SceneKit: What’s New

Session 604

Thomas Goossens, SceneKit engineer
Amaury Balliet, SceneKit engineer
Anatole Duprat, SceneKit engineer
Sébastien Métrot, SceneKit engineer
SceneKit

High level API for 3D graphics

Supported on every Apple platform
## Related Sessions

<table>
<thead>
<tr>
<th>Session</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>What's New in SceneKit</td>
<td>WWDC 2014</td>
</tr>
<tr>
<td>Enhancements to SceneKit</td>
<td>WWDC 2015</td>
</tr>
<tr>
<td>Advances in SceneKit Rendering</td>
<td>WWDC 2016</td>
</tr>
</tbody>
</table>
Camera enhancements
Camera enhancements

Tessellation and subdivision surfaces
Camera enhancements
Tessellation and subdivision surfaces
Animation improvements
Camera enhancements
Tessellation and subdivision surfaces
Animation improvements
Developer tools
Camera enhancements
Tessellation and subdivision surfaces
Animation improvements
Developer tools
Related technologies
Demo
Camera Enhancements
Camera Enhancements

Physically Based Camera API

Depth of field

Object motion blur

Screen space ambient occlusion

Camera control
// Legacy Camera API

// Move away from legacy projection model
camera.yFov = 60.0
camera.xFov = 60.0
//Physically Based Camera

// configure field of view...
camera.fieldOfView = 60.0 //degrees

// ...or focal
camera.focalLength = 50.0 //mm
camera.sensorHeight = 24.0 //mm
Physically Based Camera
Depth of field

Approximate physical photo camera

// configure the depth of field
camera.wantsDepthOfField = true
camera.focusDistance = 0.8 //meters
camera.fStop = 5.6
Physically Based Camera

Bokeh

Automatic bokeh

Works best with HDR camera

```javascript
// configure the camera for HDR
camera.wantsHDR = true
```
Physically Based Camera
Bokeh

Automatic bokeh

Works best with HDR camera

```javascript
// configure the camera for HDR
camera.wantsHDR = true
```
Physically Based Camera

Bokeh

// configure the depth of field
camera.apertureBladeCount = 5
Physically Based Camera

Bokeh

// configure the depth of field
camera.apertureBladeCount = 5
// Activate Motion Blur
camera.motionBlurIntensity = 1.0
Object Motion Blur
Object Motion Blur
Ambient Occlusion
Ambient Occlusion
Principle

NEW
Ambient Occlusion

Principle

45%
Ambient Occlusion
Screen space ambient occlusion

depth

normal
Ambient Occlusion
Screen space ambient occlusion
Ambient Occlusion
Screen space ambient occlusion
// Ambient Occlusion
// Screen space ambient occlusion

// Activate SSAO
camera.screenSpaceAmbientOcclusionIntensity = 1.0

// Configure SSAO
camera.screenSpaceAmbientOcclusionRadius = 5 // scene units
camera.screenSpaceAmbientOcclusionBias = 0.03 // scene units
camera.screenSpaceAmbientOcclusionDepthThreshold = 0.2 // scene units
camera.screenSpaceAmbientOcclusionNormalThreshold = 0.3
Demo

Anatole Duprat, SceneKit engineer
Camera Control

Thomas Goossens, SceneKit engineer
Camera Control
Common challenges

Object inspection and scene browsing

Camera with a behavior
Camera Control
Object inspection and scene browsing

Previously
• Directly manipulate the camera position, rotation or transform
• For debugging use \texttt{allowsCameraControl} API
Camera Control
Object inspection and scene browsing

Introducing `SCNCameraController`

Built-in support for common controls modes

Default camera controller provided by `SCNView`

// turn on camera control
scnView.allowsCameraControl = true

// configure the camera control behaviour
scnView.defaultCameraController.interactionMode = .orbitTurntable
scnView.defaultCameraController.inertiaEnabled = true
scnView.defaultCameraController.maximumVerticalAngle = 45 // degrees
Camera Control
SCNCameraController

- Orbit Turntable
- Orbit Arcball
- Orbit Angle Mapping
- Fly
- Pan
- Roll
- Frame nodes
- Dolly
- Inertia
- Truck
Camera Control
SCNCameraController examples

