

CarPlay 应用程序 编程指南

2021 年 3 月 31 日

📄 开发商

目录

Introduction	错误!未定义书签。
概观	四
CarPlay 应用授权	四
CarPlay 应用指南	四
发展环境	七
配置您的权利	七
CarPlay 音频应用和向后兼容性	9
CarPlay 通信应用和向后兼容性	9
模拟器	9
模板	11
行动表	12
警报	12
接触	13
格子	13
信息	14
目录	14
正在播放	15
兴趣点	16
标签栏	17
资产	17
构建您的 CarPlay 应用程序	18
启动	18
创建列表模板	20
创建一个正在播放的模板	21
播放音频	21
iPhone 锁定时工作	22
CarPlay 导航应用	22
模板	22
基础视图	22
地图	23

搜索	24
口声控制	25
嵌板	26
行程预览	26
路线选择	26
指南和行程估计	27
导航警报	28
启动	28
路线指引	29
选择目的地	30
试映	30
选择路线并开始引导.....	30
显示行程信息和即将到来的机动.....	31
结束指南	33
键盘和列表限制	33
语音提示	33
音频会话配置	33
激活和停用音频会话.....	34
提示样式	34
CarPlay 仪表盘	34
创建场景代理	35
绘制您的 CarPlay 仪表盘内容	35
仪表盘场景清单	35
发布您的 CarPlay 应用程序.....	37

介绍

CarPlay 是一种更智能、更安全的在车内使用 iPhone 的方式。CarPlay 会在开车时拿着你想用 iPhone 做的事情，放在你汽车的内置显示屏上。

除了获取方向、拨打电话、发送和接收消息以及听音乐之外，CarPlay 还支持以下类别的应用：

- 声音的
- 沟通

- 电动汽车充电
- 航行
- 停车
- 快速点餐

本指南介绍了如何创建一个 CarPlay 应用程序。

概观

用户从应用商店下载 CarPlay 应用，并像其他应用一样在 iPhone 上使用。当带有 CarPlay 应用程序的 iPhone 连接到 CarPlay 车辆时，应用程序图标会出现在 CarPlay 主屏幕上。CarPlay 应用程序不是单独的应用程序——您可以将 CarPlay 支持添加到现有的应用程序中。

CarPlay 应用程序旨在让外观和感觉像 iPhone 上的应用程序，但其 UI 元素与内置的 CarPlay 应用程序相似。

您的应用程序使用 CarPlay 框架向用户呈现用户界面元素。iOS 管理 UI 元素的显示，并处理与汽车的接口。您的应用程序不需要管理不同屏幕分辨率的用户界面元素布局，也不需要支持不同的输入硬件，如触摸屏、旋钮或触摸板。

CarPlay 应用程序必须满足 CarPlay 权利附录中定义的基本要求，并且必须遵循 CarPlay 应用程序指南。

有关一般设计指导，请参见 CarPlay 应用程序的人机界面指南。

CarPlay 应用授权

所有 CarPlay 应用都需要 CarPlay 应用授权。

要申请 CarPlay 应用授权，请前往 <http://developer.apple.com/carplay>，提供关于您的应用的信息，包括 CarPlay 应用类别。您还必须同意 CarPlay 权利附录。

苹果会审核你的请求。如果您的应用程序符合 CarPlay 应用程序的标准，苹果将为您苹果开发者帐户分配一个 CarPlay 应用程序权限，并通知您。

CarPlay 应用指南

所有 CarPlay 应用程序都必须遵守以下准则。**所有应用的指南**

1. 您的 CarPlay 应用程序必须主要设计为向用户提供指定的功能(例如。CarPlay 音频应用必须主要设计为提供音频播放服务，CarPlay 停车应用必须主要设计为提供停车服务等。
2. 永远不要指示用户拿起他们的 iPhone 来执行任务。如果出现错误情况，例如需要登录，您可以让用户知道该情况，以便他们在安全时采取措施。但是，用户信息不得包含要求用户操作其 iPhone 的措辞。
3. 所有的 CarPlay 用户流都必须在不与 iPhone 交互的情况下实现。
4. 所有 CarPlay 用户流都必须在驾驶时有意义地使用。不要在 CarPlay 中包含与主要任务无关的特性(例如。不相关的设置、维护功能等。
5. 没有游戏或社交网络。

6. 永远不要在 **CarPlay** 屏幕上显示消息、文本或电子邮件的内容。
7. 将模板用于其预期目的，并且仅用指定的信息类型填充模板(例如，必须使用列表模板来呈现供选择的列表，必须使用正在播放屏幕中的专辑插图来显示专辑封面等)。
8. 所有语音交互都必须使用 **SiriKit** 处理(除了 **CarPlay** 导航应用程序，见下文)。

音频应用的附加指南

1. 千万不要在 **CarPlay** 屏幕上显示歌词。
2. 文本到语音系统(网络阅读器、电子邮件阅读器等))。是不允许的。

通信应用的附加指南

1. 通信应用程序必须提供短格式文本消息(而不是电子邮件)功能、**VoIP** 呼叫功能或两者兼有。
2. 必须使用 **SiriKit** 处理消息。您的应用程序必须支持发送消息、搜索消息和设置消息属性。
3. **VoIP** 呼叫必须使用 **CallKit** 和 **SiriKit** 进行处理。您的应用程序必须支持启动音频通话，并搜索用户的通话记录。

电动汽车充电应用的附加指南

1. 电动汽车充电应用程序必须提供与驾驶相关的有意义的功能(例如，你的应用不能只是 **EV** 充电器的列表)。
2. 在地图上显示位置时，不要暴露电动汽车充电器以外的位置。

导航应用的附加指南

1. 基础视图必须专门用于绘制地图。不要在基础视图中绘制窗口、警报、面板、覆盖图或用户界面元素。例如，不要在基础视图中绘制车道引导信息。相反，使用提供的模板绘制车道引导信息作为辅助机动。
2. 将每个提供的模板用于其预期目的。例如，策略图像必须代表一个策略，不能代表其他内容或用户界面元素。
3. 提供一种进入平移模式的方法。如果您的应用程序支持平移，您必须在地图模板中包含一个按钮，允许用户进入平移模式，因为拖动手势并非在所有车辆中都可用。拖动手势只能用于平移地图。
4. 如有要求，立即终止路线指引。例如，如果用户使用车辆的内置导航系统开始路线引导，您的应用代表将收到取消通知，必须立即停止路线引导。
5. 正确处理音频。语音提示必须与车辆的音频系统(如调频收音机)同时工作，当没有音频播放时，您的应用程序不应不必要地激活音频会话。
6. 确保您的地图适合每个受支持的国家。
7. 开诚布公，回应反馈。如果苹果或汽车制造商对设计或功能有意见，苹果可能会联系您。

8. 语音控制必须仅限于导航功能。

停车应用程序的附加指南

1. 停车应用程序必须提供与驾驶相关的有意义的功能(例如。你的应用不能只是一个停车位置列表)。

2. 在地图上显示位置时，不要暴露停车以外的位置。**快速点餐应用的附加指南**

1. 快速订餐应用程序必须是快速服务餐厅(QSR)应用程序，主要设计用于面向驾驶的订餐(例如。驾车通过，接机)时。该类别不适用于一般零售应用程序(例如。超市、路边皮卡)。
2. 快速点餐应用程序必须提供与驾驶相关的有意义的功能(例如。你的应用不能只是商店位置的列表)。
3. 仅简化订购。不要显示完整的菜单。您可以显示最近订单的列表，或每个限制为 12 个项目的收藏夹。
4. 在地图上显示位置时，不要暴露快速服务餐厅以外的位置。

下面的例子展示了如何在 CarPlay 中构建一个快速点餐应用。该应用程序提供了四个选项卡，允许用户选择商店，查看最近的订单或喜爱的项目列表，并确认订单信息。图标和文本可以自定义。位置、列表和信息屏幕限于 12 个项目。快速点餐用户流程应该简单，仅限于最常见的任务。只显示最重要和相关的信息。



