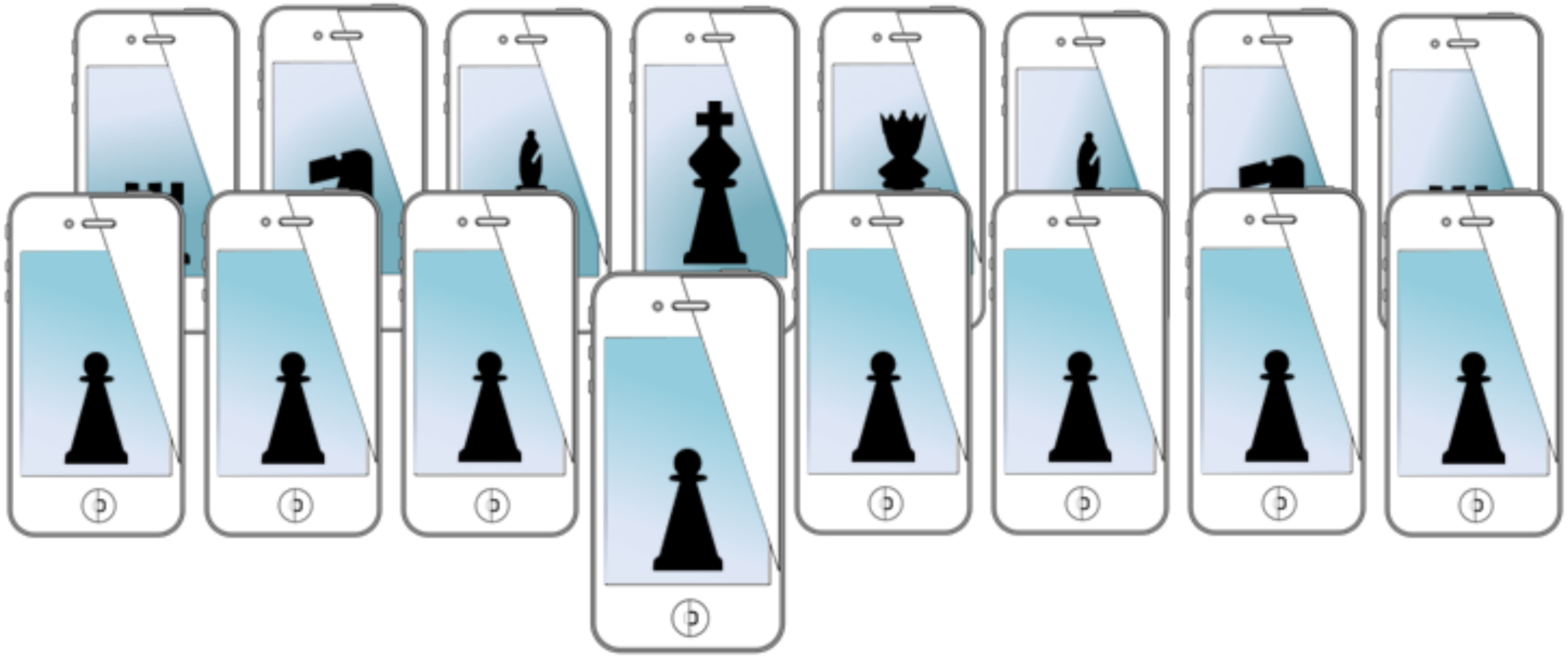
MOBILE SENSING LEARNING



CS5323 & 7323

Mobile Sensing and Learning

CoreML and ARKit

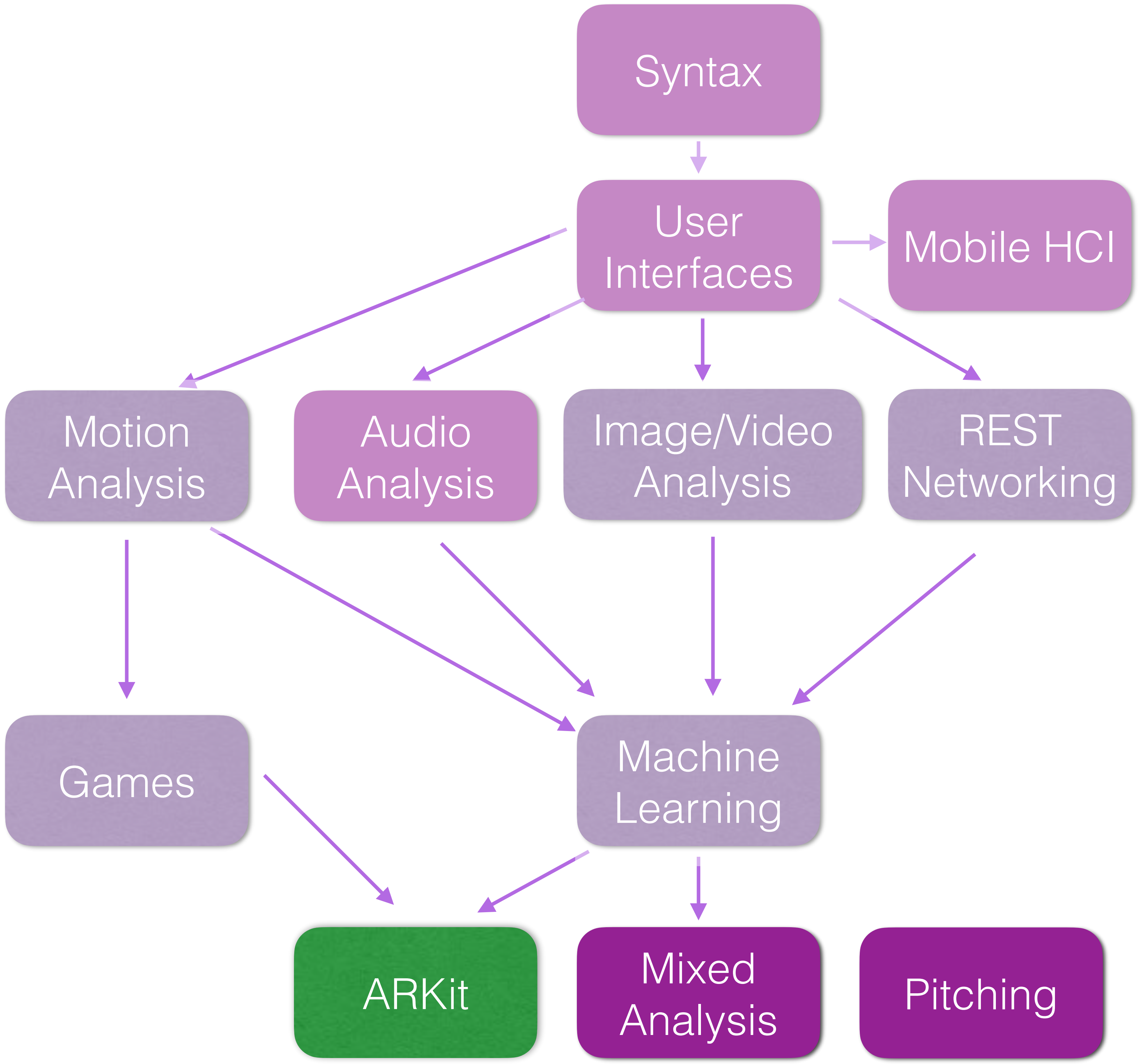
Eric C. Larson, Lyle School of Engineering,
Computer Science, Southern Methodist University

