Article

Displaying the Game Center dashboard

Provide an interface for players to navigate to their Game Center data from your game.

Overview

The *dashboard* provides a central location from which players can browse and manage their Game Center data. The player can access information about their profile, as well as leaderboards, achievements, and challenges. The dashboard also displays highlights in these areas as the players navigate.

Use the `GKGameCenterViewController` class to present the dashboard in a specific state — such as the *main dashboard*,_ _from which the player can navigate to any area, including the player profile or a list of leaderboards. For visionOS games, the dashboard appears anchored to the window, scene, or view relative to where you present the view controller.

Alternatively, to add the access point to a fixed location in your game that allows the player to open the dashboard whenever they want, see Adding an access point to your game.

For design guidance, see Human Interface Guidelines > Technologies > Game Center > Custom dashboard links.
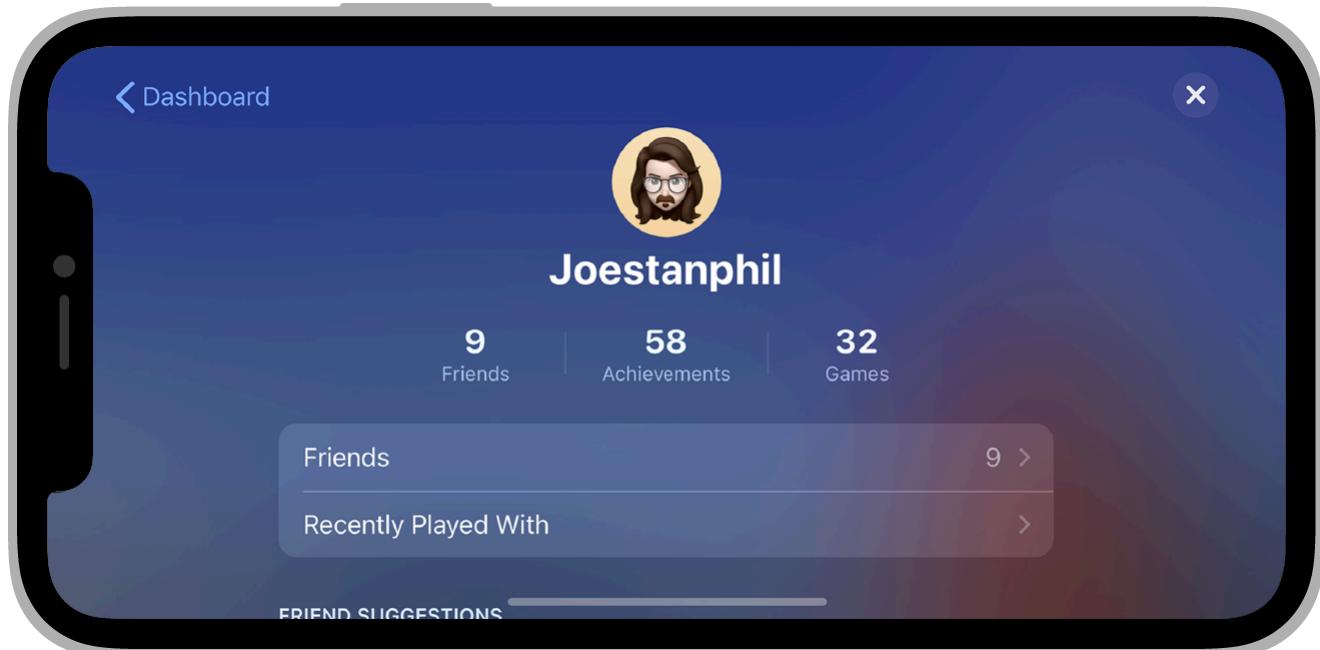
Present the main dashboard

To display the main dashboard, pass `dashboard` as the state parameter to the `init(state:)` method when you create the view controller, set the delegate of the view controller to your object, and then present it:

```swift
// Display the dashboard.
let viewController = GKGameCenterViewController(state: .dashboard)
viewController.gameCenterDelegate = self
present(viewController, animated: true, completion: nil)
```

Show the player's profile

In the profile area, the player can see their achievements, find friends, and see what games their friends are playing. They can also edit their Game Center settings, such as their nickname and avatar.