快速点餐应用程序中的标签栏

发展环境

为 CarPlay 创建应用程序需要 Xcode 和苹果开发者计划帐户。

配置您的权利

一旦您获得了 CarPlay 应用程序权利，请创建一个包含 CarPlay 应用程序功能的新配置文件。

1. 登录您的苹果开发者帐户 <https://developer.apple.com/account/>。
2. 选择证书、标识符和配置文件，然后单击边栏上的标识符。
3. 选择与您的应用程序相关联的应用程序标识，或创建新的应用程序标识。
4. 向下滚动到页面底部附近的附加功能。
5. 为您的应用启用所有必要的 CarPlay 应用功能。
6. 单击右上角的保存。
7. 继续设置配置文件，并为您的应用程序标识创建新的设置配置文件。有关其他信息，请参见开发人员帐户帮助中的参考|使用托管功能进行资源调配|应用程序标识符。

<https://help.apple.com/developer-account/>

创建新的配置文件后，将其导入 Xcode。Xcode 和模拟器需要一个支持 CarPlay 的配置文件。

在 Xcode 中，在您的项目中创建一个授权 .plist 文件(如果您还没有的话)。将您的 CarPlay 应用程序权利密钥作为布尔密钥添加。下面的例子是一个只支持 CarPlay 框架的 CarPlay 音频应用。

```
< key > com . apple . developer . carplay-audio
</key > < true/>
```

在 Xcode 中，在“签名和功能”下，关闭“自动管理签名”，在“构建”下设置确保代码签名授权设置为授权文件的路径。

使用与您选择的设置配置文件匹配的权利密钥。

权利钥匙

CarPlay 音频应用(CarPlay 框架)	com . apple . developer . carplay-音频
App 支持 CarPlay 框架。如果您的应用程序支持 CarPlay 框架和媒体播放器框架，请同时包括 CarPlay 音频应用程序权利。	
需要 iOS 14 或更高版本。	

CarPlay 音频应用(媒体播放器框架)	可播放-内容
应用程序支持媒体播放器框架(已弃用)。如果您的应用程序支持 CarPlay 框架和媒体播放器框架，请同时包括 CarPlay 音频应用程序权利。	

CarPlay 通信应用 App 支持 CarPlay 框架、CallKit、SiriKit 消息传递意图和 SiriKit VoIP 呼叫意图。 需要 iOS 14 或更高版本。	<code>com . apple . developer . carplay-communication</code>
CarPlay 消息应用 应用程序支持 SiriKit 消息传递意图。	<code>com . apple . developer . carplay-消息传递</code>
CarPlay 网络电话呼叫应用 App 支持 CallKit 和 SiriKit VoIP 通话意图。	<code>com . apple . developer . carplay-calling</code>
CarPlay 电动汽车充电应用 App 支持 CarPlay 框架。 需要 iOS 14 或更高版本。	<code>com . apple . developer . carplay-充电</code>
CarPlay 导航应用 App 支持 CarPlay 框架。 需要 iOS 12 或更高版本。	<code>com . apple . developer . carplay-maps</code>
CarPlay 停车应用 App 支持 CarPlay 框架。 需要 iOS 14 或更高版本。	<code>com . apple . developer . carplay-停车</code>
CarPlay 快速订餐应用 App 支持 CarPlay 框架。 需要 iOS 14 或更高版本。	<code>com . apple . developer . carplay-快速订购</code>

CarPlay 音频应用和向后兼容性

从 iOS 14 开始，CarPlay 音频应用可能会使用 CarPlay 框架来呈现定制的用户界面。使用 CarPlay 框架的 CarPlay 音频应用程序必须包含 `com . apple . developer . CarPlay-audio` 权限。

使用媒体播放器框架的 CarPlay 音频应用必须包括

`com . apple . developer . playable-内容授权`。如果您希望您的 CarPlay 音频应用程序支持 iOS 13 和更早版本，您必须支持媒体播放器框架并包含此权利。支持 Media Player 框架的 CarPlay 应用将继续在 iOS 14 上运行。

CarPlay 音频应用可以支持 CarPlay 框架、Media Player 框架，或者两者都支持。确保包含正确的权利(或同时包含这两种权利)以匹配您正在使用的框架。

CarPlay 通信应用和向后兼容性

从 iOS 14 开始，CarPlay 通信应用可能会使用 CarPlay 框架来呈现用户界面。使用 CarPlay 框架的 CarPlay 通信应用程序必须包含 `com . apple . developer . CarPlay-通信` 权限。

CarPlay 通信应用类别取代了旧的 CarPlay 消息传递应用

CarPlay VoIP 通话应用类别。您应该包括

```
com . apple . developer . carplay-messaging and com . apple .  
developer . carplay-
```

呼叫授权以支持 iOS 13 及更早版本。在 iOS 13 和更早版本上，您的应用程序将无法呈现用户界面，但用户仍然可以使用 SiriKit 和 CallKit 与您的应用程序进行交互。

如果使用 carplay 框架，只包括 com . apple . developer . CarPlay-通信权限。

模拟器

模拟器允许你在第二个窗口运行你的 CarPlay 应用程序。第二个窗口充当汽车的显示屏，允许您以类似于连接到 CarPlay 系统时的方式与之交互。

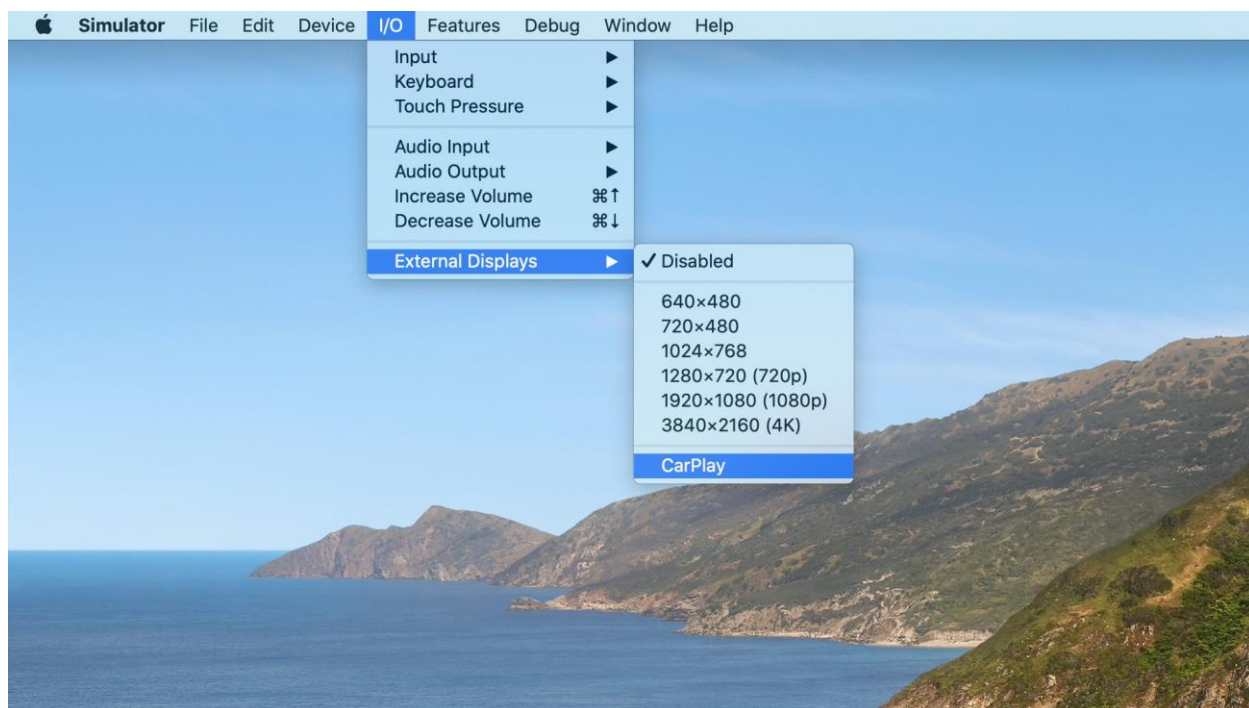
虽然模拟器在开发过程中很有用，但模拟器中没有某些 CarPlay 功能，您不应该将它作为开发应用程序的唯一方法。

模拟器中无法测试的一些功能：

- **iPhone 锁定时进行测试。**大多数用户在 iPhone 锁定时会与 CarPlay 交互。即使 iPhone 被锁定，您的应用程序也应该正常运行。
- **测试 Siri。**CarPlay 严重依赖 Siri。确保您的应用程序在 Siri 交互过程中按预期工作。
- **测试音频行为。**你的应用程序必须是一个好的音频公民。请记住，当 CarPlay 处于活动状态时，其他音频源可能正在播放。停用您的音频会话，除非您实际上正在播放音频。例如，当汽车的调频收音机正在播放时，导航语音提示应导致调频收音机音量水平暂时降低，然后在语音提示播放完毕后升高。