course logistics and agenda

- agenda:
  - Apple vision API (if needed)
  - sceneKit review
  - ARKit
  - ARKit demo

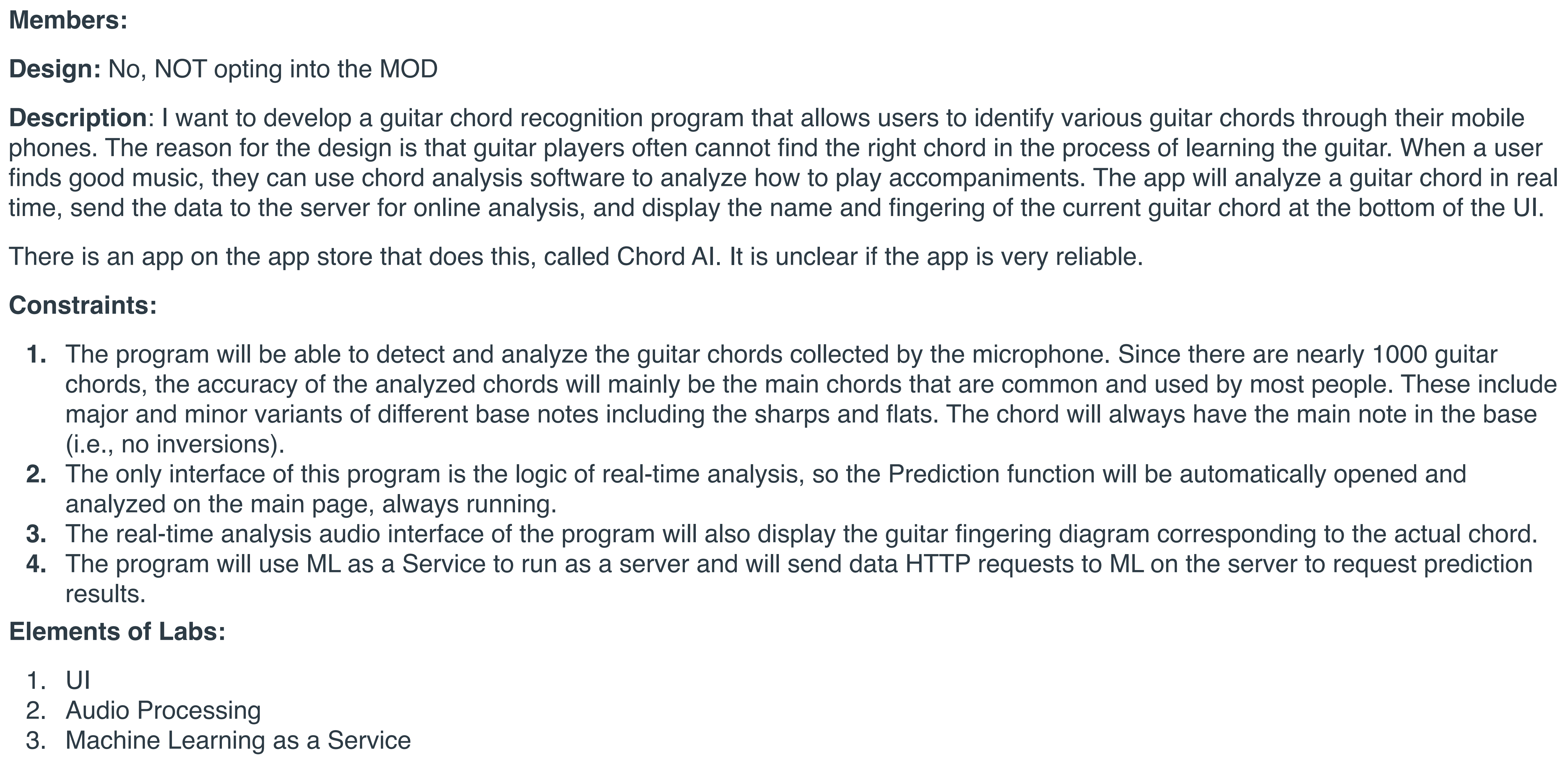
https://developer.apple.com/videos/play/wwdc2017/602/

class overview



Lab Five and Proposals

- Questions on lab five
- Questions on proposals



Members:

Design: No, NOT opting into the MOD

Description: I want to develop a guitar chord recognition program that allows users to identify various guitar chords through their mobile phones. The reason for the design is that guitar players often cannot find the right chord in the process of learning the guitar. When a user finds good music, they can use chord analysis software to analyze how to play accompaniments. The app will analyze a guitar chord in real time, send the data to the server for online analysis, and display the name and fingering of the current guitar chord at the bottom of the UI.

There is an app on the app store that does this, called Chord AI. It is unclear if the app is very reliable.

Constraints:

1. The program will be able to detect and analyze the guitar chords collected by the microphone. Since there are nearly 1000 guitar chords, the accuracy of the analyzed chords will mainly be the main chords that are common and used by most people. These include major and minor variants of different base notes including the sharps and flats. The chord will always have the main note in the base (i.e., no inversions).
2. The only interface of this program is the logic of real-time analysis, so the Prediction function will be automatically opened and analyzed on the main page, always running.
3. The real-time analysis audio interface of the program will also display the guitar fingering diagram corresponding to the actual chord.
4. The program will use ML as a Service to run as a server and will send data HTTP requests to ML on the server to request prediction results.