Orbit Turntable
Camera Control
SCNCameraController examples

Orbit Arcball
Camera Control
SCNCameraController examples

Fly
Camera Control

Common challenges

Object inspection and scene browsing

Camera with a behavior
Camera Control

Constraints

Chain SCNConstraint objects to define a camera behavior
Camera Control

Constraints

Chain **SCNConstraint** objects to define a camera behavior

- SCNLookAtConstraint
- SCNTransformConstraint
- SCNBillboardConstraint
- SCNIKConstraint
Camera Control

Constraints

Chain `SCNConstraint` objects to define a camera behavior

- SCNLookAtConstraint
- SCNTransformConstraint
- SCNBillboardConstraint
- SCNIKConstraint
- SCNDistanceConstraint
- SCNReplicatorConstraint
- SCNAccelerationConstraint
- SCNSliderConstraint
- SCNAvoidOccluderConstraint
Camera Control

Constraints

SCNDistanceConstraint

- TargetNode
- MinDistance
- MaxDistance
Camera Control
Constraints

SCNReplicatorConstraint

TargetNode

Replicated Position

Offset
Camera Control

Constraints

- SCNAccelerationConstraint
- Last Frame Position
- Resolved Position
- Max Velocity
- Next Frame Target Position

NEW
Camera Control
Building a behavior from constraints
Camera Control
Building a behavior from constraints

SCNLookAtConstraint
Camera Control
Building a behavior from constraints

SCNReplicatorConstraint

SCNLookAtConstraint
Camera Control
Building a behavior from constraints

SCNReplicatorConstraint

SCNLookAtConstraint
Camera Control
Building a behavior from constraints

- SCNDistanceConstraint
- SCNLookAtConstraint
- SCNAccelerationConstraint
Camera Control
In the «Fox 2» demo
Camera Control
Node manipulation helpers

Math helpers on `SCNNode` for common transformations

Vector conversion from node to node

`SCNNode` transforms properties available as SIMD types

// Working with SIMD
aNode.simdPosition += aNode.simdTransform * anotherNode.simdPosition
Tessellation and Subdivision Surfaces

Amaury Balliet, SceneKit engineer
Tessellation and Subdivision Surfaces

How tessellation works

New features based on tessellation

Subdivision surfaces
Tessellation
Motivation

Asset comes as a low-resolution model (coarse mesh)

Decreased memory bandwidth

High resolution model generated from the coarser model

• High resolution model not stored in memory
• Generated on-the-fly by the GPU
• Control over the amount of detail generated
Tessellation

Tessellation factors

How Tessellation Works

Start with a patch

Triangle Patch

```c
glPatchParameteri(GL_PATCH_VERTICES, 3);

glDrawArrays(GL_PATCHES, …)
```
Tessellation

Tessellation factors
How Tessellation Works

Set the outer tessellation levels

\[
\begin{align*}
gl_{\text{TessLevelOuter}}[0] &= 2.0 \\
gl_{\text{TessLevelOuter}}[2] &= 2.0 \\
gl_{\text{TessLevelOuter}}[1] &= 2.0
\end{align*}
\]

Tessellation

Tessellation factors
How Tessellation Works

Set the outer tessellation levels

- `gl_TessLevelOuter[0] = 4.0`
- `gl_TessLevelOuter[1] = 2.0`
- `gl_TessLevelOuter[2] = 2.0`

Tessellation

Tessellation factors
How Tessellation Works

Set the outer tessellation levels

\[
\begin{align*}
\text{gl}_\text{TessLevelOuter}[0] &= 6.0 \\
\text{gl}_\text{TessLevelOuter}[1] &= 2.0 \\
\text{gl}_\text{TessLevelOuter}[2] &= 2.0
\end{align*}
\]
How Tessellation Works

Set the outer tessellation levels

```
gl_TessLevelOuter[0] = 6.0
gl_TessLevelOuter[1] = 2.0
gl_TessLevelOuter[2] = 2.0
```

Tessellation

Tessellation factors
How Tessellation Works

Set the outer tessellation levels

```
gl_TessLevelOuter[0] = 6.0

Tessellation factors```
How Tessellation Works

Set the outer tessellation levels

```gl_TessLevelOuter[0] = 6.0
gl_TessLevelOuter[2] = 2.0
gl_TessLevelOuter[1] = 4.0```

