

Best Practices on Instant Games

Building games for Instant Games requires that you build your game in HTML5. Here's a list of Best Practices while developing your Instant Game:

Load times and performance

Initial loading shouldn't take more than 5 seconds: Instant Games need to be "instant", and players will tend to churn away if the initial loading takes too long. The total size of your bundle can be up to 200MB, but we'll only load the files explicitly required by your `index.html` during initial loading. So make sure to use that time to only load essential assets for the first session and postpone loading of other assets for when they are needed.

Report initial loading progress: During initial loading, you should inform us on your loading progress by using `setProgress`.

Avoid secondary loading screens: Once the native loading ring shows completion at 100%, the player shouldn't be thrown into another waiting experience - they should be able to play right away.

Optimize for mobile: Although Instant Games works on desktop browsers, the majority of sessions are from Mobile devices. It's a good idea to optimize rendering and aspect ratio for popular iOS and Android devices.

Load resources in parallel with `initializeAsync`: You shouldn't wait for the `initializeAsync` promise to resolve before downloading your resources. You can download in parallel.

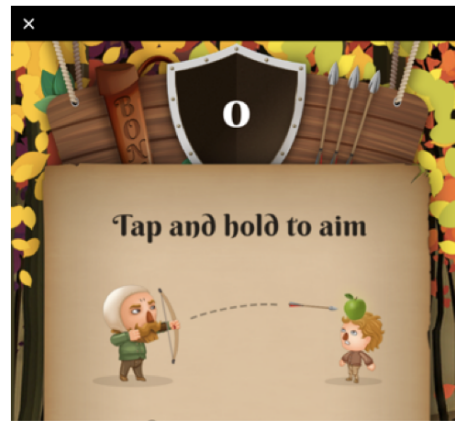
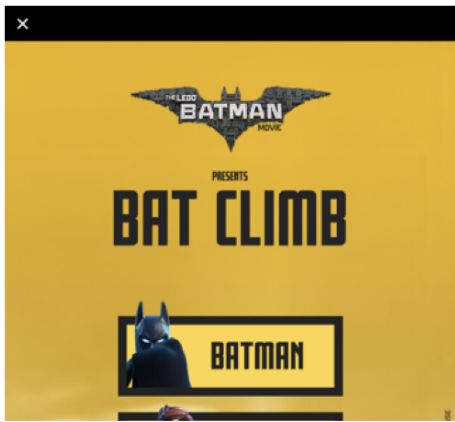
First time player experience

Include a short, non-intrusive tutorial for first-time players, if your game is not self-explanatory.

Allow seasoned players to come back to the tutorial if they choose to. It can be that they haven't played the game in a while, or that they want to show the tutorial to a friend. Take note not to force the tutorial in every session, but give experienced players the opportunity to jump right into play.

Consider group settings for new players who join the group later. You should ensure that these players see the tutorial on their first play.

Use playable tutorials over text if possible. The best tutorials are those that users do not know are tutorials.

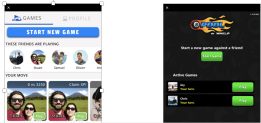



In-game player experience

Make sure to handle interruptions by implementing the `onPause` callback. This will allow your game to gracefully handle situations in which the player was interrupted (by a notification, phone call, app switch, etc.)

Make sure all list items in the lobby have a CTA: if you're developing a turn-based asynchronous game it's a very good practice to implement a **lobby screen**, where players can easily navigate all current and suggested matches. For this lobby screen consider the following sections with corresponding CTAs.

Section Name	User Filter	Suggested CTA	CTA Action
Suggested Matches	Connected players who are active in other threads but don't have any active matches with the current player	"Start new match"	<code>createAsync()</code> to start a new match in the new context

Section Name	User Filter	Suggested CTA	CTA Action
Waiting for your turn 	Matches in which the other players are waiting for the current player to play their turn	"Play your turn"	<code>switchAsync()</code> to play in that context
Waiting for their turn 	Matches in which the current player is waiting for other players to play their turn	"Nudge"	<code>switchAsync()</code> and then <code>updateAsync()</code> with a nudging message
Finished matches	Previous matches that have ended already	"Rematch"	<code>switchAsync()</code> to start a new match in that context

Enable single-player mode for players who are waiting for their turn, or have run out of challenges for the day.

Tailor the experience for the context type: since an Instant Game can be played in many different types of contexts (one-to-one conversations, group conversations, news feed posts, etc.), make sure that your game always reads `FBInstant.context.getType()` and loads the appropriate experience for that context type.

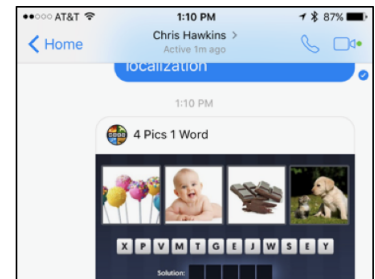
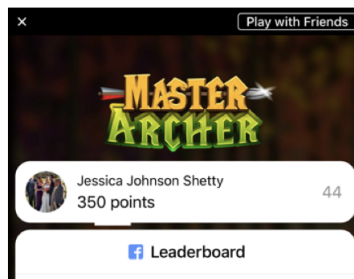
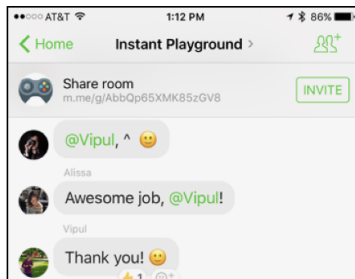
Localize your game: players tend to engage more with the game if they can play in a language that's natural to them. The table below will help you decide which languages to translate your game content to:

Languages	Percentage of players
EN, ES, VI, PT, TH, FR, PL, TW	90% of Instant Games players

Languages	Percentage of players
Adding to the languages above: TR, ID, IT, HU, DE, AR, GR, DA, CZ, NL, NO, SE, RU, SK, ZH, HR, JP	99% of Instant Games players

Social play experience

Encourage group play: players interacting in large groups (3 people or more) tend to show much better retention, so even if your game was designed to be 1 vs. 1, you should make sure to provide a good in-game experience for people playing in larger groups. You can encourage group play by designing competitive features such as per-group leaderboards or collaborative features such as boss raids.



