

第 1 部分 Unity 的基本概念

第 1 部分将探讨 Unity 游戏引擎的基本原理，并介绍 Unity 中脚本的一些关键概念，为开发游戏做好准备。

第 1 章

初识 Unity

在开始使用 Unity 开发游戏之前，需要了解 Unity。很多人，尤其是对游戏和游戏开发感兴趣的人，都知道 Unity 是一个广泛使用的游戏引擎，并且可能已经玩过很多用 Unity 开发的游戏，但可能并不熟悉如何使用 Unity 来开发游戏。例如，Unity 有许多可用版本，如何选择合适的版本？Unity 提供了不同的订阅计划，怎样选择合适的订阅计划？

如果以前从未使用过 Unity，那么有必要先学习如何使用 Unity 编辑器。除了 Unity 编辑器之外，Unity 引擎还提供了哪些功能来帮助游戏开发者开发游戏呢？了解 Unity 中的特性也非常重要。对于 .NET 开发人员，可能很熟悉 Visual Studio，需要知道如何使用 Visual Studio 来开发一个 Unity 游戏，但是开发一个 Unity 游戏是不同于开发一个 .NET 应用程序的。

本章将回答这些问题。首先介绍如何选择合适的 Unity 版本，并概述如何通过 Unity Hub 或 Unity 安装程序下载和安装 Unity。然后介绍如何根据实际情况选择合适的订阅计划，此时需要已经安装 Unity 并打开 Unity 编辑器。对于刚刚开始使用 Unity 编辑器的读者，很可能不知道如何使用它。所以，本章先探索 Unity 编辑器的使用，再讨论 Unity 提供的不同特性，接着介绍 Unity 中的 .NET 配置文件和 Unity 提供的脚本后端，最后介绍如何使用 Visual Studio 来开发 Unity 游戏。

1.1 技术要求

在开始学习之前，先检查使用的系统是否可以运行 Unity 编辑器。表 1.1 给出了运行 Unity 编辑器的最低要求。

表 1.1 运行 Unity 编辑器的最低要求

操作系统	操作系统版本	CPU	图形 API
Windows	Windows 7(SP1+)、 Windows 10-64 位版	支持 SSE2 指令集的 x64 架构	支持 DX10-、DX11-、DX12- 的 GPU
macOS	High Sierra 10.13+	支持 SSE2 指令集的 x64 架构	支持 Metal 的 Intel 和 AMD GPU
Linux	Ubuntu 20.04、 Ubuntu 18.04、 CentOS 7	支持 SSE2 指令集的 x64 架构	支持 OpenGL 3.2+ 或 Vulkan 的 NVIDIA 和 AMD GPU

1.2 开始使用 Unity 编辑器

无论是独立的游戏开发者，还是公司团队中的工作人员，在安装或下载 Unity 之前都需要做以下两件事。

- (1) 选择合适的 Unity 版本。
- (2) 选择合适的订阅计划。

1.2.1 选择合适的 Unity 版本

Unity 版本如图 1.1 所示。如今，Unity 每年都提供两种不同的发布版本：LTS（长期支持）版本和 Tech Steam（技术更迭）版本。

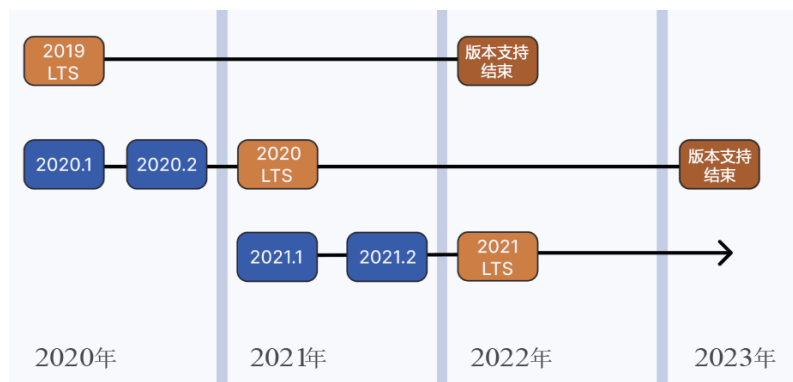


图 1.1 Unity 版本

接下来将解释这两种版本，以了解如何选择合适的版本。

LTS 版本为开发人员的项目提供了最大的稳定性和充分支持，该版本是每年技术更迭到最后生成的长期支持稳定版本。在 LTS 版本中，没有新的特性或 API 更改。LTS 版本的更新解决了崩溃问题，并修复了错误和各种小问题。每年 Unity 都会发布新的 LTS 版本，每个版本从发布之日起提供两年的相关支持。

因此，对于正在寻求性能和稳定性，以及项目已经在生产过程中或正在进行开发过程中的读者来说，使用最新的 LTS 版本来确保最佳的性能和稳定性是一个很好的选择。

注意

在撰写本书时（2022年4月），有两个 LTS 版本，即 Unity 2020 LTS 和 Unity 2019 LTS（即 2019.4）。Unity 2020 LTS 是最新的 LTS 版本，并具有与 Unity 2020.2 技术更迭版本相同的特性集，而 Unity 2019 LTS 是旧的 LTS 版本。