Tessellation

Tessellation factors
Tessellation

Tessellation factors
Tessellation

Tessellation factors

How Tessellation Works

Set the outer tessellation levels

```gl
gl_TessLevelOuter[0] = 6.0

gl_TessLevelOuter[2] = 2.0

gl_TessLevelOuter[1] = 4.0
```
How Tessellation Works

Set the outer tessellation levels
- `gl_TessLevelOuter[0] = 6.0`
- `gl_TessLevelOuter[2] = 4.0`
- `gl_TessLevelOuter[1] = 4.0`

Tessellation

Tessellation factors
How Tessellation Works

1. Set the outer tessellation levels:
   - `gl_TessLevelOuter[0]` = 6.0
   - `gl_TessLevelOuter[2]` = 4.0
   - `gl_TessLevelOuter[1]` = 4.0

Tessellation

Tessellation factors
How Tessellation Works

Set the inner tessellation levels

\[
\begin{align*}
\text{gl}_\text{TessLevelOuter}[0] &= 6.0 \\
\text{gl}_\text{TessLevelOuter}[2] &= 4.0 \\
\text{gl}_\text{TessLevelOuter}[1] &= 4.0 \\
\end{align*}
\]

\[
\begin{align*}
\text{gl}_\text{TessLevelInner}[0] &= 2.0 \\
\end{align*}
\]

Tessellation factors
How Tessellation Works

Set the inner tessellation levels
- gl_TessLevelOuter[0] = 6.0
- gl_TessLevelOuter[2] = 4.0
- gl_TessLevelOuter[1] = 4.0
- gl_TessLevelInner[0] = 3.0

Tessellation
Tessellation factors
How Tessellation Works

Set the inner tessellation levels
- \( gl_{-}TessLevelOuter[0] = 6.0 \)
- \( gl_{-}TessLevelOuter[2] = 4.0 \)
- \( gl_{-}TessLevelOuter[1] = 4.0 \)
- \( gl_{-}TessLevelInner[0] = 6.0 \)
Tessellation

Tessellation factors

How Tessellation Works

Evaluate vertex attributes into tessellated primitives

- gl_TessLevelOuter[0] = 6.0
- gl_TessLevelOuter[2] = 4.0
- gl_TessLevelOuter[1] = 4.0
- gl_TessLevelInner[0] = 6.0
let tessellator = SCNGeometryTessellator()
geometry.tessellator = tessellator
Tessellation
Uniform tessellation

```swift
let tessellator = SCNGeometryTessellator()
gometry.tessellator = tessellator

// Uniform tessellation
tessellator.edgeTessellationFactor = 3.0
tessellator.insideTessellationFactor = 3.0
```
Tessellation
Local space tessellation

```swift
let tessellator = SCNGeometryTessellator()
geometry.tessellator = tessellator

// Local space tessellation
tessellator.isAdaptive = true
tessellator.maximumEdgeLength = 0.01 // in local space
```
Tessellation
Screen space tessellation

let tessellator = SCNGeometryTessellator()
geometry.tessellator = tessellator

// Screen space tessellation
tessellator.isAdaptive = true
tessellator.isScreenSpace = true
tessellator.maximumEdgeLength = 50 // pixels
Tessellated Triangle Patch

Tessellation
Tessellation
Tessellation and Subdivision Surfaces

Tessellation overview

New tessellation-based geometry APIs

Subdivision surfaces
Shader Modifiers

Fully supported in tessellation pipeline

Allow for completely custom effects

// Shader modifier for the "geometry" entry point
float3 p = _geometry.position.xyz;
float disp = sin(p.x + 5.0 * scn_frame.time) * cos(p.y + 2.5 * scn_frame.time);
_geometry.position.xyz += _geometry.normal * disp;
Geometry Smoothing

// Geometry smoothing
tessellator.smoothingMode = .pnTriangles
Geometry Smoothing

// Geometry smoothing
tessellator.smoothingMode = .pnTriangles
Displacement Mapping
Displacement Mapping
Height maps
Displacement Mapping

Height maps
Displacement Mapping

Height maps
Displacement Mapping
Height maps
Displacement Mapping

Height maps
// Displacement Mapping

// Height map
material.displacement.contents = "volcano-height-map.png"
material.displacement.textureComponents = .red
Displacement Mapping
Vector displacement maps