Elements of Labs:

1. UI
2. Audio Processing
3. Machine Learning as a Service

SceneKit review: 3D scenes

- need some basic knowledge to understand augmented reality and digital scene
- SceneKit allows you to create a 3D world and add physics, nodes, lighting, etc.
    - very powerful
- basic workflow:
    - setup world
    - add nodes

work flow in 3D scenes

SCNScene

rootNode

→

SCNNode

Name
Camera
Light
Geometry

childNodes[]

→

SCNGeometry

Name
Materials[]
Text
Shape (plane, box, sphere, cylinder…)

→

SCNMaterial

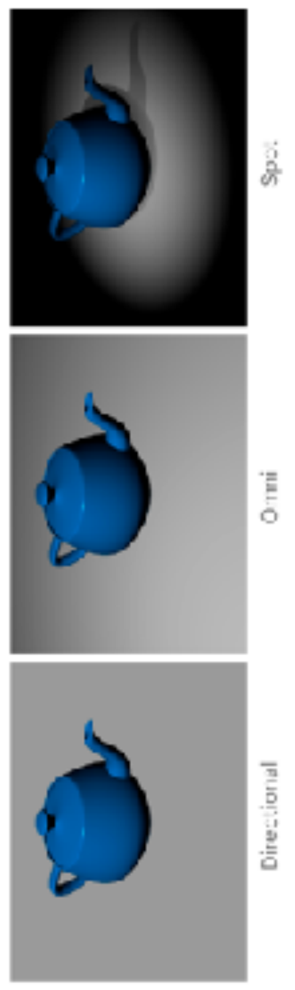
Name
Diffuse
Ambient
Specular
Reflective
…

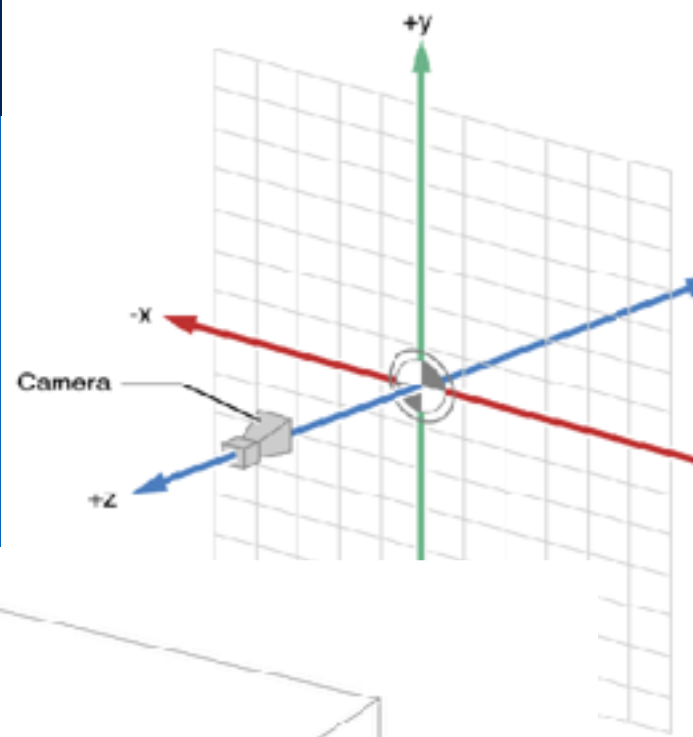
SCNLight

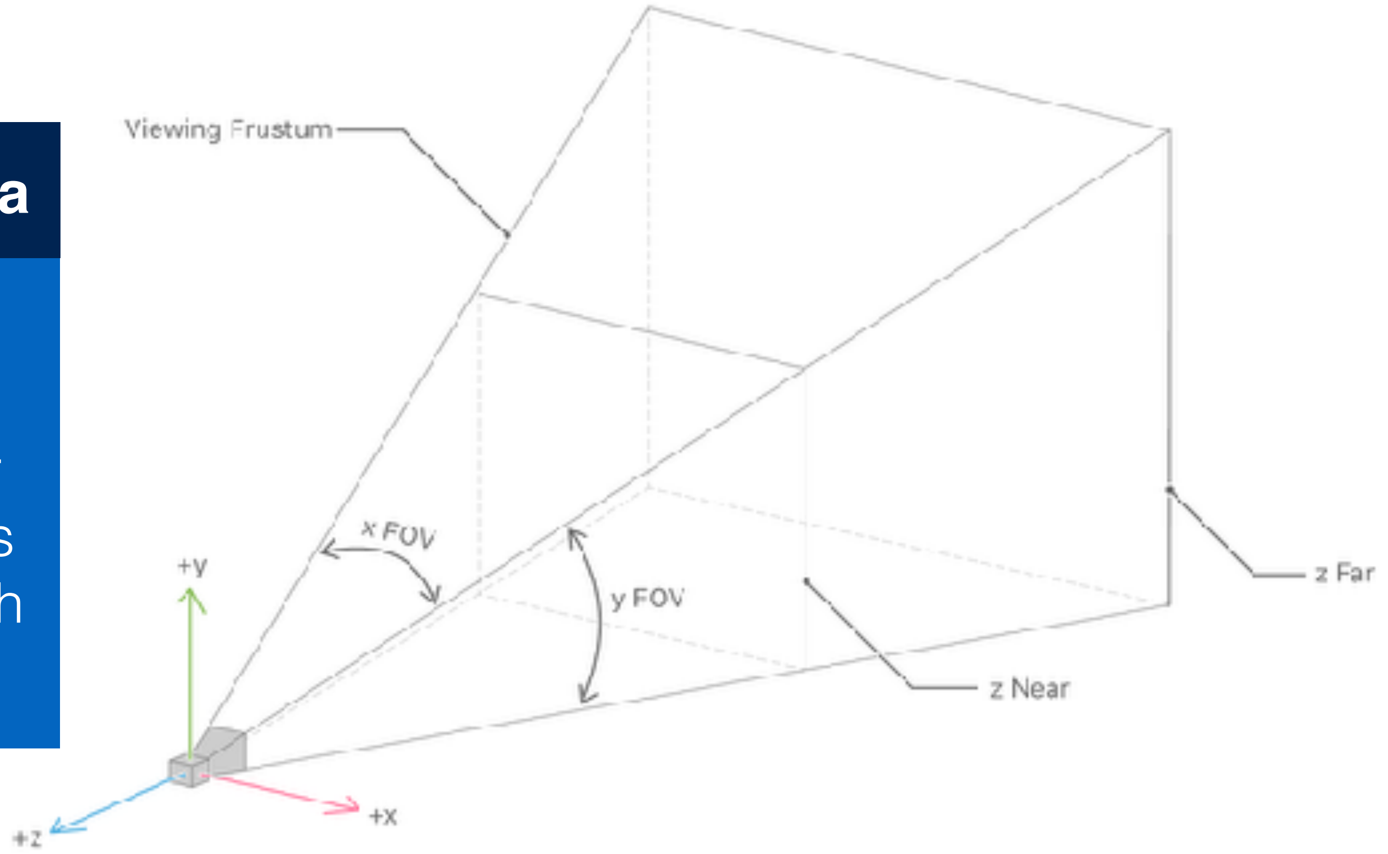
Type (ambient, omni, spot, directional)
Color
Temperature
Intensity
…

