Scroll View School

Hands-On Challenges

Scroll View School
Hands-On Challenges

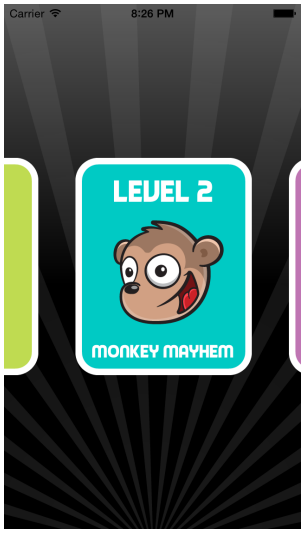
Challenge H: A UIKit Level Selector

As you saw in the lecture, you can definitely implement a level selection scene in Sprite Kit if you'd like, using the power of UIScrollView. But you can do it in UIKit if you'd like as well, since Sprite Kit works seamlessly with UIKit and Storyboards.

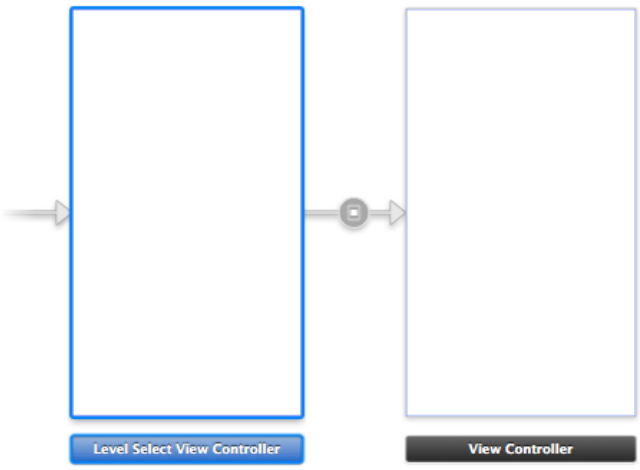
In this challenge, you will reimplement the level selection scene for your Sprite Kit game, but this time using UIKit.

Part 1: Tap Gesture Recognizer

Start by opening the starter project for this challenge, and build and run. You'll see that you can scroll through the list of levels as usual and tap a level to select it and move to a Sprite Kit scene:



However, this time the project is made completely with UIKit! Open **Main.storyboard** and you will see there are two view controllers:



When you create a Sprite Kit game the template gives you one view controller with a `SKView` that contains your game to start, but you can always add additional view controllers like the one shown here for main menus, etc.

The level select view controller has a view called `RWTLevelSelectView` that contains the code to create the scroll view and its subviews, and lay them out on the screen.

This code should be review for concepts you've learned earlier in this series. Your first part of this challenge is to review this code and make sure you understand how it works. Refer back to other parts of this series and other video tutorials if necessary.

Once you're satisfied you have a good feel for how it works, implement `scrollViewDidEndDragging:` to force the scroll view to stop on level boundaries. Refer back to the demo code or lecture if necessary. Good luck! :]

Uber Haxx0r Challenge: Pan Gesture Recognizer

Currently when you select a level, it simply does a fade out transition between the two scenes. It would be better if when you selected a level, it zoomed in to the center of the screen.

To do this, implement the commented "TODO" lines in the challenge starter project. Note that you'll have to take the view out of the scroll view hierarchy and make it a direct subview of the `RWTLevelSelectView` so that it's easier to center the view. You'll also need to do some coordinate system conversions with `convertPoint:toView:`. This is your last challenge for this series – good luck and I hope had a good time and learned a lot! :]