NEW
Displacement Mapping
Vector displacement maps
Displacement Mapping
Vector displacement maps
Displacement Mapping

Vector displacement maps
// Displacement Mapping

// Vector displacement map
material.displacement.contents = "water-drop-displacement-map.exr"
material.displacement.textureComponents = .all
Tessellation and Subdivision Surfaces

Tessellation overview

New tessellation-based geometry APIs

Subdivision surfaces
Subdivision Surfaces

Coarse mesh
Subdivision Surfaces

Subdivision level: 1
Subdivision Surfaces

Subdivision level: 2
Subdivision Surfaces

Subdivision level: 3
Subdivision Surfaces

Subdivision level: 4
Subdivision Surfaces

Subdivision level: 5
Subdivision Surfaces
Iterative refinement
Subdivision Surfaces
Edge and vertex sharpness
Subdivision Surfaces

Easier and faster to create by artists

Reduced file sizes and faster load times

Dynamic control on quality of rendered geometry
Subdivision Surfaces

Original support for CPU-based subdivision

Better support for per-face data interpolation (e.g. texture coordinates)

// Enable subdivision surfaces
geometry.subdivisionLevel = 1
OpenSubdiv on Metal
Subdivision Surfaces

Leverages tessellation with Metal-based GPU backend

Both uniform and feature-adaptive subdivision are now supported

Efficient implementation for animated models with skinning and morphing
Subdivision Surfaces

Coarse mesh
Subdivision Surfaces
Refined mesh

NEW
Subdivision Surfaces

Creases

NEW
Subdivision Surfaces
Feature-adaptive and uniform subdivision
Subdivision Surfaces
Feature-adaptive and uniform subdivision
Subdivision Surfaces
Feature-adaptive and uniform subdivision
Subdivision Surfaces
Feature-adaptive and uniform subdivision
Subdivision Surfaces
Feature-adaptive and uniform subdivision
Subdivision Surfaces
Feature-adaptive and uniform subdivision
// Subdivision Surfaces
// Adaptive subdivision on the GPU

// Enable subdivision surfaces
geometry.subdivisionLevel = 1
geometry.wantsAdaptiveSubdivision = true

// Enable tessellation
let tessellator = SCNGeometryTessellator()
geometry.tessellator = tessellator
Subdivision Surfaces
Faster skinning and morphing

Only the coarse mesh is animated

Leverages the GPU for morphing, skinning and refinement
// Subdivision Surfaces
// Asset authoring

// Preserve topology when importing from files
let scene = try! SCNScene(url: url, options: [.preserveOriginalTopology: true])

// Use quads when creating geometries programmatically
let element = SCNGeometryElement(data: elementData,
    primitiveType: .polygon,
    primitiveCount: 1,
    bytesPerIndex: MemoryLayout<UInt8>.size)
Tessellation and Subdivision Surfaces

macOS
All configurations

iOS
A9 chip
Animation Enhancements
Introducing `SCNAnimation` and `SCNAnimationPlayer`

Mutability while playing
- Speed
- Pausing
- Blending

Bridged with Core Animation

Available on all platforms
// Animations
// Old approach

// start walking
character.addAnimation(animation, forKey: "walk")

// later: stop walking, start jumping
character.removeAnimation(forKey: "walk")
character.addAnimation(jumpAnimation, forKey: "jump"
// Animations
// New approach

// configure the players
character.addAnimationPlayer(walkAnimationPlayer, forKey: "walk")
character.addAnimationPlayer(jumpAnimationPlayer, forKey: "jump")

// start walking
walkAnimationPlayer.play()

// later: stop walking, start jumping
walkAnimationPlayer.stop()
jumpAnimationPlayer.play()
// Animations
// New approach

// at any time: mutate the players
walkAnimationPlayer.speed = characterSpeed // walk slower or faster
runAnimationPlayer.blendFactor = walkRunFactor // `walkRunFactor` of walk
    // `1.0 - walkRunFactor` of run
Animations
Animation blending

Step
Walk
Run
Animations
Animation blending

Speed

Blend Factors
Step
Walk
Run
Animations

Performance

Starting, pausing and stopping animations is much faster

Efficient evaluation of skeletal animation
Animations

Performance

Starting, pausing and stopping animations is much faster

Efficient evaluation of skeletal animation

5700 Bones
1.22 ms
Animations
Performance

Starting, pausing and stopping animations is much faster

Efficient evaluation of skeletal animation
Developer Tools

Sébastien Métrot, SceneKit engineer
FPS Gauge

Split per categories

Real-time

Integrated in Xcode

Great for performance overview
SceneKit Instrument

Understand performance issues

Record a trace of SceneKit’s behavior

Accurate per-frame performance analysis
SceneKit Instrument

SceneKit Application Instrument profiles the performances of iOS, tvOS and MacOS SceneKit applications by providing tracing informations from the application.