技术更迭版本为想要探索最新的正在开发的特性的开发者们提供了一个选项，让他们可以使用这些新特性为未来的项目做准备。与 LTS 版本不同，技术更迭版本每年发布两次（通常在第一季度和第四季度发布），并且仅在下一个技术更迭版本正式发布之前得到支持。

因此，对于正在准备下一个项目或正在致力于原型设计和实验的读者来说，可以尝试使用技术更迭版本。

注意

在撰写本书时（2022年4月），最新的技术更迭版本是 Unity 2021.2。

本书选择了 LTS 版本——Unity 2020.3。

1.2.2 选择合适的订阅计划

Unity 是一个广泛使用的游戏引擎，许多独立游戏开发者都使用 Unity 进行游戏开发。但从技术上讲，Unity 并不是一个免费的游戏引擎。本节将介绍 Unity 提供的几种订阅计划，读者可以根据自身情况选择合适的订阅计划。

Unity 提供了一系列的订阅计划，从针对个人学习者的免费个人计划到针对大型组织的企业计划。

每个 Unity 订阅计划都有不同的资格要求，具体如下。

（1）个人计划（Personal plan）是免费的，包括了 Unity 的所有基本功能。如果是个人工作者，在过去 12 个月里通过 Unity 项目获得的收入或资金低于 10 万美元，可以选择该计划。此外，学生或教育工作者在通过身份验证后可以获得 GitHub 学生开发包。

（2）Plus 计划是付费计划，提供了更多的功能和培训资源，如高级云诊断和初始屏幕自定义。如果在过去的 12 个月里通过使用 Unity 获得的收入或资金超过 10 万美元但低于 20 万美元，则应该选择该计划。

（3）Pro（专业）计划也是付费计划。与 Plus 计划相比，使用 Pro 计划可以从 Unity 获得更多的技术支持。如果你的组织在过去 12 个月里从任何来源获得超过 20 万美元的收入，那么必须使用 Pro 计划或企业计划。

（4）企业计划（Enterprise plan）是专门为至少有 20 名成员的团队设计的，提供比 Pro 计划更多的支持。例如，一位来自 Unity 的客户成功经理将被分配到你的组织，提供指导、协调资源并作为内部倡导者。

1.2.3 下载和安装 Unity 编辑器

有两种方式可以下载和安装 Unity 编辑器。第一种推荐的方式是使用 Unity Hub。Unity Hub 是一个管理工具，用来管理所有的 Unity 项目和 Unity 的安装。可以采取以下步