强烈建议使用支持无线 CarPlay 的汽车或售后系统开发和测试 CarPlay 应用程序。这将允许您无线运行 CarPlay，同时使用闪电到 USB 电缆将 iPhone 连接到 Mac 上的 Xcode。

要在模拟器中访问 CarPlay，请选择输入/输出、外部显示器和 CarPlay 来显示 CarPlay 屏幕。



在第二个窗口中显示 CarPlay

如果您正在开发一个 CarPlay 导航应用程序，您可以使用额外的选项来配置模拟器。要启用额外选项，请在启动模拟器之前在终端中输入以下命令。默认值写入 `com.apple.iphonestimulator`

```
carplayextractions-bool YES
```

如果您为模拟器配置了额外的选项，您可以修改 CarPlay 屏幕的宽度、高度和比例。

默认情况下，CarPlay 屏幕配置为 800 像素宽、480 像素高的显示器。在这种配置中，CarPlay 将使用 2.0 的比例。这是许多 CarPlay 系统的典型特征。在更高的分辨率下，CarPlay 自动使用 3.0 的比例。

建议您使用以下设置测试您的 CarPlay 导航应用程序。特别是，您需要确保基础地图在所有分辨率和比例下都能正确渲染。

宽度和高度	规模	
最低限度 (尽可能小的 CarPlay 屏幕)	748 像素 x 456px 像素	2.0
标准 (许多 CarPlay 屏幕的典型默认分辨率)	800 像素 x 480px 像素	2.0

高分辨率 (典型的大型 CarPlay 屏幕)	1920 像素 x 720px 像素 3.0
肖像 (垂直 CarPlay 屏幕示例)	768px x 1024px 2.0。

模板

CarPlay 应用程序是由 iOS 在 CarPlay 屏幕上呈现的一组固定的用户界面模板构建而成的。

CarPlay 应用程序负责选择在屏幕上显示哪个模板(控制器)，并提供要在模板内部显示的数据(模型)。iOS 负责在 CarPlay(视图)中呈现信息。

CarPlay 支持通用模板，如警报、列表和标签栏。它还支持为特定功能设计的模板，如联系人、地图和正在播放的屏幕。每个 CarPlay 应用类别可以访问一组有限的模板。对模板的访问由您的应用授权决定。例如，音频应用程序可能只使用提醒、网格、列表、正在播放和标签栏模板。

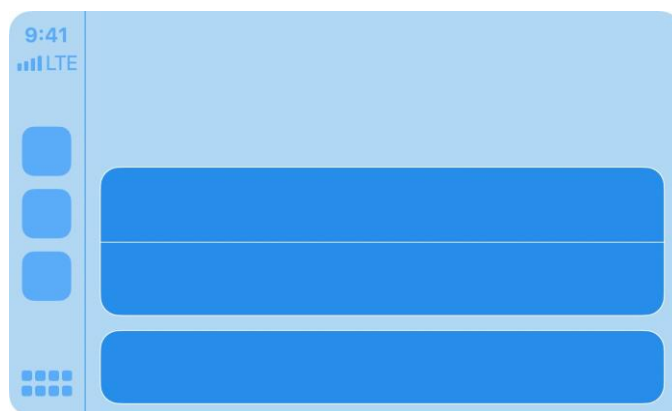
如果您试图使用您的权利不支持的模板，运行时将出现异常。

	声音的	沟通	电动汽车充电	航行	停车	快餐 排序
行动表		"	"	"	"	"
警报	"	"	"	"	"	"
接触		"		"		
格子	"	"	"	"	"	"
信息		"	"	"	"	"
目录	"	"	"	"	"	"
地图				"		
正在播放	"					
兴趣点			"		"	"
搜索				"		
标签栏	"	"	"	"	"	"
口声控制				"		

可以推到屏幕上的模板数量是有限制的。所有应用程序都被限制在最多 5 个模板的深度，包括根模板。快速点餐应用仅限于 2 个模板深度。

行动表

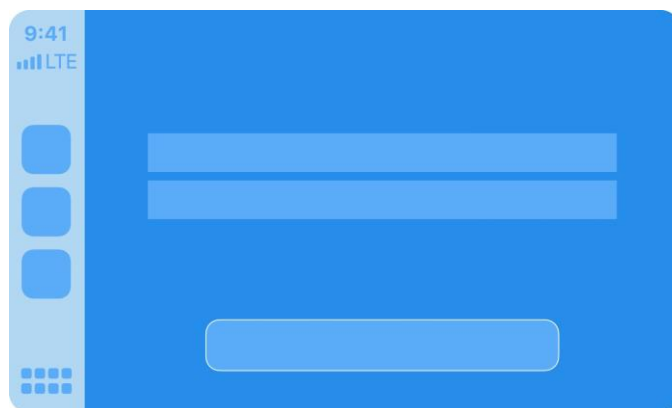
动作表是响应控件或动作而出现的特定警报样式，它提供了一组与当前上下文相关的两个或多个选项。使用行动表让人们发起任务，或者在执行潜在的破坏性操作之前请求确认。行动表模板可用于通信、电动汽车充电、导航、停车和快速点餐应用。



行动表

警报

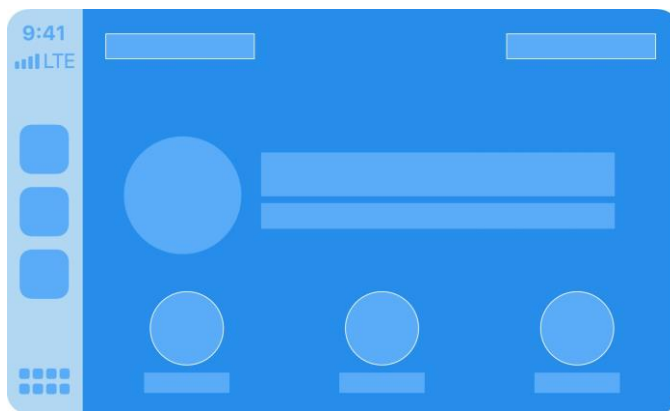
提醒传达与应用程序状态相关的重要信息。警报由标题和一个或多个按钮组成。您可以提供不同长度的标题，并让 **CarPlay** 选择最适合可用屏幕空间的标题。如果基础条件允许，可以通过编程消除警报。所有应用程序都可以使用警报模板。



警报

接触

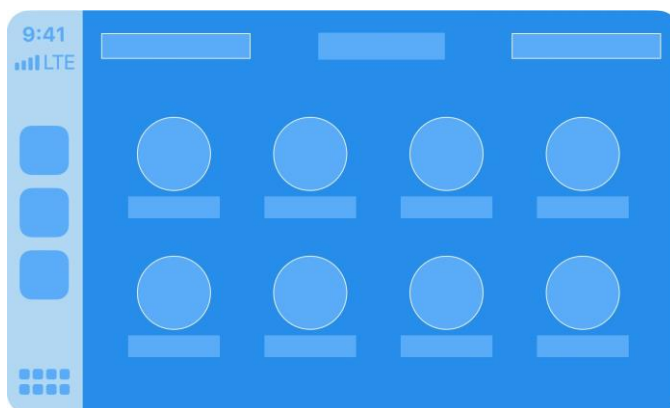
联系人允许您展示有关个人或企业的信息。联系人由图像、标题、副标题和操作按钮组成。使用操作按钮让用户执行与当前联系人相关的任务，例如打电话或发送消息。联系人模板可用于通信和导航应用程序。



接触

格子

网格是一种特定样式的菜单，最多显示八个选项，由一个图标和一个标题表示。使用网格模板让人们从固定的类别列表中进行选择。网格还包括一个带有标题、前导按钮和尾随按钮的导航栏，这些按钮可以显示为图标或文本。网格模板可以被所有应用程序使用。