SCNCamera

Location
FoV
zNear / zFar
Euler Angles
Focal Length
…







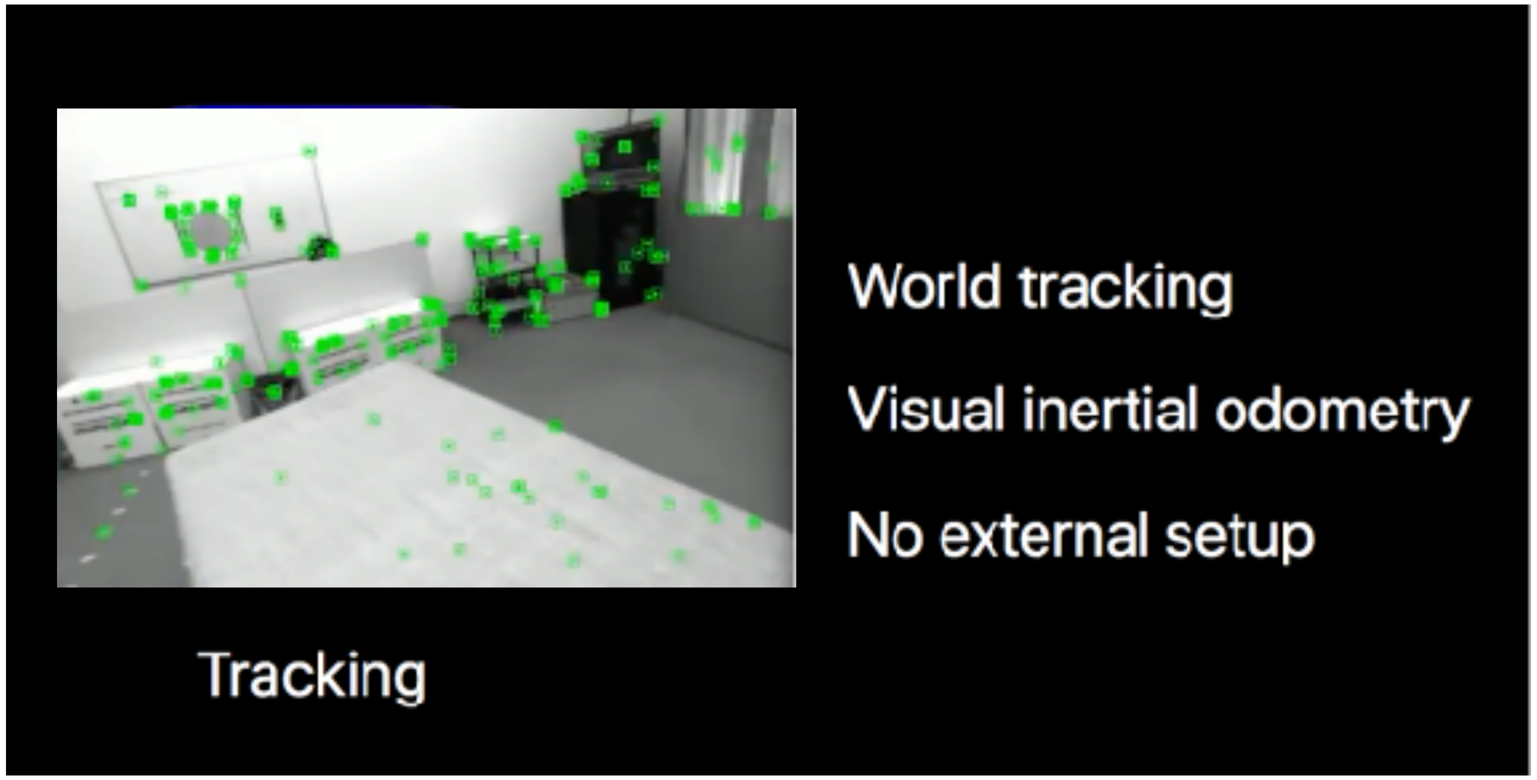
SCNNode is the base for nearly everything in simulation env.

but… we want the SCNScene
to be the real world
ARKit anchors the SCNScene to objects in the immediate environment

```swift
// Setup view
let view = self.view as SCNView
view.scene = scene
```

```swift
// Setup view
let view = self.view as ARSCNView
view.scene = scene
```

