

# What's New in Game Center

Session 508

**Jeff Watkins**

iOS Engineer

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

**Huge Success**

1

Discoverable

2

Polished

3

Fun

1

Discoverable

2

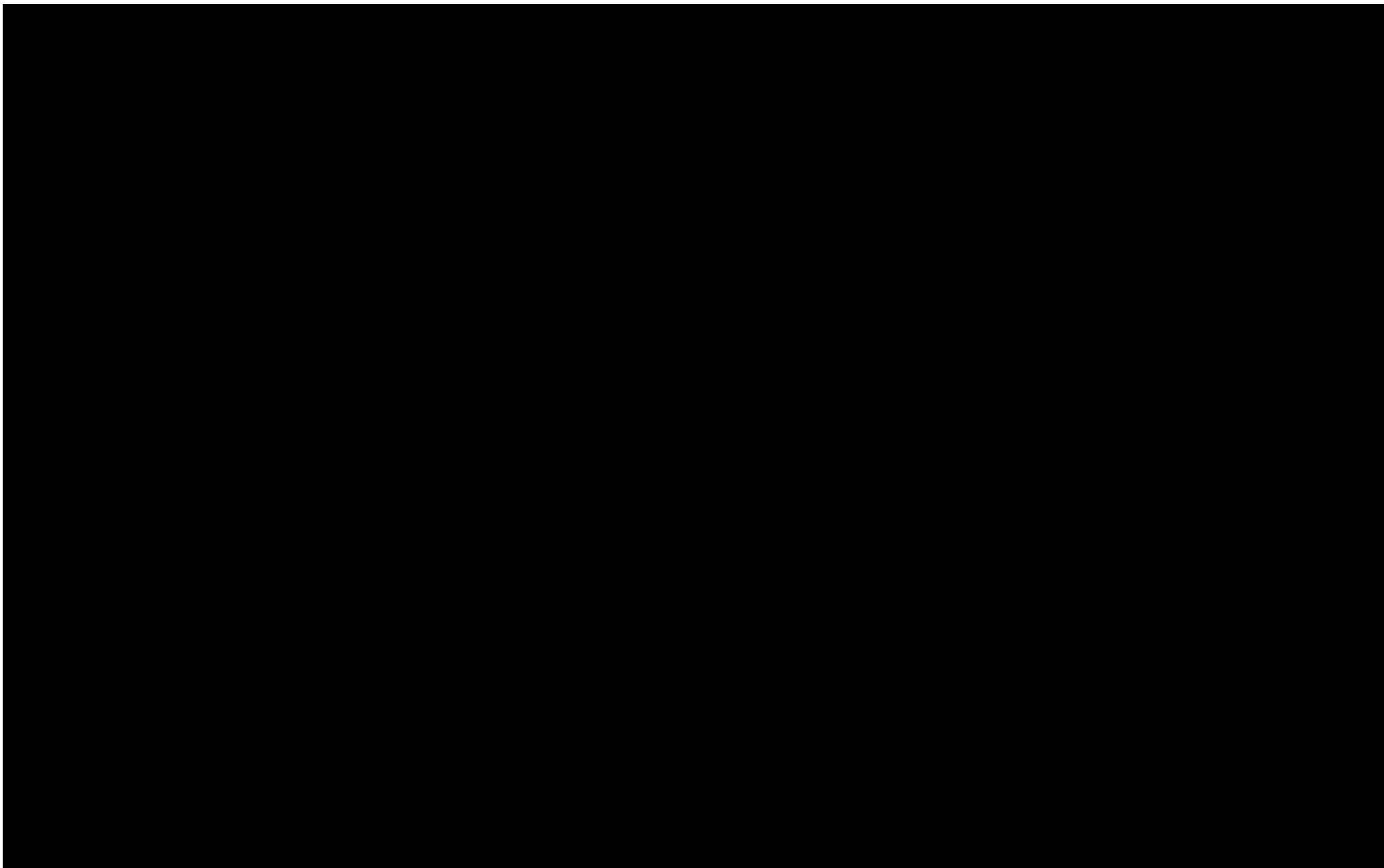
Polished

3

Fun

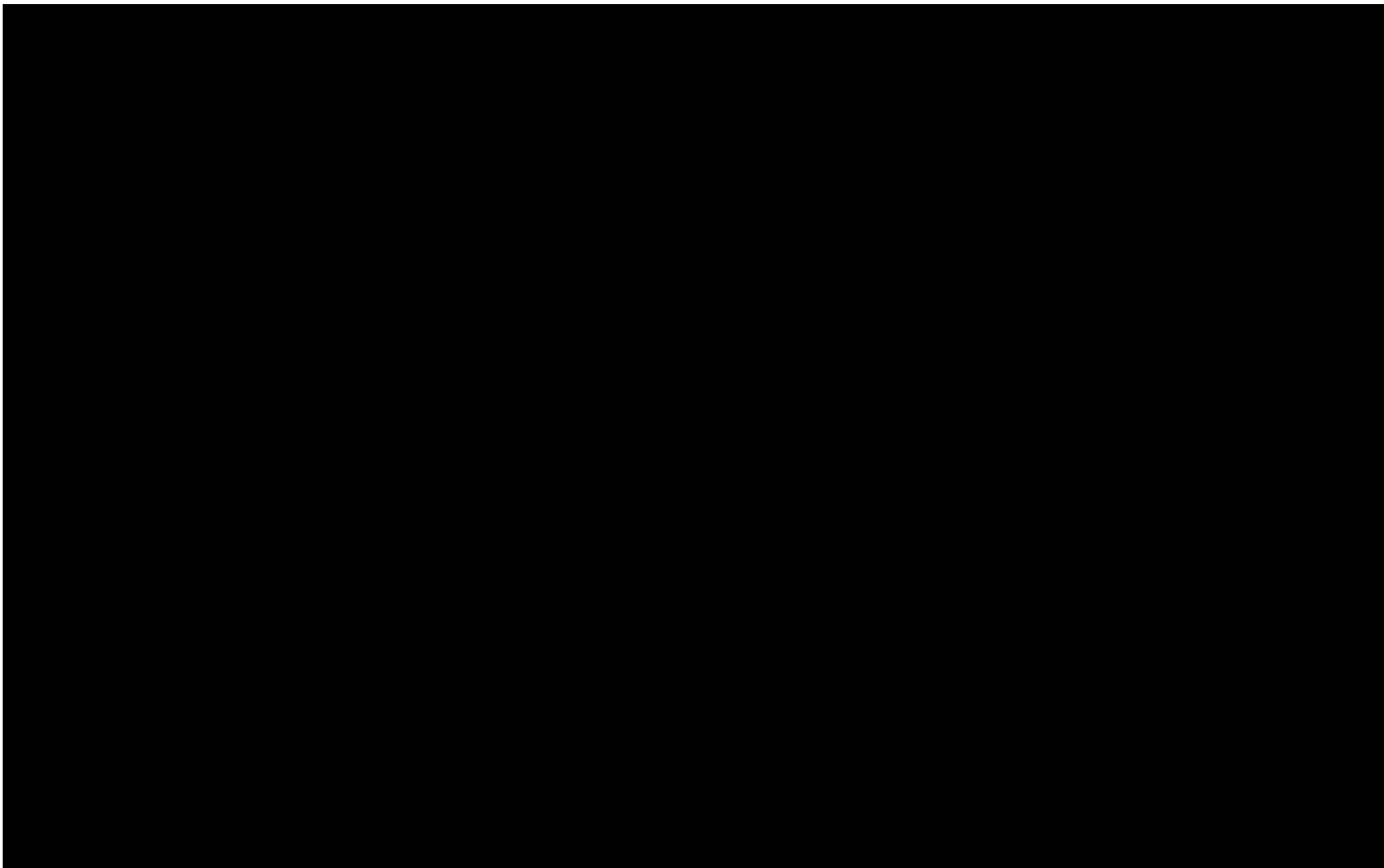


Game Center



**130 Million**

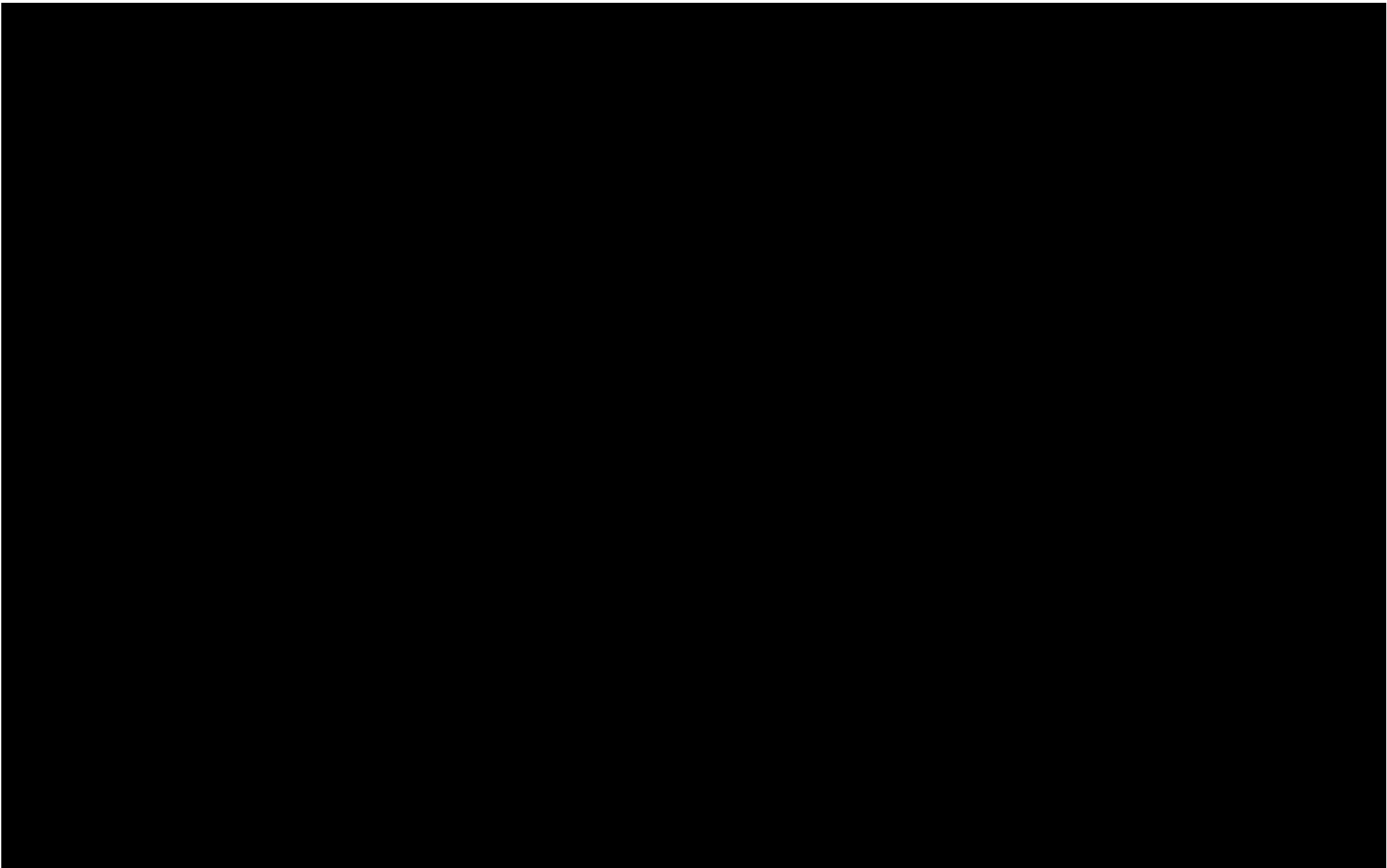