格子

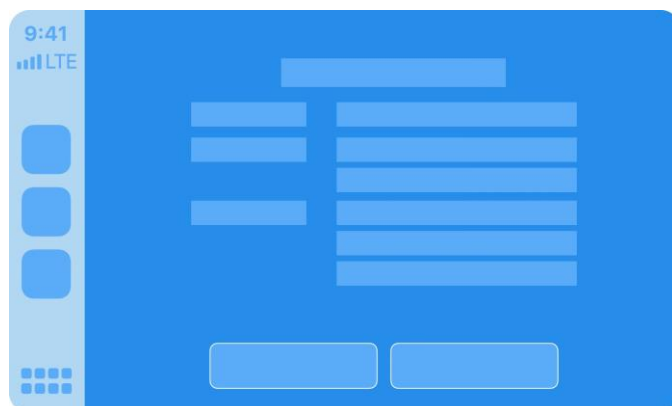
信息

信息屏幕是一种特定样式的列表，它通过可选的页脚按钮呈现有限数量的静态标签。标签可以出现在一列或两列中。

使用信息模板显示重要信息。例如，电动汽车充电应用程序可以显示关于充电站的信息，例如可用性，而快速食品订购应用程序可以显示订单摘要，例如提货地点和时间。

由于标签数量有限，因此只显示完成任务所需的最重要的摘要信息。

信息模板可用于通信、电动汽车充电、停车、导航和快速点餐应用。



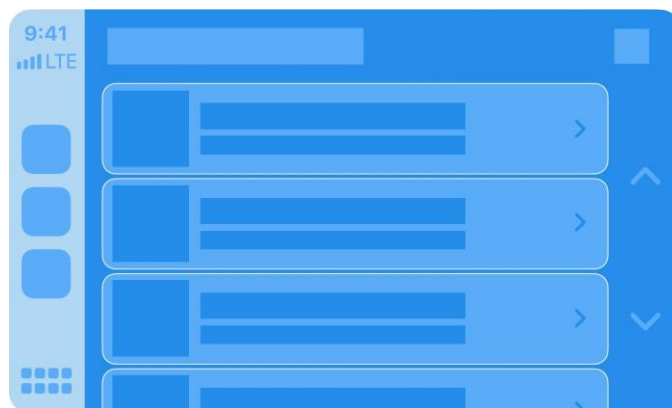
信息

目录

列表将数据显示为滚动的单列表格，表格中的行可以分为多个部分。列表非常适合基于文本的内容，并且可以用作导航分层信息的手段。

列表中的每个项目可以包括诸如图标、标题、副标题、公开指示符、进度指示符、回放状态或阅读状态等属性。如果您只需要显示带有文本的图标，或者选择特定的列表项，如音频应用程序的图像行列表项或通信应用程序的消息列表项，请使用通用列表项。

一些汽车动态限制列表最多 12 个项目。您的应用程序总是可以检查当前允许的最大值，但是您总是需要准备好处理只能显示 12 个项目的情况。超过最大值的项目将不会显示。所有应用程序都可以使用列表。





带有音频应用程序图像行列表项的列表

正在播放

“正在播放”屏幕显示有关当前正在播放的音频的信息，如标题、艺术家、经过的时间和专辑插图。它还允许人们使用回放控制按钮来控制你的应用程序。

现在播放的屏幕是可定制的，您应该根据自己的需要进行调整。例如，您可以提供指向即将播放的曲目的链接，可以使用自己的图标自定义播放控制按钮，并且可以为固定长度的音频或开放式音频(如实时流)配置经过时间指示器。

现在播放模板很特别，因为用户可以直接从 **CarPlay** 主屏幕或通过应用程序导航栏中的“现在播放”按钮访问它。您必须随时准备填充正在播放的模板。

只有列表模板可以被推到正在播放的模板之上。例如，如果您的应用程序在正在播放的模板中启用了“播放下一个”按钮，您可以通过显示包含即将到来的播放队列的列表模板来做出响应。现在播放的模板可以在音频应用程序中使用。



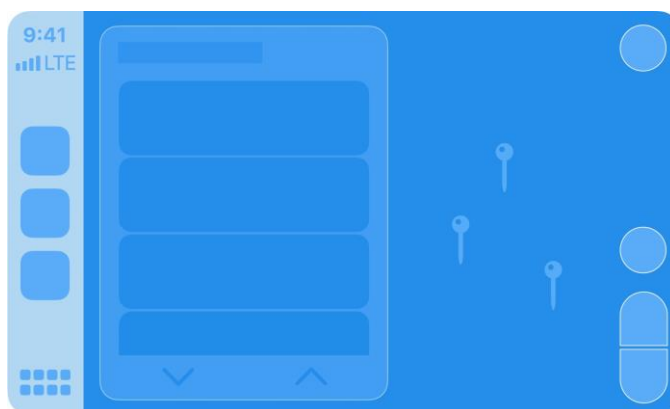
正在播放

兴趣点

兴趣点屏幕允许用户浏览地图上附近的位置，并选择一个位置进行进一步的操作。

兴趣点模板包括由 **MapKit** 框架提供的地图，以及包含多达 **12** 个位置列表的覆盖图。位置列表应仅限于最相关或附近的位置。

兴趣点模板可用于电动汽车充电、停车和快速点餐应用。



兴趣点

标签栏

标签栏是其他模板的通用容器，其中每个模板占据标签栏中的一个标签。

用户可以使用标签栏在不同的模板之间快速切换。

标签栏最多可以有 **5** 个标签，虽然这可能会在将来发生变化，所以您不应该依赖于这个数字。相反，观察 **iOS** 提供的最大选项卡数。在音频应用的情况下，一个标签总是被现在播放的按钮占据。标签栏模板可以被所有应用程序使用。



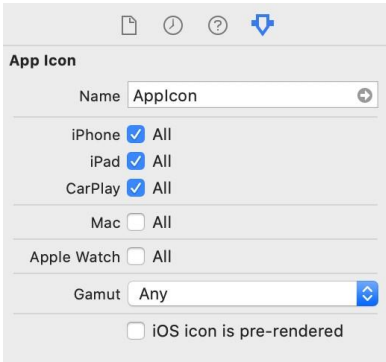
标签栏

资产

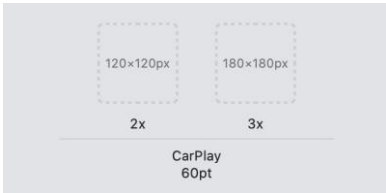
CarPlay 资产包括您的应用图标和模板中使用的图像，如图标和按钮。以 2 倍和 3 倍的比例为 CarPlay 以及明暗界面创建资产。

您在 CarPlay 中的应用图标应该看起来与 iPhone 上显示的应用图标相似。

要设置您的应用程序图标，请在 Xcode 中打开 CarPlay 资产，并填充 CarPlay 图像池。



打开 CarPlay 应用程序图标资产



CarPlay 2x 倍和 3 倍图像池

创建图像时，请使用以下尺寸指南。

	最大尺寸 (3x) 像素	最大尺寸 (2x) 像素	以磅为单位的最大尺寸	像素
CarPlay 应用图标	60pt x 60pt	180 像素 x 180px	120 像素 x 120px	像素
联系人操作按钮	50 磅 x 50pt 磅	150 像素 x 150px	100 像素 x 100px	像素
网格图标	40pt x 40pt	120 像素 x 120px	80px x 80px	像素
现在播放动作按钮	20 磅 x 20pt 磅	60px x 60px	40px x 40px	
标签栏图标	24pt x 24pt	72px 72px	48px x 48px	

如果需要知道 CarPlay 屏幕比例，可以使用特质集合 `carTraitCollection` 获取显示比例。不要使用 `carTraitCollection` 中的其他参数，一定要获取汽车屏幕的比例(而不是 iPhone 屏幕的比例)。

提供可显示的图像资源，包括明暗界面。为不同外观提供图像的最佳方式是使用资产目录。资产目录中的每个资产代表同一图像的不同变体。或者，您可以使用 `UIImageAsset` 在运行时将图像的不同变体组合成一个包。

构建您的 CarPlay 应用程序

启动

所有的 CarPlay 应用程序都必须采用场景并声明一个 CarPlay 场景来使用 CarPlay 框架。您可以动态声明一个场景，也可以在您的 `Info.plist` 文件中包含一个静态场景清单。下面是一个场景清单的例子，它声明了一个 CarPlay 场景。您可以将其添加到应用程序的 `Info.plist` 文件的顶层。

```
< key > ui applicationcenemanifest </key >
<字典>
<键>用户配置</键>
<字典>
<!--声明设备场景--> <键> uiwindowsscensessionroleuployment </键>
<数组>
<字典>
<键>用户用户名</键>
<字符串> UIWindowScene </字符串>
<键>用户配置名称</键>
<字符串>电话</字符串>
< key > uissenedlegaclassname </key >
<字符串> my appwindowscenedelegate </字符串>
</dict >
</array >
<!--声明 CarPlay 场景--> <键> cptemplateapplicationcenesessionroleuployment </键>
<数组>
<字典>
<键>用户用户名</键>
<字符串> CPTemplateApplicationScene </字符串>
<键>用户配置名称</键>
<字符串> my app-Car </字符串>
< key > uissenedlegaclassname </key >
```

```

<字符串> MyApp.CarPlaySceneDelegate</string>
</dict>
</array>
</dict>
</dict>