骤来安装 Unity Hub 和 Unity 编辑器。

(1) 访问 Download Unity 页面 (<https://unity3d.com/get-unity/download>)，如图 1.2 所示。

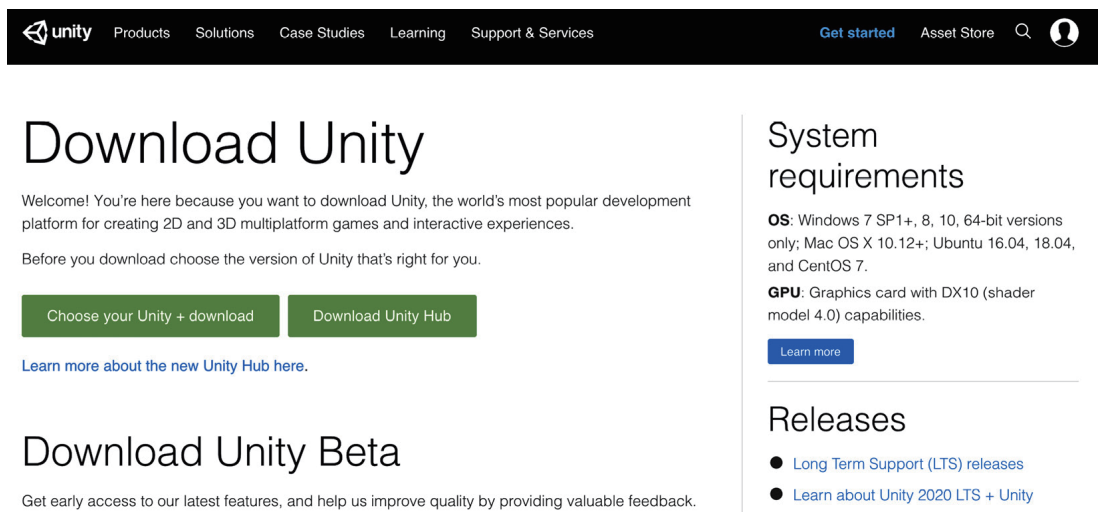


图 1.2 Download Unity 页面

Unity Hub 支持 Windows、macOS、Ubuntu 和 CentOS。

(2) 安装 Unity Hub 非常简单，只需要选择安装 Unity Hub 的文件夹，如图 1.3 所示。然后，单击 Install 按钮。

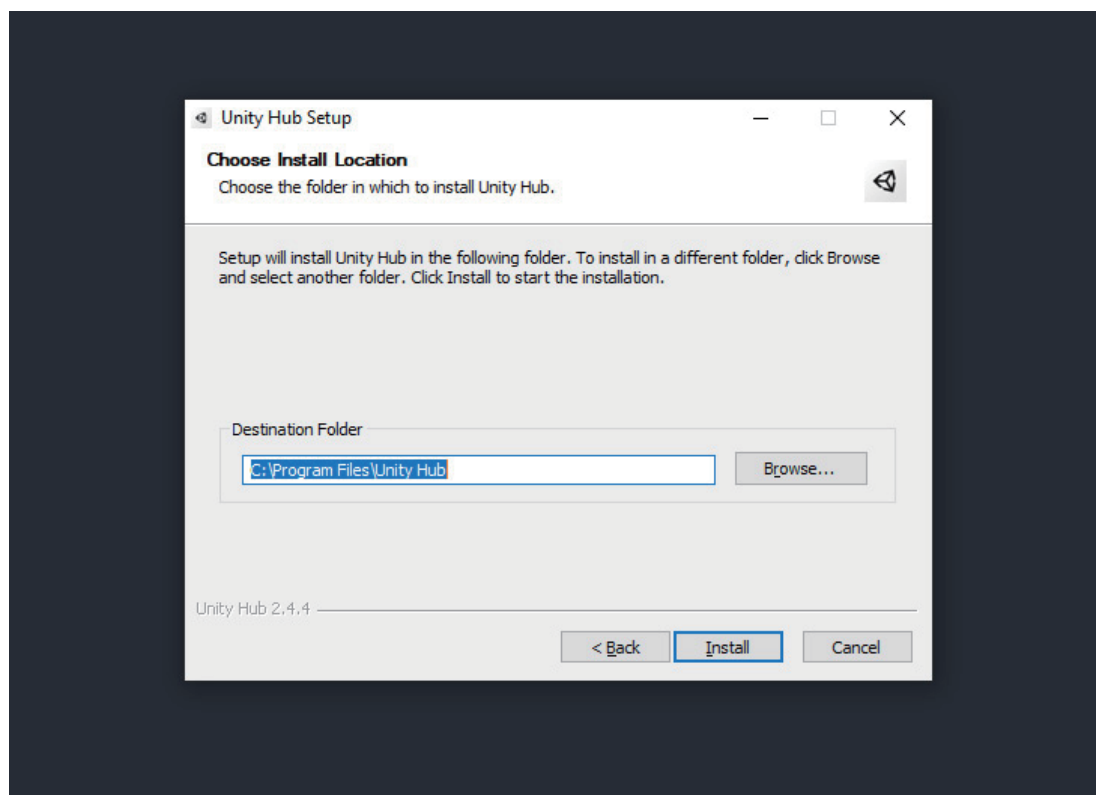


图 1.3 Unity Hub 安装

(3) 完成 Unity Hub 的安装后，勾选 Run Unity Hub 复选框，然后单击 Finish 按钮以启动 Unity Hub，如图 1.4 所示。

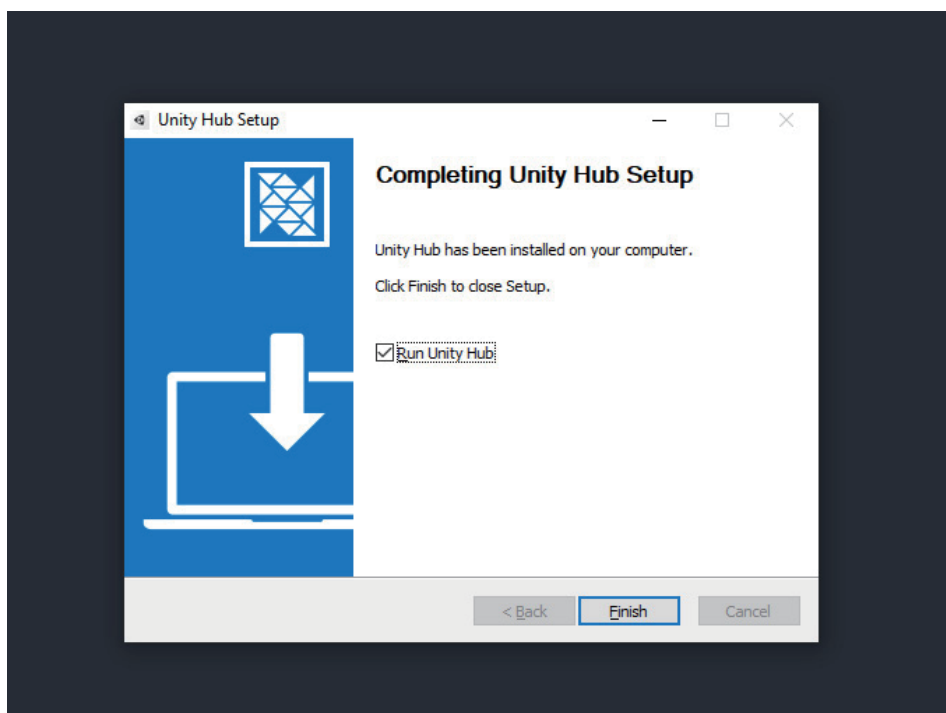


图 1.4 完成 Unity Hub 安装

在撰写本书时，使用的是当前最新版本的 Unity Hub（版本 3.0.0）。如果使用过以前版本的 Unity Hub，会发现新版本的 Unity Hub 的启动页面是完全不同的，如图 1.5 所示。

