SpineEducation Documentation

## Maya Ramamurthy, Katrine Rachitsky, Manaar Hyder, Randa Mohsen

### Dr. Anand, Colby Oitment

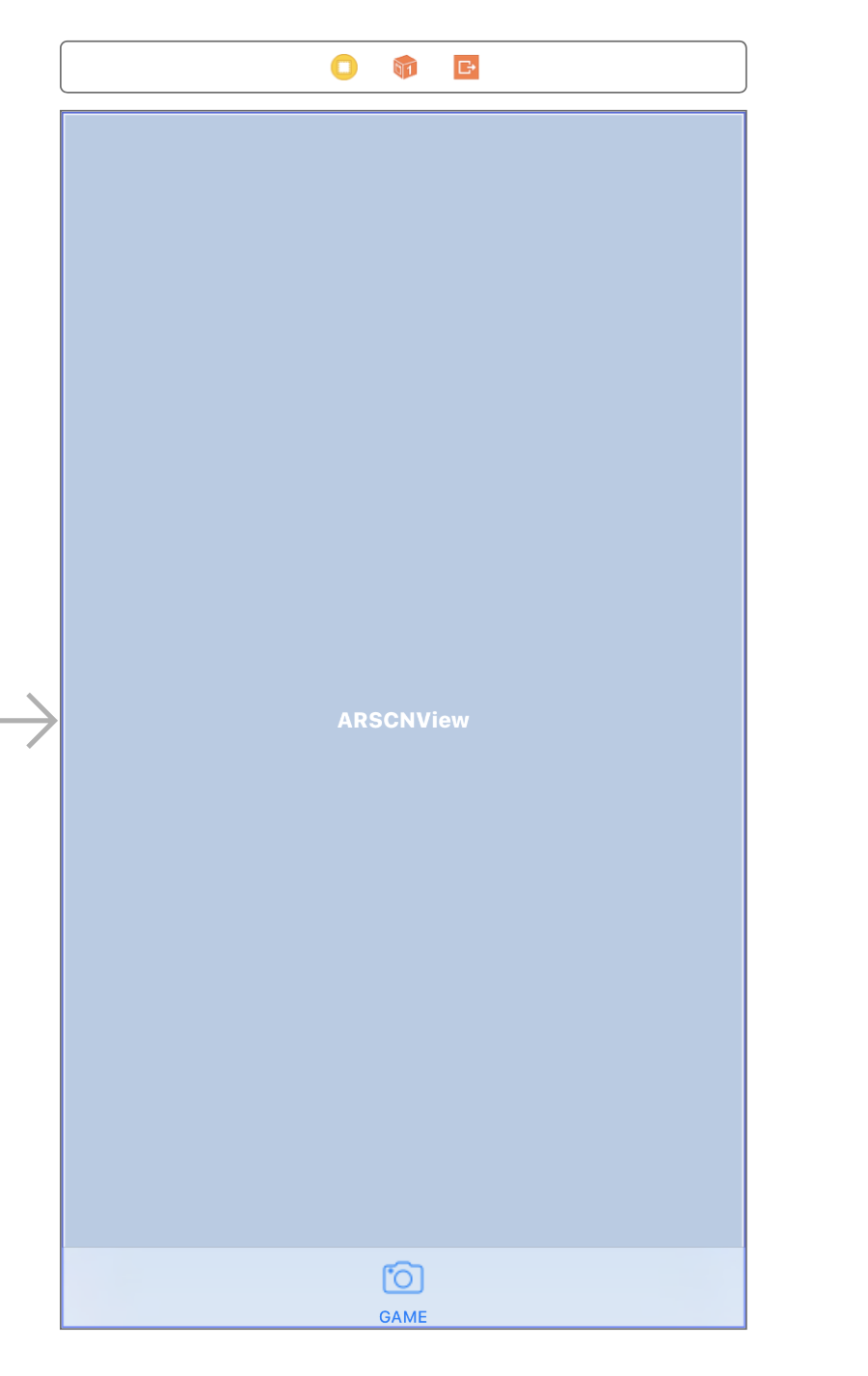
#### CS 4ZP6

# **Augmented Reality (AR) Component**

## *AR Code Base:*

## **Storyboard Component:**

The Storyboard for this component is fairly simple, with just an *ARSCNView* component added into the screen to allow for the AR function to work. This part is tied into the “GAME” component of the menu options on the bottom bar.



