

# iOS Games—Part 1

Introduction to new technologies

Session 402

**Nate Begeman**

Graphics & Media

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

# It Is a Great Time to Make Games



# Part 1 — New Technologies



# Part 2—Game Design

## Designing great games for iOS

- Audio
- Input
- Quality
- And so much more

# Game Center



# New Game Center Features

Gaming is social



- Profile photos
- Game recommendations
- Friend recommendations

# New Leaderboard Features

Gaming is competitive



- Add friends
- Rate games
- Custom category icons

# New Achievement Features

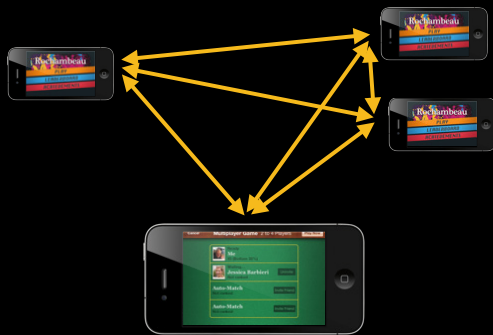
Gaming is competitive



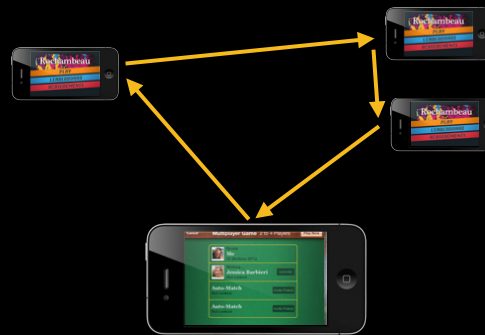
- Completion banners
- Achievement Leaderboards