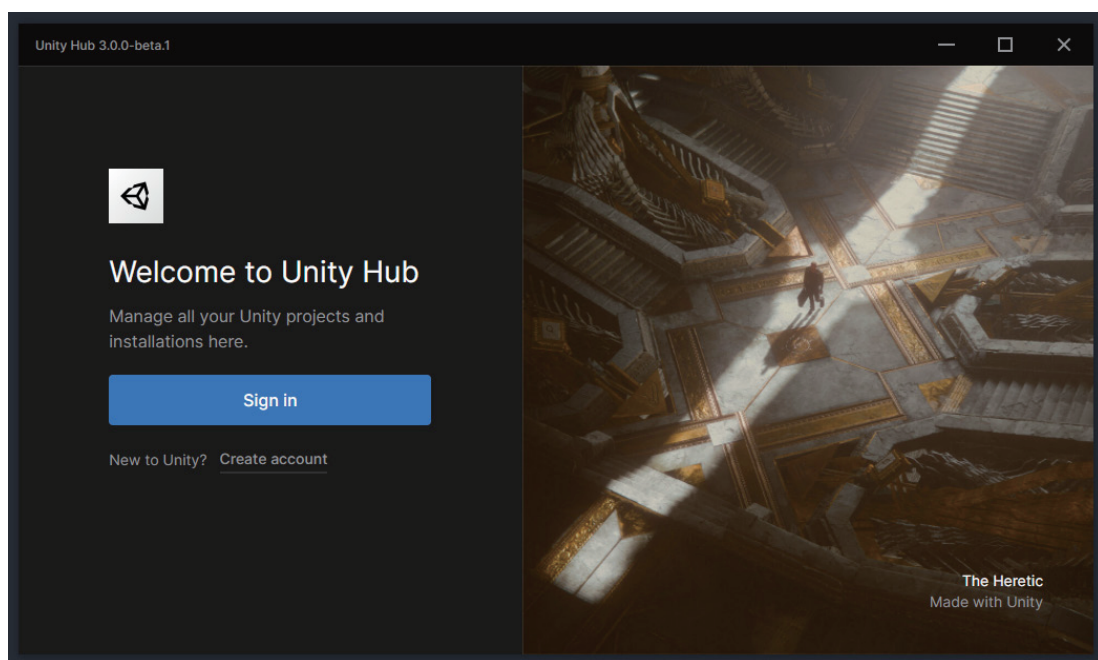


图 1.5 Unity Hub 的启动页面

(4) 需要一个 Unity 账户才能访问 Unity 编辑器和 Unity Hub。如果没有 Unity 账户，则需要创建一个账户。

(5) 第一次登录 Unity Hub 时，需要激活许可证，如图 1.6 所示。单击 Manage licenses 按钮，在打开的 Preference 对话框中选择左侧的 Licenses 选项。

