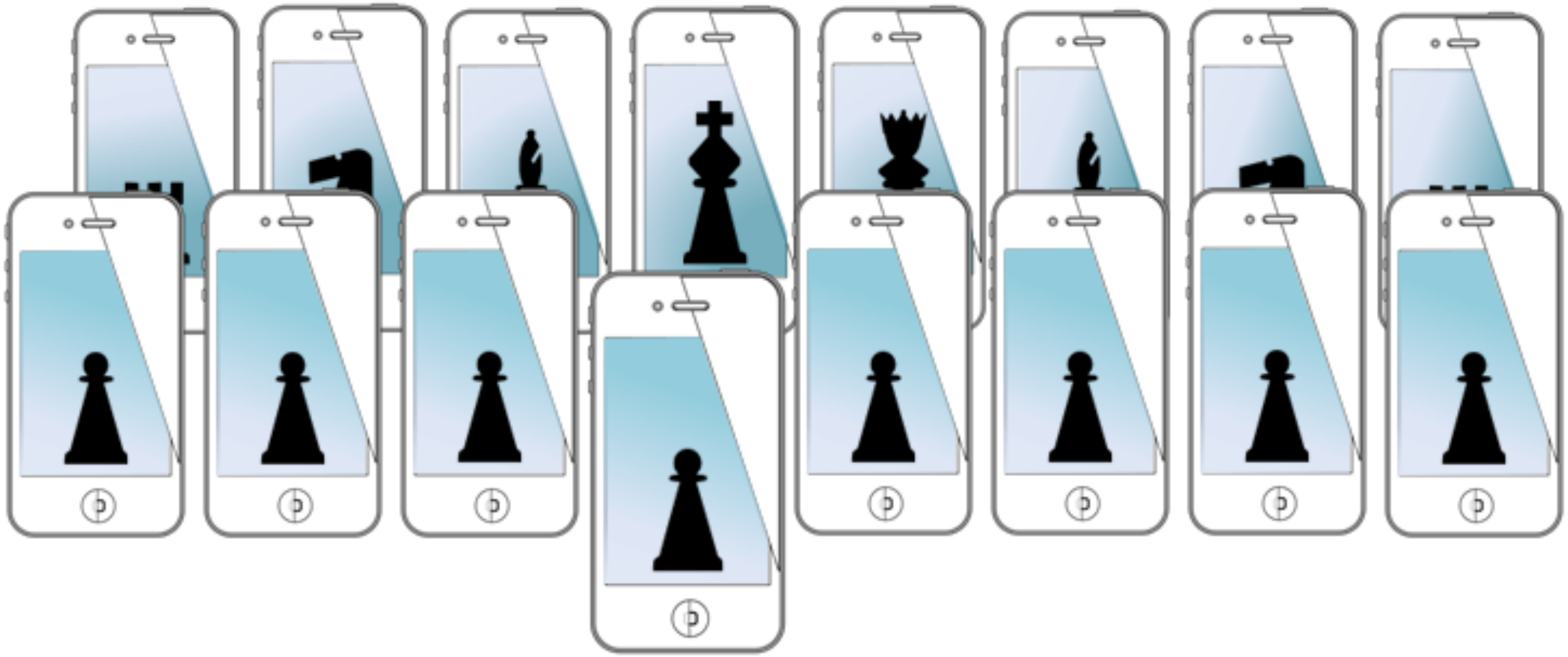
MOBILE SENSING LEARNING



CS5323 & 7323

Mobile Sensing and Learning

SceneKit Demonstration

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

course logistics/agenda

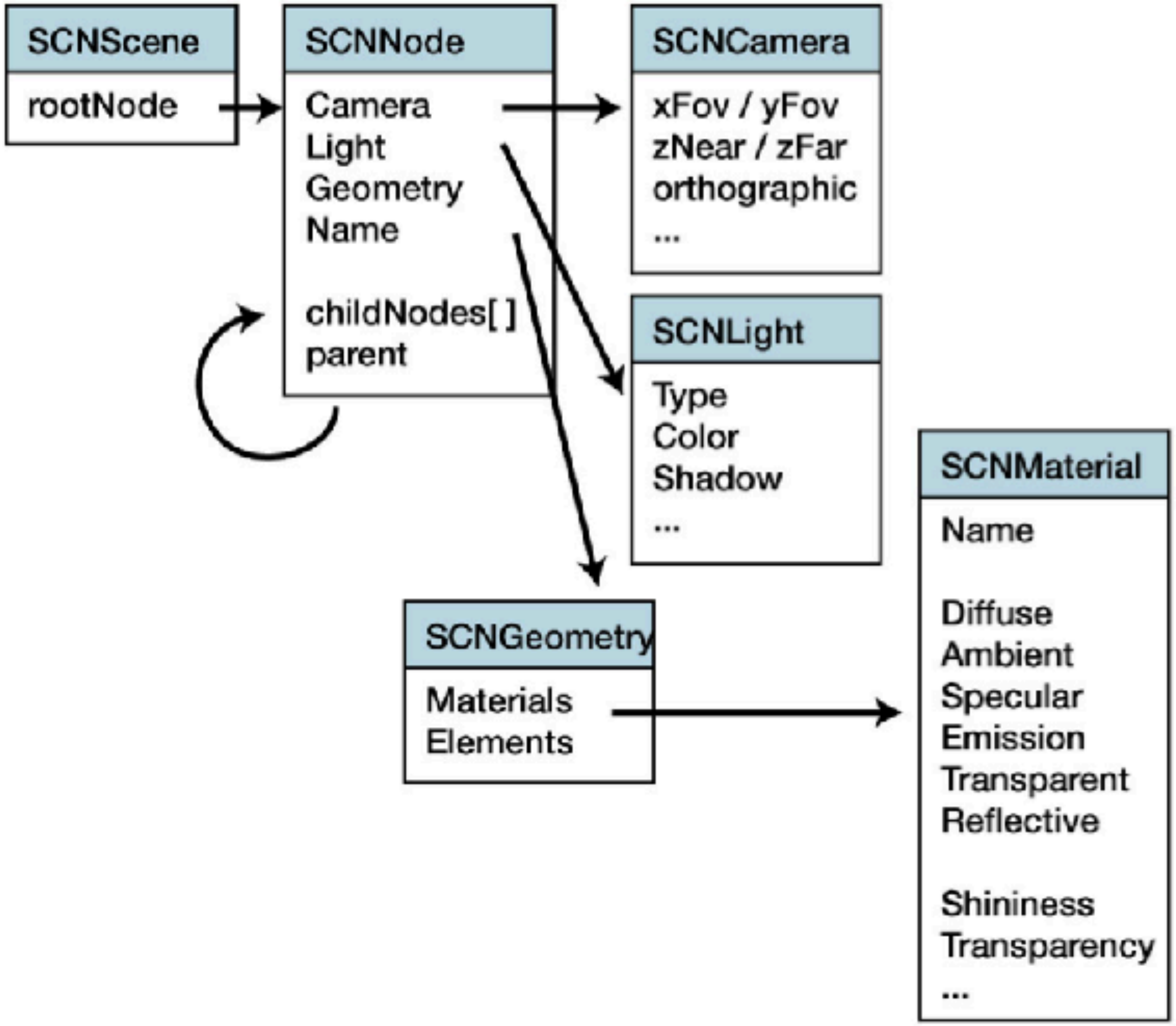
- logistics:
    - A2 due soon!
- today:
    - Finish SceneKit (if needed)
    - image processing basics
    - core image filtering

SceneKit: 3D scenes

- 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

Camera
Light
Geometry
Name

childNodes[ ]
parent

SCNCamera

xFov / yFov
zNear / zFar
orthographic
...

SCNLight

Type
Color
Shadow
...