```

在上面的例子中，应用程序声明了两个场景——一个用于 iPhone 屏幕，一个用于 CarPlay 屏幕。

充当场景委托的类的名称在清单中由定义

UISceneDelegateClassName。您的代表必须符合

cpTemplateApplicationSceneDelegate。听听滴滴连接和滴滴断开的方法，了解你的应用程序何时在 CarPlay 屏幕上启动。请记住，您的应用程序可能只能在 CarPlay 屏幕上启动。

请注意，导航应用程序还为 CarPlay Dashboard 声明了一个场景，并监听不同的方法。有关详细信息，请参见 CarPlay 导航应用程序。

当您的应用程序启动时，您将收到一个 CPInterfaceController，它管理 CarPlay 屏幕上的所有模板。握住控制器，因为您需要它来管理模板，例如显示正在播放的屏幕或警报。

启动时，您还必须指定根模板。在下面的例子中，应用程序指定了一个 CPListTemplate 作为根模板。

导入 CarPlay

```

类 CarPlaySceneDelegate:UIResponder, cpTemplateApplicationSceneDelegate { var
interface controller:cpinterface controller?

```

// CarPlay 已连接

```

func template application scene(_ template application scene:cpTemplate application scene, did
connect interface controller:cpinterface controller){ self . interface controller =
interface controller let list template:cplist template =...
interface controller . setroottemplate(列表模板, 动画:true)
}

```

// CarPlay 已断开连接

```

func template application scene(_ template application scene:cpTemplate application scene, did
disconnect interface controller:cpinterface controller){ self . interface controller = nil
}
}

```

创建列表模板

下面的示例演示如何创建一个列表，该列表包含带有标题和副标题的单个列表项。

当用户选择一个列表项时，您的列表项处理程序将被调用。您应该在这里采取适当的措施，例如在音频应用程序的情况下开始音频回放。如果您启动异步工作，并且没有立即调用完成块，**CarPlay** 将显示一个微调器，让用户知道您的应用程序正忙。当您准备好继续时，您必须调用完成块来告诉 **CarPlay** 移除微调器。

```
导入 CarPlay

let item = CPLListItem(text:"我的标题", detailText:"我的副标题")
item.listItemHandler = { item, completion, [弱自我] in //开始异步播放...

self . interface controller . pushtemplate(cpnowplayingtemplate . shared(), 动画:true)完成()
}

let section = CPLListSection(items:[item])
let listTemplate = CPLListTemplate(标题:"相册", sections:[section])
self . interface controller . pushtemplate(listTemplate, 动画:true)
```

创建一个正在播放的模板

现在播放的模板是一个共享实例，因此您需要获取它并配置它的属性。

当界面控制器连接到您的应用程序时，请执行此操作，因为 **iOS** 可以代表您显示共享的“正在播放”模板。例如，当用户轻按 **CarPlay** 主屏幕上或应用程序导航栏中的“正在播放”按钮时，**iOS** 将立即显示共享的正在播放模板。

这个例子展示了一个在当前播放模板上配置播放速率按钮的应用。

```
导入 CarPlay 类 CarPlaySceneDelegate:UIResponder,

cptemplateapplicationcenedelegate {

func template application cene(_ template application cene:cptemplate application cene, DidConnect
interface controller:cpinterface controller){ 0

let now playing template = cpnow playing template . shared()
```

```
let rate button = cpnowplaygplaybackrate button(){ 0
//更改播放速率!
}

now playing template . updatenow playing 按钮([rateButton])
}
}
```

播放音频

如果您的应用程序播放音频，请确保它与车内的音频源配合良好。

准备好播放音频后，立即激活您的音频会话。当您激活音频会话时，车内的其他音频源将停止。例如，如果用户正在收听汽车的调频收音机，当他们启动您的应用程序时，他们不会期望调频收音机停止。调频收音机应该继续播放，直到他们使用您的应用程序明确选择另一个音频流。不要在应用程序启动时激活音频会话。相反，等到你真的需要播放音频。

iPhone 锁定时工作

当 iPhone 处于锁定状态时，会频繁使用 CarPlay。全面测试您的应用程序，以确保在 iPhone 锁定时它能按预期工作。

在 iPhone 锁定的情况下，当启动或运行时，您将无法访问以下任何内容。

- 使用 `NSFileProtectionComplete` 或 `nsfilecoprotection complete unlespen` 保存的文件。
- 具有 `kSecAttrAccessibleWhenPasscodesetthisdeviceonly`、`kSecAttrAccessibleWhenUnlocked` 或 `kSecAttrAccessibleWhenUnlockedDeviceOnly` 属性的钥匙串项目。

CarPlay 导航应用

下一节描述如何创建一个 CarPlay 导航应用程序。

CarPlay 导航应用具有不同于其他 CarPlay 应用类别的附加 UI 元素和功能。如果您没有创建导航应用程序，请跳过此部分。

模板

CarPlay 导航应用使用额外的模板来显示地图信息、键盘和语音控制反馈。

基础视图

所有的 CarPlay 导航应用都是从基础视图开始的。基础视图是您绘制地图的地方。创建基础视图，并在 CarPlay 启动时将其附加到提供的窗口。

您需要能够在车辆屏幕上以各种长宽比和分辨率绘制地图。有关使用不同显示器配置进行测试的更多信息，请参见模拟器。基础视图只能用于绘制地图，不能用于绘制警报、覆盖图或其他用户界面元素。出现在屏幕上的所有用户界面元素，包括导航栏和地图按钮，都必须使用提供的模板来实现。您的应用程序不会在基础视图中接收直接点击或拖动事件。基本视图可用于导航应用程序。



基础视图

地图

地图模板是一个控制图层，显示为基础视图的覆盖图，允许用户操作地图。它由一个导航栏和地图按钮组成，绘制成单独的覆盖图。默认情况下，导航栏在用户与应用程序交互时出现，在一段时间不活动后消失。您可以自定义此行为，包括是否隐藏地图按钮。

导航栏包括最多两个前导按钮和两个尾随按钮，可以用图标或文本指定。

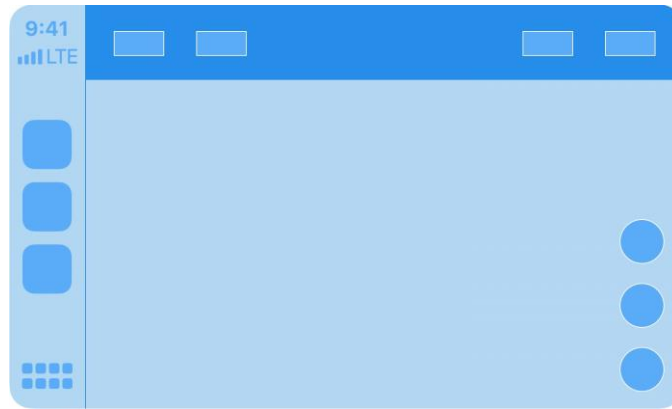
您还可以指定最多四个以图标形式显示的地图按钮。使用地图按钮提供缩放和平移功能。虽然许多汽车支持通过直接操作汽车的触摸屏来平移，但也有一些汽车只支持通过旋钮或触摸板事件来平移。**CarPlay** 通过“平移模式”支持这些汽车。如果您的应用程序支持任何平移功能，您必须将其

其中一个地图按钮分配为平移按钮，以允许

用户进入平移模式，您必须响应中的平移功能

`CMapTemplate`。

基本视图可用于导航应用程序。

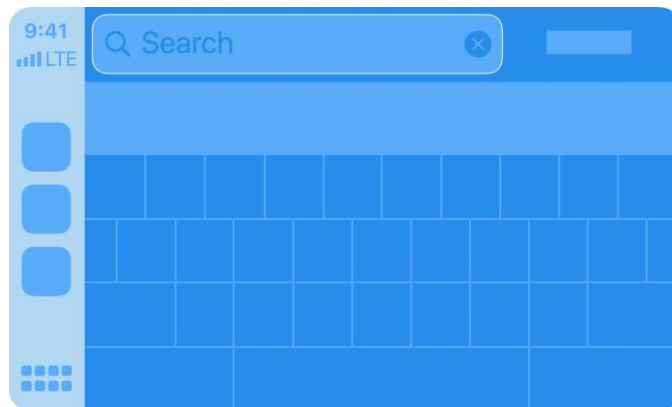


地图

搜索

搜索模板显示文本输入字段、搜索结果列表和键盘。您的应用程序通过响应 `updatedSearchText` 并使用 `CPListItem` 元素数组更新搜索结果列表来解析文本。当用户通过响应 `selectedResult` 从列表中选择项目时，您也必须采取措施。