To present the local player's profile, pass `localPlayerProfile` as the state parameter to the `init(state:)` method:

```
// Display the player's profile.
let viewController = GKGameCenterViewController(state: .localPlayerProfile)
viewController.gameCenterDelegate = self
present(viewController, animated: true, completion: nil)
```

Present the list of leaderboards

You can display a list of leaderboards in the dashboard that allows the player to navigate to the leaderboards they want to see. When the player selects a leaderboard, the dashboard displays the details for that leaderboard.
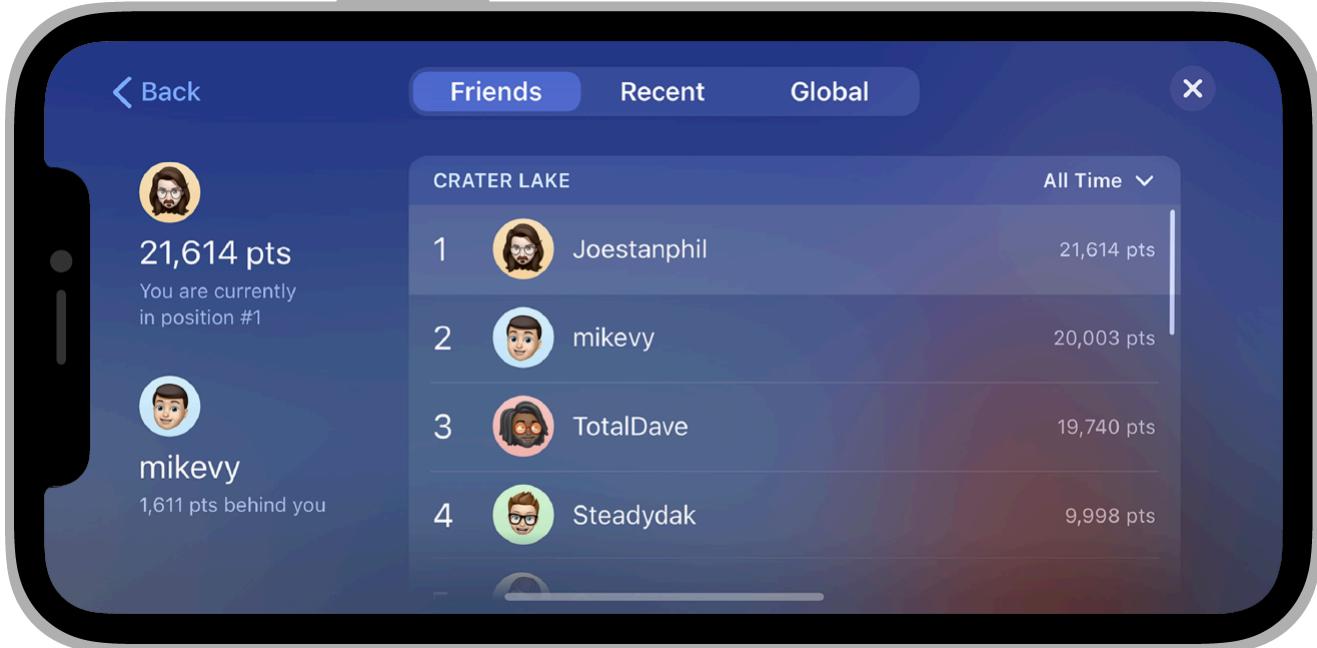
To present a list of leaderboards, pass `leaderboards` as the state parameter to the `init(state:)` method:

```
// Display a list of leaderboards.
let viewController = GKGameCenterViewController(state: .leaderboards)
viewController.gameCenterDelegate = self
present(viewController, animated: true, completion: nil)
```