ARKit basics

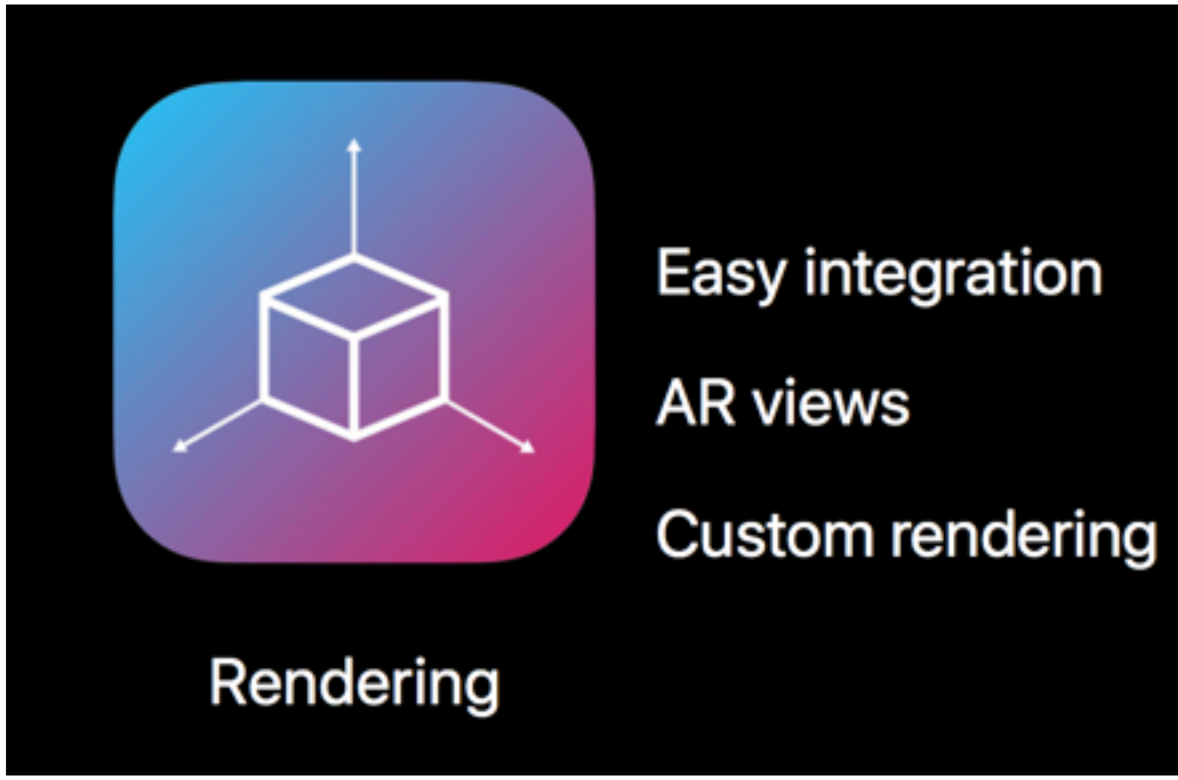


World tracking

Visual inertial odometry

No external setup

Tracking

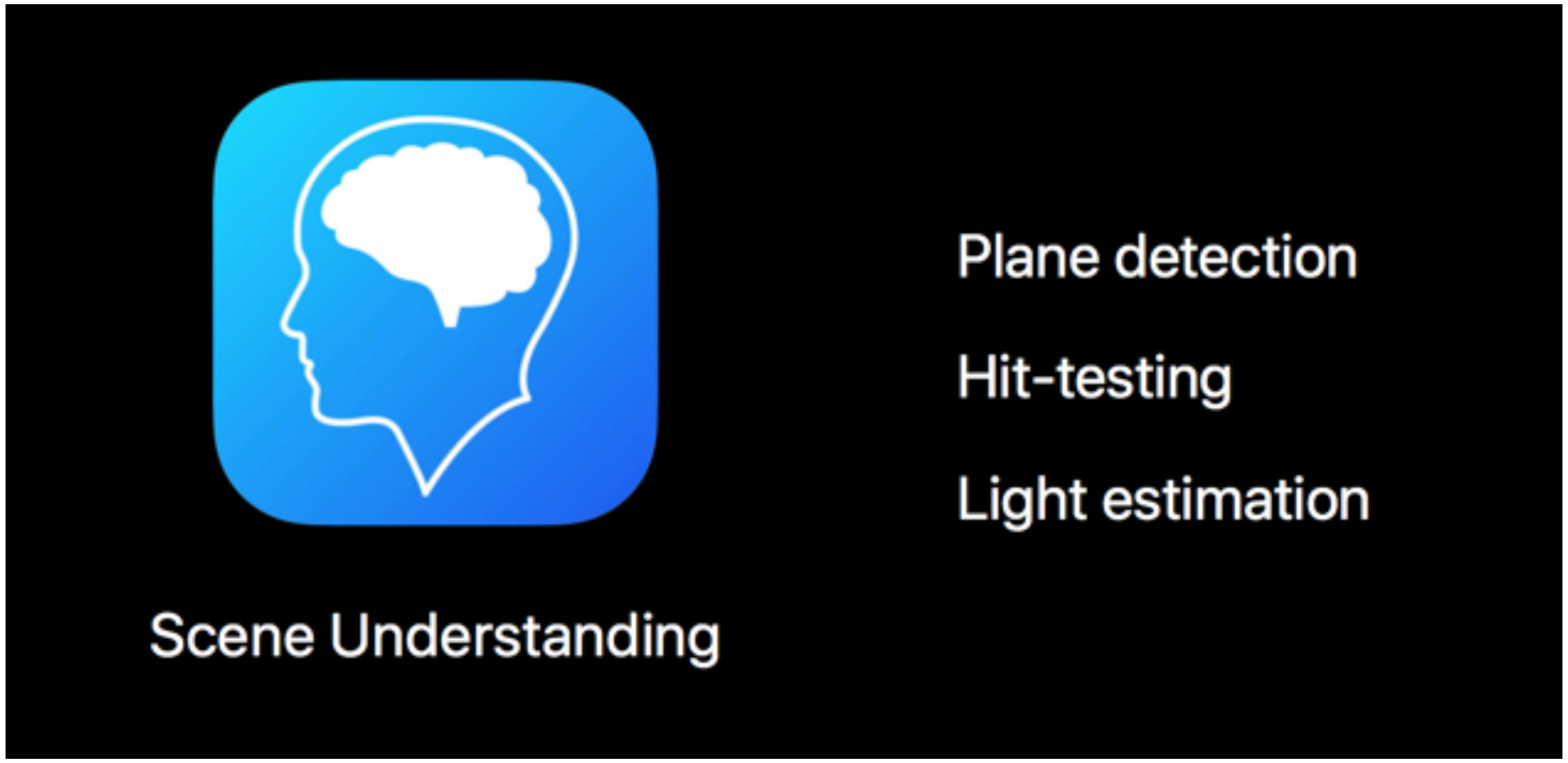


Easy integration

AR views

Custom rendering

Rendering



Plane detection

Hit-testing

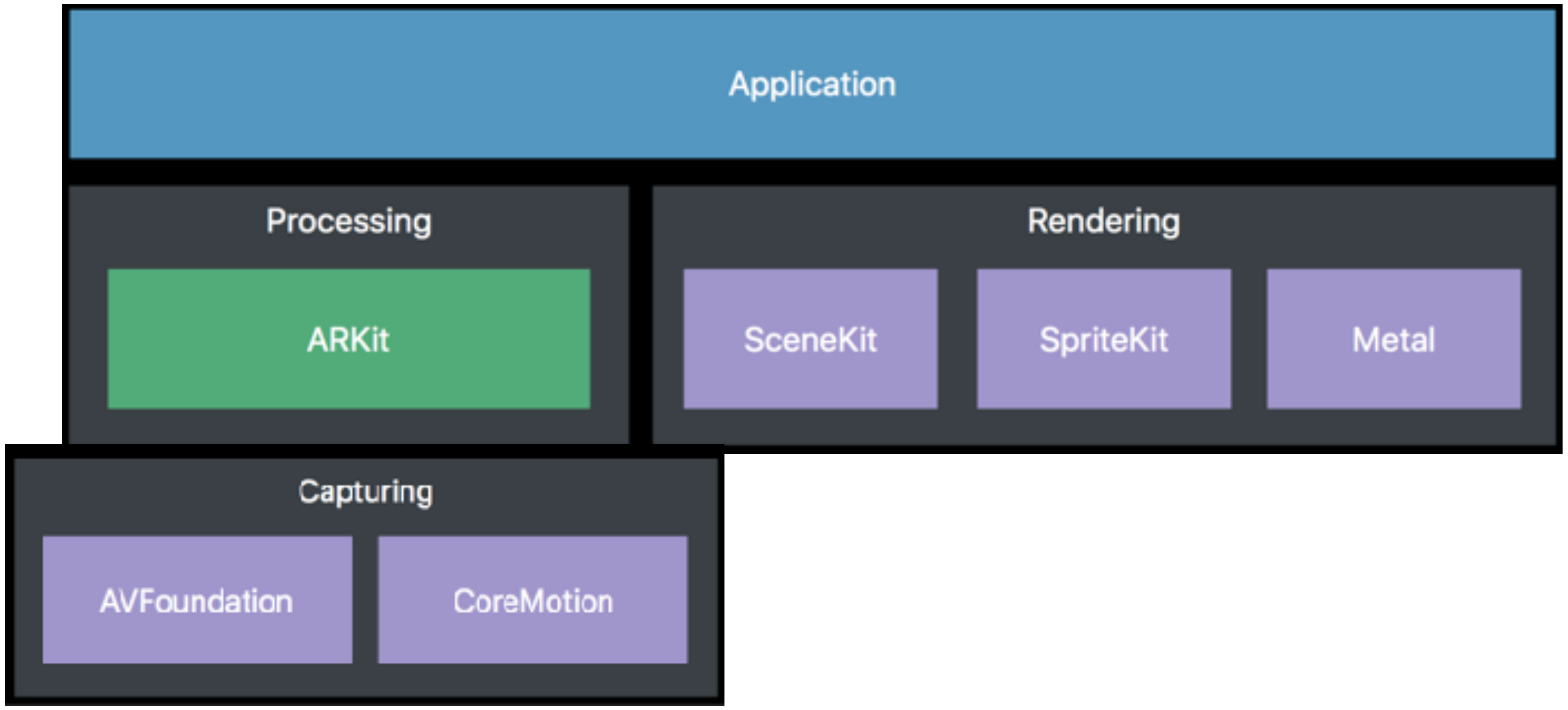
Light estimation

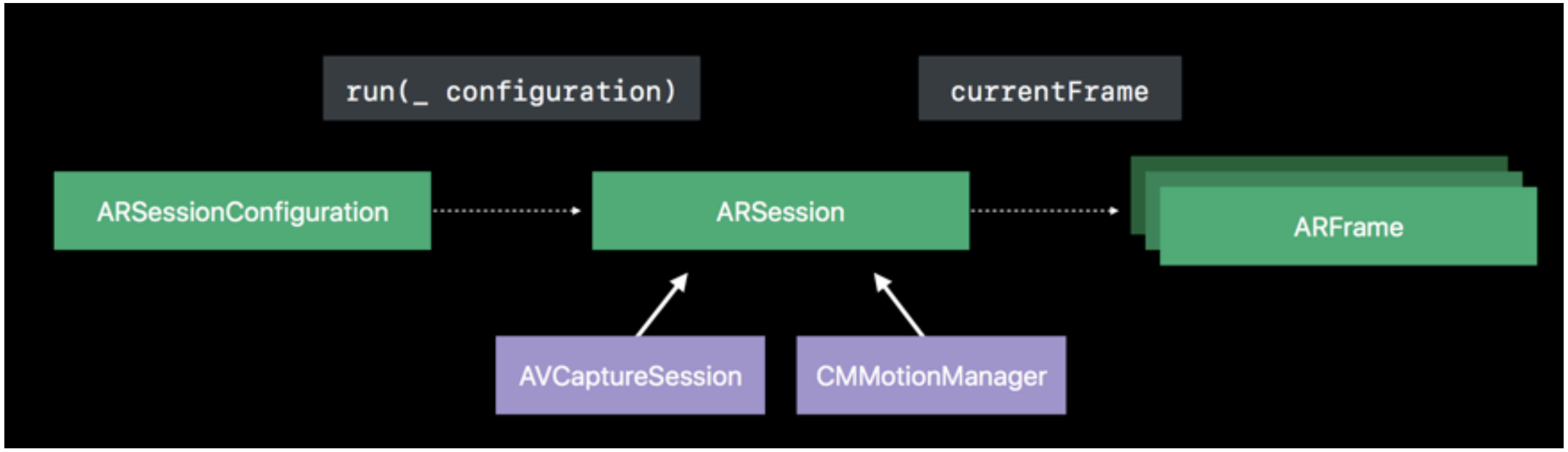
Scene Understanding

https://developer.apple.com/videos/play/wwdc2017/602/

ARKit system integration

https://developer.apple.com/videos/play/wwdc2017/602/





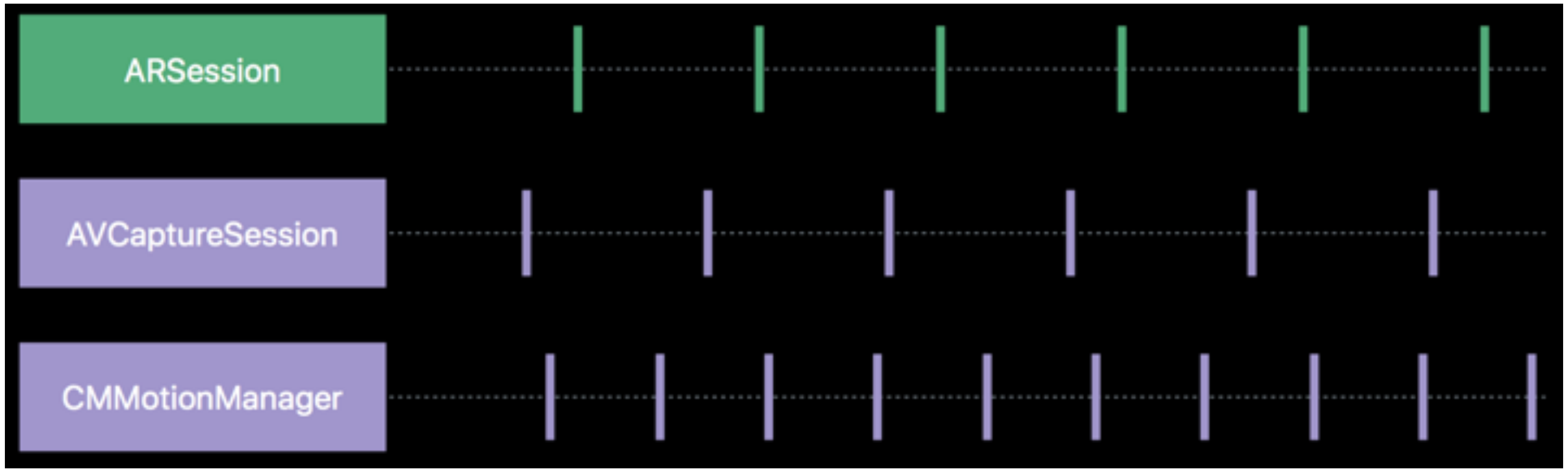
session basics

```
let session = ARSession()
let conf = ARWorldTrackingSessionConfiguration()
session.run(conf)

// Run your session
session.run(configuration)

// Pause your session
session.pause()

// Resume your session
session.run(session.configuration)

// Change your configuration
session.run(otherConfiguration)
