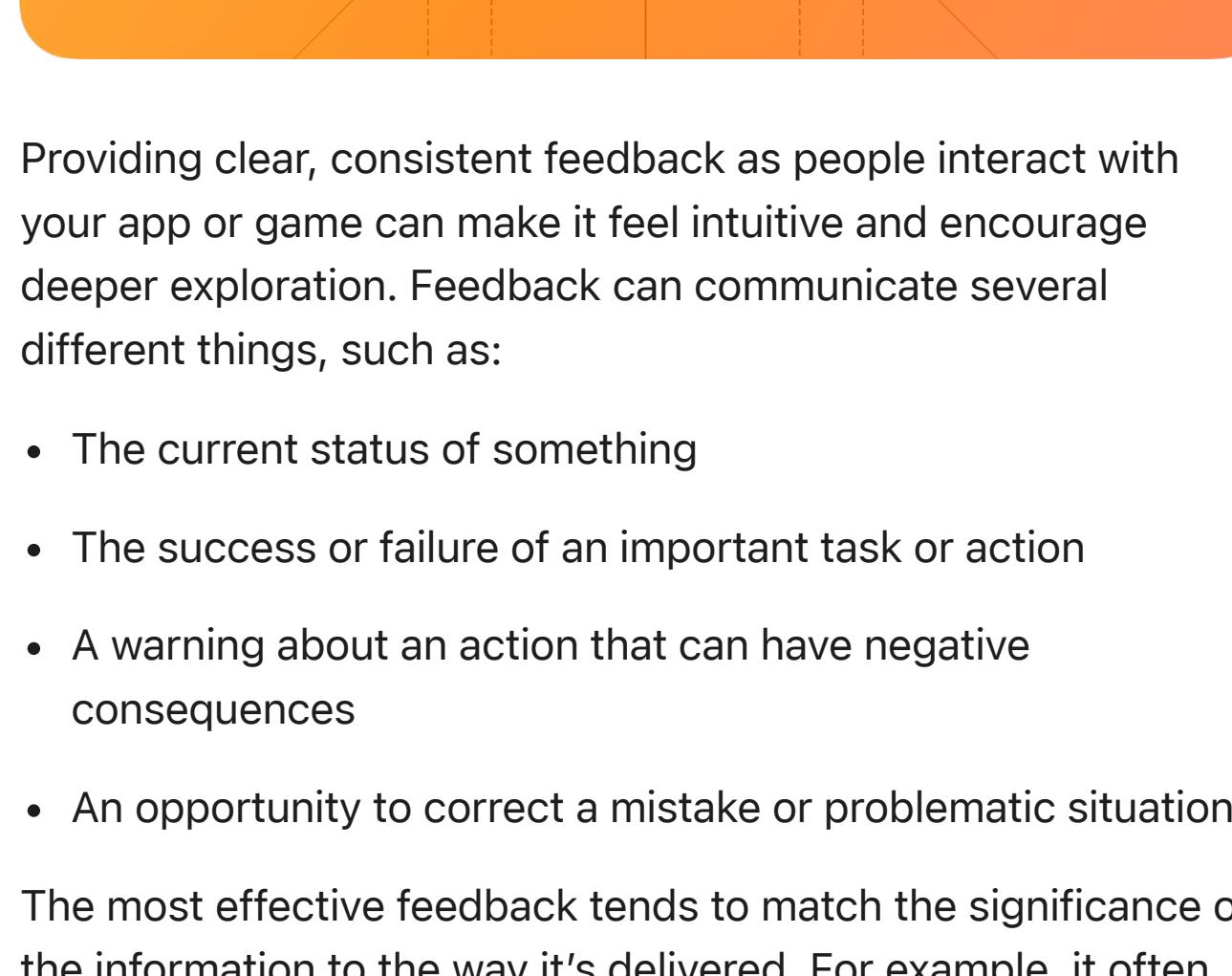


Feedback

Feedback helps people know what's happening, discover what they can do next, understand the results of actions, and avoid mistakes.



Supported platforms



Feedback

Best practices

Platform considerations

Resources

Providing clear, consistent feedback as people interact with your app or game can make it feel intuitive and encourage deeper exploration. Feedback can communicate several different things, such as:

- The current status of something
- The success or failure of an important task or action
- A warning about an action that can have negative consequences
- An opportunity to correct a mistake or problematic situation

The most effective feedback tends to match the significance of the information to the way it's delivered. For example, it often works well to display status information in a passive way so that people can view it when they need it. In contrast, a warning about possible data loss needs to interrupt people so they have a chance to avoid the problem.

Best practices

Make sure all feedback is accessible. When you use multiple ways to provide feedback, you reach more people and give them the opportunity to receive the feedback in ways that work for them. For example, when you provide feedback using color, text, sound, and haptics, people can receive it whether they silence their device, look away from the screen, or use VoiceOver. (For guidance on providing haptic feedback, see [Playing haptics](#).)

Consider integrating status feedback into your interface. When status feedback is available near the items it describes, people get important information without having to take action or leave their current context. For example, Mail in iOS and iPadOS describes the most recent update and displays the number of unread messages in the toolbar of the mailbox screen, making the information unobtrusive but easy for people to check when they're interested.

Use alerts to deliver critical — and ideally actionable — information. By design, alerts disrupt the current context, so you need to match the importance of the information to the level of interruption. Alerts can lose their impact if you use them too often or to deliver unimportant information. For guidance, see [Alerts](#).

Warn people when they initiate a task that can cause data loss that's unexpected and irreversible. In contrast, don't warn people when data loss is the expected result of their action. For example, the Finder doesn't warn people every time they throw away a file because deleting the file is the expected result.

When it makes sense, confirm that a significant action or task has completed. For example, people appreciate getting feedback that confirms a successful Apple Pay transaction. It's generally best to reserve this type of confirmation for activities that are sufficiently important — because people typically expect their action or task to succeed, they only need to know when it doesn't.

Show people when a command can't be carried out and help them understand why. For example, if people request directions without specifying a destination, Maps tells them that it can't provide directions to and from the same location.

Platform considerations

No additional considerations for iOS, iPadOS, macOS, tvOS, or visionOS.

watchOS

Avoid displaying an indeterminate progress indicator — such as a loading indicator — in a watchOS app. An animated indicator can make people think they need to continue paying attention to the display, which isn't a good user experience. To provide a better experience, reassure people that they'll receive a notification when the process completes.

Resources

Related

[Playing audio](#)

[Playing haptics](#)

[Motion](#)

Developer documentation

[Animation and haptics](#) — UIKit

Videos

Designing Fluid Interfaces

Essential Design Principles