SCNGeometry

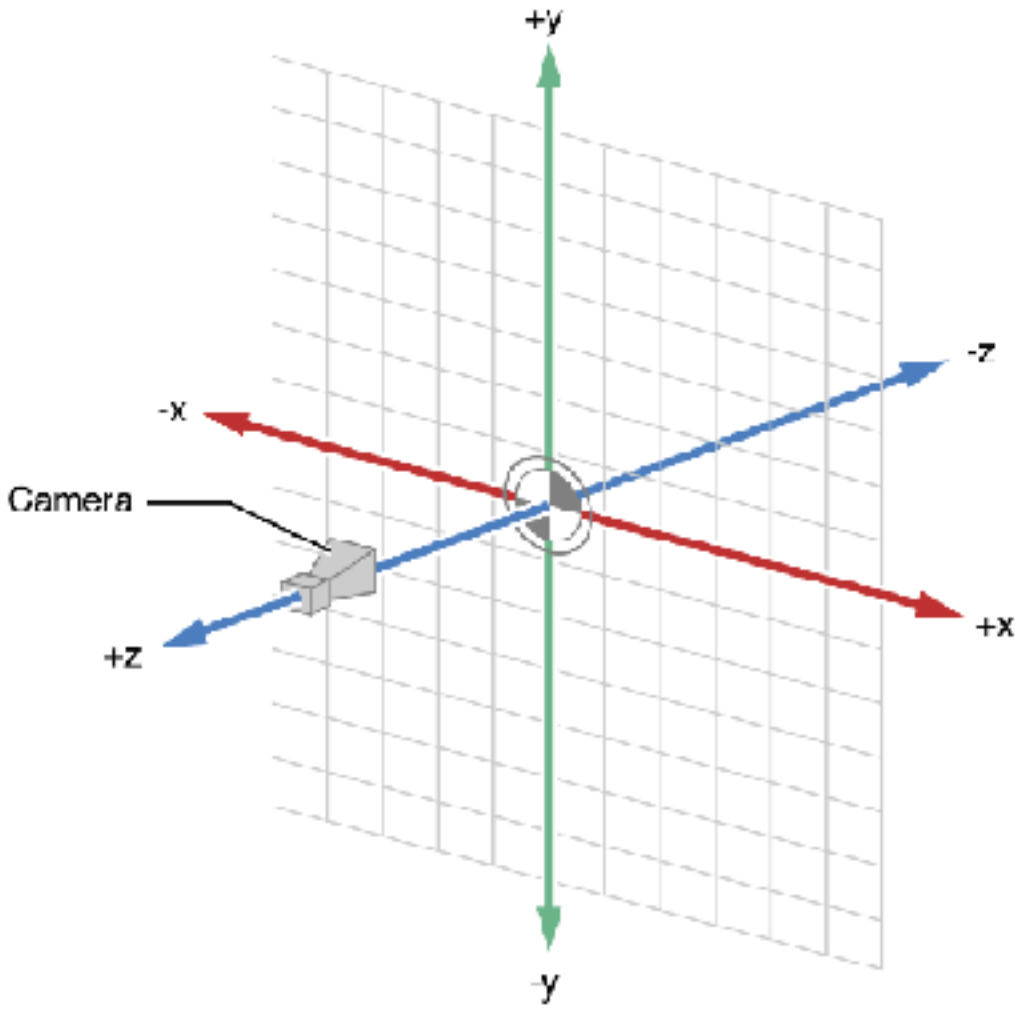
Materials
Elements

SCNMaterial

Name

Diffuse
Ambient
Specular
Emission
Transparent
Reflective

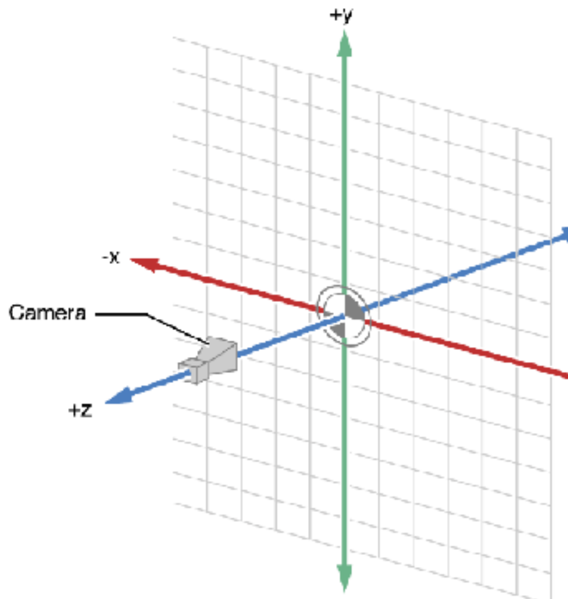
Shininess
Transparency
...