## **ViewController.swift:**

There are several methods in this class which allow the AR component to work:

override func viewDidLoad()

This is the first function that is called upon in ViewController.swift. This function is used to set up the key components for AR, such as initializing the Scene, and displaying the first instruction for tapping the screen to show the spine to the user.

override func touchesBegan()

This function is where the majority of the AR Surgery occurs, as it is where the app detects the user’s taps onto the screen and responds accordingly, depending on the value of a variety of variables. There are a few global variables initialized in ViewController.swift, mainly used to indicate the state in which the program is currently in.

if (!targetExists)

If the target does not exist already, it will be added to the scene. Moreover, it will be added as a child node of the user perspective, so that it remains in the middle of the screen, regardless of where the user moves in 3D space. The target not existing also indicates that the spine has not yet been displayed to the user (this is likely the first screen tap), so the spine will be added to the scene as a child node of the scene, so that it will behave as a real 3D object in 3D space. Additionally, the user will be given a few seconds to reposition the spine by tapping the screen so that the spine is recreated in any direction the user points the camera of their device.

(Calls on the createSpine() and the showTarget()functions)

if (targetExists && !targetLocked)

Values of true and false for these variables, respectively, indicate that the above target has been created, however the start point for the surgery has not yet been set. This will call upon the SCNHitTestOption to determine if the user has selected a valid part of the spine. If not, the user will be able to continue tapping until they have. Once they do select a valid component of the spine, this point will be ‘locked in’ and saved for use in determining trajectory.

if (targetLocked && !trajectoryExists)

This will check if the trajectory exists (if it has been determined at this stage in the code), and, if not, will call upon the drawTrajectory() function to draw the trajectory.

func showTarget()

This function will make an image of a 2D target (make2dNode) into a node and

attach it to sceneView.pointOfView?.addChildNode(bullseyeNode) so that it appears

on the camera position.

  func showUserInstruction ()

This function allows easy drawing of instructions on the screen by sending it an

offset value for proper positioning and the string, holding any desired instruction.

  func make2dNode()

Creates a 3D node object from a 2D image

  override func viewWillAppear()

Creates a session configuration for AR.

override func viewWillDisappear()

Pauses the view’s session (ie if switch screens can reload current progress)

func drawTrajectory ()

Using the current locked target position (where the user selected), to draw a

line from camera to the position.

func createSpine(position : SCNVector3)

        Creates the spine node at the given position, and attaches it to the scene.

    override func didReceiveMemoryWarning()

      Release any cached data, images, etc that aren't in use.

    func session()

        Present an error message to the user if failed to load session

    func sessionWasInterrupted()

        Inform the user that the session has been interrupted, for example, by

presenting an overlay

func sessionInterruptionEnded()