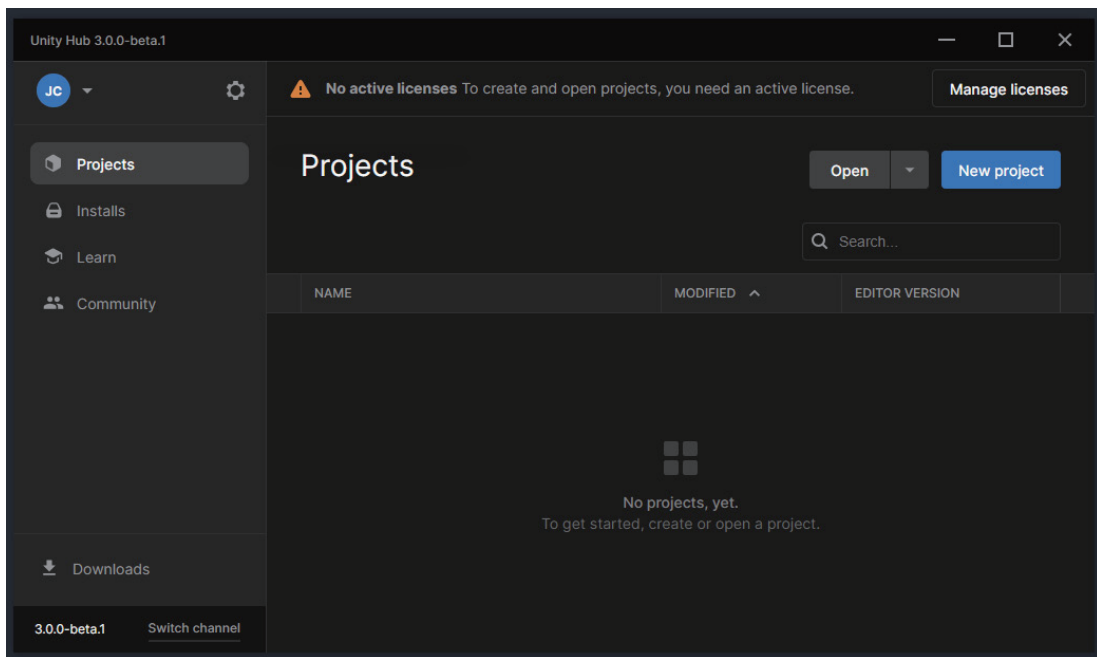


图 1.6 单击 Manage licenses 按钮

(6) Licenses 视图中提供了两个按钮，可用于添加许可证，可以单击右上角的 Add 按钮或下方的 Add license 按钮，如图 1.7 所示。

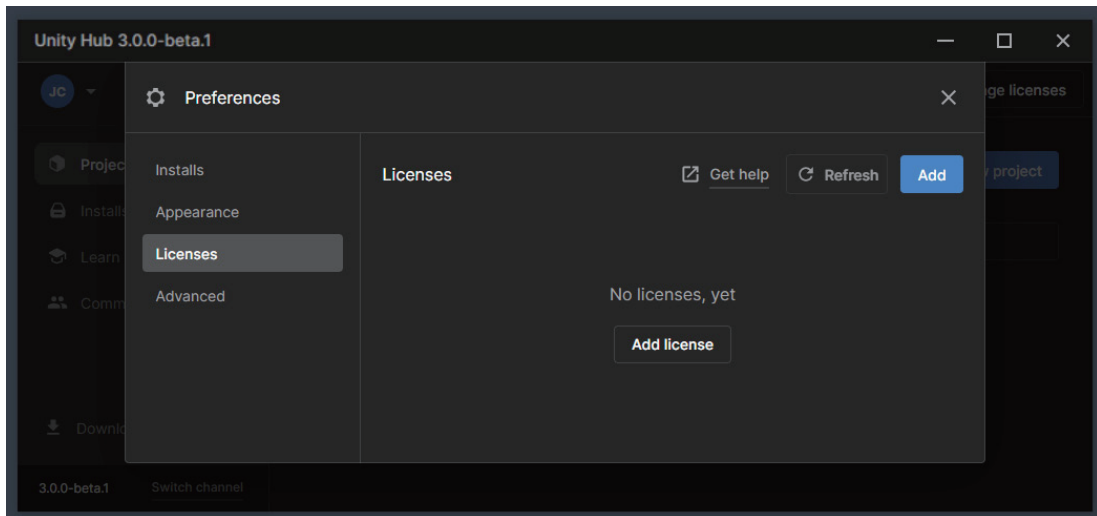


图 1.7 Licenses 视图

然后，在弹出的 Add new license 对话框中，有不同的订阅计划选项来激活许可证，如图 1.8 所示。1.2.2 节已经讨论了不同的 Unity 订阅计划。

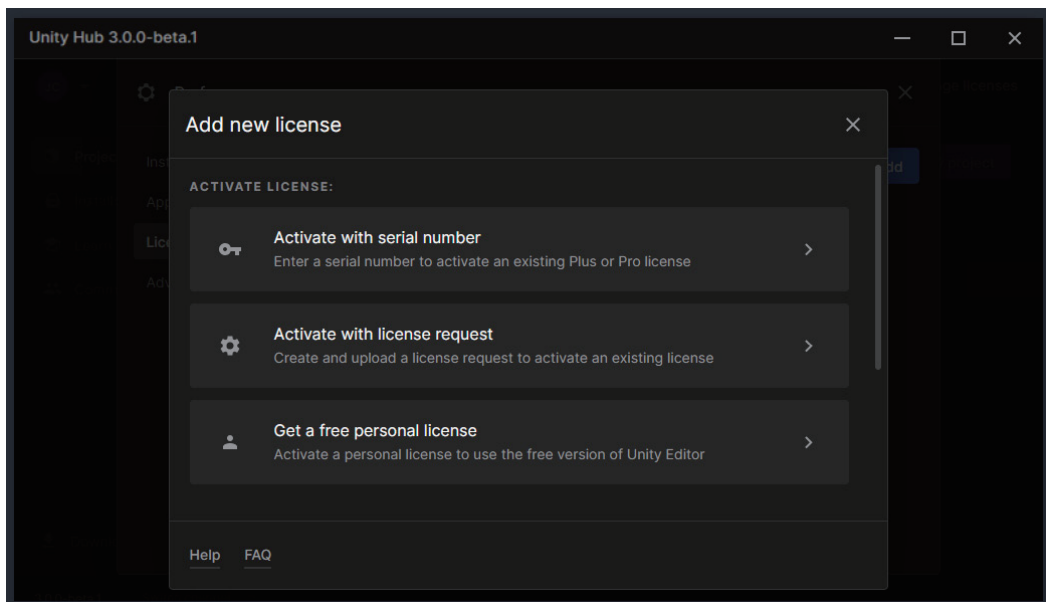


图 1.8 Add new license 对话框

(7) 添加新许可证后，可以开始探索 Unity Hub。如图 1.9 所示，从 Projects 视图中，有一个由 Unity Hub 管理的 Unity 项目列表。可以单击 Projects 视图右上角的 New project 按钮创建一个全新的项目，也可以单击 Open 按钮导入现有项目。