setting up a world

```
// Setup scene
scene = SCNScene()
scene.physicsWorld.speed = 1

// Setup camera position
cameraNode = SCNNode()
cameraNode.camera = SCNCamera()
cameraNode.position = SCNVector3(x: 0, y: 0, z: 30)
scene.rootNode.addChildNode(cameraNode)

// add a plane to the view that users must bounce the ball on
//setup the geometry of node (as a plane)
let wall = SCNPlane(width: 10.0, height: 10.0)
wall.firstMaterial?.doubleSided = true
wall.firstMaterial?.diffuse.contents = UIColor.redColor() // make it red!!

// add the plane to the world as a static body (no dynamic physics)
wallNode = SCNNode()
wallNode.geometry = wall
wallNode.physicsBody = SCNPhysicsBody.staticBody()
wallNode.position = SCNVector3(x: 0.0, y: 0.0, z: -5)

scene.rootNode.addChildNode(wallNode)

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

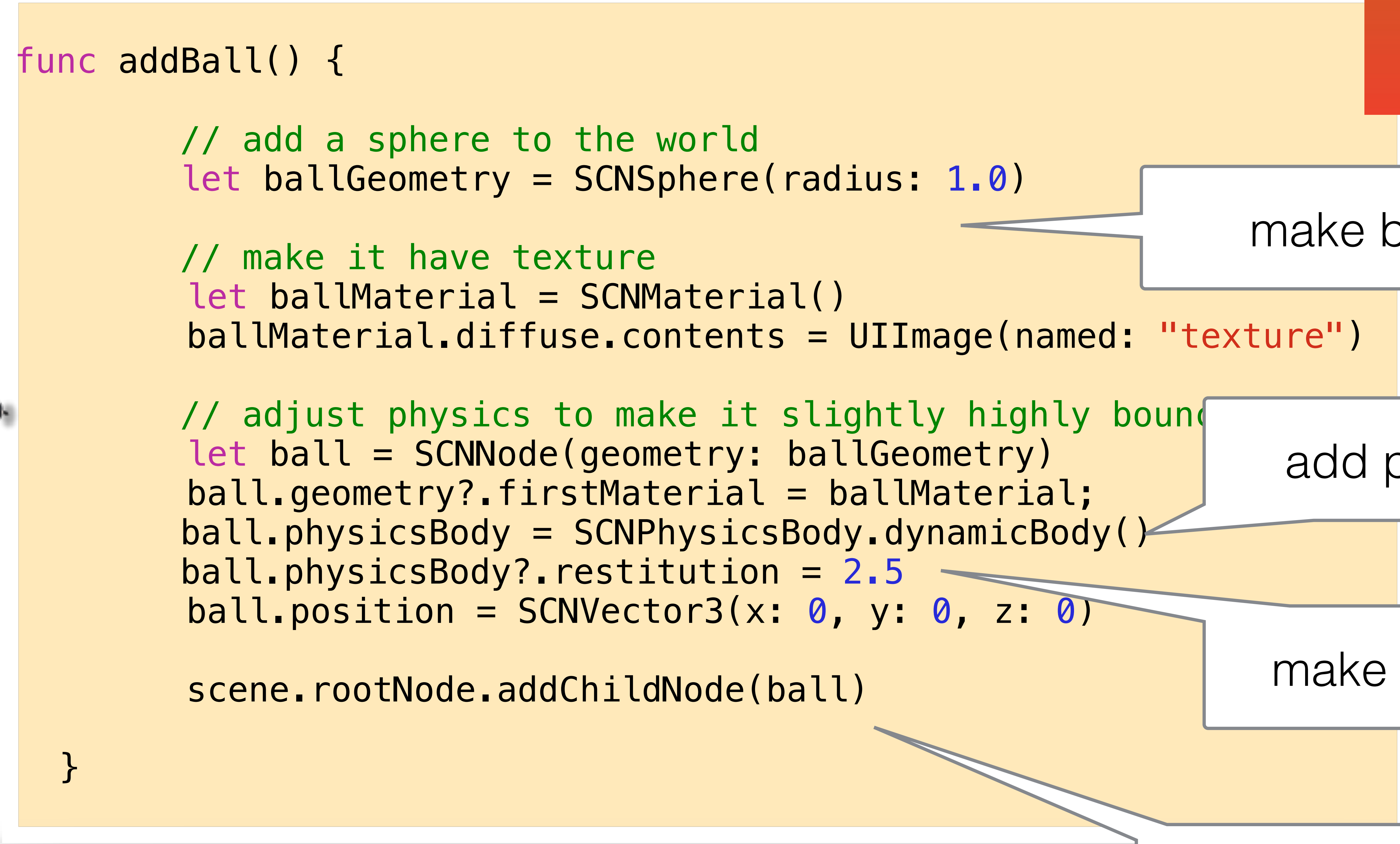
add camera



add plane

make this scene the world

add to world