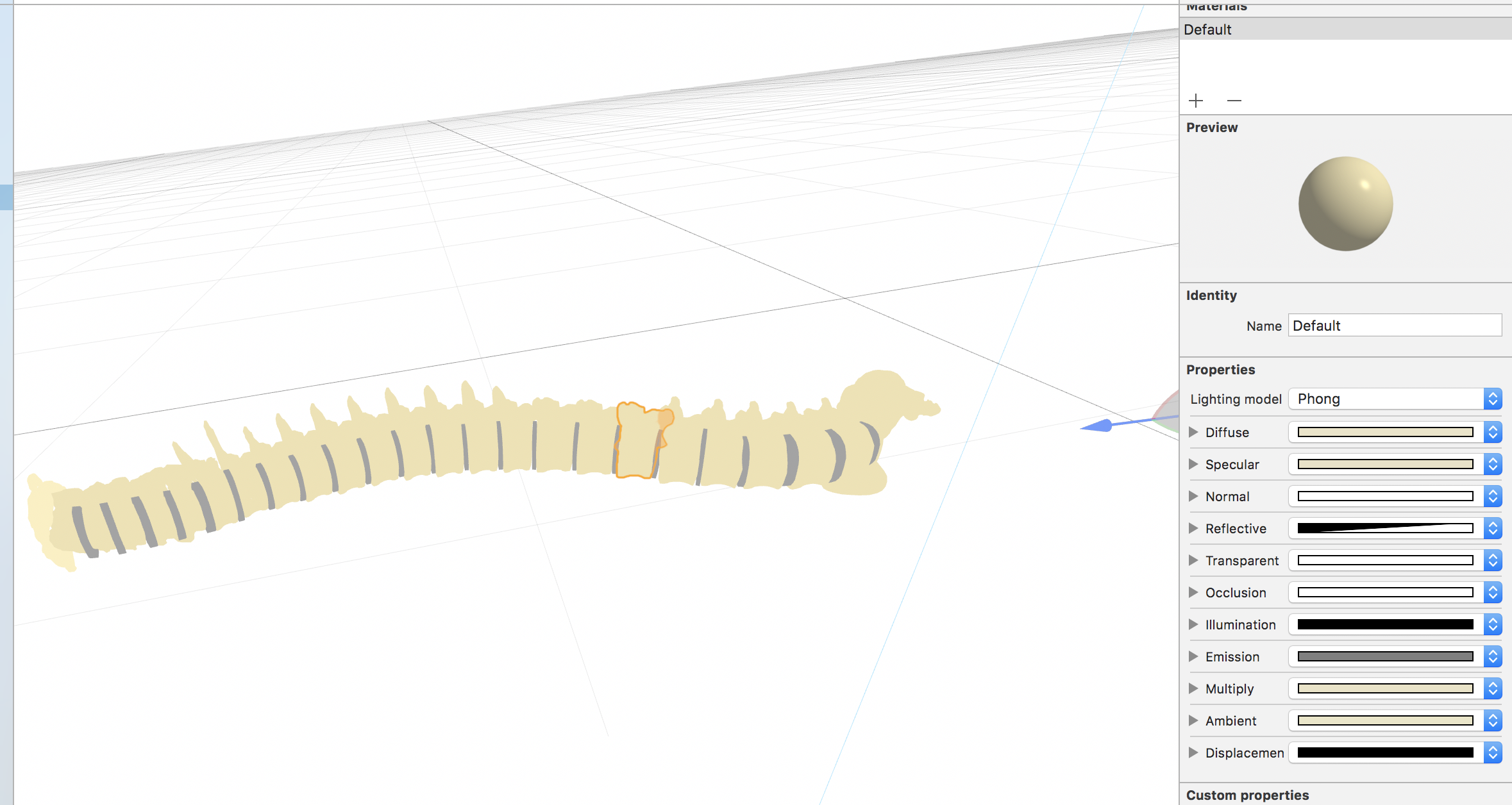
        Reset tracking and/or remove existing anchors if consistent tracking is

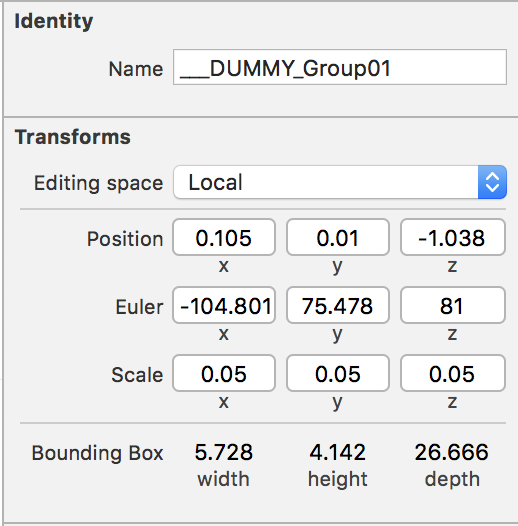
required

# *Spine Models:*

The current spine model is 4Cylinders.dae. This file has been modified in Blender to add in cylinders for the surgery components. These cylinders indicate acceptable pedicle start points as well as encompassing the entirety of the acceptable trajectory for a screw to enter the spine (if the start point or trajectory lie outside of a cylinder, they can be deemed inaccurate and the user has failed to properly execute the procedure). Additionally, it has the following XCode properties attached to it in order to add textures, materials to it, and colouring. The cylinders are marked as “hidden” so that, while they exist on the spine model, the user cannot see them, which adds difficulty.

