

[GameKit](#) / Adding an access point to your game

Article

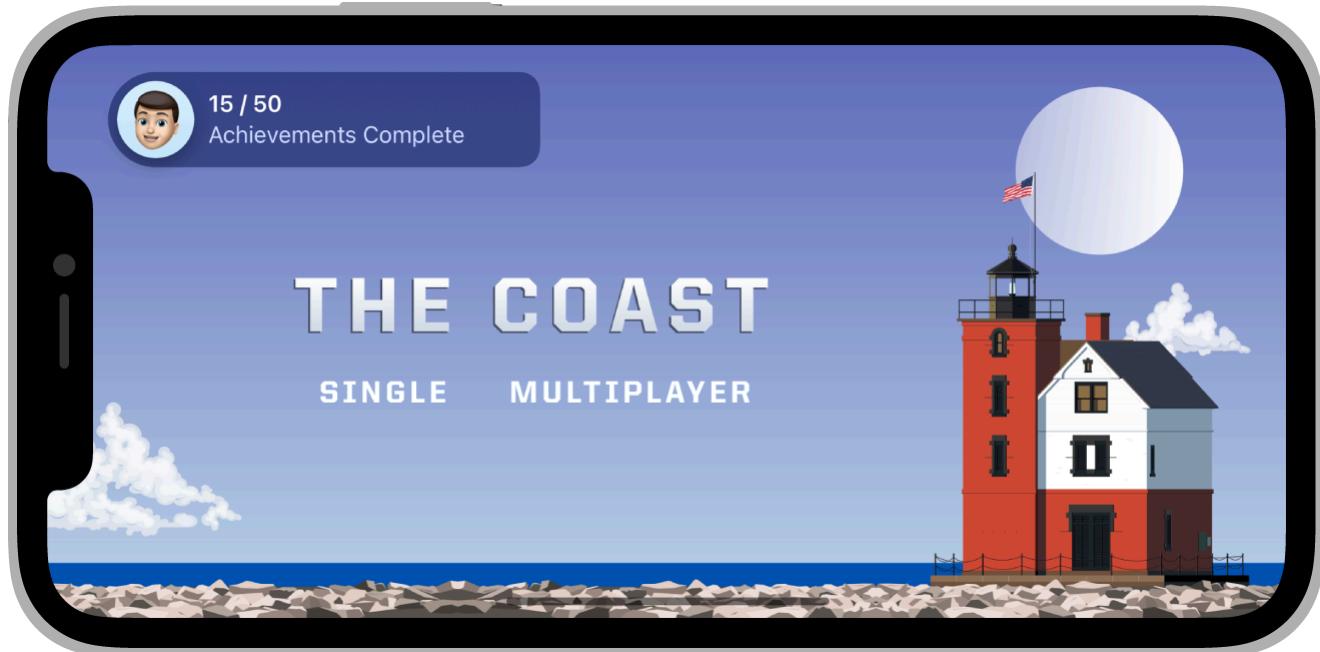
# Adding an access point to your game

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

## Overview

You can add an *access point* to your game that provides a way for players to manage their profile, and view leaderboards, achievements, and challenges.

The access point initially shows player highlights, like how many achievements they've earned and where they stand in leaderboards. Then the access point collapses into the player's avatar and remains on your game's screen. When the player taps or clicks the avatar, GameKit displays the dashboard so players can drill down into the details of their Game Center data, including more highlights and statistics. You can choose where to display the access point, and select the highlights to display.



You can also display the dashboard on its own, or a particular section of the dashboard. For more information, see [Displaying the Game Center dashboard](#). For design guidance, see [Human](#)

## Configure the access point

You can place the access point in a corner of the screen and choose whether to show highlights when the access point first appears, such as the number of achievements or the player's rank on the default leaderboard. Start by obtaining the shared `GKAccessPoint` object using the `shared` class property.

If you don't want the access point to appear on the main window, set it to another window using the `parentWindow` property. For example, set the parent window to your title screen or main menu. For volumetric games on visionOS, you need to set the parent window for the access point to appear. For immersive games on visionOS, it appears below the HUD and in front of the person by default.

Use the `location` property to set the corner in which the access point appears. The default location is the upper-left corner (`GKAccessPoint.Location.topLeading`). For volumetric and immersive games on visionOS, if you don't set the parent window, GameKit ignores the `location` property (see [Configure the access point on visionOS](#)).

```
// Place the access point on the upper-left corner.  
GKAccessPoint.shared.location = .topLeading
```

For right-to-left languages, such as Arabic and Hebrew, the system flips the location. For example, `GKAccessPoint.Location.topLeading` specifies the upper-right corner and `GKAccessPoint.Location.bottomLeading` specifies the bottom-left corner.

