Sound

Mobile Application Development in iOS

School of EECS

Washington State University

Instructor: Larry Holder
Sound

- **MediaPlayer** framework to play songs
- **AV Foundation** framework to play audio
- **Audio Toolbox** framework to play system audio with synchronization capabilities
- **SpriteKit and SceneKit** audio
Media Player

- Add MediaPlayer framework
- Import MediaPlayer
- Create music player controller
- Add songs to queue based on query
- Play

```swift
#import <MediaPlayer/MediaPlayer.h>
MPMusicPlayerController *myPlayer = [MPMusicPlayerController applicationMusicPlayer];
[myPlayer setQueueWithQuery: [MPMediaQuery songsQuery]];  // all songs
[myPlayer play];
```
AV Foundation

- Add AVFoundation framework
- Import AVFoundation
- Create AVPlayer (strong) from URL (mp3 file)
- Play
- Must recreate AVPlayer each time

```cpp
#import <AVFoundation/AVFoundation.h>
@property (strong) AVPlayer* boingPlayer;
_boingPlayer = [AVPlayer playerWithURL:
    [[NSBundle mainBundle] URLForResource: @"boing" withExtension:@"mp3"]];
[_boingPlayer play];
```
Audio Toolbox

- Add AudioToolbox framework
- Import AudioToolbox
- Create CFURLRef to sound file
- Create sound id using AudioServicesCreateSystemSoundID
- Play using AudioServicesPlaySystemSound

Constraints:
- No longer than 30 seconds in duration
- In linear PCM or IMA4 (IMA/ADPCM) format
- Packaged in a .caf, .aif, or .wav file

```c
CFBundleRef mainBundle = CFBundleGetMainBundle();
CFURLRef boingURL = CFBundleCopyResourceURL(mainBundle,
    CFSTR("boing"), CFSTR("wav"), NULL);
SystemSoundID boingSoundID;
AudioServicesCreateSystemSoundID(boingURL, &boingSoundID);
AudioServicesPlaySystemSound(boingSoundID);
AudioServicesPlaySystemSound(kSystemSoundID_Vibrate); // vibrate device
```
SpriteKit Audio

- **SKAudioNode**
  - Add to scene in scene editor using Audio Node
  - Or, create using initWithFileNamed method
    - Add node to scene using `[self addChild:node]`
  - Plays continuously while game not paused

```swift
-(void)didMoveToView:(SKView *)view {
    // setup sound using SKAudioNode
    SKAudioNode *backgroundMusicNode = [[SKAudioNode alloc] initWithFileNamed:@"WSU-Fight-Song.mp3"];
    [self addChild:backgroundMusicNode];
}
```
SpriteKit Audio

• SKAction
  – Create using `playSoundFileNamed` method
  – Add action to any SKNode in scene to play once

```swift
- (void)didBeginContact:(SKPhysicsContact *)contact {
  SKAction* glassbreakAction = [SKAction playSoundFileNamed:@"glassbreak.mp3"
                             waitForCompletion:NO];
  // if either contacting body is a brick, then remove it from scene
  if ([contact.bodyA.node.name isEqualToString:@"brick"]){
    [contact.bodyA.node removeFromParent];
    [ball runAction:glassbreakAction];
    bricksLeft--;}
  if ([contact.bodyB.node.name isEqualToString:@"brick"]){
    [contact.bodyB.node removeFromParent];
    [ball runAction:glassbreakAction];
    bricksLeft--;}
}
```
SceneKit Audio

- No Audio Node in scene editor (yet)

- SCNAudioPlayer
  - Create using audioPlayerWithSource method
  - Add to node in scene using addAudioPlayer method
  - Set audioSource.loop = YES for continuous play

```objective-c
// background music
SCNAudioSource* backgroundMusic = [SCNAudioSource audioSourceNamed:@"WSU-Fight-Song.mp3"];
backgroundMusic.loops = YES;
SCNAudioPlayer* backgroundMusicPlayer = [SCNAudioPlayer audioPlayerWithSource:backgroundMusic];
[paddleNode addAudioPlayer:backgroundMusicPlayer]; // or any node in the scene
```
SceneKit Audio

- SCNAction
  - Create using playAudioSource method
  - Add action to any SCNNode in scene to play once

```swift
-(void)physicsWorld:(SCNPhysicsWorld *)world didBeginContact:(SCNPhysicsContact *)contact {
    SCNAudioSource* glassbreakSource = [SCNAudioSource audioSourceNamed:@"glassbreak.mp3"];
    SCNAction* glassbreakAction = [SCNAction playAudioSource:glassbreakSource waitForCompletion:NO];
    if ([contact.nodeA.name isequalToString:@"brick"]
        [contact.nodeA removeFromParentNode];
        [ballNode runAction:glassbreakAction];
        bricksLeft--;
    }
    if ([contact.nodeB.name isequalToString:@"brick"]
        [contact.nodeB removeFromParentNode];
        [ballNode runAction:glassbreakAction];
        bricksLeft--;
    }
    if (bricksLeft == 0)
        [self gameOver];
}
Resources

• Using Audio