Players





**5 Billion**

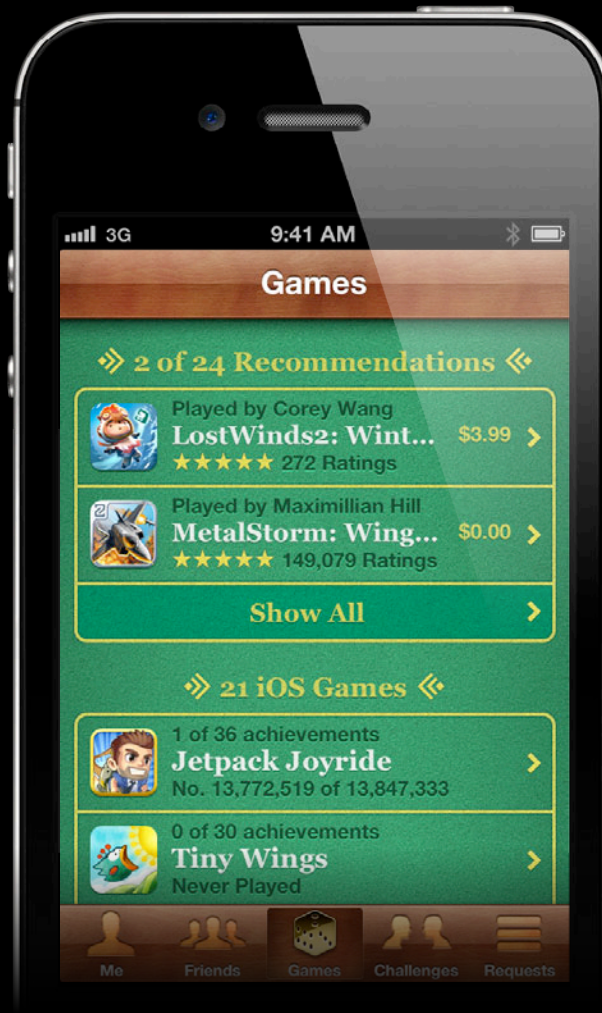
Scores per week



67

Top 100

# Game Recommendations



# Friend Recommendations



# Friend Recommendations



**Friends**

# Tell a Friend





# App Store Rating



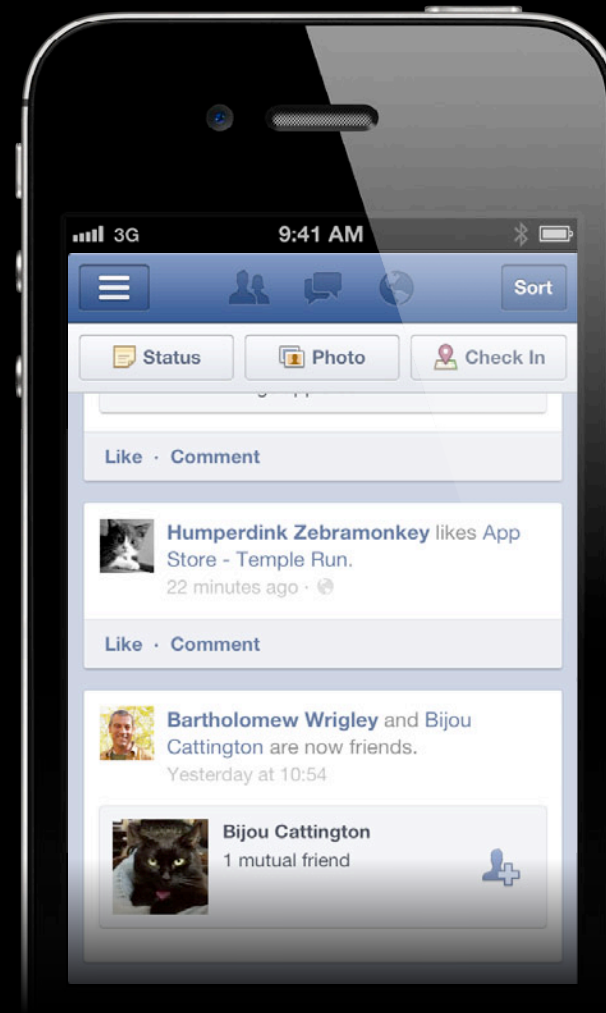
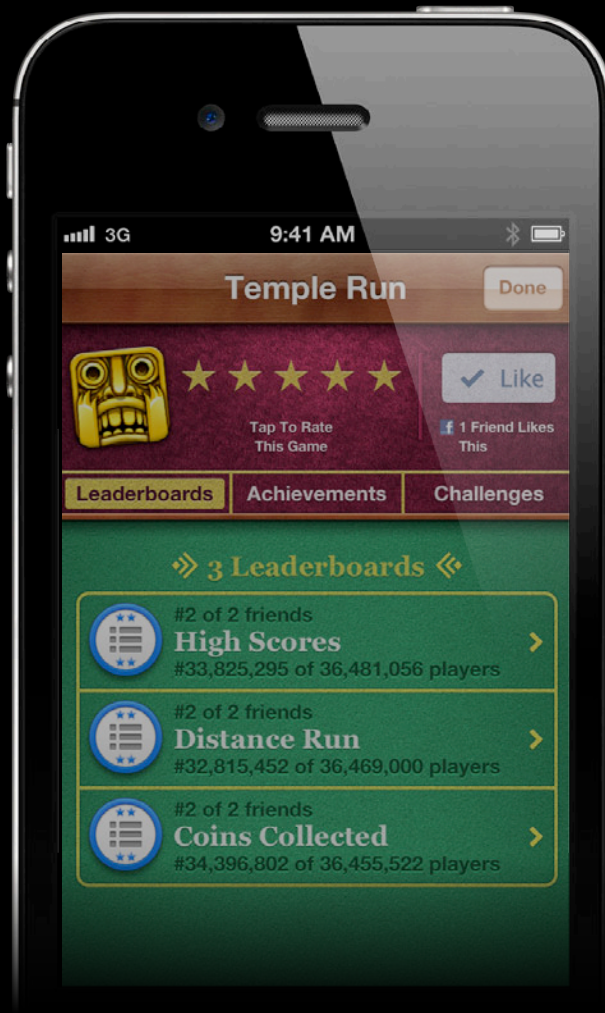
# Facebook Like Button



