

Framework

GameKit

Enable players to interact with friends, compare leaderboard ranks, earn achievements, and participate in multiplayer games.

iOS 3.0+ | iPadOS 3.0+ | Mac Catalyst 13.1+ | macOS 10.8+ | tvOS 9.0+ | visionOS 1.0+ | watchOS 3.0+

Overview

Use the GameKit framework to implement Game Center social-gaming network features. Game Center is an Apple service that provides a single account that identifies players across all their games and devices. After players sign in to Game Center on their device, they can access their friends and use Game Center features you implement.



Before you can use GameKit classes, you must enable Game Center in your project and initialize the local player in your code; otherwise, your game receives a [GKError.Code.not](#)

Authenticated error.

If you have an existing Unity project, you can access the GameKit framework using the [Apple Unity Plug-ins](#).

Implement Game Center features

After you've enabled Game Center, you can implement many useful features to enhance the gaming experience.

You can add leaderboards that let players see how well they rank amongst friends and players all over the world. Create recurring leaderboards to organize regular competitions that provide players more chances to earn the top score. As players progress through your game, you can reward them with achievements that encourage them to keep playing.

GameKit supports real-time and turn-based multiplayer experiences. Players can choose automatic matching or invite their friends to join a game. You can support turn-based gaming in which a match plays out over a series of alternating turns, and players can receive invitations even when your game isn't in the foreground.

GameKit also provides user interface components for your players to see highlights and access their Game Center data directly in your game. The access point provides a way for players to open a dashboard in which they can browse their profile, leaderboards, and achievements, as well as manage their friends list.

For designing Game Center features in your app, see [Human Interface Guidelines > Technologies > Game Center](#).

Topics

Essentials

Initializing and configuring Game Center

Enable Game Center, configure features, and test them locally in your Xcode project.

Authenticating a player

Confirm player credentials and device capabilities and check for account restrictions.

Improving the player experience for games with large downloads

Provide ample content in your base installation and then use on-demand resources and the Background Assets API to handle additional content.

A Boolean value that indicates whether users of the app may see and compare achievements on a leaderboard, invite friends, and start multiplayer games.

Players

Connecting players with their friends in your game

Give players the ability to connect and interact with friends in your game.

Saving the player's game data to an iCloud account

Save game data during play or after a game in the player's iCloud account that's accessible from any device.

Protecting the player's privacy using scoped identifiers

Use the scoped identifiers that GameKit provides you as player IDs when transmitting or saving player data.

`class GKLocalPlayer`

The local player who signs in to Game Center on the device running the game.

`class GKPlayer`

A remote player who the local player running your game can invite and communicate with through Game Center.

`class GKBasePlayer`

A class that provides common data and methods for the different player objects.

`protocol GKLocalPlayerListener`

A protocol that handles events for Game Center players.

`static let GKPlayerAuthenticationDidChangeNotificationName: NSNotification.Name`

A notification that posts after GameKit authenticates the local player.

`static let GKPlayerDidChangeNotificationName: NSNotification.Name`

A notification that posts when a player object's data changes.

Game Center interfaces

Adding an access point to your game

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

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.

Leaderboards

Encourage progress and competition with leaderboards

Let players measure their own progress and compare their skills with friends and others.

Creating recurring leaderboards

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

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.

Achievements

Rewarding players with achievements

Use achievements to motivate players and engage them more in your game.

`class GKAchievement`

An achievement you can award a player as they make progress toward and reach a goal in your game.

`class GKAchievementDescription`

An object containing the text and artwork used to present an achievement to a player.

Challenges

Creating engaging challenges from leaderboards

Encourage friendly competition by adding challenges to your game.

Choosing a leaderboard for your challenges

Understand what gameplay works well when configuring challenges in your game.

`class GKChallengeDefinition`

An object that represents the static metadata you define for the challenge.

~~GKShowChallengeBanners~~

A Boolean value that indicates whether GameKit can display challenge banners in a game.

Deprecated

Activities

Creating activities for your game

Use activities to surface game content to players and encourage them to connect with each other.

`class GKGameActivity`

An object that represents a single instance of a game activity for the current game.

`class GKGameActivityDefinition`

An object that represents the static metadata you define for the activity.

`protocol GKGameActivityListener`

An object that responds to activity events.

Real-time games

Creating real-time games

Develop games where multiple players interact in real time.

Finding multiple players for a game

Discover and invite other players to participate in a real-time game.

Exchanging data between players in real-time games

Send data between players in a real-time multiplayer game.

Adding voice chat to multiplayer games

Enable players to voice chat with all, or groups of, players in a multiplayer game.

Matchmaking rules

Game Center applies different type of rules you create in a particular order to find the best matches.

`class GKMatchRequest`

An object that encapsulates the parameters to create a real-time or turn-based match.

`class GKMatchmaker`

An object that creates matches with other players without presenting an interface to the players.

`class GKMatchmakerViewController`

An interface that allows a player to invite other players to a real-time game and automatch to fill any empty slots.

`protocol GKInviteEventListener`

A protocol that handles invite events from Game Center.

`class GKInvite`

An invitation to join a match sent to the local player from another player.

`class GKMatch`

A peer-to-peer network between a group of players that sign into Game Center.

Turn-based games

Creating turn-based games

Develop games where multiple players take turns and can exchange data while waiting for their turn.

Starting turn-based matches and passing turns between players

Let Game Center store and forward match data between players in a turn-based game.

File: Sending messages to players in turn-based games

Notify players of match events by sending messages and game data.

File: Exchanging data between players in turn-based games

Add the ability for players to exchange game data and send messages while waiting for their turns.

`class GKTur`

An interface that allows a player to invite other players to a turn-based match and automatch to fill any empty slots.

`class GKTur`

An object that encapsulates the match data for games where players take turns.

`class GKTur`

A participant in a turn-based match.

`protocol GKTur`

The protocol that handles turn-based and data-exchange events between participants in a match.

`class GKTur`

Exchange request information that participants send in a turn-based match.

`class GKTur`

Details about a recipient's response to an exchange request.

`GKGameCenterBadgingDisabled`

A Boolean value indicating whether GameKit can add badges to a turn-based game icon.

Errors

`struct GKError`

The error structure used by this framework.

`enum Code`

Error codes for the GameKit error domain.

`let GKErrorDomain: String`

The error domain for general game errors.

Deprecated

☰ Deprecated symbols

Review unsupported symbols and their replacements.