请注意，许多汽车限制键盘何时显示。有关详细信息，请参见键盘和列表限制。搜索模板可以在导航应用程序中使用。



搜索

语音控制

语音控制模板允许您在语音控制会话期间提供视觉反馈。**CarPlay** 导航应用可以提供语音控制功能，但必须仅限于导航功能。此外，只要语音控制音频会话处于活动状态，导航应用程序就必须显示语音控制模板。

语音控制模板可用于导航应用程序。其他 **CarPlay** 应用程序必须使用 **SiriKit** 或 **Siri** 快捷键来提供语音控制功能。



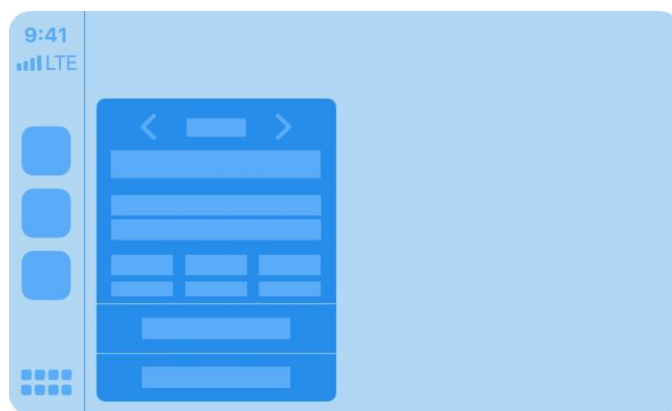
语音控制

嵌板

CarPlay 导航应用利用面板在地图上叠加信息。这包括行程预览、路线选择、路线指引和导航提醒。您不需要直接创建面板。相反，使用提供的 API 来触发它们。

行程预览

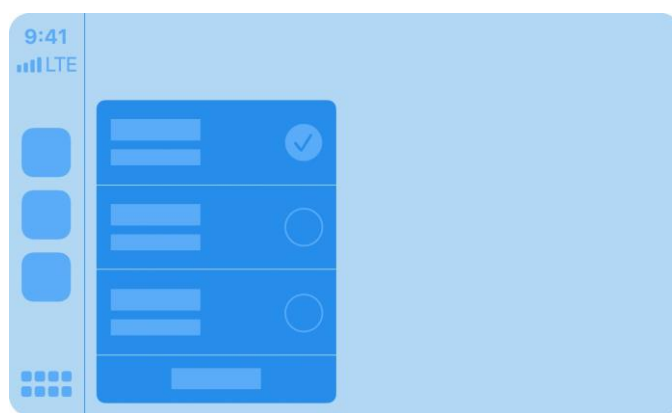
行程预览面板允许用户从最多 12 个目的地的列表中进行选择，通常是作为目的地搜索的结果。当用户预览行程时，在基础视图中显示该行程的可视化表示。



行程预览

路线选择

路线选择面板允许用户从特定旅程的潜在路线列表中进行选择。每条路线都应该有清晰的描述，以便用户可以选择自己喜欢的路线。例如，路线的摘要和可选描述可以是“途经 280 号州际公路南段”和“交通不拥挤”。当用户预览路线时，在基础视图中显示该路线的可视化表示。



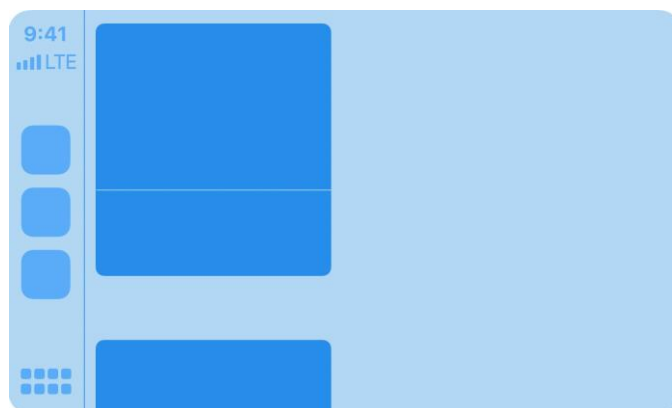
路线选择

指南和行程估计

制导和行程估计面板显示即将到来的机动和行程估计。

一次显示一个机动，但是如果机动快速连续出现，可以显示两个机动。第二次操纵也可用于显示第一次操纵的车道引导或交叉路口图像。

每个动作可以包括一个符号、指令文本、估计的剩余距离和时间。除了提供即将到来的演习，你应该不断更新整体行程估计。



指南和行程估计

提供机动图像时，请遵循以下指南。对于机动符号，提供明暗界面的变体。

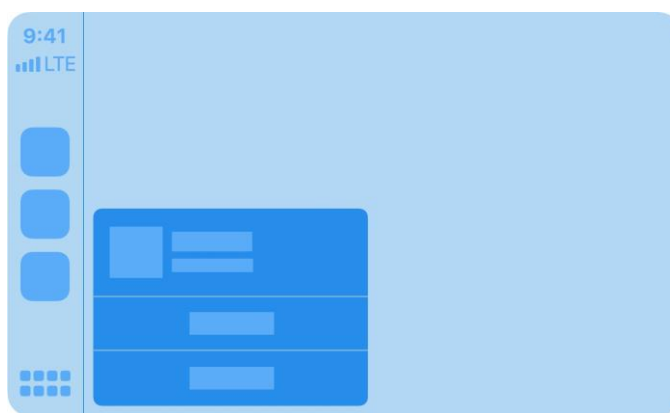
	最大尺寸 (3x) 像素	最大尺寸 (2x) 像素	以磅为单位的最大尺寸	像素
第一个机动符号 (一行上的符号和说明)	50 磅 x 50pt	磅	150 像素 x 150px 像素	100 像素 x 100px 像素
第一个机动符号 (两行符号和说明)	120 磅 x 50pt	磅	360 像素 x 150px 像素	240 像素 x 100px 像素
第二个机动符号(符号和说明)	18pt x 18pt		54px x 54px	36px x 36px
第二个符号(仅符号)	120 磅 x 18pt	磅	360 像素 x 54px 像素	240 像素 x 36px 像素
CarPlay 仪表板交叉点图像 像素	140 磅 x 100pt	磅	420 像素 x 300px 像素	280 像素 x 200px 像素

导航警报

导航警报提供重要的实时反馈，并可选地为用户提供做出影响当前路线的决定的机会。例如，如果前方有意外流量，并且您建议用户选择备用路线，则应显示警报。如果您的应用程序在后台运行，导航提醒会触发通知。

使用图像和简短的文本描述向用户提供信息。您可以提供主要和次要操作按钮。例如，用户可以选择保持当前路线，或者选择替代路线。

您还可以指定警报在自动解除之前可见的持续时间。



导航警报

启动

CarPlay 导航应用可以声明两个 CarPlay 场景，一个用于 CarPlay 中的主应用窗口，一个用于 CarPlay Dashboard。如果您正在使用场景清单，请用关于 CarPlay 仪表板场景的信息扩展它。有关详细信息，请参见仪表板场景清单。为 CarPlay 场景和 CarPlay 仪表板场景提供代理。听听滴滴连接和滴滴断开的方法，了解你的应用程序何时在每个场景中启动。在主应用程序窗口中，将使用接收接口控制器和窗口的滴滴连接和滴滴断开方法调用您的 `cptemplateapplicationcendelegate`。

CPInterfaceController 和一个 CPWindow 对象。

对于主应用程序视图，在 CarPlay 会话期间保留对界面控制器和地图内容窗口的引用。

```
self . interface controller = interface controller self .  
carwindow = window
```

接下来，创建一个新的视图控制器，并将其分配给窗口的根视图控制器。使用视图控制器将地图内容作为窗口中的基础视图进行管理。

```
让 rootview controller = myrootview controller() 窗口 . rootview controller  
= rootview controller
```

