

开始使用
关于GameplayKit
游戏架构设计
打造伟大的游戏玩法
修订历史

## 关于GameplayKit

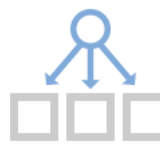
GameplayKit是用于在iOS、OS X和tvOS中构建游戏的基本工具和技术的集合。构建、发展和维护复杂的游戏需要精心策划的设计——GameplayKit提供架构工具，帮助您以最小的努力设计模块化、可扩展的游戏架构。创建伟大的游戏还需要部署复杂的算法来解决常见游戏机制背后的问题——GameplayKit还提供此类算法的标准实现，允许您将更多时间花在使您的游戏玩法独特的功能上。

由于GameplayKit独立于高级游戏引擎技术，您可以将其与任何这些技术相结合来构建完整的游戏：2D游戏的Sprite工具包，3D游戏的SceneKit，或使用Metal或OpenGL ES的自定义或第三方游戏引擎。对于图形需求较低的游戏，您甚至可以将GameplayKit与UIKit（在iOS或tvOS中）或AppKit（在OS X中）一起使用。

GameplayKit提供了七个核心功能领域，您可以将其组合或独立使用来创建游戏：



**随机化。**使用这些强大、灵活的标准算法实现作为多种游戏机制的基石。



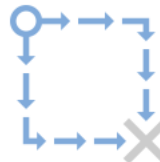
**实体和组成部分。**通过构建此架构，设计更多可重用的游戏代码。



**状态机。**使用此架构在游戏设计中解开复杂的过程代码。



**Minmax 策略师。**为您的回合制游戏和人工智能玩家对象创建一个模型，这些对象使用该模型来计划最佳动作。



**寻找路径。**将游戏世界描述为一个图表，允许GameplayKit为游戏角色规划最佳路线。



**代理人、目标和行为。**使用此模拟，让游戏角色根据高级目标移动自己，并对周围环境做出反应。



**规则系统。**将游戏设计与可执行代码分开，以加快游戏开发周期，或实现模糊逻辑推理，为游戏添加现实行为。

### 实例

本书中讨论的游戏示例可以作为示例代码提供，因此您可以下载每个游戏作为完整项目，以查看GameplayKit（和其他技术）的运行情况：

- 盒子：GameplayKit实体组件基础知识**——GameplayKit实体组件系统的基本演示，在**实体和组件**一章中讨论。
- 分配器：GameplayKit状态机基础知识**——GameplayKit状态机系统的基本演示，在**状态机**一章中讨论。
- 探路者：GameplayKit探路基础知识**——GameplayKit寻路系统的基本演示，在**探路**一章中讨论。
- 迷宫：GameplayKit入门**——一款简化的经典街机游戏，使用GameplayKit中的许多设计和游戏功能。该项目在**实体和组件**、**状态机**、**寻路**和**规则系统**章节中讨论。
- FourInARow：使用GameplayKit Minmax策略师进行对手人工智能**——使用UIKit构建的简单棋盘游戏（仅限iOS），说明使用GKMinmaxStrategist类和相关协议，在**Minmax策略师**中讨论。
- AgentsCatalog：在GameplayKit中使用代理系统**——演示了GKAgent类和代理可以遵循的几个单个目标，以及如何将目标组合成复杂的行为。**代理、目标和行为**一章讨论了这个项目。
- DemoBots：使用SpriteKit和GameplayKit构建跨平台游戏**——一款功能齐全的游戏，使用GameplayKit中几乎所有系统，并演示使用SpriteKit设计和构建多级游戏项目的技术、点播资源和Xcode 7中的新功能。该项目的部分内容在**代理、目标、行为和规则系统**章节中讨论。

### 另请参阅

本指南中的示例游戏使用UIKit、SpriteKit和SceneKit框架构建。在与其中一项技术一起使用GameplayKit之前，您应该熟悉相关概念和工具，并通常熟悉为苹果平台构建应用程序。有关您计划使用哪种游戏引擎技术的更多信息，请参阅以下文档：

- [SpriteKit编程指南](#)和[SpriteKit框架参考](#)
- [场景套件框架参考](#)
- [开始为tvOS](#)和[Xcode开发iOS应用程序（Swift）](#)或[应用程序编程指南](#)