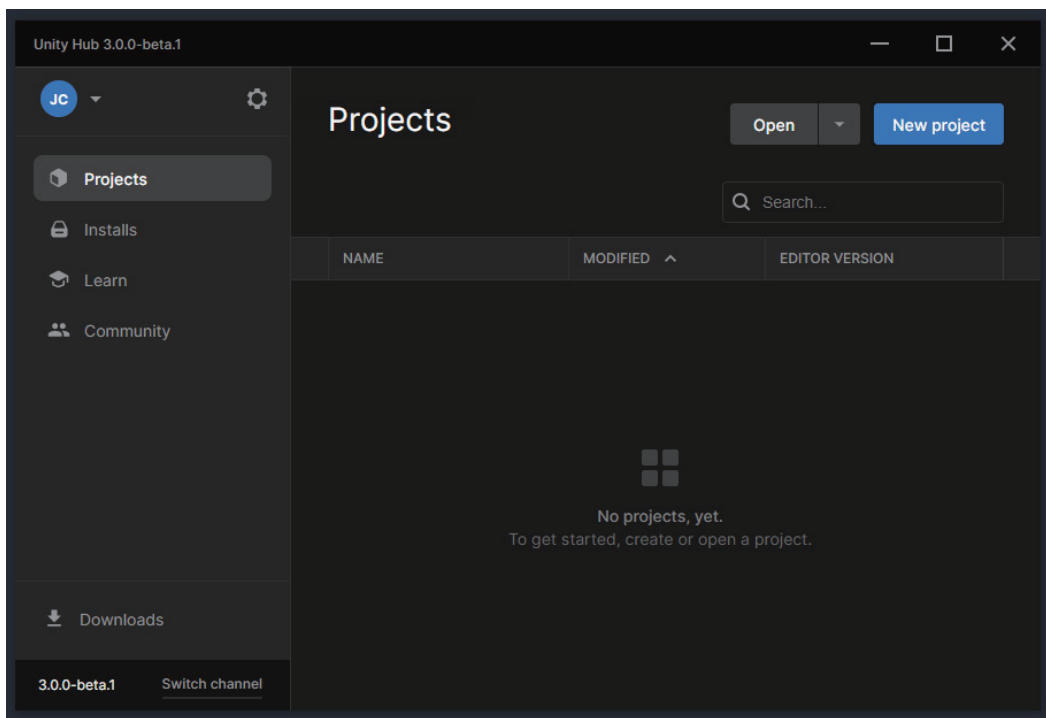


图 1.9 Projects 视图

(8) 要安装 Unity 编辑器，需要打开 Installs 视图，如图 1.10 所示，可以在其中管理多个版本的 Unity 编辑器的安装。

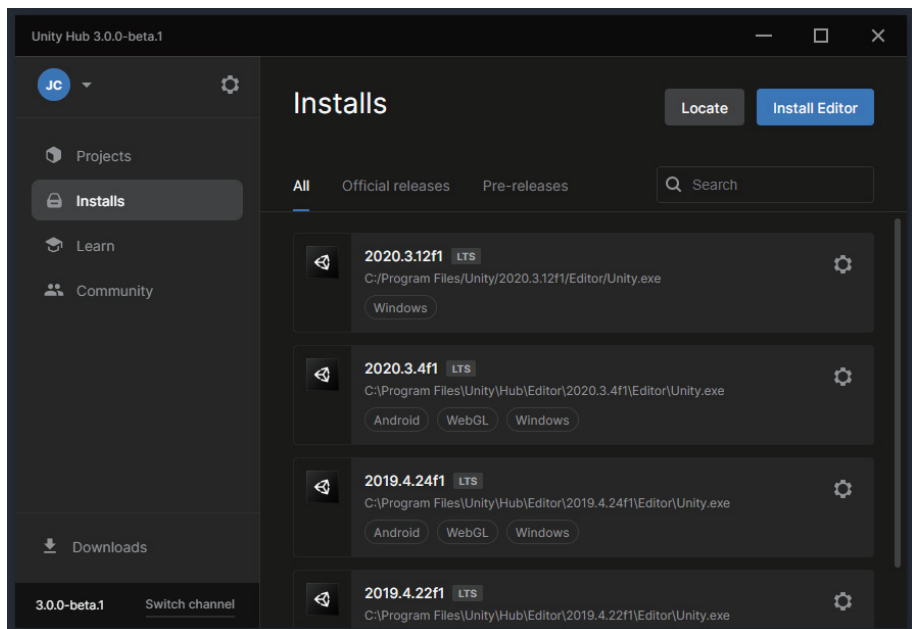


图 1.10 Installs 视图

该视图中有一个由 Unity Hub 安装和管理的 Unity 编辑器列表。与 Project 视图类似，可以下载并安装一个 Unity 编辑器，也可以导入一个不是由 Unity Hub 管理的现有 Unity 编辑器，如使用 Unity 安装程序安装的 Unity 编辑器。

(9) 单击 Installs 视图上的 Install Editor 按钮，打开 Install Unity Editor 面板，如图 1.11 所示。可以看到每个版本的最新版本。