```
func addBall() {

        // add a sphere to the world
        let ballGeometry = SCNSphere(radius: 1.0)

        // make it have texture
        let ballMaterial = SCNMaterial()
        ballMaterial.diffuse.contents = UIImage(named: "texture")

        // adjust physics to make it slightly highly bounc
        let ball = SCNNode(geometry: ballGeometry)
        ball.geometry?.firstMaterial = ballMaterial;
        ball.physicsBody = SCNPhysicsBody.dynamicBody()
        ball.physicsBody?.restitution = 2.5
        ball.position = SCNVector3(x: 0, y: 0, z: 0)

        scene.rootNode.addChildNode(ball)

  }
```

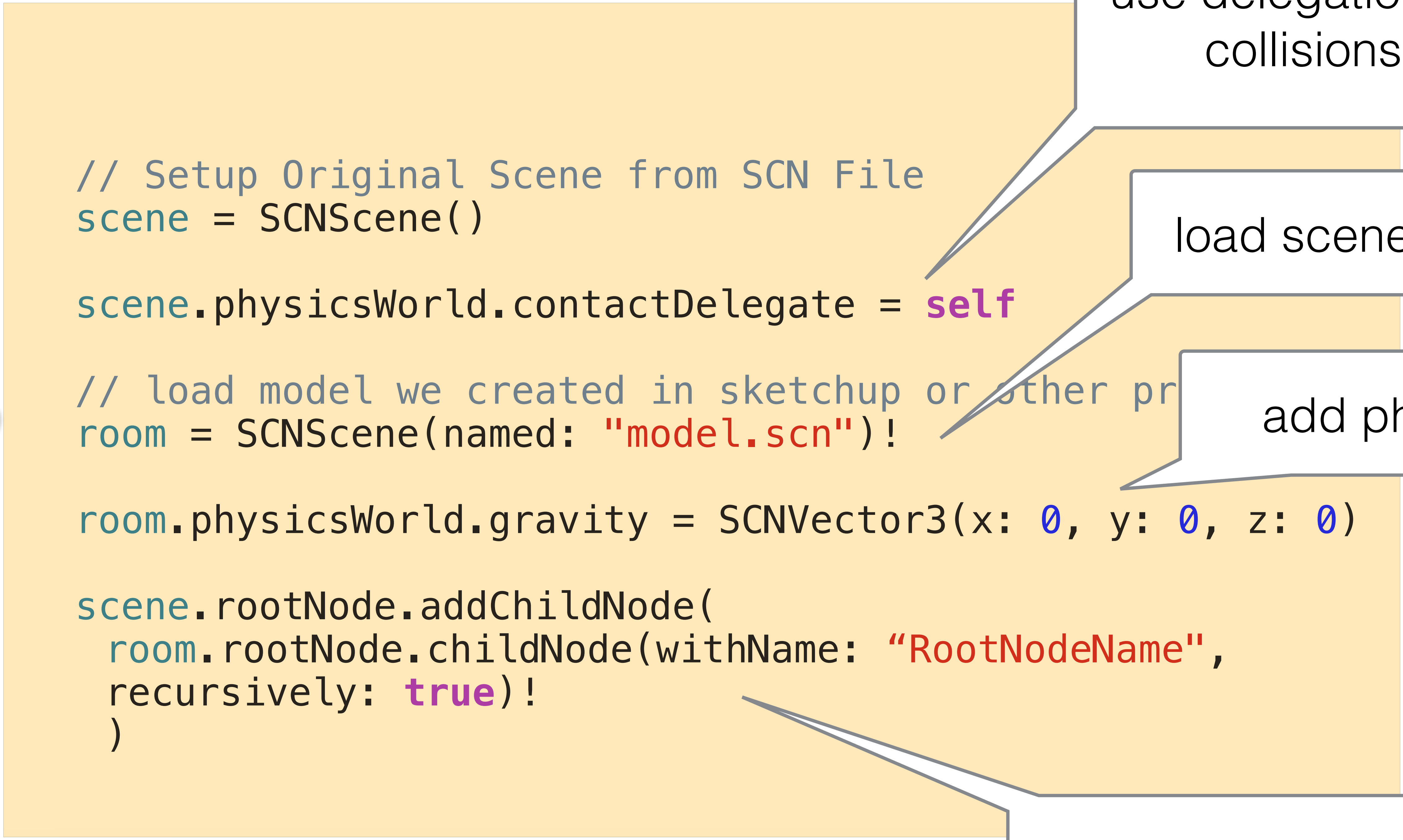
make ball

add physics

make bouncy

add to world

pre-made scenes



