

[GameKit](#) / Creating recurring leaderboards

Article

# Creating recurring leaderboards

Create a leaderboard for your game that ranks player scores based on a schedule.

## Overview

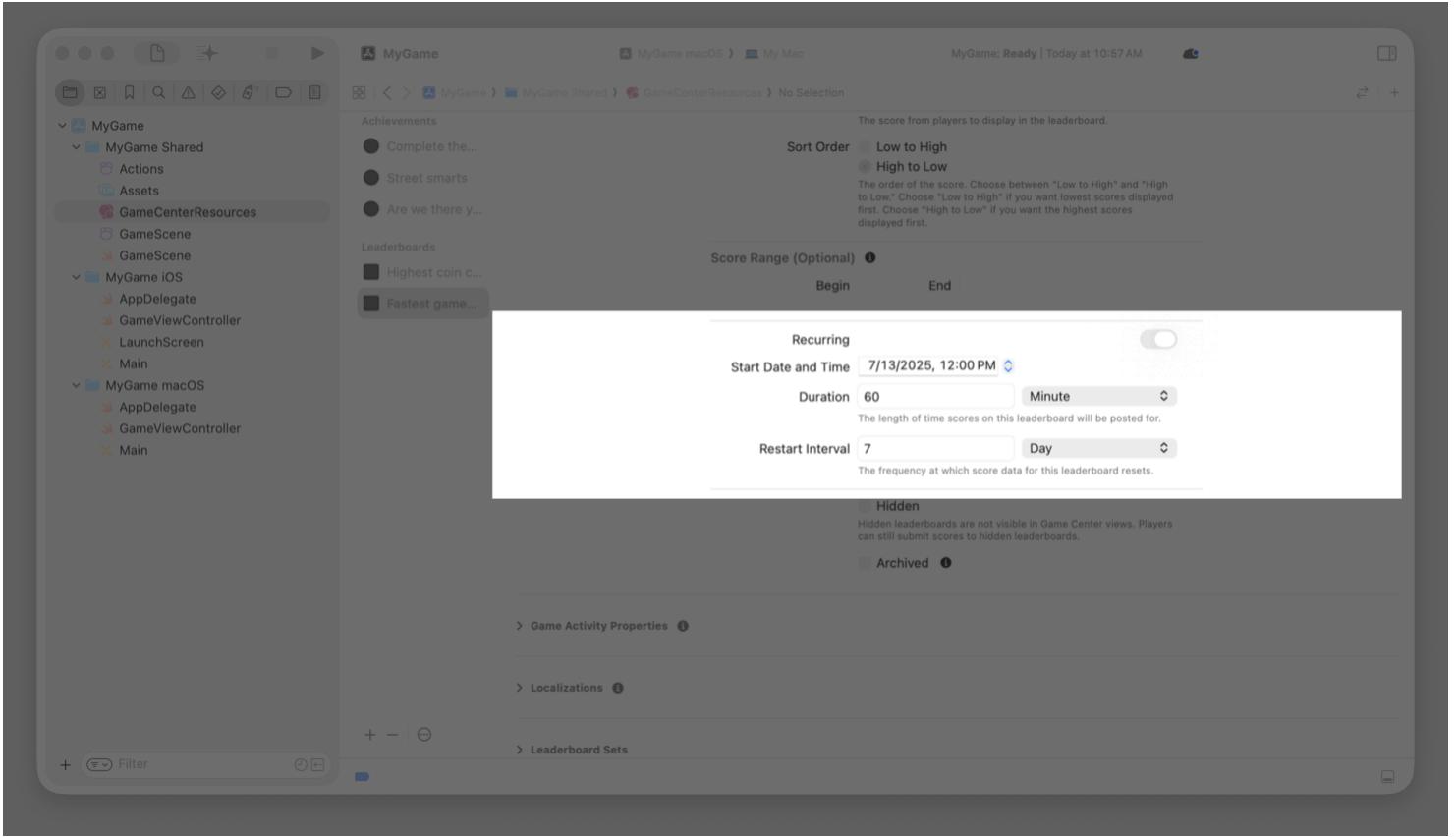
Use a *recurring leaderboard* to organize regular competitions or encourage players to score higher in your game. Unlike a classic leaderboard that never resets, a recurring leaderboard represents score rankings during a period of time. Each leaderboard in the sequence is called an occurrence.

For design guidance, see [Human Interface Guidelines > Technologies > Game Center > Leaderboards](#).

## Configure and test recurring leaderboards

Configure recurring leaderboards in Xcode before accessing them in your code and testing locally with Game Progress Manager. When you're ready to deploy your configuration, sync your updates with App Store Connect. For more information about configuring and testing Game Center features, see [Initializing and configuring Game Center](#).

