includes Storyboard which allows for drag-and-drop creation of user interfaces.
Objective-C and Swift	Latest introduction of script-like Swift promises to make development of feature-rich code even more accessible to non-programmers.
Cocoa Touch	A native framework for capturing touch-events when user interacts with iOS device.
SpriteKit	A native framework for easily creating 2D games. Includes a physics engine for simulating collisions, impulses, & dissipating forces.
SceneKit	A native framework for creating 3D games, removing the need to program the GPU directly.
OpenGL ES	If needed, one can also program the iOS GPU directly using OpenGL ES.
Unity, Unreal, & others	Third Party development environments for 3D games. Includes full suite of helper-tools to simplify your workflow. Many are free to use if distribution is limited to your classroom.

A Subjective List of Best Resources

Resources for Learning Object-Oriented Programming

- Apple Developer Documentation: *Object-Oriented Programming with Objective-C*.
- McLaughlin, Pollice & David West, *Head First Object-Oriented Analysis & Design*, 2006, O'Reilly.

Resources for Learning Objective-C

- Kochan, *Programming in Objective-C*, 2013, Addison-Wesley Professional.
- Hillegass & Ward, *Objective-C Programming: The Big Nerd Ranch Guide*, 2013, BNR Guides.

Resources for Learning iOS and Cocoa Touch Programming

- Beginning iOS with Objective-C ... a 7-day, onsite training session.
<https://training.bignerdranch.com/classes/beginning-ios>
- Conway & Hillegass, *iOS Programming: The Big Nerd Ranch Guide*, 2014, Big Nerd Ranch Guides.
- Neuburg, *Programming iOS*, 2014, O'Reilly.
- Sadun, *iOS Drawing: Practical UIKit Solutions*, 2013, Addison-Wesley Professional.
- Lockwood, *iOS Core Animation: Advanced Techniques*, 2013, Addison-Wesley Professional.

Resources for Learning Sprite Kit

- Wenderlich, *iOS Games by Tutorials*, 2013, Razeware LCC.

Resources for Learning OpenGL ES

- Apple Developer Documentation: *OpenGL ES Programming Guide for iOS*
- Ginsberg & Purnomo, *OpenGL ES 3.0 Programming Guide*, 2014, Addison-Wesley Professional.
- Wolf, *OpenGL 4 Shading Language Cookbook*, 2013, Packt Publishing.