# Facebook Like Button



# Facebook Like Button



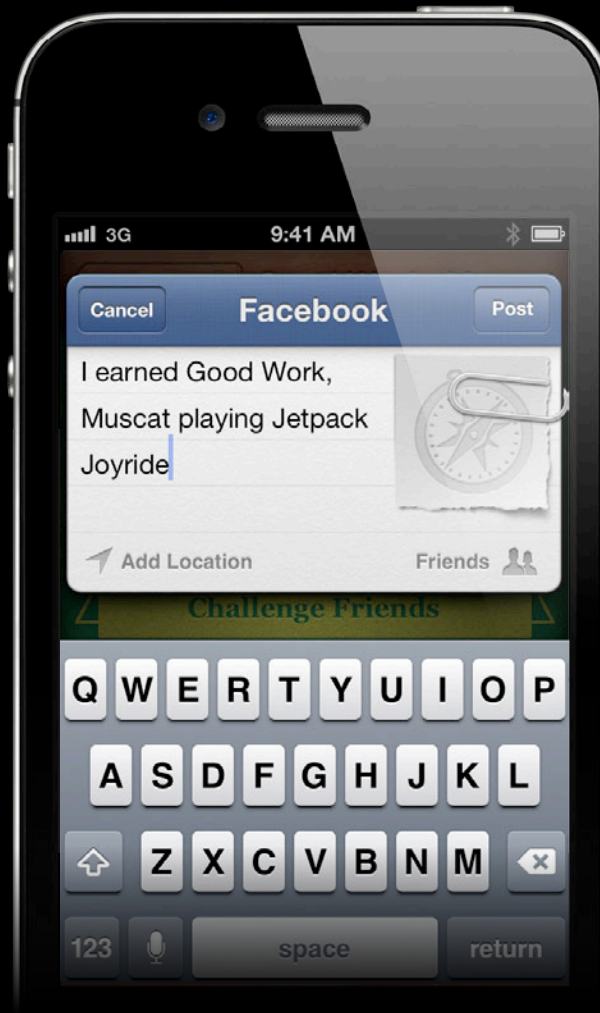
# Share Scores and Achievements



# Share Scores and Achievements



# Share Scores and Achievements

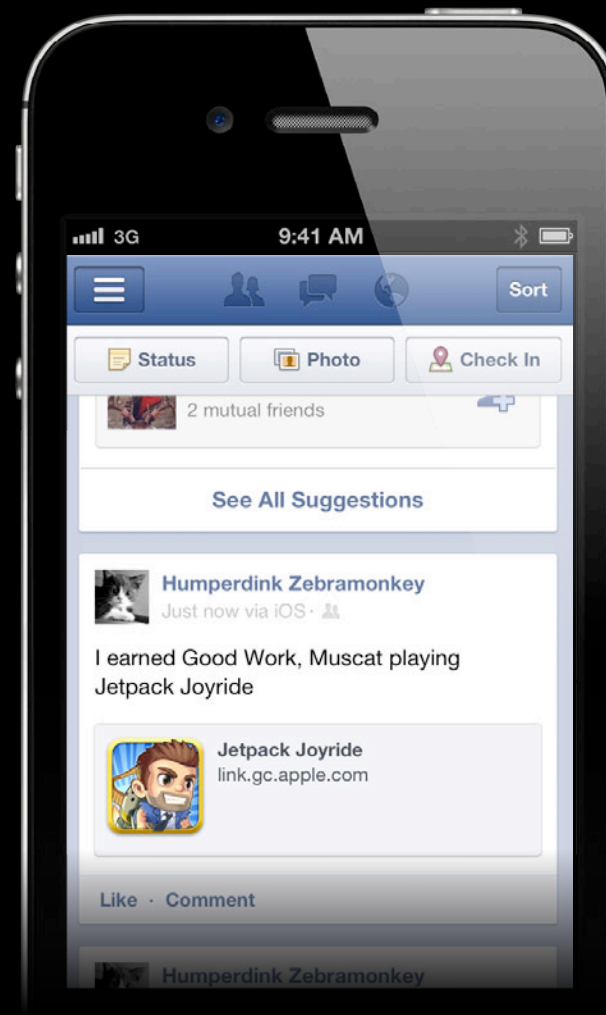


# Share Scores and Achievements





# Share Scores and Achievements



# Challenges



# Challenges



# Challenges



# Challenges



# Challenges



# Challenges



# Challenges

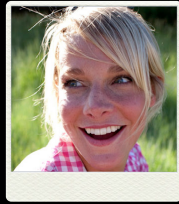




**Multiplayer**



Aideen

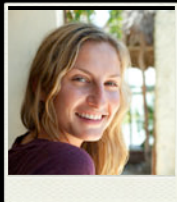


Aideen

# Peer-to-Peer Multiplayer



Aideen



Deirdre

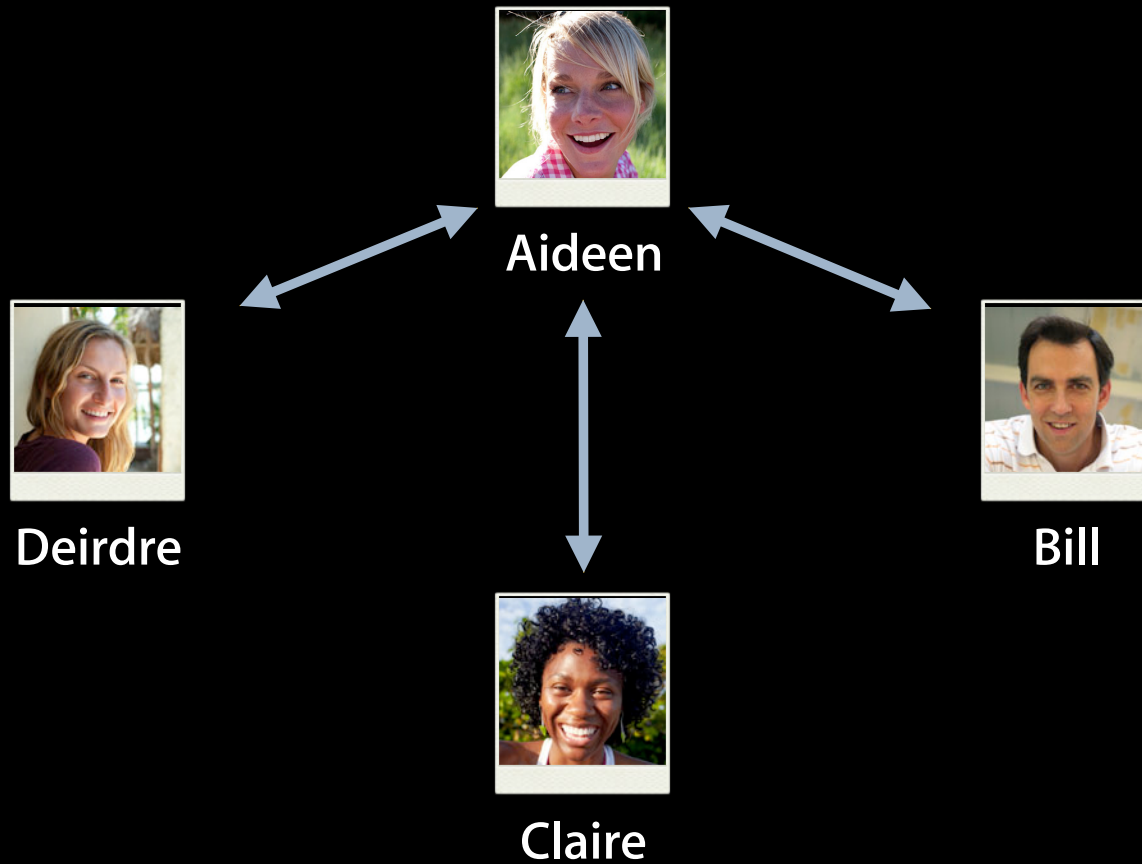


Bill

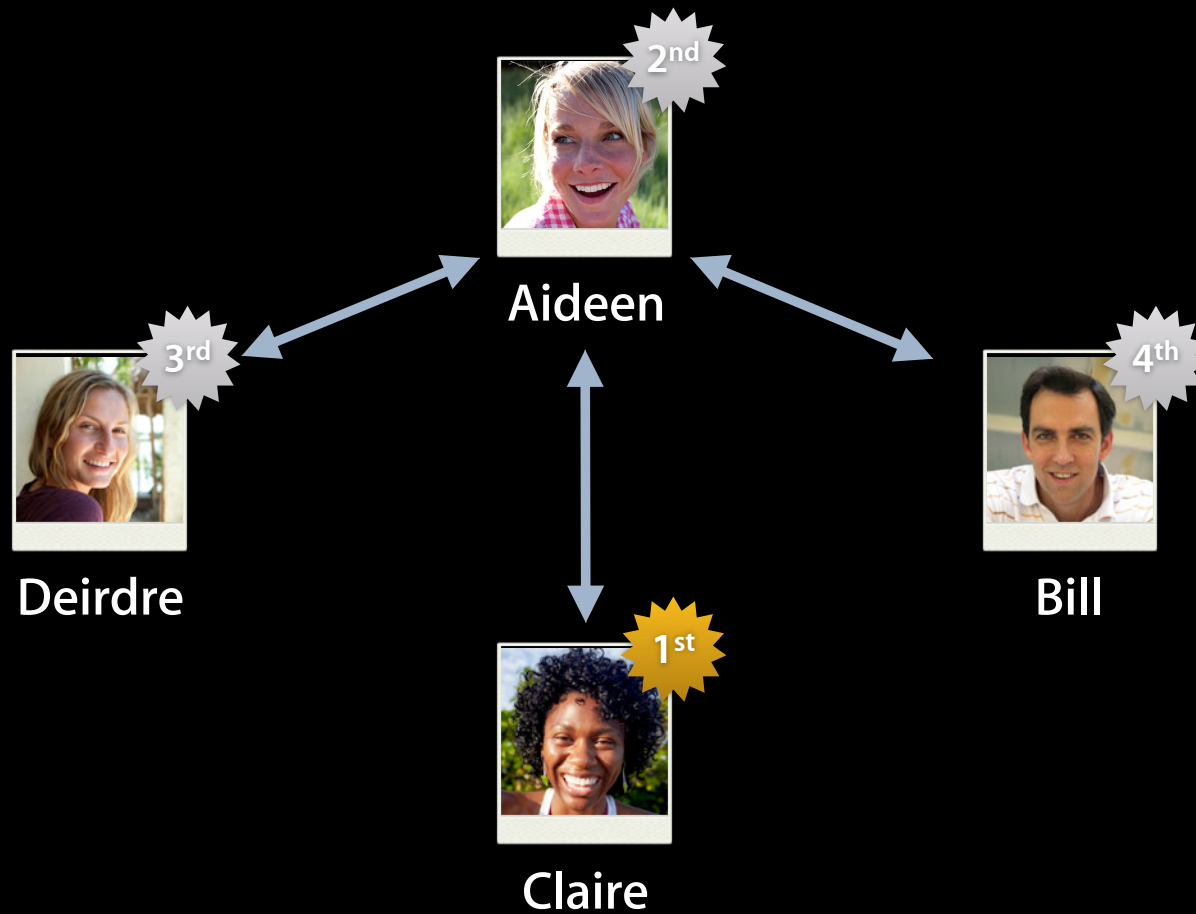


Claire

# Peer-to-Peer Multiplayer



# Peer-to-Peer Multiplayer



# Play It Again

## GKMatch and GKTurnBasedMatch

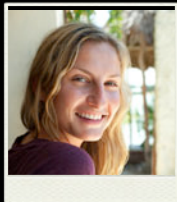


```
[match rematchWithCompletionHandler:^(GKMatch *rematch, NSError *error) {  
    if (error) {  
        // couldn't rematch...  
        return;  
    }  
    // start a new game  
}];
```