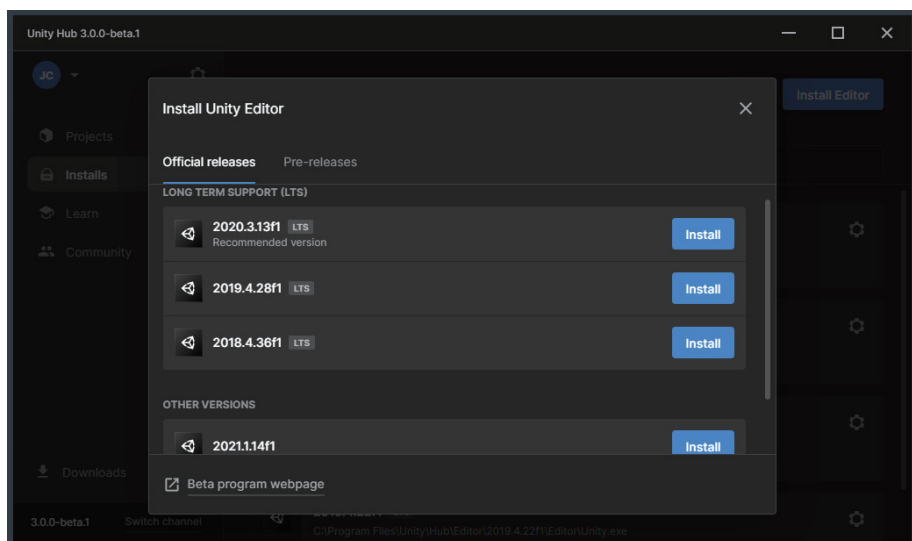


图 1.11 Install Unity Editor 面板

注意

因为 Unity 2018 LTS 的支持周期已经结束，所以不要安装它。

书中使用 Unity 2020 LTS 版本，所以需要安装 Unity 2020.3.13f1，如图 1.12 所示。

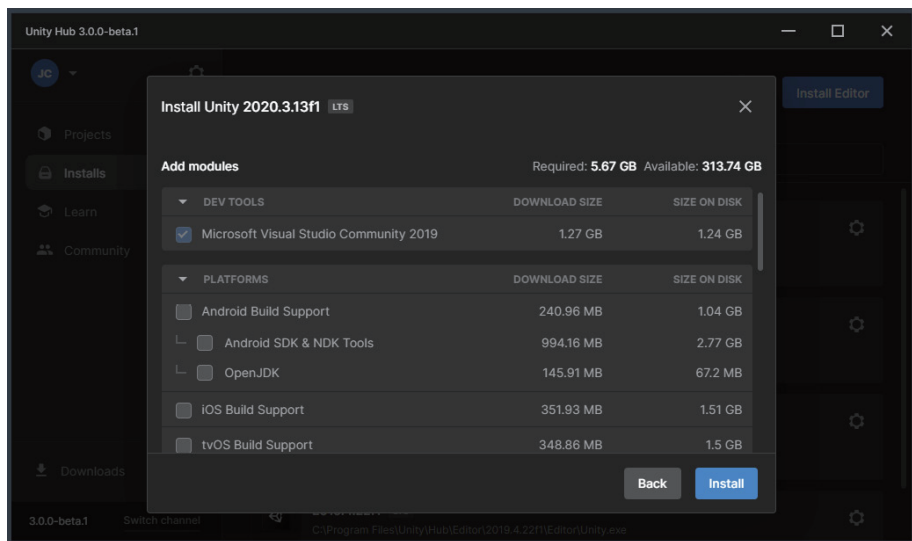


图 1.12 安装 Unity 2020.3.13f1

然后，选择需要安装的模块。可以看到，Microsoft Visual Studio Community 2019 是默认安装的，这是本书进行 Unity 游戏开发的集成开发环境（Integrated Development Environment, IDE）。

注意

如果要更改安装位置，可以在 Preferences 对话框的 Installs 视图中进行更改。

有时，可能需要一个无法通过 Unity Hub 获得的特定版本，如一些较旧的 Unity 版本。此时，可以通过第二种方式来安装 Unity 编辑器，即通过 Unity 安装程序，步骤如下。

(1) 要下载 Unity 的以前版本，需要访问 Unity download archive 页面（<https://unity3d.com/get-unity/download/archive>），如图 1.13 所示。