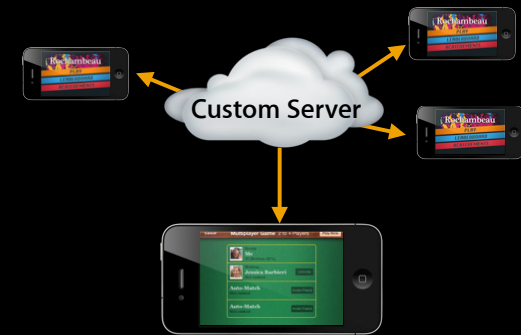
# Styles of Multiplayer



Peer-to-Peer



Turn-Based



Server-Based

# Turn-Based Match Opportunities



- Sports
- Board games
- Puzzles
- Be creative

# Word for Word

Turn-based match example

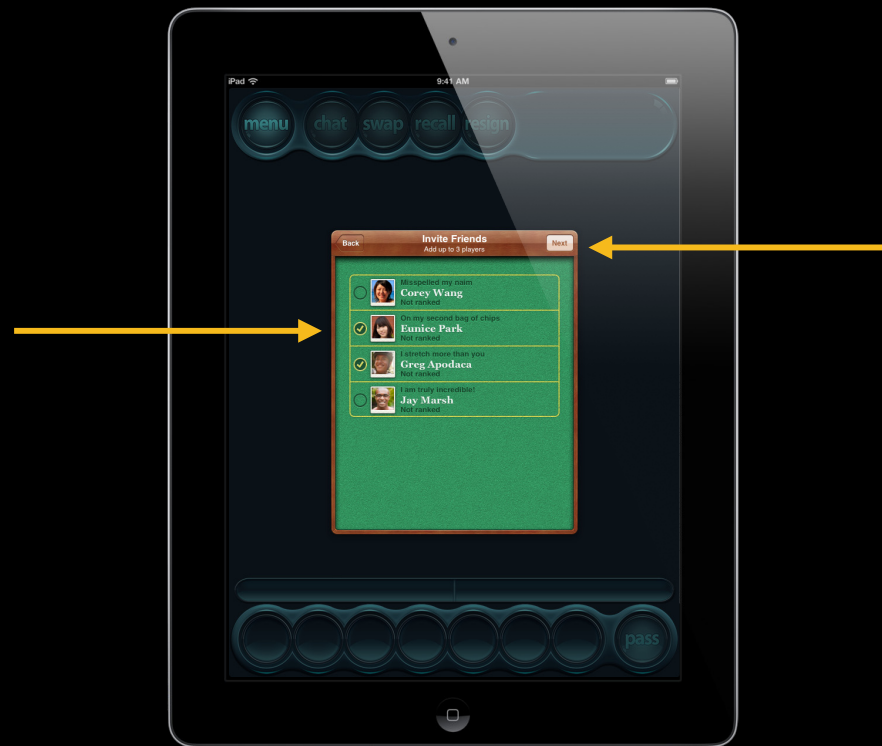
# Starting a Match



# Starting a Match



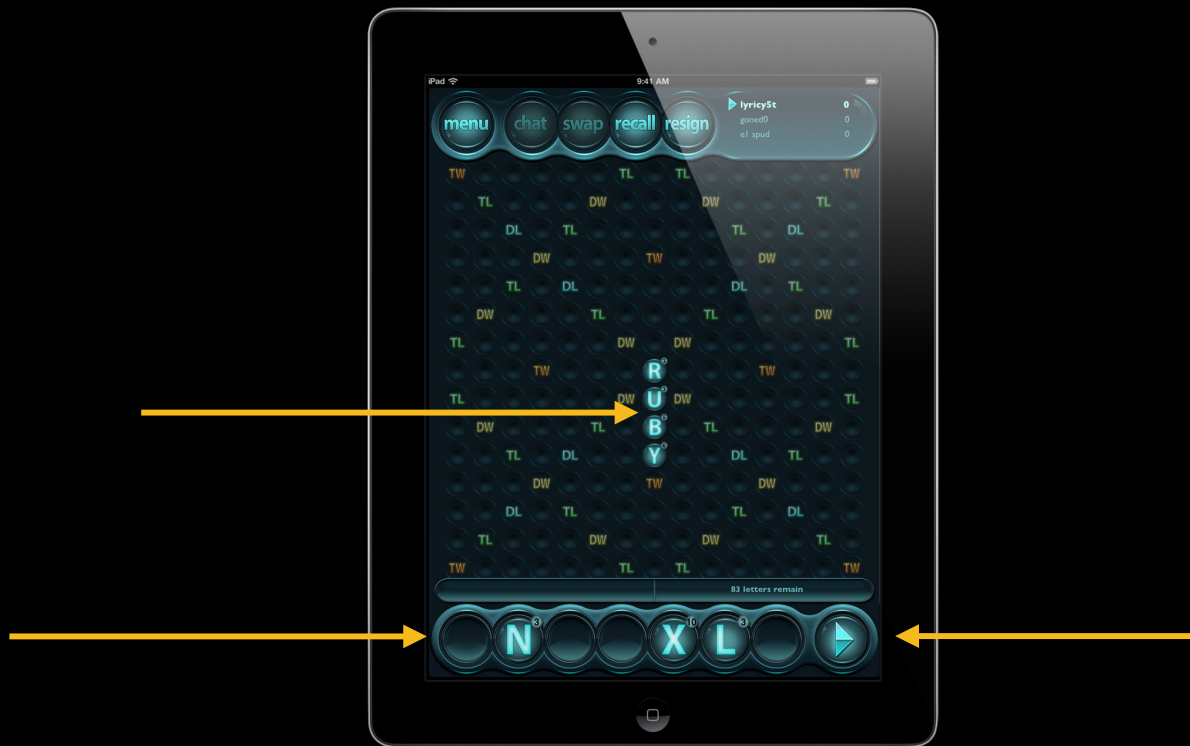
# Invite Two Additional Players



# Ready to Begin

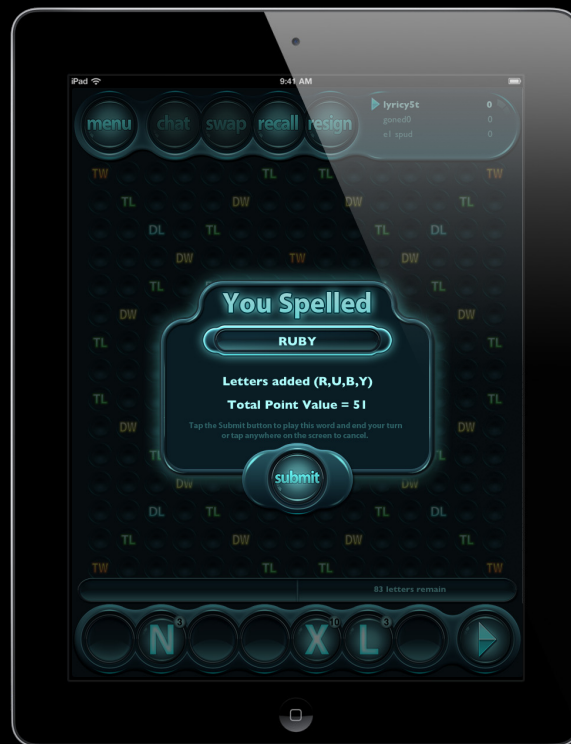


# First Turn





# First Turn



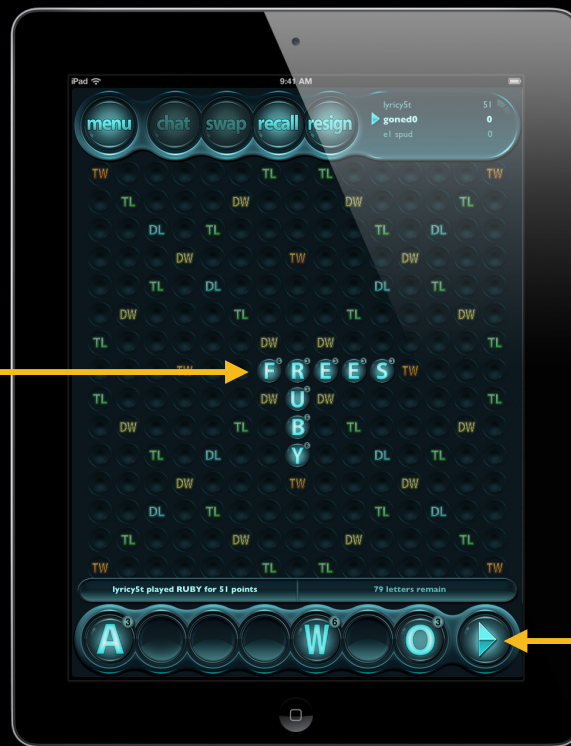
# Invite Notification



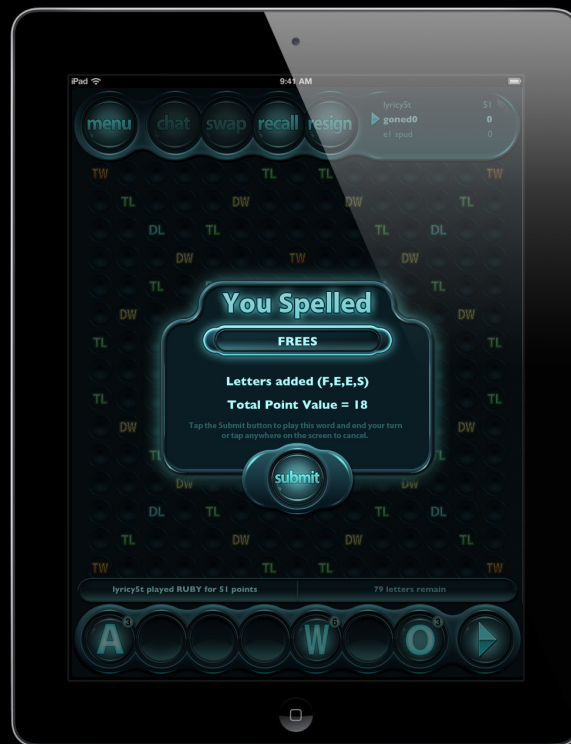
# Available Games



# Second Turn



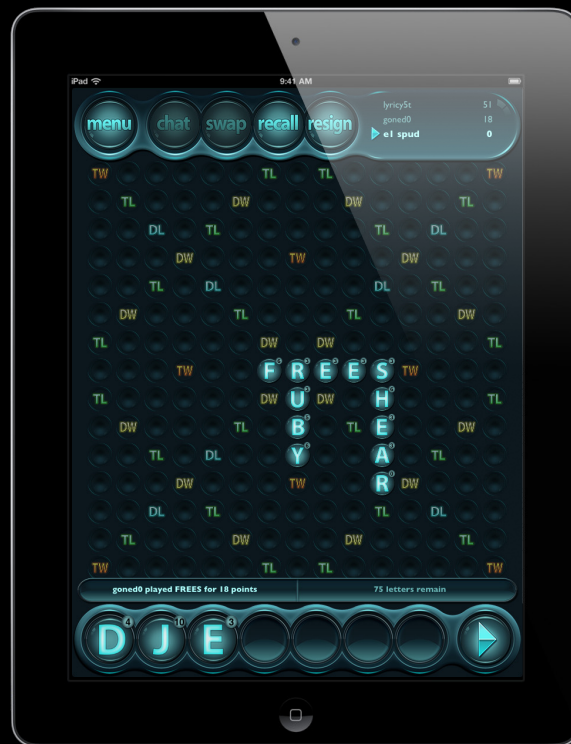
# Second Turn



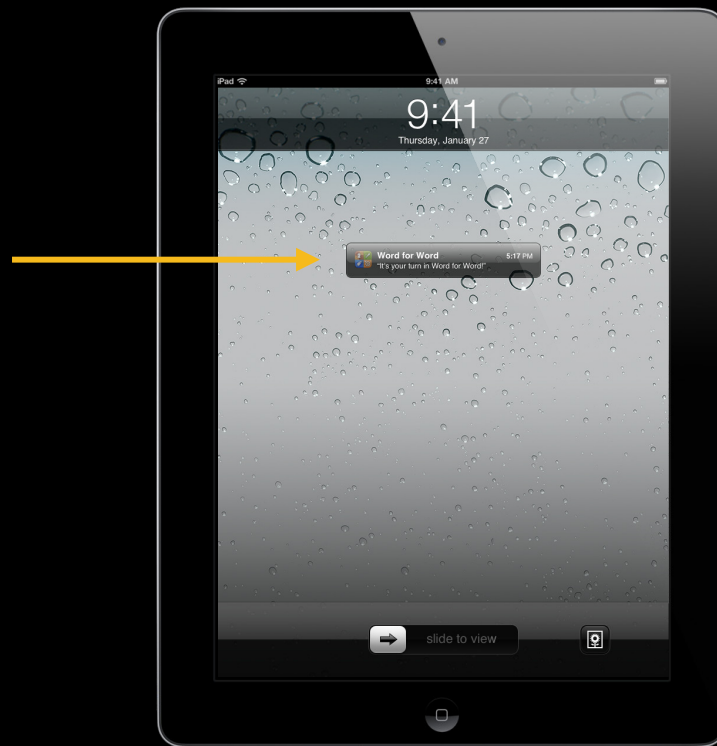
# Invite Notification



# Third Turn



# Turn Notification





# Word for Word

Turn-based gaming basics—Code examples