# Turn-Based Multiplayer



Aideen



Deirdre



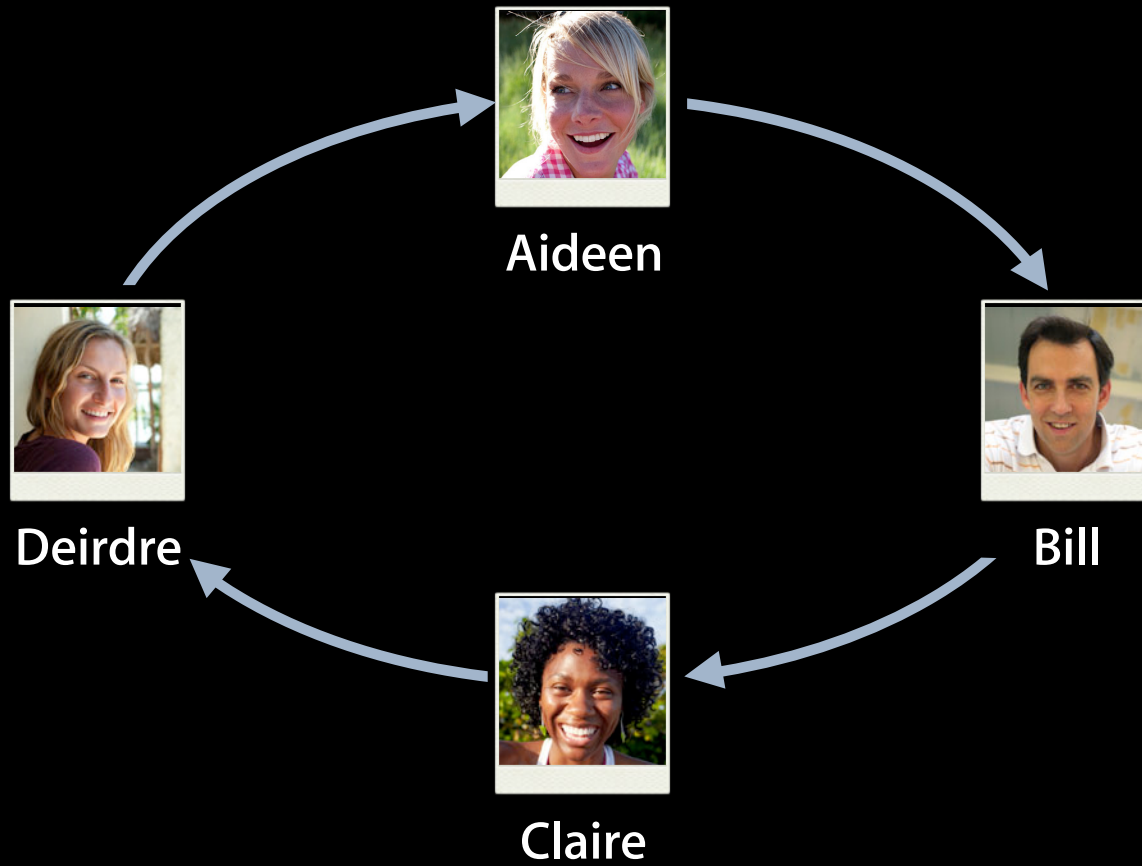
Bill



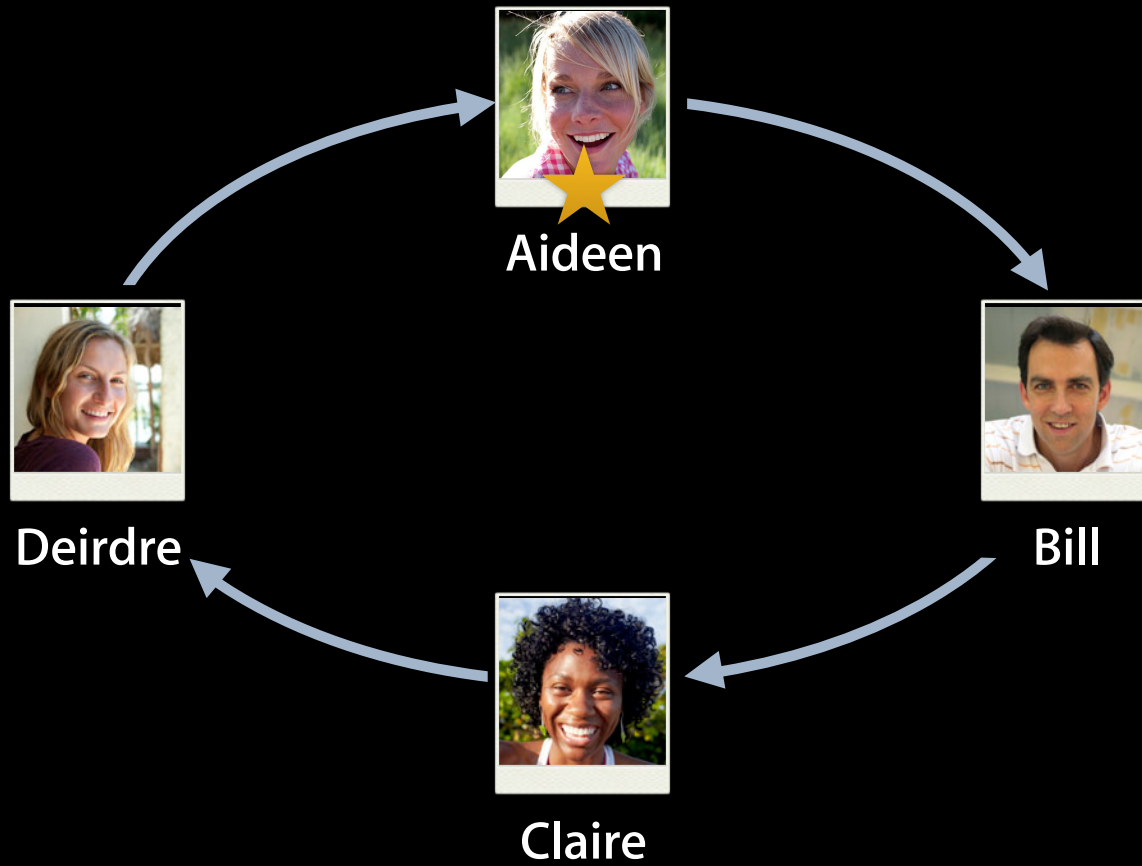
Claire



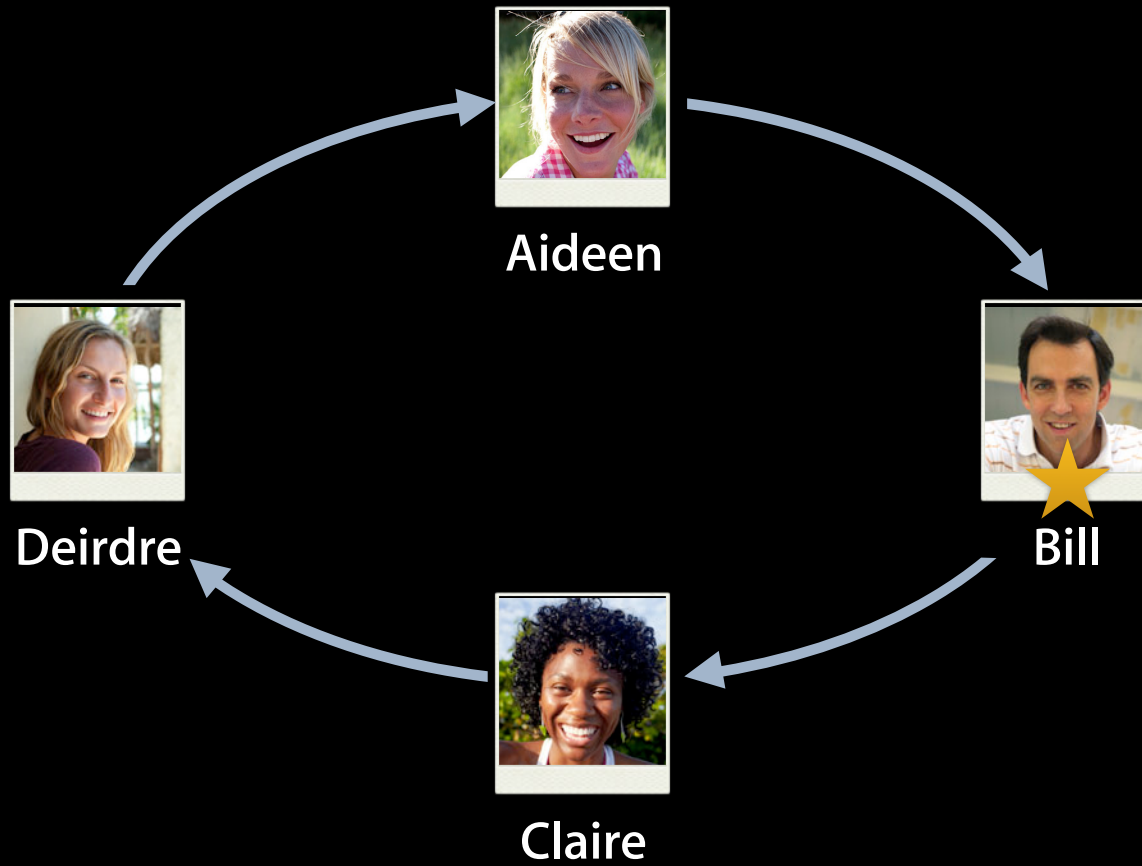
# Turn-Based Multiplayer



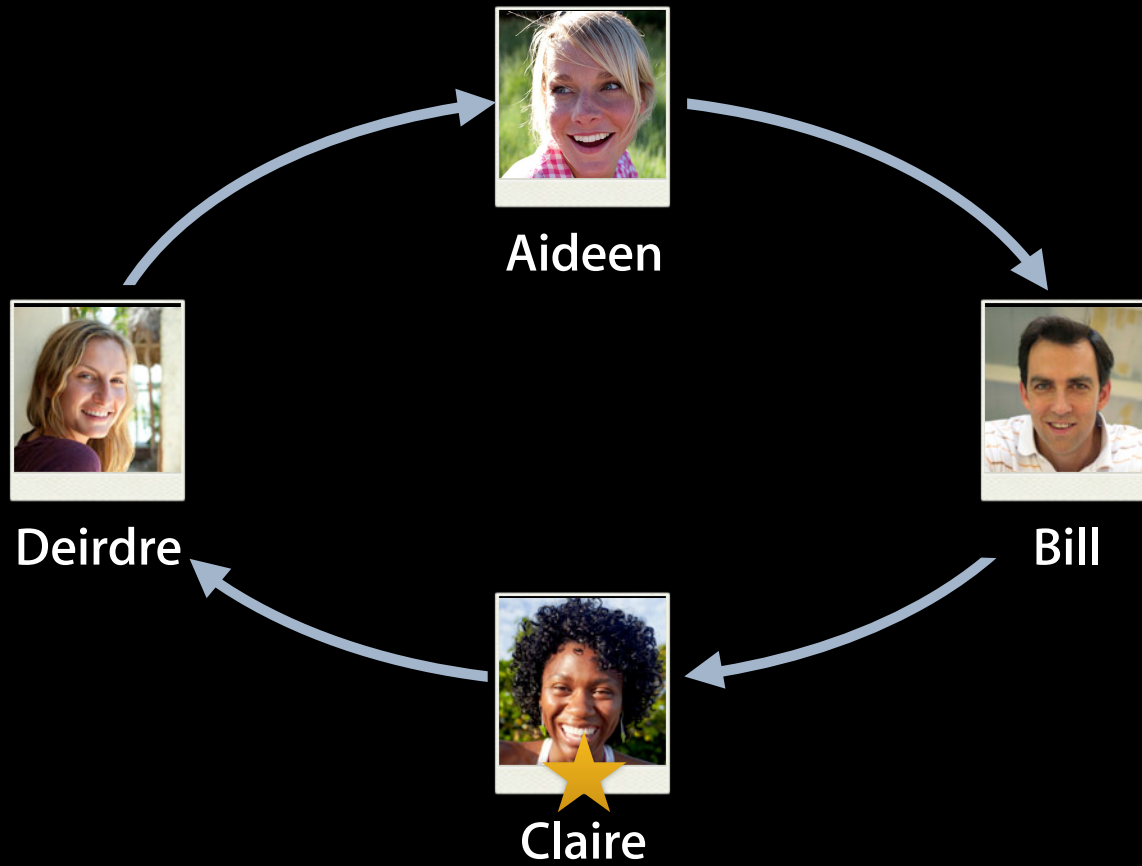
# Turn-Based Multiplayer



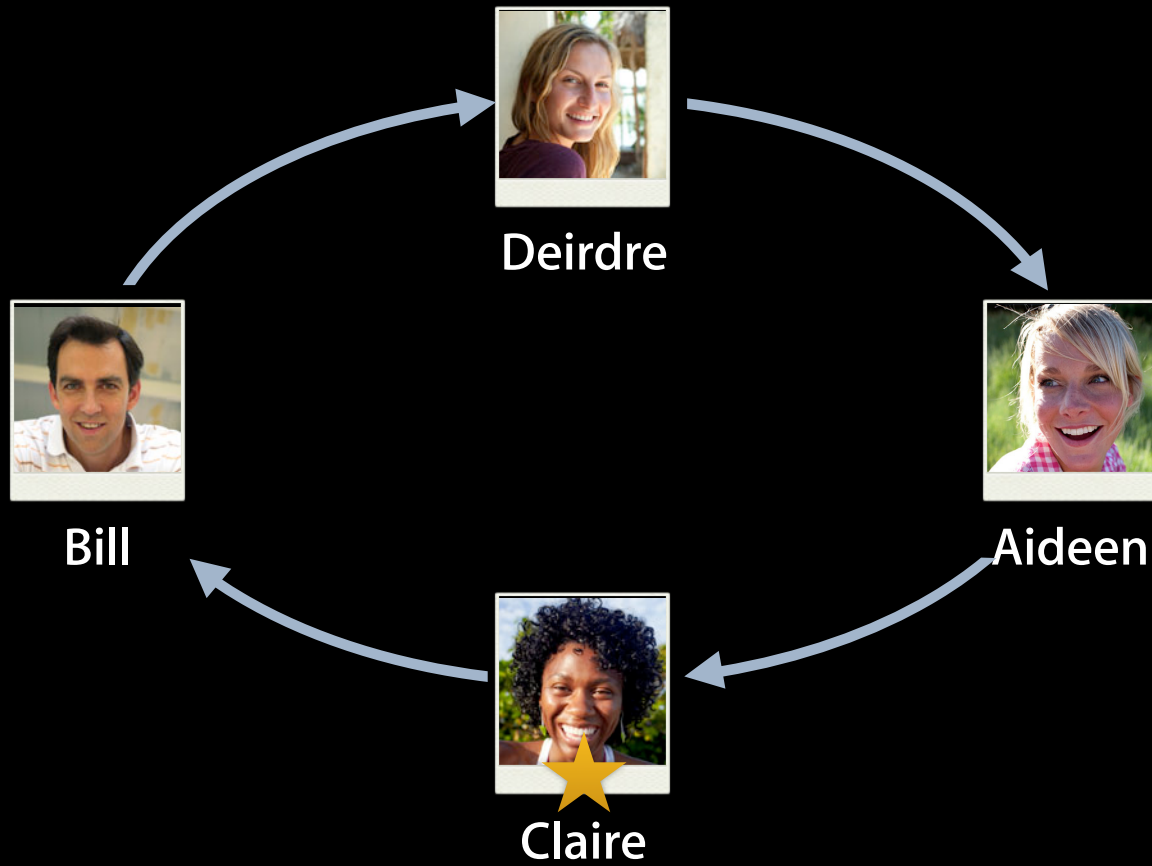
# Turn-Based Multiplayer



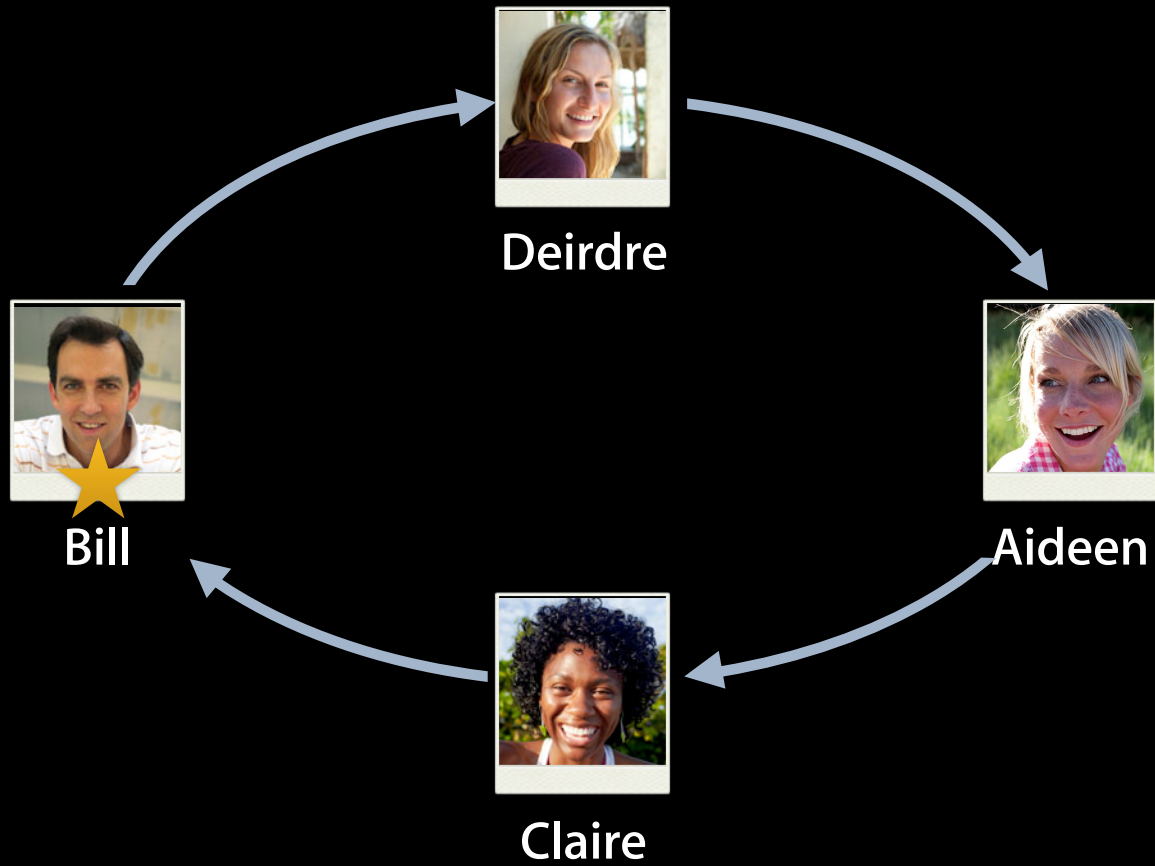