Open an Existing File...  Cancel  Choose
SceneKit Instrument

NEW
SceneKit Instrument

NEW
SceneKit Instrument

NEW
SceneKit Instrument

Example: Slow shader compilation
SceneKit Instrument

Example: Slow shader compilation
SceneKit Instrument

In combination with the Metal Instrument trace
SceneKit Capture
Debugging with SceneKit
SceneKit Capture
Debugging with SceneKit

View debugger enhancements

Captures the current state of your scene
SceneKit Capture
Debugging with SceneKit

View debugger enhancements

Captures the current state of your scene
SceneKit Capture

View debugger enhancements

Captures the current state of your scene
SceneKit Capture
View debugger enhancements

Captures the current state of your scene
SceneKit Capture
View debugger enhancements

Captures the current state of your scene
SceneKit Scene Editor

New features
SceneKit Scene Editor

New features
SceneKit Scene Editor

New features
SceneKit Scene Editor

New features
SceneKit Scene Editor

New features

New Shader Modifier Editor
Edit shader and material
Supports custom material properties
SceneKit Scene Editor

More features

Displacement

Tessellation

Support for new constraints

Cascaded shadows

Procedural sky

Override Materials for reference nodes
SceneKit Scene Editor

More features

Displacement

Tessellation

Support for new constraints

Cascaded shadows

Procedural sky

Override materials for reference nodes
SceneKit Scene Editor

More features

Displacement

Tessellation

Support for new constraints

Cascaded shadows

Procedural sky

Override materials for reference nodes
SceneKit Scene Editor

More features

Displacement
Tesselllation
Support for new constraints
Cascaded shadows
Procedural sky
Override materials for reference nodes
SceneKit Scene Editor

More features

Displacement

Tessellation

Support for new constraints

Cascaded shadows

Procedural sky

Override materials for reference nodes
SceneKit Scene Editor

More features

**Displacement**

**Tessellation**

**Support for new constraints**

**Cascaded shadows**

**Procedural sky**

**Override materials for reference nodes**
SceneKit Scene Editor

Material overrides

Scene

SCNNodeReference

SCNNodeReference

MaxTheRedPanda.scn
SceneKit Scene Editor

Material overrides

Scene

SCNNodeReference

Override Material

SCNNodeReference

MaxTheRedPanda.scn
Related Technologies

Thomas Goossens, SceneKit engineer
Augmented Reality

ARKit

ARSCNView

‘Out of the box’ solution for AR
Augmented Reality

ARKit

All SceneKit available via ARSCNView

- Scene graph
- Camera
- Post process
- Particles…
Augmented Reality

// Run the AR session
arView-session.run(configuration)

// Called on the view's delegate
// when a new anchor is found by ARKit
func renderer(_ renderer: SCNSceneRenderer,
            didAdd node: SCNNode,
            for anchor: ARAnchor)
{
    // attach a 3D node to the anchor node
    node.addChildNode(maxTheRedPanda)
}
Augmented Reality

ARKit

New support for background video

• Camera feed using `AVCaptureDevice`
• Video using `AVPlayer`
• Automatic with `ARSCNView`

```swift
// Setup background video
let captureDevice: AVCaptureDevice = ...
scene.background.contents = captureDevice
```
Augmented Reality
Drop shadow trick
Augmented Reality
Drop shadow trick
Augmented Reality
Drop shadow trick
Augmented Reality

Drop shadow trick

Material Inspector
Augmented Reality

Drop shadow trick

Material Inspector
Augmented Reality
Drop shadow trick

Light Inspector
Augmented Reality
Drop shadow trick

Light Inspector
Augmented Reality
Drop shadow trick
Related Technologies
GameplayKit entity/components