You configure a recurring leaderboard like a classic leaderboard, but you also enable the Recurring setting, and configure time-related properties. Under the Recurring section, set a start date for the first occurrence. Then, enter the duration for each occurrence to establish the period in which players can earn scores. To specify the frequency of the occurrences, enter a restart interval. Occurrences are sequential and don't overlap, so the restart interval must be equal to or greater than the duration. To create a time delay between occurrences, set the restart interval to a number that is greater than the duration.



For example, if the restart interval and duration are both 24 hours, the recurring leaderboard runs daily with no gaps between occurrences. To create a 1-hour contest every Sunday at noon, set the start date to Sunday at noon, then set the restart interval to 7 days and the duration to 60 minutes. To create a 15-minute competition every hour, set the restart interval to 60 minutes and the duration to 15 minutes.

### Note

If you've already pushed your configuration changes to App Store Connect, removing a leaderboard or leaderboard set from the local configuration file doesn't remove the leaderboard or leaderboard set from App Store Connect.

Use the Game Progress Manager to test leaderboards on your local device before you push the configuration update to App Store Connect. After selecting a leaderboard, you can add players to the leaderboard with a score value. Click Reset Leaderboards to simulate what happens in your game when the leaderboard occurrence ends. You can't access previous occurrences of a recurring leaderboard in Debug Mode.

To learn more about the information you enter in App Store Connect, see [Leaderboard properties](#).

## Submit a score to the current occurrence

Use the leaderboard ID you entered in Xcode to specify the current occurrence when submitting a score. Don't submit scores to past occurrences leaderboards.

To submit a score to one or more leaderboards, including recurring leaderboards, use the [submitScore\(\\_ :context:player:leaderboardIDs:completionHandler:\)](#) class method in [GKLeaderboard](#). Pass one or more leaderboard IDs, the score, context, and player. The context parameter is an optional value for game-specific data that you can store, and fetch later with the score. If there's an occurrence at the time you submit the score, the occurrence retains either the best score or the most recent score.

```
// Submit a score to one or more leaderboards
try await GKLeaderboard.submitScore(points,
                                      context: 0,
                                      player: GKLocalPlayer.local,
                                      leaderboardIDs: ["my.recurring.leaderboard.id"])
```

Alternatively, submit the score to the recurring leaderboard by loading the leaderboard using the [loadLeaderboards\(IDs:completionHandler:\)](#) method in [GKLeaderboard](#). Then use the [submitScore\(\\_ :context:player:completionHandler:\)](#) instance method to submit the score to the recurring leaderboard.

```
// Submit a score to the current occurrence of a recurring leaderboard
let leaderboards = try await GKLeaderboard.loadLeaderboards(IDs: ["my.recurring.leaderboard"])
guard let leaderboard = leaderboards.first,
      let endDate = leaderboard.startDate?.addingTimeInterval(leaderboard.duration)
try await GKLeaderboard.submitScore(points,
                                      context: 0,
                                      player: GKLocalPlayer.local,
                                      leaderboardIDs: ["my.recurring.leaderboard.id"])
```

#### Note

You can only use the [submitScore\(\\_ :context:player:completionHandler:\)](#) instance method for recurring leaderboards.

However, submitting a score with this method fails if the leaderboard isn't active, so check the start date and duration properties of the leaderboard before calling the [submitScore\(\\_ :context:player:completionHandler:\)](#) method. To get the start date of the next occurrence, use the [nextStartDate](#) property.

# Access the previous occurrence

If you want the scores and rankings from the previous occurrence, you can get the occurrence using the `loadPreviousOccurrence(completionHandler:)` method in `GKLeaderboard`. First, load the current occurrence using the `loadLeaderboards(IDs:completionHandler:)` class method, then call `loadPreviousOccurrence(completionHandler:)` to load the previous occurrence.

```
// Load current occurrence of a recurring leaderboard
let leaderboards = try await GKLeaderboard.loadLeaderboards(IDs: ["my.recurring.leaderboard"])

if let current = leaderboards.first {
    // Load previous occurrence using the current occurrence.
    try await current.loadPreviousOccurrence()

    // Do something with the previous occurrence.
}
```

## Note

Game Center keeps expired occurrences for up to 30 days. However, a player can view scores only on the current occurrence and one previous occurrence. The previous occurrence is the most recent expired occurrence in which the player submitted a score.

## See Also

### Leaderboards

- 📄 Encourage progress and competition with leaderboards
  - Let players measure their own progress and compare their skills with friends and others.
- {} Adding Recurring Leaderboards to Your Game
  - Encourage competition in your games by adding leaderboards that have a duration and repeat.

`class GKLeaderboard`

A leaderboard for a game that Game Center stores.

`class GKLeaderboardSet`

Organizes leaderboards into logical and coherent groups.

```
class GKLeaderboardScore
```

Information about a player's score on a leaderboard.