# Turn-Based Multiplayer



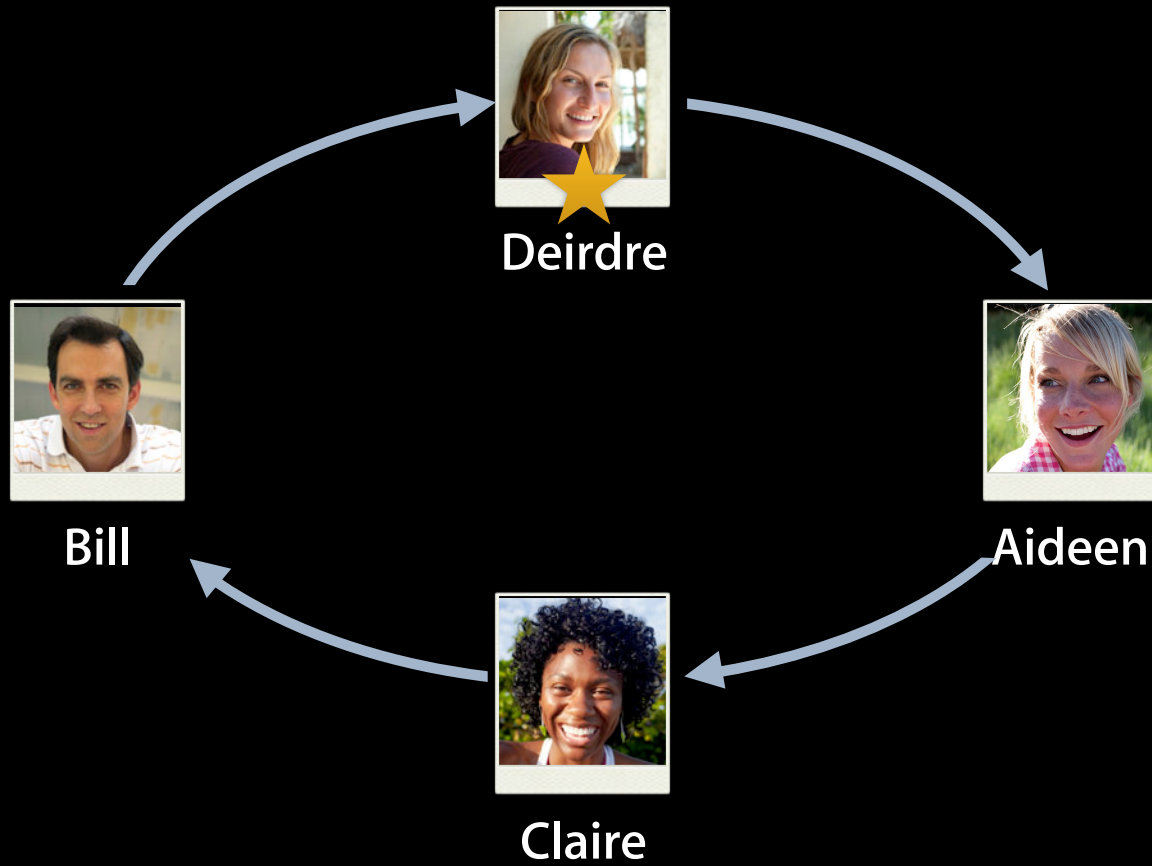
# Turn-Based Multiplayer



# Turn-Based Multiplayer



# Turn-Based Multiplayer



# Turn-Based Multiplayer

## Turn timeouts

- Specify order of participants
  - `endTurnWithNextParticipants:`
    - `turnTimeout:`
    - `matchData:`
    - `completionHandler:`
- Prevent abandoned games