# Turn-Based Game Code Example

## Turn-based match protocol

```
// Our Turn-based Match.  
GKTurnBasedMatch *_match;  
  
// Tell GameKit we'll be handling its events  
[GKTurnBasedEventHandler sharedTurnBasedEventHandler].delegate = self;  
  
// Three kinds of events  
- (void)handleInviteFromGameCenter:(NSArray *)playersToInvite;  
- (void)handleTurnEventForMatch:(GKTurnBasedMatch *)match;  
- (void)handleMatchEnded:(GKTurnBasedMatch *)match;
```

# Turn-Based Game Code Example

## Choose next player

```
// Golf example, pick the next player farthest from pin.  
int playerIndex = farthestFromPin();
```

```
// Word for Word example, rotate through indices.  
int playerIndex = ++currentIndex % numPlayers;
```

```
// Get the Participant for this index.  
GKTurnBasedParticipant *nextParticipant =  
    [_match.participants objectAtIndex:playerIndex];
```

# Turn-Based Game Code Example

## Advance to next player

```
- (void)submitTurnButtonWasPressed
{
    // Serialize data for this turn.
    NSData *gameTurnData = myGenerateTurnData();

    // send to server (advance turn to next player)
    [self.gkTurnBasedMatch endTurnWithNextParticipant:nextParticipant
     matchData:gameTurnData completionHandler:^(NSError *error)
     {
         if (error == nil)
             success stuff;
     }];
}
```

# Turn-Based Game Code Example

It is your turn

```
// Conforming to GKTurnBasedEventHandlerDelegate protocol
- (void)handleTurnEventForMatch:(GKTurnBasedMatch *)match
{
    // Unpack your data

    // refresh game UI here to reflect new data
}
```

# Turn-Based Game Code Example

## Tips

- Send up to 4kB turn data
- Each match has a unique ID
- Notifications on your turn