To display highlights when the access point appears, set the `showHighlights` property to `true`:

```
GKAccessPoint.shared.showHighlights = true
```

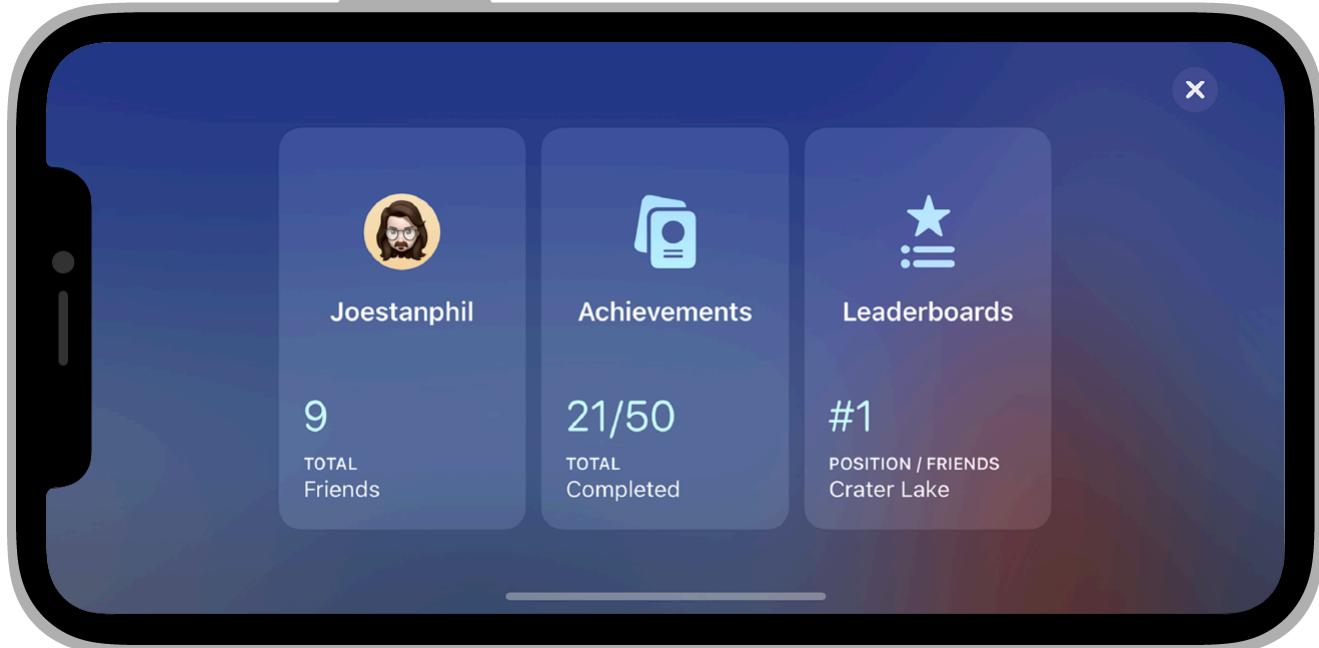
When you're done configuring the access point, set the `isActive` property to `true`:

```
GKAccessPoint.shared.isActive = true
```

Hide the access point when displaying game intros or settings. To give your players a consistent experience, see [Access Point](#) in the [Human Interface Guidelines](#).

## Adapt your game to the access point

You can observe access point properties to adjust your game when the access point either changes sizes or just before it presents the dashboard.



To adjust your graphics when the access point size changes (for example, while it displays highlights), observe the `frameInScreenCoordinates` property. Be sure to convert the screen coordinates of the access point to your view's coordinates.

```
// Observe when the access point changes its size.  
let observation = GKAccessPoint.shared.observe(  
    \.frameInScreenCoordinates  
) { [weak self] _,_ in  
    let screenFrame = GKAccessPoint.shared.frameInScreenCoordinates  
    let accessPointFrame = myView.convert(screenFrame, from: nil)  
    // Adjust your layout.  
}
```

To make changes to your game before the player interacts with the dashboard, observe the `isPresentingGameCenter` property. For example, you should pause animations when the player clicks or taps the access point to display the dashboard.

```
// Observe when the access point displays the dashboard.  
let observation = GKAccessPoint.shared.observe(  
    \.isPresentingGameCenter  
) { [weak self] _,_ in  
    self.paused = GKAccessPoint.shared.isPresentingGameCenter  
}
```

# Change the focus and display the dashboard programmatically

If you handle the input from an Apple TV remote and game controller yourself, you need to programmatically change the focus to the access point and display the dashboard when the player taps or clicks it.

You can use the `frameInScreenCoordinates` property to change the focus when the player navigates the area behind the access point:

```
// Track and update focus.  
func trackController(position: CGPoint) {  
    let screenFrame = GKAccessPoint.shared.frameInScreenCoordinates  
    let accessFrame = myView.convert(screenFrame, from: nil)  
    // If the point is in the access point, turn on feedback.  
    accessPointElement.focusFeedback = CGRectContainsPoint(accessFrame, position)  
}
```

Then use the `trigger(state:handler:)` method to show the dashboard when the player selects the access point.

## Configure the access point on visionOS

The location of the access point and dashboard varies depending on the type of visionOS game. To change the default behavior below, set the `location` and `parentWindow` properties before you set `isActive` to `true`. For example, if you don't set the parent window of volumetric games, the access point doesn't appear.

Type	Access point behavior
Compatible	Appears over the main or parent window in the specified location.
Native	Appears outside of the main or parent window in the specified location.
Volumetric	If the parent window is non-nil, appears outside of it in the specified location. If the parent window is nil (the default value), it doesn't appear.
Immersive	If the parent window is non-nil, appears outside of it in the specified location. If the parent window is nil (the default value), it appears below the HUD in front of the person and tracks their head position.

For all types of visionOS games, when the person displays the dashboard using the access point, it appears in a separate floating container app that the person can position in the space. However, if you use the [GKGameCenterViewController](#) class to present the dashboard (see [Displaying the Game Center dashboard](#)), it appears anchored to the window, scene, or view relative to where you present the view controller.

---

## See Also

### Game Center interfaces

 [Displaying the Game Center dashboard](#)

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

`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.