```
// Setup Original Scene from SCN File
scene = SCNScene()

scene.physicsWorld.contactDelegate = self

// load model we created in sketchup or other pr
room = SCNScene(named: "model.scn")!

room.physicsWorld.gravity = SCNVector3(x: 0, y: 0, z: 0)

scene.rootNode.addChildNode(
  room.rootNode.childNode(withName: “RootNodeName",
  recursively: true)!
  )
```

use delegation for collisions

load scene model

add physics

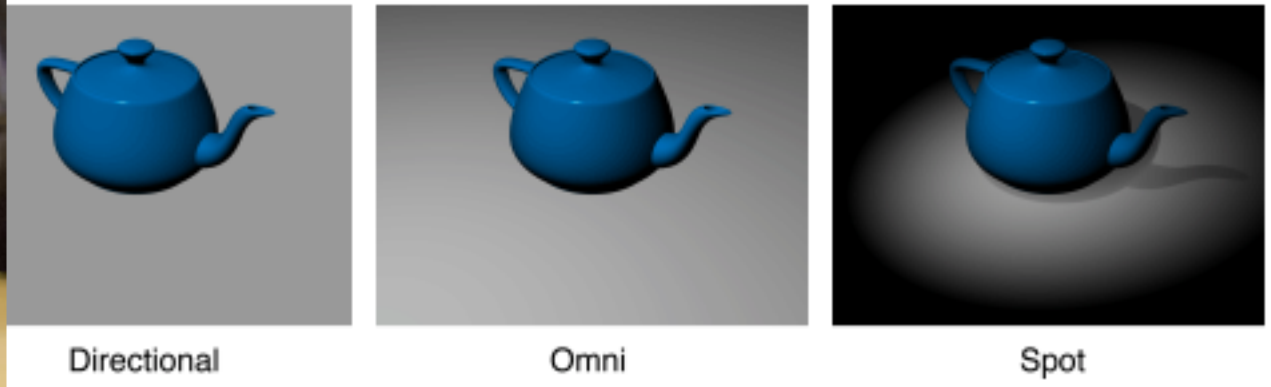
add node from model into scene

physics in world

```
motionManager.startDeviceMotionUpdatesToQueue(
   NSOperationQueue.currentQueue())
        {
            (deviceMotion, error) -> Void in

            let accel = deviceMotion.gravity
            let userAccel = deviceMotion.userAcceleration

            let accelX = Float(9.8 * accel.x + userAccel.x*9.8)
            let accelY = Float(9.8 * accel.y + userAccel.y*9.8)
            let accelZ = Float(9.8 * accel.z + userAccel.z*9.8)

            self.scene.physicsWorld.gravity =
                   SCNVector3(x: accelX, y: accelY, z: accelZ)

        }
```

similar to SpriteKit
but in three dimensions!!

When we move to **Augmented Reality**, SceneKit is the engine for adding **Virtual Elements** to the Actual World!



device motion demo 3

- SceneKit VR
- Scavenger!
- Hockey!