# Game Center Sessions

Introduction to Game Center

Mission  
Tuesday 4:30PM

Multiplayer Gaming with Game Center

Mission  
Wednesday 10:15AM

Turn-Based Gaming with Game Center

Mission  
Wednesday 11:30AM

Introduction to Game Center

Russian Hill  
Friday 9:00AM

# Game Center Wrap Up



- Friends
- Leaderboards
- Achievements
- Multiplayer



# iCloud



# Games Need Storage



- Automatic syncing
- Per user storage
- Perfect for gaming

# Storage Between Devices

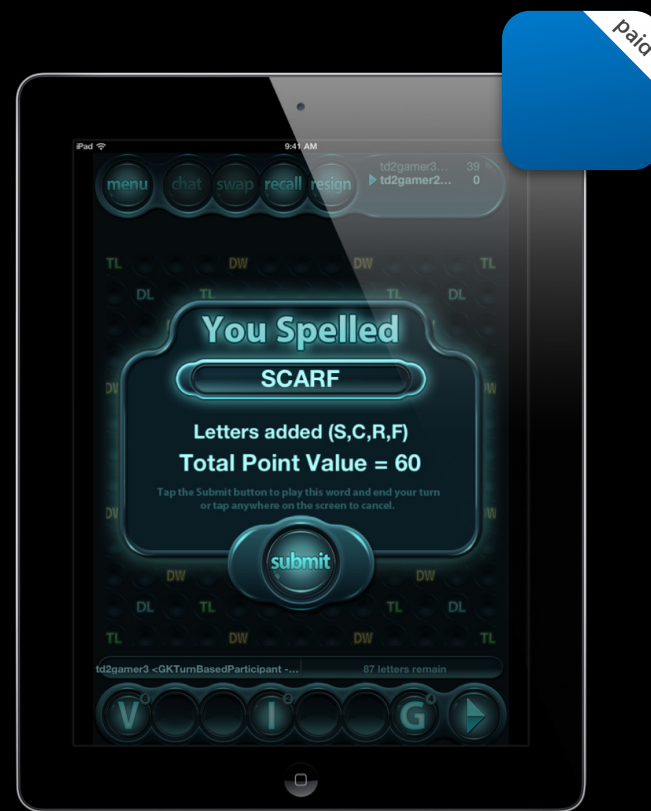
## iCloud scenarios



- Game state goes where you do
  - iPad at home
  - iPhone at work

# Storage Between Versions

## iCloud scenarios



# Storage Between Platforms

## iCloud scenarios



# Cloud Storage

Code examples

# iCloud Basics

## Getting a connection

```
// Let's get a connection to the cloud!  
NSUbiquitousKeyValueStore *cloudStore =  
    [NSUbiquitousKeyValueStore defaultStore];  
  
// Saving a simple value  
[cloudStore setBool:TRUE forKey:@"com.my.app.myAwesomeBoolKey"];
```

# iCloud Basics

## Packing up game state

```
// Serialize Data
NSData *gameState =
    [NSKeyedArchiver archivedDataWithRootObject:self.currentGameState];

// Save Game State to cloud
[cloudStore setData:gameState forKey:@"com.my.app.radGameState"];
```



# iCloud Basics

## Using NSCoder

- Game state object conforms to NSCoder protocol
  - No hassle cross-platform object transfer
    - 32/64b
    - iOS 5/Lion
- 
- `(void)encodeWithCoder:(NSCoder *)aCoder;`
  - `(id)initWithCoder:(NSCoder *)aDecoder;`

# iCloud Basics

## Using NSCoder

```
- (void)encodeWithCoder:(NSCoder *)aCoder
{
    [aCoder encodeInteger:self.PlayerIndex forKey:@"Index"];
    [aCoder encodeInteger:self.lastTurnAction forKey:@"Action"];
}

- (id)initWithCoder:(NSCoder *)aDecoder
{
    [super init];
    self.PlayerIndex = [aDecoder decodeIntegerForKey:@"Index"];
    self.lastTurnAction = [aDecoder decodeIntegerForKey:@"Action"];
}
```

# iCloud Basics

## Registering for notifications

```
[[NSNotificationCenter defaultCenter]
 addObserver:[MyAppClass class]
 selector:@selector(handleSyncedDataChanged:)
 name:NSUbiquitousKeyValueStoreDidChangeExternallyNotification
 object:cloudStore];
```