- GKScene
- GKEntity
- GKComponent

- SCNView
- SCNNode
GameplayKit Entity/Components

Use GameplayKit to drive SceneKit

Integrated in Xcode

<table>
<thead>
<tr>
<th>Component</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>chaseDeceller</td>
<td>-0.25</td>
</tr>
<tr>
<td>chaseDistance</td>
<td>4.8</td>
</tr>
<tr>
<td>chaseSpeed</td>
<td>3</td>
</tr>
<tr>
<td>chaseAcceller</td>
<td>2.16</td>
</tr>
<tr>
<td>wanderSpeed</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Related Technologies
Model I/O

Improved support for USD
Better material bridging
Support for animations

From Art to Engine with Model I/O
Executive Ballroom
Friday 2:50PM
Related Technologies
UIFocus support

SCNNode conforms to UIFocusItem

```swift
node.focusBehavior = .none // or .occluding or .focusable
// in ViewController
override func didUpdateFocus(in context: UIFocusUpdateContext,
    with coordinator: UIFocusAnimationCoordinator) {
    if let node = context.nextFocusedItem as? SCNNode {
        // …
    }
}
```
UIFocus Support
UIFocus Support
UIFocus Support
UIFocus Support
UIFocus Support
UIFocus Support
UIFocus Support
UIFocus Support
UIFocus Support
Rendering Additions
Point Cloud Rendering

geometry.pointSize = size
geometry.minimumPointScreenSpaceRadius = 5.0
geometry.maximumPointScreenSpaceRadius = 1.0
Point Cloud Rendering

```javascript
geometry.pointSize = size
geometry.minimumPointScreenSpaceRadius = 5.0
geometry.maximumPointScreenSpaceRadius = 1.0
```
Transparency Modes
Single layer, dual layer

.default  .singleLayer  .dualLayer
Transparency Modes

Dual layer
Cascaded Shadow Maps

- MaxShadowDistance
- zFar
- cascadeSplittingFactor
- zNear
// Cascaded Shadow Maps
// Cascades

// activate shadow cascade
light.shadowCascadeCount = 4

// configure shadow map sizes and cascade splitting
light.shadowMapSize = CGSize(width: 512, height: 512)
light.shadowCascadeSplittingFactor = 0.25
Cascaded Shadow Maps
Summary
More Information

<table>
<thead>
<tr>
<th>Session</th>
<th>Location</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introducing Metal 2</td>
<td>Executive Ballroom</td>
<td>Tuesday 1:50PM</td>
</tr>
<tr>
<td>Introducing ARKit: Augmented Reality for iOS</td>
<td>Hall 3</td>
<td>Tuesday 5:10PM</td>
</tr>
<tr>
<td>Debugging with Xcode 9</td>
<td>Hall 2</td>
<td>Wednesday 10:00AM</td>
</tr>
<tr>
<td>SceneKit in Swift Playgrounds</td>
<td>Grand Ballroom B</td>
<td>Thursday 9:00AM</td>
</tr>
<tr>
<td>Focus Interaction in tvOS 11</td>
<td>Grand Ballroom A</td>
<td>Thursday 9:00AM</td>
</tr>
<tr>
<td>Metal 2 Optimizing and Debugging</td>
<td>Grand Ballroom B</td>
<td>Thursday 3:10PM</td>
</tr>
<tr>
<td>Going Beyond 2D with SpriteKit</td>
<td>Executive Ballroom</td>
<td>Friday 10:00AM</td>
</tr>
<tr>
<td>From Art to Engine with Model I/O</td>
<td>Executive Ballroom</td>
<td>Friday 2:50PM</td>
</tr>
<tr>
<td>Lab</td>
<td>Location</td>
<td>Time</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>----------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Model I/O and GameplayKit lab</td>
<td>Technology Lab G</td>
<td>Tues 9:00AM–12:00PM</td>
</tr>
<tr>
<td>SceneKit Lab</td>
<td>Technology Lab G</td>
<td>Wed 1:00PM–4:10PM</td>
</tr>
<tr>
<td>ARKit Lab</td>
<td>Technology Lab A</td>
<td>Wed 1:00AM–3:10PM</td>
</tr>
<tr>
<td>ARKit Lab</td>
<td>Technology Lab A</td>
<td>Thur 12:00PM–3:10PM</td>
</tr>
</tbody>
</table>