Sharing and inviting should be meaningful: players respond much better to an invite if the message they receive has meaningful social context. For instance a message like "Your friend is stuck in this level, can you help them?" and that opens the game directly on the corresponding level is recommended over a generic "Your friend invited you to play this game".

Custom updates should be relevant and contextual, for instance a custom update that shows a player overtaking another in the leaderboard is preferred over one that only shows the new score of the player.



Use data payloads on custom updates to provide a contextual experience, for instance, if a custom update message says "Your friend is asking for your help to defeat this boss", the CTA of the message should take the invited player directly to that boss battle and not to the initial screen of the game.

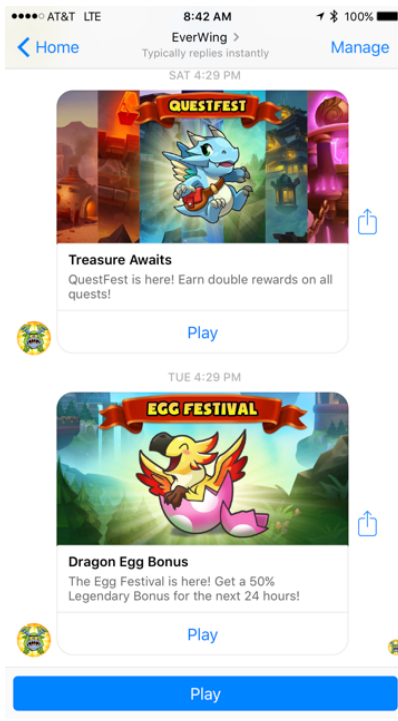
Re-engagement experience

Game Bots should provide *timely*, *relevant* and *valuable* messages to players:

Timely: Build an option screen in your game client that gives the players control to switch bot messages on/off and also gives them control about what time of the day to receive their messages.

Relevant: Bot messages should have some game or social context. Messages like "your expedition has finished, come back to collect your rewards" is preferred over messages like "you haven't played in a while, please come back to the game".

Valuable: Make sure your players have the right incentives open the game via a bot message by using the message payload to reward them in-game with something valuable. A bot message is usually not valuable if it opens your game in the start screen.



Further best practices with game bots

Do:

Provide **relevant**, **timely** and **valuable** updates to the players. For more information, visit our [Best Practices](#) section.

Give the user control (for example, by confirming they would like to be notified, and with what frequency).

Use entry point data on play buttons to load the game in contextually relevant ways.

Name the bot the same as the game.

Make use of social updates like turn reminders, tournaments results, timed rewards and challenges.

Make sure your players have the right incentives open the game via a bot message by using the message payload to reward them in-game with something valuable. A bot message is usually not valuable if it opens your game to the start screen.

Use a persistent menu to provide common actions, such as launching the game.

Set default action to use `game_play` on custom updates, so that the entire image takes you into the game.

Use bots to announce new features or content.

Optimize time of day for message sends per user, being sensitive to timezones.

Follow the general [best practices for Messenger Bots](#).

Do not:

- ✗ Send a message immediately after the player closes the game.
- ✗ Send messages to re-engage the player with no context (e.g.: "Come back to the game now!"). Instead prefer re-engagement messages with rich context (e.g.: "Your scout has come back with more info")
- ✗ Adopt the voice of other Facebook users or mislead players to believe their friends are communicating with them.
- ✗ Continue to send a user bot messages when they repeatedly do not engage. Policy limits will apply and block your message from being sent. **Current limits are 5 messages over 10 days of last game play session.** [More information in section 9.4 of our Platform Policy Docs](#)
- ✗ Link to any app store.

Games

Game Services

Mobile Games

Instant Games

Getting Started

Guides

Rich Gameplay Features

Best Practices

Lightweight games

Game performance

Ads Monetization

Using Leaderboards

Game Bots and Server Communication

Analytics

Engine Recommendations

In App Purchases

FAQ

SDK

PC Games SDK

Games on Facebook

App Center

Facebook Gameroom

Gaming Community

LANGUAGES

Products

[Facebook Login](#)

[Sharing on Facebook](#)

[Games](#)

[Facebook App Ads](#)

SDKs

[iOS SDK](#)

[Android SDK](#)

[JavaScript SDK](#)

[PHP SDK](#)

[Unity SDK](#)

Tools

[Graph API Explorer](#)

[Open Graph Debugger](#)

[Object Browser](#)

[JavaScript Test Console](#)

[API Upgrade Tool](#)

[Facebook Analytics](#)

Support

[Platform Status](#)

[Developers Group](#)

[Marketing Partners](#)

[Bugs](#)

News

[Blog](#)

[Success Stories](#)

[Videos](#)