# iCloud Basics

## Retrieving and unpacking data

```
+ (void)handleSyncedDataChanged:(NSNotification *)note
{
    NSData *syncedGameState =
        [cloudStore dataForKey:@" com.my.app.radGameState"];

    // Calls -initWithCoder
    [self.currentGameState =
        [NSKeyedUnarchiver unarchiveObjectWithData:syncedGameState];
}
```

# iCloud Sessions

iCloud Storage Overview

Presidio  
Tuesday 11:30AM

Taking Advantage of File Coordination

Pacific Heights  
Tuesday 4:30PM

Storing Documents in iCloud Using iOS 5

Presidio  
Wednesday 3:15PM

# Cloud Storage Is Versatile



- Not just game state
  - App prefs
  - Statistics
  - Personalization

# iOS Games

GL Kit



# iOS Games

## GL Kit



- UIKit integration
- Fast math library
- Easy texture loading
- Effects library

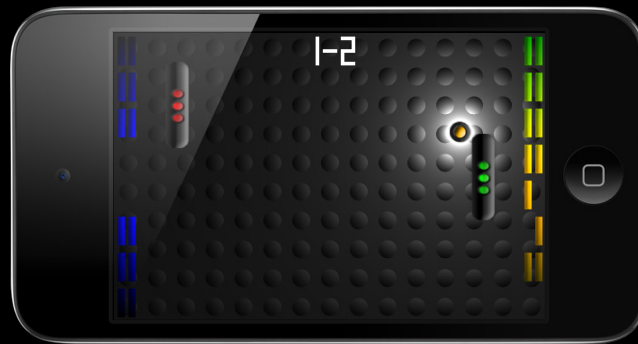


# Shock

GLKit Basics—Code examples

# Shock

Texturing, lighting, and presenting a GL ES view



# GL Kit

## Texture loading before

```
GLuint texID;
textureImage = [UIImage imageNamed:@"earth"].CGImage;
width = CGImageGetWidth(textureImage);
height = CGImageGetHeight(textureImage);

if(textureImage)
{
    ...
}
```

# GL Kit

## Texture loading before

```
if(textureImage) {
    textureData = (GLubyte *) malloc(width * height * 4);
    textureContext = CGContextCreate(textureData, ...);
    CGContextDrawImage(textureContext, CGRectMake(0, 0, width, height),
        textureImage);
    CGContextRelease(textureContext);

    glGenTextures(1, &texID);
    glBindTexture(GL_TEXTURE_2D, texID);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, width, height, 0, GL_RGBA,
        GL_UNSIGNED_BYTE, textureData);
}
```

# GL Kit

## Texture loading after

```
NSURL *EarthURL =
    [NSURL fileURLWithPath:[NSBundle mainBundle]
           pathForResource:@"earth" ofType:@"png"]];

// Texture information returned, not texture itself.
GLKTextureInfo *texInfo =
    [GLKTextureLoader textureWithContentsOfURL:EarthURL
                       options:nil error:nil];

// textureName is our texture id, see header for other properties
GLuint texId = [texInfo textureName];
```

# GL Kit

## Texture loading after

```
GLuint texID =  
[[GLKTextureLoader textureWithContentsOfURL:  
    [NSURL URLWithString: [[NSBundle mainBundle]  
                           pathForResource:@"earth"  
                           ofType:@"png"]]  
    options: nil error: nil] textureName];
```

# GL Kit

## Lighting before

```
// Load and bind our texture
[self loadTexture:@"bumps.png" withID:&backgroundTex];
glActiveTexture(GL_TEXTURE0);
glBindTexture(GL_TEXTURE_2D, backgroundTex);

// Set our ball position and texture sampler.
glUseProgram(lightProgram);
glUniform2f(glGetUniformLocation([program programObject], "ballPos"),
           ballPosition.x, ballPosition.y);
sampler_loc = glGetUniformLocation([self glProgramObj], "sampler");
glUniform1i(sampler_loc, 0);

glDrawArrays(GL_TRIANGLES, 0, count);
```

# GL Kit

## Lighting before

```
precision mediump float;
varying vec2 texcoord;
uniform sampler2D sampler;
uniform vec2 ballPos;
void main() {
    vec3 ball = vec3(ballPos, 3.0);
    vec3 diff = vec3(gl_FragCoord.xy, 3.0) - ball;
    vec3 LightPos = normalize(diff);
    vec3 N = texture2D(sampler, texcoord).rgb;
    float attenuation = 20.0/length(diff) + 0.05;
    gl_FragColor = vec4 ((c+0.7)*attenuation, (c+0.7)*attenuation,
                        (c+0.7*attenuation, 1.0);
}
```