Display a single leaderboard

You can present a specific leaderboard in the dashboard that allows players to see how they rank against friends, players with whom they played recently, and players from all over the world. They can use the filter in the header area to adjust the scope of the leaderboard by a time period, or show a specific occurrence of a recurring leaderboard. They can scroll to the top of the list by tapping on the name in the header.
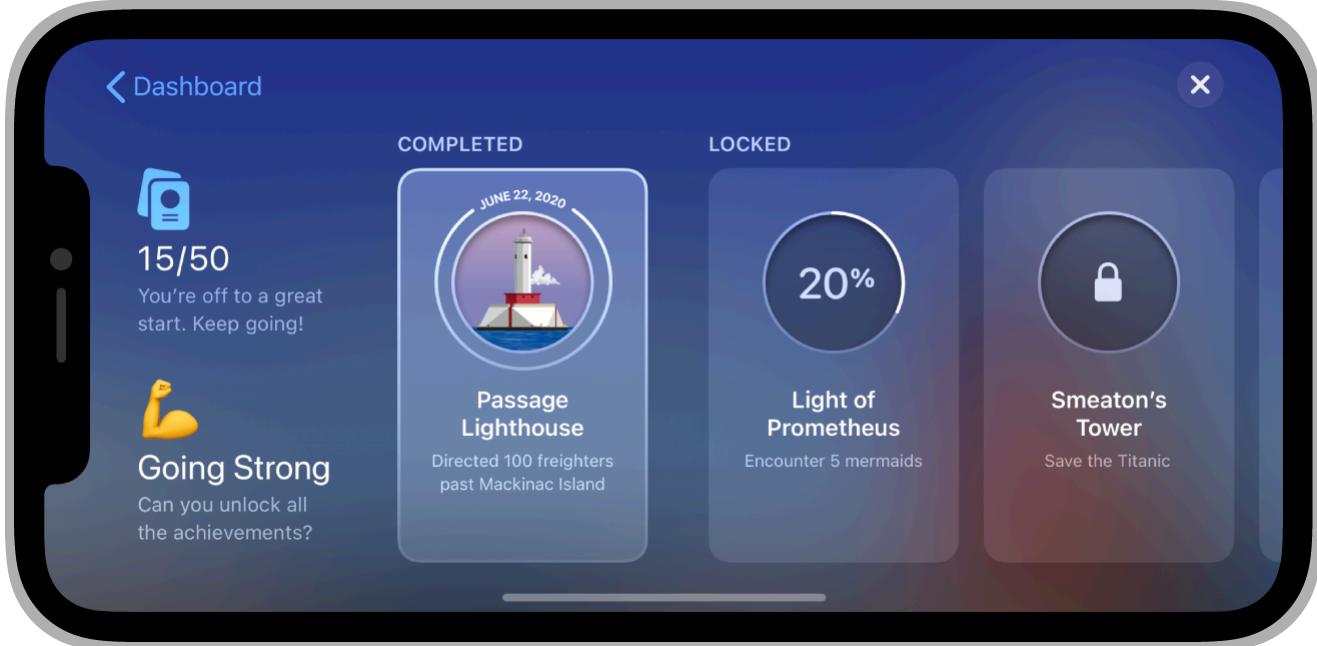
To present a single leaderboard, pass the leaderboard ID (the identifier you entered in App Store Connect when creating the leaderboard) along with the player scope and time scope when you create the view controller:

```
// Display scores for a specific leaderboard.
let viewController = GKGameCenterViewController(
                leaderboardID: "grp.xyz.laketahoe",
                playerScope: .global,
                timeScope: .allTime)
viewController.gameCenterDelegate = self
present(viewController, animated: true, completion: nil)
```

The player can filter and scope scores by player (Friends, Recent, or Global) and time period (for example, All Time).

Display the player's achievements

You can show the player a list of achievements they received and achievements they've yet to complete. An *achievement* is a collectible item indicating that the player successfully reached a particular goal in your game. In the dashboard, an achievement appears as either locked, in-progress, completed, or hidden.

To present the achievements, pass `achievements` as the state parameter to the [init(state:)](#) method:

```
// Display a list of achievements.
let viewController = GKGameCenterViewController(state: .achievements)
viewController.gameCenterDelegate = self
present(viewController, animated: true, completion: nil)
```

Dismiss the dashboard

Implement the [gameCenterViewControllerDidFinish(_:)](#) delegate method in the [GKGameCenterControllerDelegate](#) protocol to dismiss the dashboard when the player closes it.

```
func gameCenterViewControllerDidFinish(_ gameCenterViewController: GKGameCenterViewC
    // Dismiss the view controller.
    gameCenterViewController.dismiss(animated:true)
}
```

See Also

Game Center interfaces

📄 Adding an access point to your game

Provide your users a convenient connection to the Game Center dashboard.

`class` GKAccessPoint

An object that allows players to view and manage their Game Center information from within your game.

`class` GKDialogController

An object that provides the ability to present the dashboard in macOS games.

`protocol` GKViewController

The abstract base protocol adopted by GameKit view controller classes.