最后，创建一个地图模板，并将其指定为根模板。

```
let root template:CMapTemplate = createRoot template()self . interface  
controller? 设置根模板(根模板, 动画:假)
```

创建一组默认的导航栏按钮和地图按钮，并将它们分配给根地图模板。通过设置引导导航工具栏按钮和跟踪导航工具栏按钮阵列来指定导航栏按钮。通过设置地图按钮阵列来指定地图按钮。

如果您的 CarPlay 导航应用程序支持平移，您创建的按钮之一必须是平移按钮，允许用户进入平移模式。在不支持通过触摸屏进行平移的车辆中，平移按钮是必不可少的。

您可以根据应用程序的状态动态更新导航栏按钮和地图按钮。例如，在活动路线引导期间，您可以选择用结束路线引导的选项替换默认导航栏按钮。

路线指引

所有的 CarPlay 导航应用都有一个选择目的地和提供路线指引的标准流程。

选择目的地。所有路线引导都是从用户选择目的地开始的，无论这是屏幕搜索、语音命令的结果，还是从列表中选择类别或目的地。

预览。选择目的地后，会向用户显示行程预览。同时，基础视图中的地图通常会显示旅程的可视化表示。当有多个匹配的目的地时，预览还支持消歧。例如，如果用户选择导航到附近的公园，预览可以显示多达 12 个附近的公园供选择。

选择路线并开始引导。一旦用户确认了目的地，他们就可以开始路线引导。如果有多条可能的路线，您可以将这些路线作为选项提供给用户选择。

显示行程信息和即将到来的演习。当用户开始路线引导时，显示实时信息，包括即将到来的机动和行程估计(距离和剩余时间)。

结束指导。路线引导一直持续到用户到达目的地，或者选择结束路线引导。

选择目的地

使用 `CPInterfaceController` 呈现允许用户指定目的地的模板。若要呈现新模板，请将 `pushTemplate` 与受支持的 `CPTemplate` 类(如 `CPGridTemplate`、`CPListTemplate`、`CPSearchTemplate` 或 `CPVoiceControlTemplate`)一起使用。

当用户选择一个项目或取消选择时，您的委托将被调用，并显示所采取的操作的信息。

您可以连续呈现多个模板来支持分层选择。例如，您可以显示一个列表模板，该模板包含列表项，这些列表项在被选中时会导致其他子列表。对于支持分层浏览的列表项，请确保将 `showsDisclosureIndicator` 设置为 `true`，并在选择列表项时推送新的列表模板。

分层选择不得超过五个深度级别。

试映

在用户选择目的地并且您准备好显示行程预览后，使用

`CPMapTemplate showTripPreviews` 提供最多 12 个 `CPTrip` 对象的数组。每个 `CPTrip` 对象代表一个旅程，该旅程由一个起点、一个终点、多达 3 个路线选择以及剩余时间和距离的估计组成。

使用 `CPRouteChoice` 来定义每个路由选择。您对每条路由的描述都以可变长度字符串数组的形式提供，按长度降序排列(最长字符串优先)。CarPlay 将显示适合屏幕可用空间的最长字符串。对于每次 `CPTrip`，请确保使用 `CPMapTemplate updateEstimates` 提供行程估计:如果剩余时间或距离发生变化，请更新估计。您还可以自定义行程预览面板中显示的开始、概览和附加路线按钮的名称。

选择路线并开始引导

当用户选择不同的路线进行预览时，将调用代表 `selectedPreviewFor:`进行预览。通过更新您的地图基础视图进行响应。

如果用户决定开始旅行，将调用代理 `startedTrip:`方法。通过启动路线指引进行响应。此时，使用 `CPMapTemplate hideTripPreviews` 关闭行程预览面板。

```
mapTemplate.hideTripPreviews()
```

接下来，使用 `CPMapTemplate startNavigationSession` 为所选行程启动导航会话，并获取表示活动导航会话的 `CPNavigationSession` 对象。

```
let session = MapTemplate . startnavigationession(for:trip)
```

在计算初始机动时，将导航会话暂停状态设置为 `CPTripPauseReasonLoading`，以便 CarPlay 可以显示正确的状态。

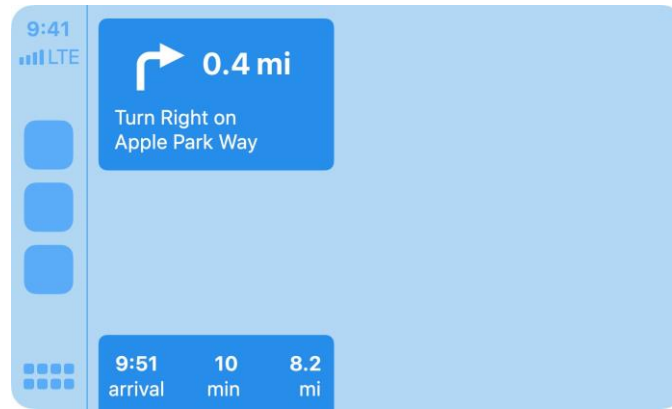
```
session.pauseTrip(对于:.CPTripPauseReasonLoading)
```

此时，更新导航栏按钮和地图按钮，为用户管理路线提供适当的操作。

显示行程信息和即将到来的机动

在逐个转弯引导期间，通过更新显示路线引导信息

更新关于即将到来的回合的信息。每个 `Each` 代表一个单独的操作，可能包括一个符号、一个指令和剩余时间和距离的估计。



在路线引导面板中显示一个操作

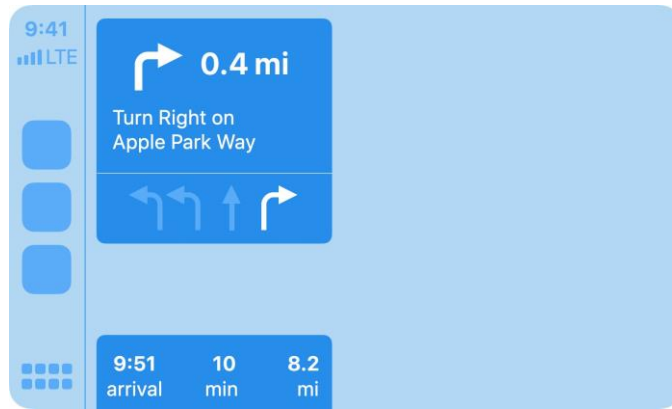
符号。如果机动有相关的符号，如右转箭头，使用符号集提供图像。该符号将显示在路线指南卡 and 任何相关通知中。您必须使用 `CPIImageSet` 提供两种图像变体——一种用于在浅色背景上渲染符号，另一种用于在深色背景上渲染符号。

指令。使用 `instructionVariants` 提供一个指令，它是一个字符串数组。使用数组提供不同长度的变体，这样 CarPlay 就可以显示最适合屏幕可用空间的指令。例如，如果机动要求您在名为“太阳圈”的街道上右转，您可以选择提供 3 种指令变体“在太阳圈右转”、“在太阳圈右转”。，和“右转”。CarPlay 将显示在可用空间中具有最长字符串长度的指令。指令数组必须按长度降序提供(最长字符串优先)。您可以选择提供 `attributedInstructionVariants`，以便在指令中包含嵌入的图像。如果需要显示特殊符号(如公路符号)作为说明的一部分，这将非常有用。请注意，包括文本大小和字体在内的其他文本属性将被忽略。如果您提供了 `attributedInstructionVariants`，请始终提供纯文本的 `InstructionVariants`，因为 CarPlay 车辆可能不总是支持属性化字符串。

始终保持机动阵列中至少一个即将到来的转弯。在快速连续进行两次机动的情况下，提供可能同时在屏幕上显示的第二次机动。

如果提供第二个操作，可以通过指定符号样式来自定义其外观。在 `CPMapTemplateDelegate` 中，当请求时，为该操作返回一个 `cp` 机动布局样式。显示样式仅适用于第二个动作。

如果您的应用程序提供车道引导信息，您必须使用第二个机动来显示车道引导。创建包含符号集的第二个操纵，暗图像和亮图像占据了整个制导面板的宽度(最大尺寸为 120pt x 18pt)，为指令变量提供一个空数组，并在 `CPMapTemplateDelegate` 中为该操纵返回一个符号样式 `cpmagnetierdisplaystyle symbolonly`。