# Game Groups



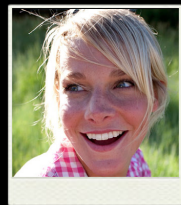
# Game Groups



# Game Groups



**Alliance**



**Armada**

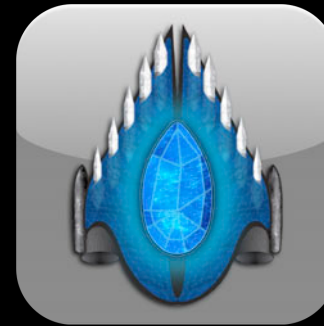


# Game Groups

Scores and Achievements



**Alliance**



**Armada**

# Game Groups

Scores and Achievements



**Alliance**



**Armada**



1

Discoverable

2

Polished

3

Fun

1

Discoverable

2

Polished

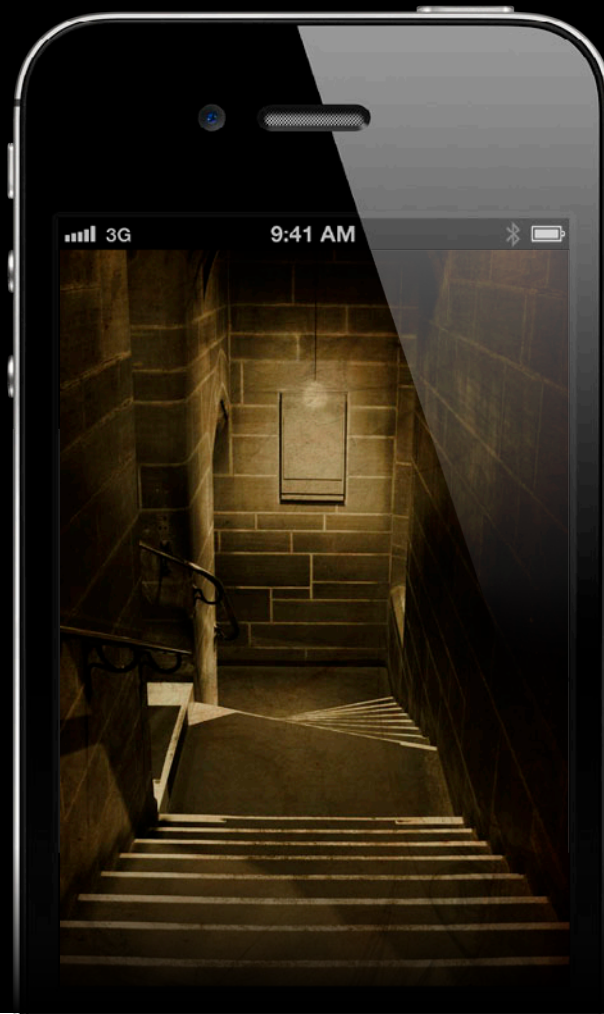
3

Fun

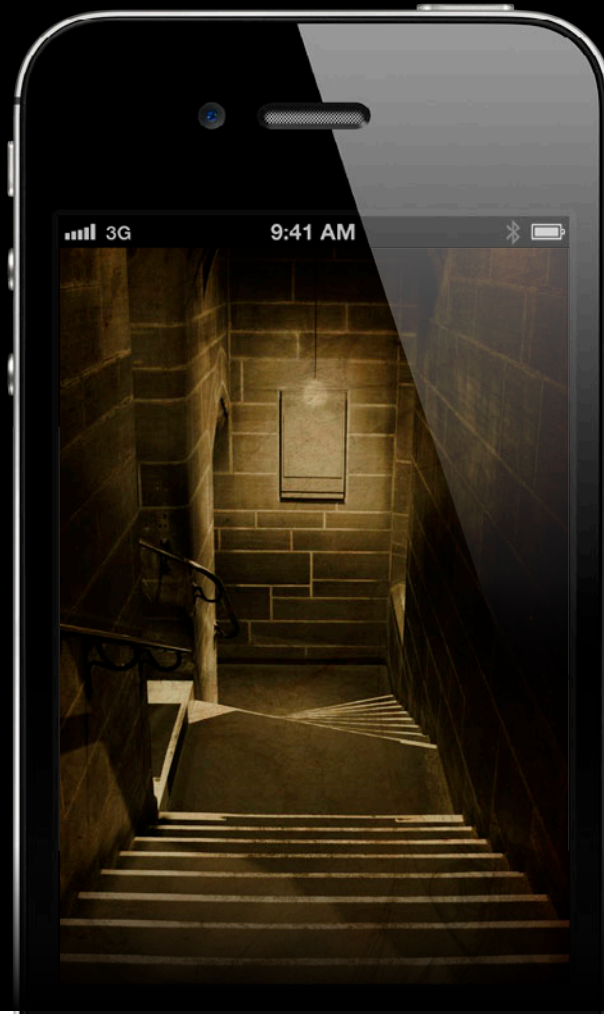


# Authentication

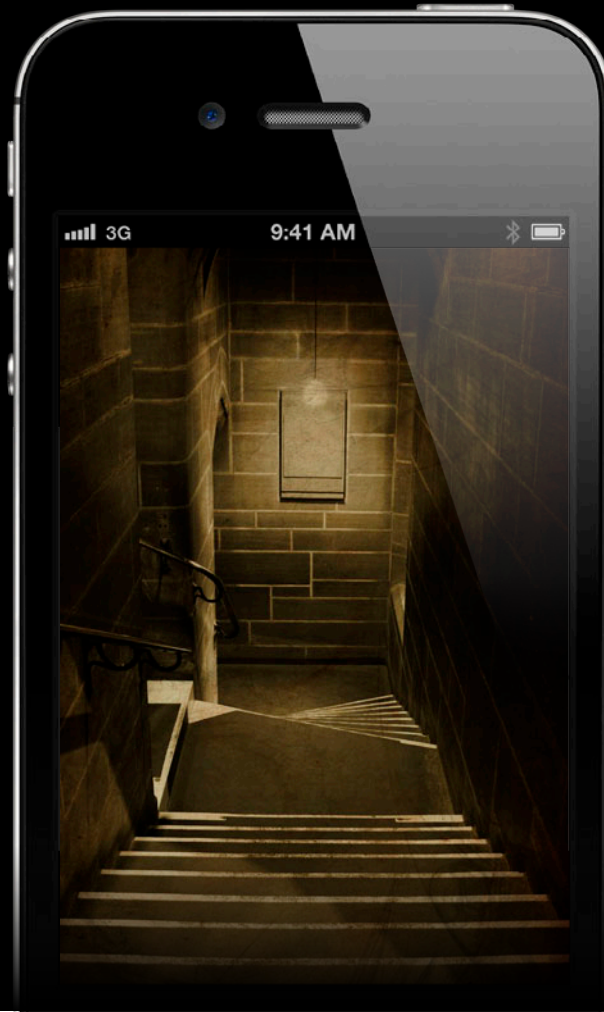
# Authentication



# Authentication



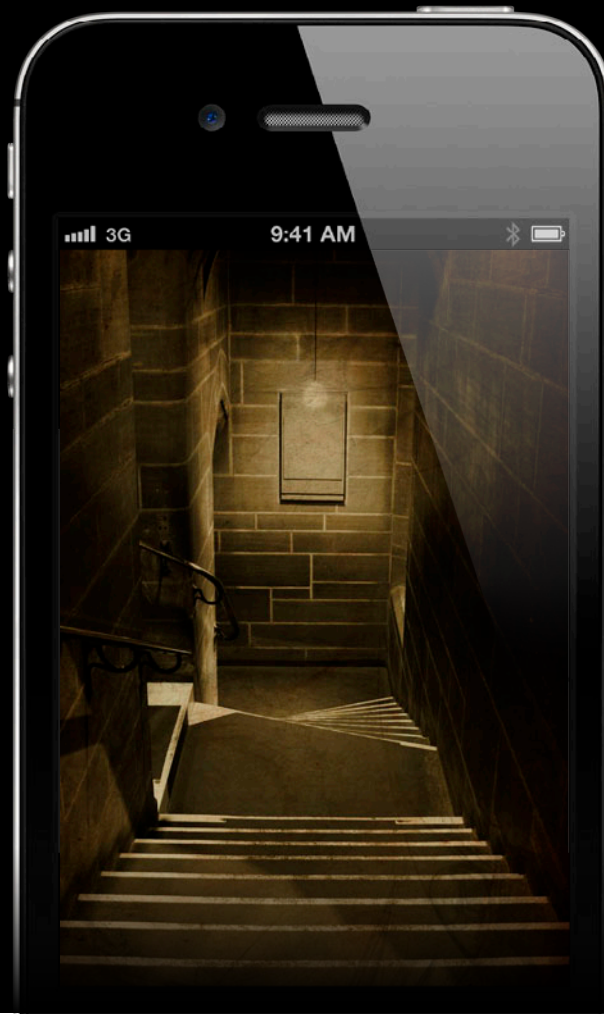
# Authentication



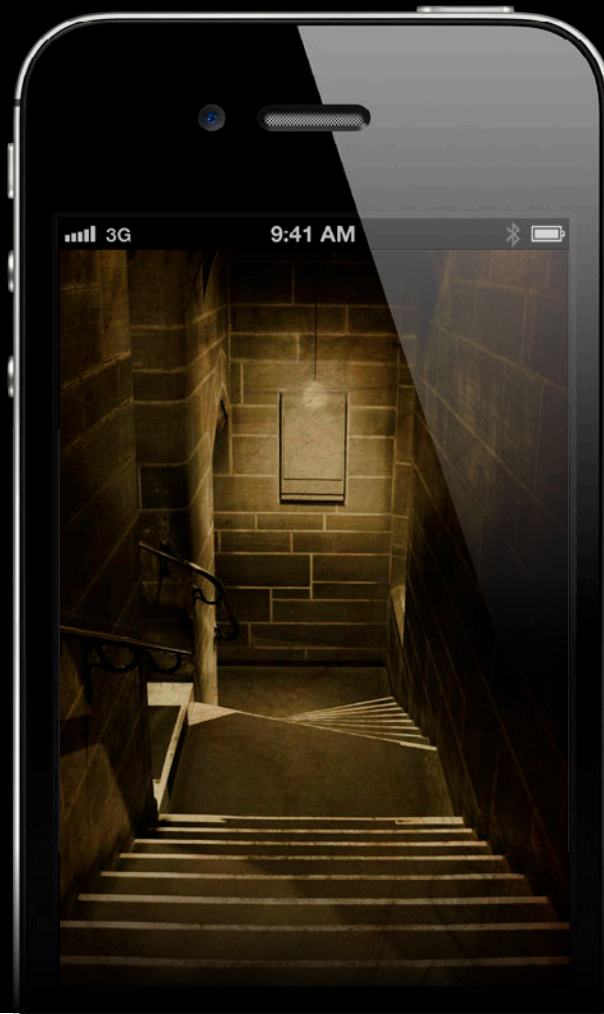
# Authentication



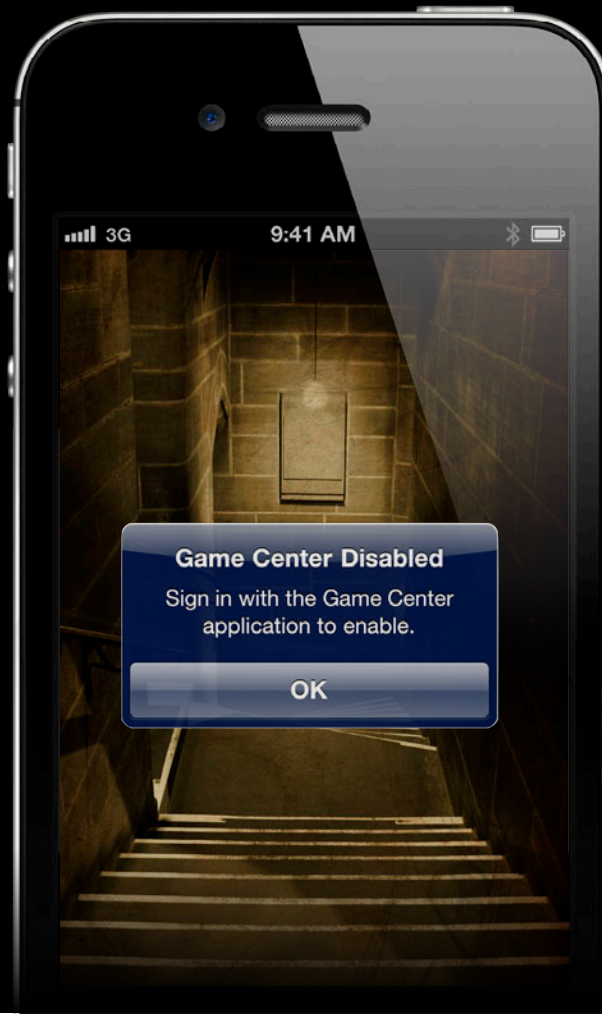
# Authentication



# Authentication

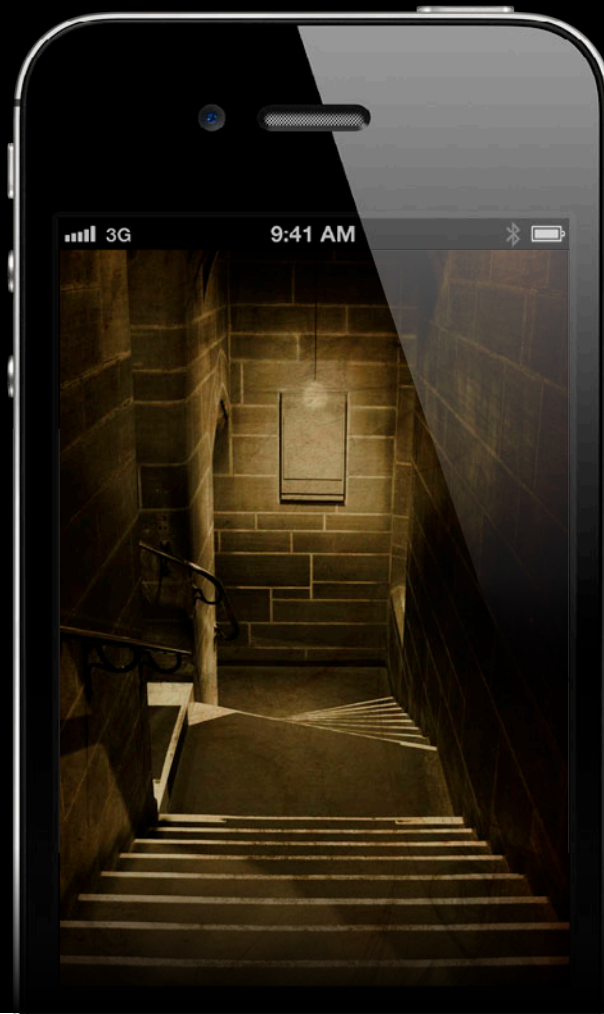


# Authentication





# Authentication



# Authentication



```
- (void)application:(UIApplication *)application
  didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    GKLocalPlayer *localPlayer = [GKLocalPlayer localPlayer];

    localPlayer.authenticationHandler = //...

    [localPlayer authenticate];
}
```

# Authentication



```
localPlayer.authenticationHandler = ^(UIViewController *viewController,  
    NSError *error) {  
  
    if (viewController) {  
        [self.viewController presentViewController:viewController  
            animated:YES  
            completion:nil];  
        return;  
    }  
  
    if (localPlayer.authenticated) {  
        [self enableGameCenterForPlayer:[GKLocalPlayer localPlayer]];  
    }  
};
```

# Authentication



```
localPlayer.authenticationHandler = ^(UIViewController *viewController,  
    NSError *error) {
```

```
    if (viewController) {  
        [self.viewController presentViewController:viewController  
            animated:YES  
            completion:nil];  
        return;  
    }
```

```
    if (localPlayer.authenticated) {  