# GL Kit

## Lighting after

```
// Create and configure GLKBaseEffect instance
GLKBaseEffect *baseEffect = [[GLKBaseEffect alloc] init];

// Set the lighting type (per-vertex by default)
baseEffect.lightingType = GLKLightingTypePerPixel;

// Set some light properties
baseEffect.light0.enabled = GL_TRUE;
baseEffect.light0.position =
    GLKVector4Make(ballPos.x, ballPos.y, 1.0f, 0.0f);
baseEffect.light0.spotCutoff = 20.0f;
```

# GL Kit

## Lighting after

```
baseEffect.texture2d0.envMode = GLKTextureEnvModeReplace;
baseEffect.texture2d0.glName =
    [[GLKTextureLoader textureWithContentsOfURL:@"bumps.png"] textureName];

// Synchronize our effect changes
[baseEffect prepareToDraw];

glDrawArrays(GL_TRIANGLES, 0, count);
```

# GL Kit

## Render loop before—EAGLView

```
// You must implement this method
+ (Class)layerClass { return [CAEAGLLayer class]; }

//The GL view is stored in the nib file. When it's unarchived it's sent -
initWithCoder:
- (id)initWithCoder:(NSCoder*)coder {
    CAEAGLLayer *eaglLayer = (CAEAGLLayer *)self.layer;
    eaglLayer.opaque = YES;
    ...
}
```

# GL Kit

## Render loop before—EAGLView

```
- (id)initWithCoder:(NSCoder*)coder {  
    ...  
    eaglLayer.drawableProperties =  
    [NSDictionary dictionaryWithObjectsAndKeys:  
        [NSNumber numberWithInt:NO], kEAGLDrawablePropertyRetainedBacking,  
        kEAGLColorFormatRGBA8, kEAGLDrawablePropertyColorFormat, nil];  
  
    context = [[EAGLContext alloc] initWithAPI:kEAGLRenderingAPIOpenGLES2];  
  
    // set up animation timer to call drawView @ 30fps  
}
```

# GL Kit

## Render loop before—EAGLView

```
- (void)drawView {  
    // Update the game state for this frame  
    [[app delegate] gameLogic];  
  
    [EAGLContext setCurrentContext:context];  
    glBindFramebuffer(GL_FRAMEBUFFER, viewFramebuffer);  
    glViewport(0, 0, &width, &height);  
  
    // Call our custom drawing code.  
    [delegate drawInRect: CGRectMake(0, 0, width, height)];  
    glBindRenderbuffer(GL_RENDERBUFFER, viewRenderbuffer);  
    [context presentRenderbuffer:GL_RENDERBUFFER];  
}
```

# GL Kit

## Render loop after—GLKView

```
- (void)applicationDidFinishLaunching:(UIApplication *)application {
    GLKView *glkView = (GLKView *)self.viewController.view;
    glkView.delegate = game;
    glkView.context = [game context];

    self.viewController.delegate = game;
    self.viewController.preferredFramesPerSecond = 30;
}

- (void)glkViewControllerUpdate:(GLKViewController *)controller {
    [delegate gameLogic];
}

- (void) glkView:(GLKView *)view drawInRect:(CGRect)rect
    // our custom drawing code
}
```