```

session basics

```swift
// Access the latest frame
func session(_: ARSession, didUpdate: ARFrame)

// Handle session errors
func session(_: ARSession, didFailWithError: Error)
```

delegation



unified sampling

https://developer.apple.com/videos/play/wwdc2017/602/

adding to the AR world

poll ARSession

create a plane

add to world

translate

```
// grab the current AR session frame from the scene, if possible
guard let currentFrame = sceneView.session.currentFrame else {
    return
}

// setup some geometry for a simple plane
let imagePlane = SCNPlane(width:sceneView.bounds.width/6000,
                          height:sceneView.bounds.height/6000)

// add the node to the scene
let planeNode = SCNNode(geometry:imagePlane)
sceneView.scene.rootNode.addChildNode(planeNode)

// update the node to be a bit in front of the camera inside the AR session


// step one create a translation transform
var translation = matrix_identity_float4x4
translation.columns.3.z = -0.1

// step two, apply translation relative to camera for the node
planeNode.simdTransform = matrix_multiply(currentFrame.camera.transform, translation )
```

operations

Figure 1-8 Matrix configurations for common transformations

$$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

Identity

$$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\tx&ty&tz&1\end{bmatrix}$$

Translate

$$\begin{bmatrix}sx&0&0&0\\0&sy&0&0\\0&0&sz&0\\0&0&0&1\end{bmatrix}$$

Scale

$$\begin{bmatrix}1&0&0&0\\0&\cos\theta&\sin\theta&0\\0&-\sin\theta&\cos\theta&0\\0&0&0&1\end{bmatrix}$$

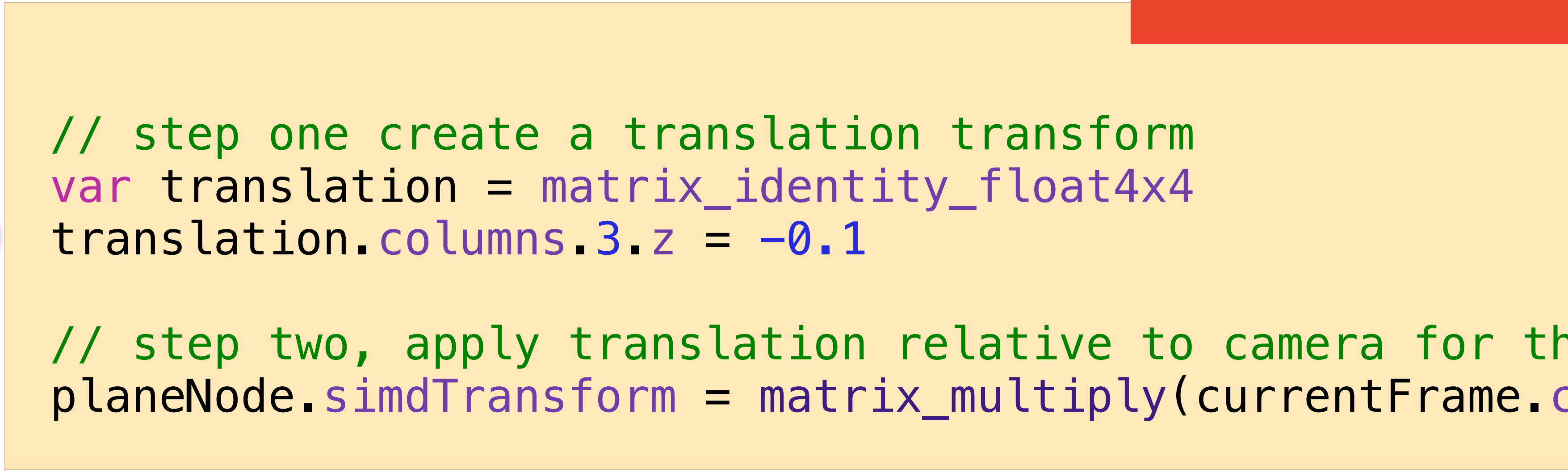
Rotate around X axis

$$\begin{bmatrix}\cos\theta&0&-\sin\theta&0\\0&1&0&0\\\sin\theta&0&\cos\theta&0\\0&0&0&1\end{bmatrix}$$