        [self enableGameCenterForPlayer:[GKLocalPlayer localPlayer]];  
    }  
};
```

# Authentication



```
localPlayer.authenticationHandler = ^(UIViewController *viewController,  
    NSError *error) {  
  
    if (viewController) {  
        [self.viewController presentViewController:viewController  
            animated:YES  
            completion:nil];  
        return;  
    }  
  
    if (localPlayer.authenticated) {  
        [self enableGameCenterForPlayer:[GKLocalPlayer localPlayer]];  
    }  
};
```

**Unified Experience**

# Unified Experience



# Unified Experience





# Unified Experience



# Unified Experience



# Unified Experience



# Unified Experience



# Unified Experience



# Unified Experience

## GKGameCenterViewController



```
// Grab Unified Experience view controller
GKGameCenterViewController *viewController = [GKGameCenterViewController
    sharedController];

// Configure delegate
viewController.gameCenterDelegate = self;

// Present Unified Experience view controller
[self presentViewController:viewController animated:YES completion:nil];
```

# Unified Experience

## GKGameCenterViewController



```
// Grab Unified Experience view controller
GKGameCenterViewController *viewController = [GKGameCenterViewController
    sharedController];

// Configure delegate
viewController.gameCenterDelegate = self;

// Present Unified Experience view controller
[self presentViewController:viewController animated:YES completion:nil];
```

# Unified Experience

## GKGameCenterViewController



```
// Grab Unified Experience view controller
GKGameCenterViewController *viewController = [GKGameCenterViewController
    sharedController];

// Configure delegate
viewController.gameCenterDelegate = self;

// Present Unified Experience view controller
[self presentViewController:viewController animated:YES completion:nil];
```