# GLKit



- Easily adopt GL ES 2.0
- Leverage powerful GPUs
- Lots of functionality

# OpenGL ES Sessions

Advances in OpenGL ES for iOS 5

Mission  
Wednesday 2:00PM

Tools for Tuning OpenGL ES Apps on iOS

Mission  
Wednesday 3:15PM

Best Practices for OpenGL ES Apps on iOS

Mission  
Wednesday 4:30PM



# iOS Games

AirPlay



# AirPlay

## Second display



- Mirroring
- Second display
- iPad 2 only

# Doodlewords

Slideware

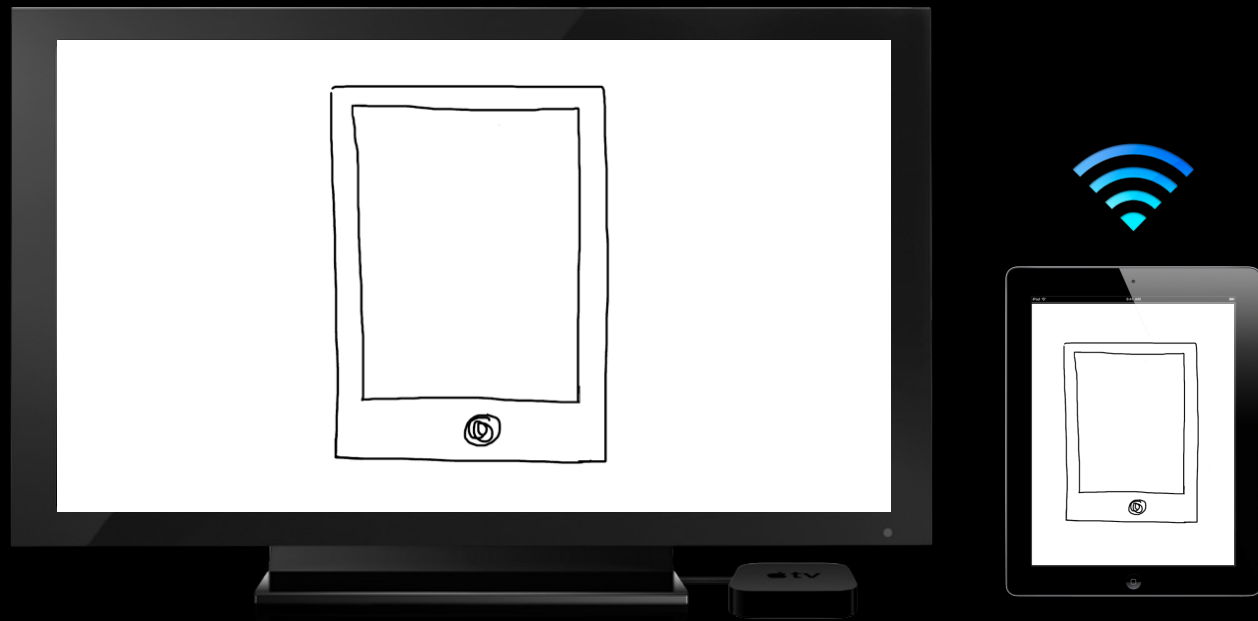
# Shared Experience



# Shared Experience



# Shared Experience



# AirPlay Second Display

Code examples

# Second Display Code Example

## Detecting at app launch

```
- (void)applicationDidFinishLaunching:(UIApplication *)application
{
    if ([[UIScreen screens] count] > 1) {
        [self prepareScreen:[UIScreen screens] lastObject]];
    }
}
```



# Second Display Code Example

## Detecting on hot plug

```
[[NSNotificationCenter defaultCenter]
    addObserver:self
        selector:@selector(screenDidConnect:)
        name:UIScreenDidConnectNotification
        object:nil];

- (void)screenDidConnect:(NSNotification *)notification {
    [self prepareScreen:[notification object]];
}
```

# Second Display Code Example

## Create a remote window

```
// associate window with 2nd screen (main screen always at index 0)
UIScreen* secondScreen = [[UIScreen screens] objectAtIndex:1];
CGRect screenBounds = secondScreen.bounds;

UIWindow *uiWindow = [[UIWindow alloc] initWithFrame:screenBounds];
self.externalWindow = uiWindow;
[uiWindow release];

self.externalWindow.screen = secondScreen;
```

# Second Display Code Example

Create a view, show the window

```
myViewController *extViewController = [[myViewController alloc]
initWithFrame:screenBounds];
    self.externalViewController = extViewController;
    [extViewController release];
    self.externalWindow.rootViewController = self.externalViewController;

// show the window
self.externalWindow.hidden = NO;
```

# Second Display Code Example

## Other tidbits

```
// Resolutions available on your second display
[UIScreen availableModes]
[UIScreen preferredMode]

// Change Brightness of the mirrored display
[[UIScreen mirroredDisplay].brightness = 0.5f];
```

# AirPlay Session

AirPlay and External Displays in iOS Apps

Presidio  
Tuesday 3:15PM

# Summary



# More Information

**Allan Schaffer**

Graphics and Game Technologies Evangelist

[aschaffer@apple.com](mailto:aschaffer@apple.com)

# Labs

Game Design for iOS Lab

Graphics, Media & Games Lab A  
Tuesday 2:00PM



# Coming Up

## Part 2

- 15 minutes from now
- Same convenient location