Rotate around Y axis

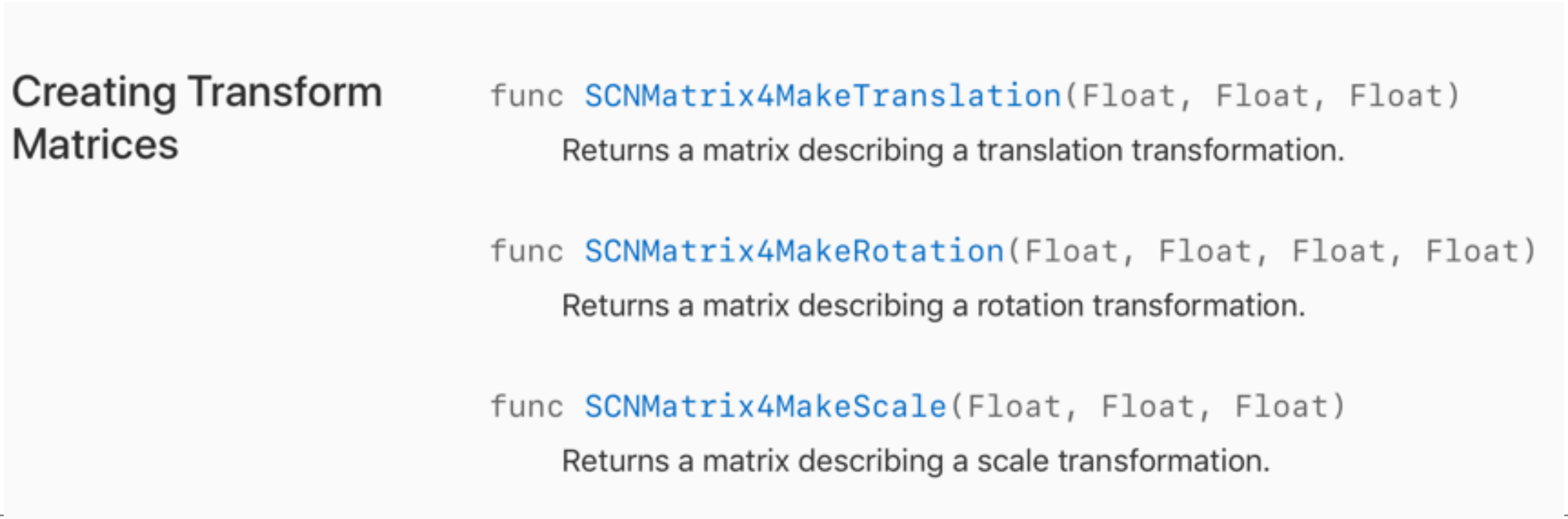
$$\begin{bmatrix}\cos\theta&\sin\theta&0&0\\-\sin\theta&\cos\theta&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

Rotate around Z axis



```
// step one create a translation transform
var translation = matrix_identity_float4x4
translation.columns.3.z = -0.1

// step two, apply translation relative to camera for th
planeNode.simdTransform = matrix_multiply(currentFrame.c
```

SIMD: single instruction, multiple data



Creating Transform Matrices

`func SCNMatrix4MakeTranslation(Float, Float, Float)`

Returns a matrix describing a translation transformation.

`func SCNMatrix4MakeRotation(Float, Float, Float, Float)`

Returns a matrix describing a rotation transformation.

`func SCNMatrix4MakeScale(Float, Float, Float)`

Returns a matrix describing a scale transformation.

ARKit and SceneKit

- extended demo

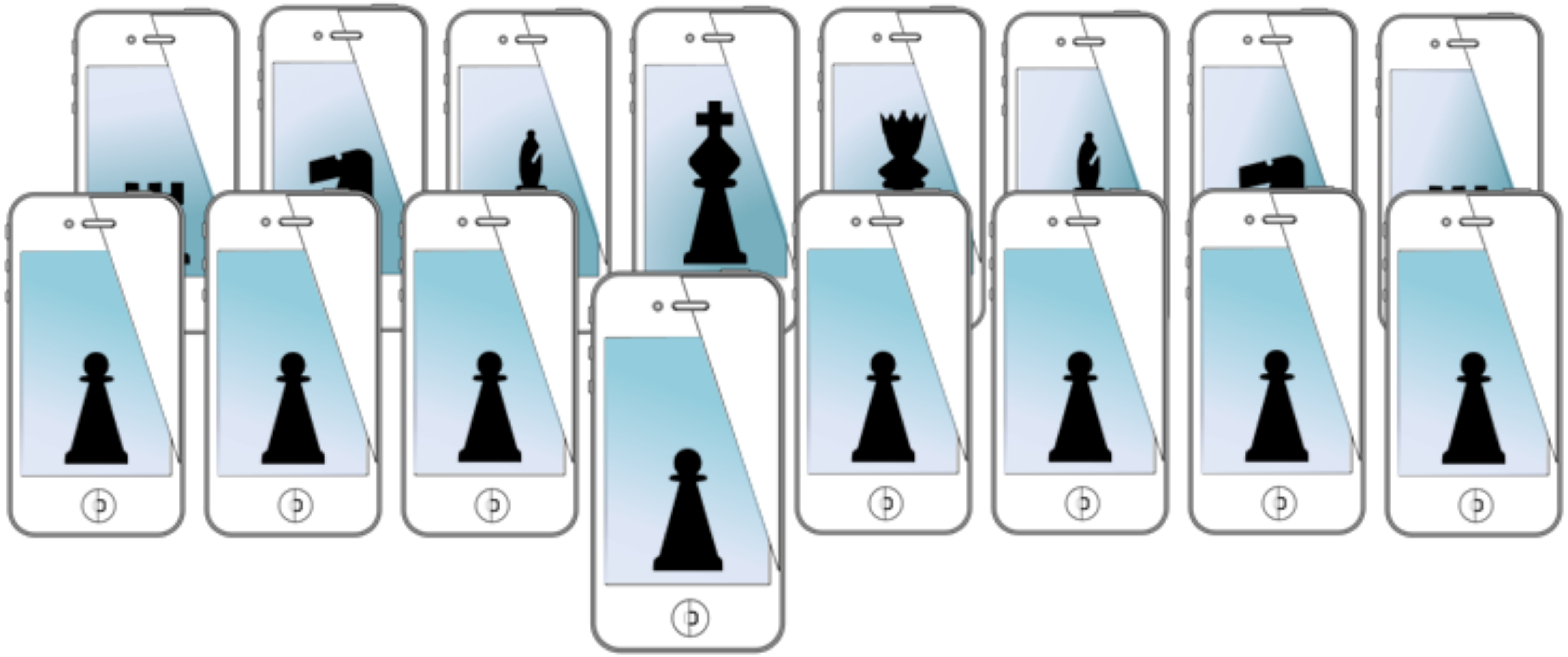
branches: (1) bare implementation,
(2) image extension, (3) stylization

and now its time for a demo

for next time…

- ARKit with Object Recognition
- Speech Recognition
- Pitching (Required)
- ~Fin~

MOBILE SENSING LEARNING



CS5323 & 7323

Mobile Sensing and Learning

ARKit and SceneKit

Eric C. Larson, Lyle School of Engineering, Computer Science, Southern Methodist University