显示带有车道引导信息的机动

您的应用程序负责持续更新每次机动的剩余时间和距离以及整个行程的估计值。使用 `cpnavigationession update estimates` 更新每次机动的估计值，使用 `CPMapTemplate updateEstimates` 更新行程的总体估计值。仅在发生重大变化时更新值，例如剩余分钟数发生变化时。

如果您需要显示与地图或导航相关的警报，请创建一个 `CPNavigationAlert`，并使用 `CPMapTemplate` 显示它。导航警报可以配置为在固定时间间隔后自动消失。它们也可能显示为通知，即使您的应用程序不在前台。

对于每个机动和导航警报，指定当您的应用程序在后台运行时，它是否应该显示为 **CarPlay** 通知。回应

应显示通知用于指定机动或导航警报行为的委托调用。在机动的情况下，您可以选择将更新行程估计作为通知的一部分。

除了路线引导面板之外，机动也可以显示在通知中，或者发送到支持在组合仪表或平视显示器中显示 **CarPlay** 元数据的车辆。

结束指南

当路线指引暂停、取消或完成时，调用中的相应方法

`CPNavigationSession`。

在某些情况下，系统可能会取消 **CarPlay** 路线引导。例如，如果汽车的原生导航系统启动路线引导，**CarPlay** 路线引导会自动终止。在这种情况下，您的代表将收到 `mapTemplateDidCancelNavigation`，您应该立即结束路线指引。

键盘和列表限制

一些汽车在驾驶时限制键盘的使用和列表的长度。**iOS** 会自动禁用键盘，并在汽车指示应该这样做时减少列表长度。但是，如果您的应用程序需要调整其他用户界面元素来响应这些更改，您可以

在限制更改时收到通知。例如，当列表长度较短时，您可能希望禁用键盘图标或调整列表项。使用 `CPSessionConfiguration` 观察有限的用户界面。

语音提示

语音提示对于路线引导体验至关重要，但您必须确保您的应用程序是一个良好的音频公民，并且与 iPhone 和车内的其他音频源配合良好。

音频会话配置

CarPlay 导航应用程序在为即将到来的机动播放语音提示时，必须使用以下音频会话配置。

1. 将音频会话类别设置为 `AVAudioSessionCategoryPlayback`。

2. 将音频会话模式设置为 `AVAudioSessionModeVoicePrompt`。

3. 将音频会话类别选项设置为

```
avaudiosessioncategoryoptions interrupted naudioandmixwith others  
和 avaudiosseoncategoryoptions duck others。
```

语音提示通过单独的音频通道播放，并与车内的音频源混合，包括车内自己的音频源，如调频收音机。

`avaudiosessioncategoryoptions interruptionstrandnaudioandmixwitothers` 允许语音提示暂停某些带有语音音频的应用程序 (如播客或有声读物)，并与音乐等其他应用程序混合。

`avaudiosessioncategoryoptions duckeothers` 允许在播放音频时，语音提示闪开 (降低音量) 其他应用程序，如音乐。

激活和停用音频会话

在您准备好播放语音提示之前，保持您的音频会话处于停用状态。仅当语音提示准备播放时，使用是呼叫设置激活。如果您知道多个音频提示将快速连续播放，您可以在短时间内保持音频会话活动。但是，当您的音频会话处于活动状态时，音乐应用程序将保持回避状态，带有语音音频的应用程序将保持暂停状态。如果没有播放音频提示，请不要保持活动状态超过几秒钟。

播放完语音提示后，按“否”调用“设置活动”以允许其他音频继续播放。

提示样式

在某些情况下，播放语音提示没有意义。例如，用户可能正在打电话，或者正在使用 Siri。

就在播放每个语音提示之前，检查音频会话的 `promptStyle`。如有必要，它将返回一个提示，以改变您应该播放的提示类型，以响应其他系统音频。

提示样式	行动	
	没有人	不要播放任何声音
	短的	播放音调
标准	播放完整的语音提示	

CarPlay 仪表板

从 iOS 13.4 开始，可以增加对 CarPlay Dashboard 的支持。在 CarPlay 仪表板中，让您的地图、即将到来的机动和仪表板按钮一目了然。CPTemplateApplicationDashboardScene 是 CarPlay 在确定您的应用程序应该出现在 CarPlay Dashboard 中时创建的 UIScene 子类。“控制面板控制器”和“控制面板按钮”允许您管理出现在 CarPlay 仪表板中的控件。

创建场景代理

首先，像定义主模板应用程序场景一样，为仪表板场景定义一个委托。此委托符合 CPTemplateApplicationDashboardSceneDelegate 和将被赋予一个 CPDashboardController 和 UIWindow 的实例。

此外，要支持 CPTemplateApplicationDashboardScene，请在应用程序场景清单中包含一个新的键，以声明支持值为 true 的 CarPlay dashboardnavigationse:cpupportsdashboardnavigationse。

绘制您的 CarPlay 仪表板内容

使用提供的窗口绘制导航相关内容，以便在 CarPlay 中显示仪表板，类似于给主模板应用场景的 CPWindow。您还可以向 CPDashboardController 提供两个 CPDashboardButton 实例，当您的应用程序不在活动导航时，它们会出现在导航卡小部件中。

人们可以通过仪表板按钮和主应用程序界面与您的应用程序进行交互。当使用 CPMAPTemplate 和 CPNavigationSession 在应用程序中开始导航时，CarPlay 会自动在导航小部件中显示机动。当导航结束时，它将恢复显示您的仪表板按钮。

仪表板场景清单

以下是支持 CarPlay 仪表板的场景清单示例。

```
< key > ui applicationcenemanifest </key >
<字典>
<键> cpsupportsdashboardnavigationse </键>
< true/>
<键>用户配置</键>
<字典>
<!--对于设备场景--> <键> uiwindowscenessionroleupation </键>
<数组>
```

```

<字典>
<键>用户用户名</键>
<字符串> UIWindowScene </字符串>
<键>用户配置名称</键>
<字符串>电话</字符串>
< key > uissenedlegaclassname </key >
<字符串> my appwindowscenedelegate </字符串>
</dict >
</array >
<!--对于主 CarPlay 场景--> <键> cptemplatesapplicationcensessionroleuployment </键>
<数组>
<字典>
<键>用户用户名</键>
<字符串> CPTemplateApplicationScene </字符串>
<键>用户配置名称</键>
<字符串> CarPlay </字符串>
< key > uissenedlegaclassname </key >
<字符串> my appcarplayscenedelegate </string >
</dict >
</array >
<!--对于 CarPlay 仪表板场景--> <键>
cptemplateapplicationdashboardscenessessionoleapplication </键>
<数组>
<字典>
<键>用户用户名</键>
<字符串> CPTemplateDashboardScene </字符串>
<键>用户配置名称</键>
<字符串> CarPlay-仪表板</字符串>
< key > uissenedlegaclassname </key >
<字符串> my appcarplaydashboardscenedelegate </字符串>
</dict >
</array >
</dict >
</dict >

```

发布您的 CarPlay 应用程序

当您准备在应用商店上发布您的 CarPlay 应用时，请遵循与任何 iOS 应用相同的流程，并使用应用商店连接提交您的应用。

确保您的应用遵循 CarPlay 应用指南。



苹果公司。
版权所有 2020 苹果公司。

保留所有权利。

未经苹果公司事先书面许可，不得以任何形式或任何方式(机械、电子、影印、录制或其他方式)复制、存储在检索系统中或传输本出版物的任何部分。但以下情况除外:任何人在此被授权将文档存储在一台计算机或设备上，仅供个人使用，并打印文档副本供个人使用，前提是文档包含苹果的版权声明。

对于本文中描述的任何技术，不授予任何明示或暗示的许可。苹果保留与本文中描述的技术相关的所有知识产权。本文档旨在帮助应用程序开发人员仅开发苹果品牌产品的应用程序。

苹果公司。
一条苹果公园路
加利福尼亚州库比蒂诺 95014 408-
996-1010

苹果是苹果公司的商标。在美国和其他国家注册。

**苹果不做任何保证或
表示，任一表达
或暗示，关于本文件，其质量，
准确性、适销性或特定用途的适用性
目的。因此，这
文件按“原样”提供，您，即读者，是
承担其质量和准确性的全部风险。在任
何情况下，苹果公司都不对直接、间接、
特殊、
由任何原因引起的附带或后果性损害
本文件中的缺陷、错误或不准确，即使被
告知此类损害的可能性。某些司法管辖区
不允许排除暗示担保或责任，因此上述排
除可能不适用于您。**