# Unified Experience

## GKGameCenterViewController



```
// Grab Unified Experience view controller
GKGameCenterViewController *viewController = [GKGameCenterViewController
    sharedController];

// Configure delegate
viewController.gameCenterDelegate = self;

// Present Unified Experience view controller
[self presentViewController:viewController animated:YES completion:nil];
```

# Unified Experience

## GKGameCenterViewController



```
// Grab Unified Experience view controller
GKGameCenterViewController *viewController = [GKGameCenterViewController
    sharedController];

// Configure delegate
viewController.gameCenterDelegate = self;

// Configure leaderboard view
viewController.viewState = GKGameCenterViewControllerStateLeaderboards;
viewController.leaderboardTimeScope = GKLeaderboardTimeScopeToday;
viewController.leaderboardCategory = topScores;

// Present Unified Experience view controller
[self presentViewController:viewController animated:YES completion:nil];
```

# Unified Experience

## GKGameCenterViewController



```
// Grab Unified Experience view controller
GKGameCenterViewController *viewController = [GKGameCenterViewController
    sharedController];

// Configure delegate
viewController.gameCenterDelegate = self;

// Configure leaderboard view
viewController.viewState = GKGameCenterViewControllerStateLeaderboards;
viewController.leaderboardTimeScope = GKLeaderboardTimeScopeToday;
viewController.leaderboardCategory = topScores;

// Present Unified Experience view controller
[self presentViewController:viewController animated:YES completion:nil];
```

# Unified Experience

## GKGameCenterViewController



```
// Grab Unified Experience view controller
GKGameCenterViewController *viewController = [GKGameCenterViewController
    sharedController];

// Configure delegate
viewController.gameCenterDelegate = self;

// Configure achievement view
viewController.viewState = GKGameCenterViewControllerStateAchievements;

// Present Unified Experience view controller
[self presentViewController:viewController animated:YES completion:nil];
```

# Unified Experience

## GKGameCenterViewController



```
// Grab Unified Experience view controller
GKGameCenterViewController *viewController = [GKGameCenterViewController
    sharedController];

// Configure delegate
viewController.gameCenterDelegate = self;

// Configure achievement view
viewController.viewState = GKGameCenterViewControllerStateAchievements;

// Present Unified Experience view controller
[self presentViewController:viewController animated:YES completion:nil];
```

# Unified Experience

## GKGameCenterViewController



```
// Grab Unified Experience view controller
GKGameCenterViewController *viewController = [GKGameCenterViewController
    sharedController];

// Configure delegate
viewController.gameCenterDelegate = self;

// Configure challenges view
viewController.viewState = GKGameCenterViewControllerStateChallenges;

// Present Unified Experience view controller
[self presentViewController:viewController animated:YES completion:nil];
```

# Unified Experience

## GKGameCenterViewController



```
// Grab Unified Experience view controller
GKGameCenterViewController *viewController = [GKGameCenterViewController
    sharedController];

// Configure delegate
viewController.gameCenterDelegate = self;

// Configure challenges view
viewController.viewState = GKGameCenterViewControllerStateChallenges;

// Present Unified Experience view controller
[self presentViewController:viewController animated:YES completion:nil];
```

# Unified Experience





# Multiplayer

## Invite friends



# Multiplayer

## Invite recent players



# Multiplayer

Invite nearby players



# Unified Experience

## GKMatchmakerViewController

```
// Create match request
GKMatchRequest *matchRequest = [[GKMatchRequest alloc] init];
matchRequest.minPlayers = 2;
matchRequest.maxPlayers = 4;

// Create matchmaker view controller
GKMatchmakerViewController *controller = [[GKMatchmakerViewController alloc]
    initWithMatchRequest:matchRequest];

// Configure delegate
controller.matchmakerDelegate = self;

// Present matchmaker view controller
[viewController presentViewController:viewController animated:YES
    completion:nil];
```

# Unified Experience

## GKMatchmakerViewController

```
// Create match request
GKMatchRequest *matchRequest = [[GKMatchRequest alloc] init];
matchRequest.minPlayers = 2;
matchRequest.maxPlayers = 4;

// Create matchmaker view controller
GKMatchmakerViewController *controller = [[GKMatchmakerViewController alloc]
    initWithMatchRequest:matchRequest];

// Configure delegate
controller.matchmakerDelegate = self;

// Present matchmaker view controller
[viewController presentViewController:viewController animated:YES
    completion:nil];
```

# Unified Experience

## GKMatchmakerViewController

```
// Create match request
GKMatchRequest *matchRequest = [[GKMatchRequest alloc] init];
matchRequest.minPlayers = 2;
matchRequest.maxPlayers = 4;
```

```
// Create matchmaker view controller
GKMatchmakerViewController *controller = [[GKMatchmakerViewController alloc]
    initWithMatchRequest:matchRequest];
```

```
// Configure delegate
controller.matchmakerDelegate = self;
```

```
// Present matchmaker view controller
[viewController presentViewController:viewController animated:YES
    completion:nil];
```

# Unified Experience

## GKMatchmakerViewController

```
// Create match request
GKMatchRequest *matchRequest = [[GKMatchRequest alloc] init];
matchRequest.minPlayers = 2;
matchRequest.maxPlayers = 4;

// Create matchmaker view controller
GKMatchmakerViewController *controller = [[GKMatchmakerViewController alloc]
    initWithMatchRequest:matchRequest];

// Configure delegate
controller.matchmakerDelegate = self;

// Present matchmaker view controller
[viewController presentViewController:viewController animated:YES
    completion:nil];
```

# Unified Experience

## GKMatchmakerViewController

```
// Create match request
GKMatchRequest *matchRequest = [[GKMatchRequest alloc] init];
matchRequest.minPlayers = 2;
matchRequest.maxPlayers = 4;

// Create matchmaker view controller
GKMatchmakerViewController *controller = [[GKMatchmakerViewController alloc]
    initWithMatchRequest:matchRequest];

// Configure delegate
controller.matchmakerDelegate = self;

// Present matchmaker view controller
[viewController presentViewController:viewController animated:YES
    completion:nil];
```



1

Discoverable

2

Polished

3

Fun

1

Discoverable

2

Polished

3

Fun



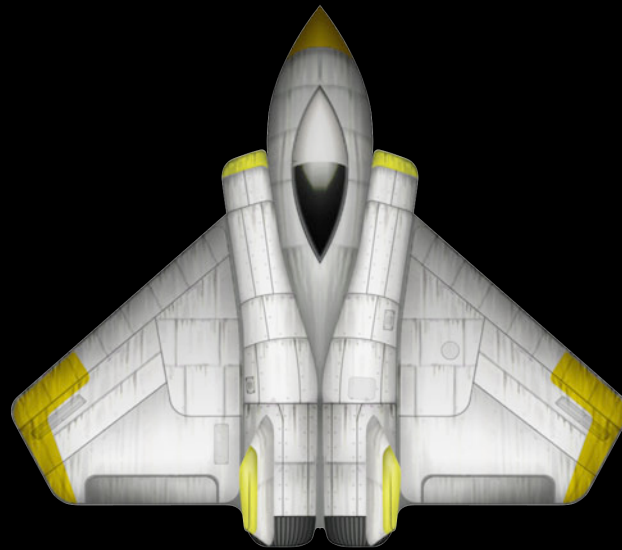
# Scores in Context

GKScore.context is 64 bits of goodness



# Scores in Context

As a bit field



# Scores in Context

As a bit field



# Scores in Context

As a bit field



# Scores in Context

As a bit field





# Scores in Context

As a bit field



# Scores in Context

As a bit field



# Scores in Context

"Ghost Lap"



# Scores in Context

"Ghost Lap"

<http://.../C515>



+ Context = Cool



+ Context + Challenges

**Awesome**

**CHALLENGE:** FINISH IN UNDER 3:15



PLAY

DECLINE



**CHALLENGE:** FINISH IN UNDER 3:15



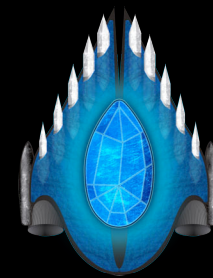
PLAY

DECLINE

# Game Groups



**Alliance**

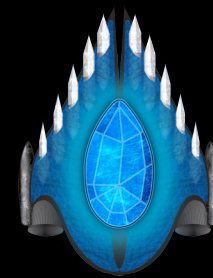


**Armada**

# Game Groups



**Alliance**

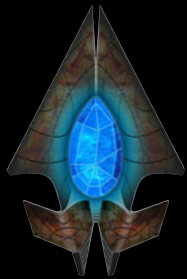


**Armada**

**Host**



# Game Groups



**Alliance**



**Armada**



**Host**



# Party Games with AirPlay



Aileen



Bill



Claire



Deirdre

# Party Games with AirPlay



Aileen



Mike



Karla



Clark

# Save to iCloud



# Save to iCloud





# Save to iCloud



Use **ID of Game Group**  
instead of **Bundle ID**

1

Discoverable

2

Polished

3

Fun

1

Discoverable

2

Polished

3

Fun

1

Discoverable

2

Polished

3

Fun

# More Information

## Allan Schaffer

Graphics and Game Technologies Evangelist  
[aschaffer@apple.com](mailto:aschaffer@apple.com)

## Documentation

Game Center for Developers  
<http://developer.apple.com/devcenter/ios/gamecenter>

## Apple Developer Forums

<http://devforums.apple.com>

# Related Sessions

Integrating Your Games with Game Center

Pacific Heights  
Wednesday 4:30PM

What's New in iTunes Connect for App Developers

Nob Hill  
Thursday 9:00AM

Multiplayer Gaming with Game Center

Pacific Heights  
Thursday 10:15AM

Building Game Center Games for OS X

Pacific Heights  
Thursday 11:30AM

# Labs

Game Center Lab

Graphics, Media & Games Lab B  
Thursday 2:00PM

Game Center Lab

Graphics, Media & Games Lab C  
Friday 9:00AM

 **WWDC2012**