Please also note in the second screenshot, the numbers corresponding to the position, euler and scale rows are not to be changed, this can ruin the proper positioning of the spine.



**

# *Getting Started with AR (Useful Resources)*

Learning How to set up AR Environment

* <https://lifehacker.com/how-to-get-started-using-apple-s-arkit-augmented-realit-1797690723>
* <https://blog.pusher.com/building-an-ar-app-with-arkit-and-scenekit/>
* <https://developer.apple.com/documentation/arkit/handling_3d_interaction_and_ui_controls_in_augmented_reality>

## Adding Materials/Textures in Blender:

* <https://www.wikihow.com/Apply-a-Material-or-Texture-in-Blender>
* <https://blender.stackexchange.com/questions/8697/how-do-i-put-an-image-texture-on-a-material-in-cycles>

## Open Source Spine Models

* <https://grabcad.com/library/human-spine-1>
* <https://www.thingiverse.com/thing:31845/#comments>
* [https://sketchfab.com/models/1ce94666922f48c9b68cd0e196b74a0a#](https://sketchfab.com/models/1ce94666922f48c9b68cd0e196b74a0a)
* <https://grabcad.com/library/model-pedicle-screw-solidworks-iges>

## *AR Related Errors:*

### **Help! The Spine Is Flying Away:**

Sorry, this cannot be helped. This error occurs due to the fact that AR works by configuring itself to the camera’s real-world surroundings, and, as such, places the object (ie the Spine) in real world configuration. This is common when the user is moving the camera too jerkily, or switching the surroundings quickly, as this will cause the spine to fly away as the program is attempting to read the surroundings and place an object in an unstable 3D space.

**What you can do:** Hold the camera steady and ensure that your environment is adequately lit (poor lighting can result in a poor read of the environment, the phone camera cannot determine the planes of the 3D space when the surroundings are indiscernible), and the AR will re-configure based on the surroundings and stabilize. If this does not work, then try relaunching the app. It is very rare that the app is unable to reconfigure once the surroundings have been stabilized and the environment has enough lighting. Also try using a different view on the camera (ie turn a different direction) as it is possible the direction you were trying was too confusing for the AR.

# **Cases Component**

# **Common XCode Related Errors**

## *Error Message:* “Code Signing Error”

### Details:

Code Signing Error: No account for team "NWVTD7H53D". Add a new account in the Accounts preference pane or verify that your accounts have valid credentials.

Code Signing Error: No profiles for 'com.<name>.Spineducation' were found:  Xcode couldn't find any iOS App Development provisioning profiles matching 'com.<name>.Spineducation'.

Code Signing Error: Code signing is required for product type 'Application' in SDK 'iOS 11.2'

### Cause:

This error occurs due to the fact that when the Git Project was pulled from Git, the name of the team was left from the last user’s details.

### Solution:

This can be fixed in the *Spineducation.xcodeproj* file by ensuring that the proper values are in each field.



## *Error Message: “Command /usr/bin/codesign failed with exit code 1”*

Details:

CodeSign /Users/me/Library/Developer/Xcode/DerivedData/MyApp-Spineducation/Build/Products/Debug-iphonesimulator/MyApp.app

CODESIGN\_ALLOCATE=/Users/me/Downloads/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/codesign\_allocate

export PATH="/Users/me/Downloads/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/usr/bin:/Users/me/Downloads/Xcode.app/Contents/Developer/usr/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin"

Signing Identity: "-"

/usr/bin/codesign --force --sign - --timestamp=none /Users/me/Library/Developer/Xcode/DerivedData/MyApp-gnoiiwnelmxzdidnijaswisrwdqe/Build/Products/Debug-iphonesimulator/MyApp.app

/Users/me/Library/Developer/Xcode/DerivedData/MyApp-gnoiiwnelmxzdidnijaswisrwdqe/Build/Products/Debug-iphonesimulator/MyApp.app: resource fork, Finder information, or similar detritus not allowed

Command /usr/bin/codesign failed with exit code 1  
Cause:

This error occurs because of cached information from pulling from Git

Solution: Run the following line in Terminal:

*xattr –rc ~/Library/Developer/Xcode/DerivedData*