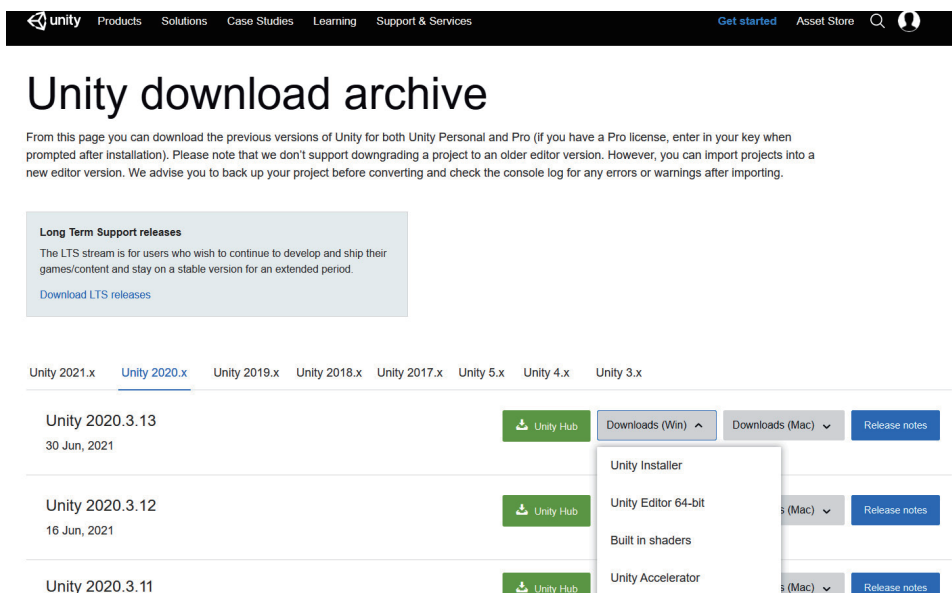


图 1.13 Unity download archive 页面

(2) Unity 安装程序界面如图 1.14 所示。单击 Next 按钮，并选择想要下载和安装的 Unity 组件。

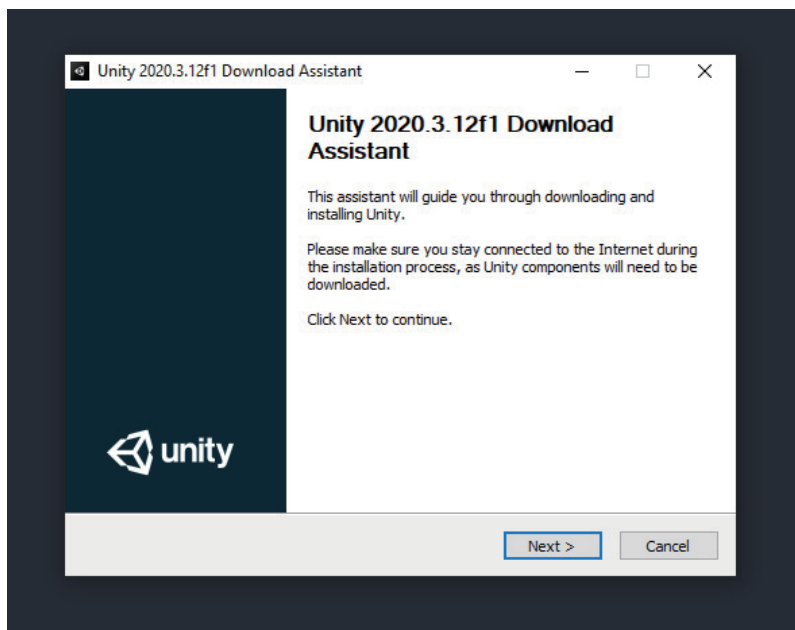


图 1.14 Unity 安装程序界面

(3) Unity 编辑器已默认选择安装。若要构建不同平台的游戏，还需要选择相应的构建支持组件，如图 1.15 所示。例如，如果想开发一款运行在 Android 设备上的游戏，需要下载并安装 Android Build Support 组件。

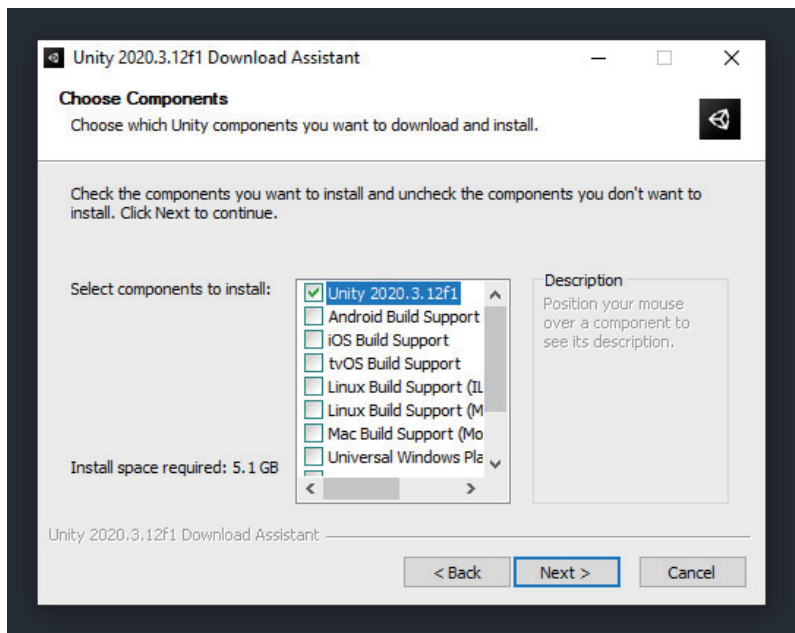


图 1.15 选择相应的构建支持组件

(4) 单击 Next 按钮，选择下载和安装的位置，如图 1.16 所示。

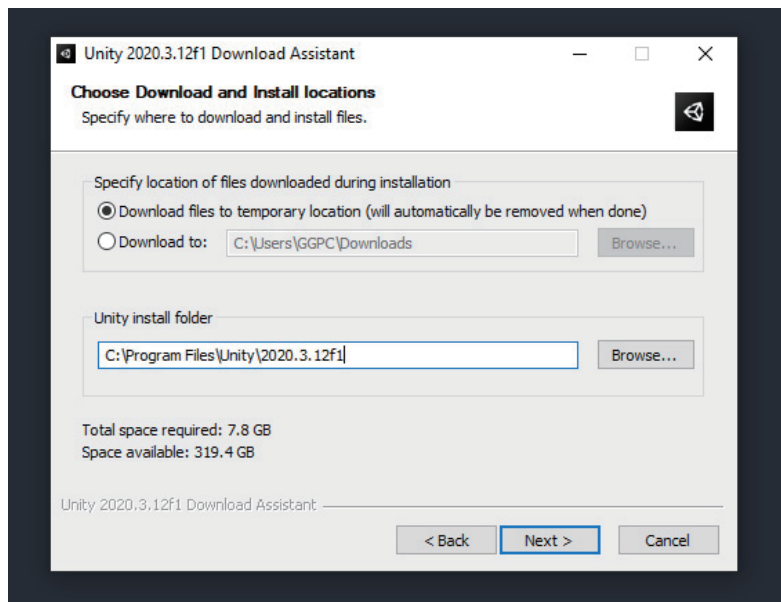


图 1.16 选择下载和安装的位置

(5) 指定位置后，单击 Next 按钮，开始下载和安装 Unity 编辑器，如图 1.17 所示。

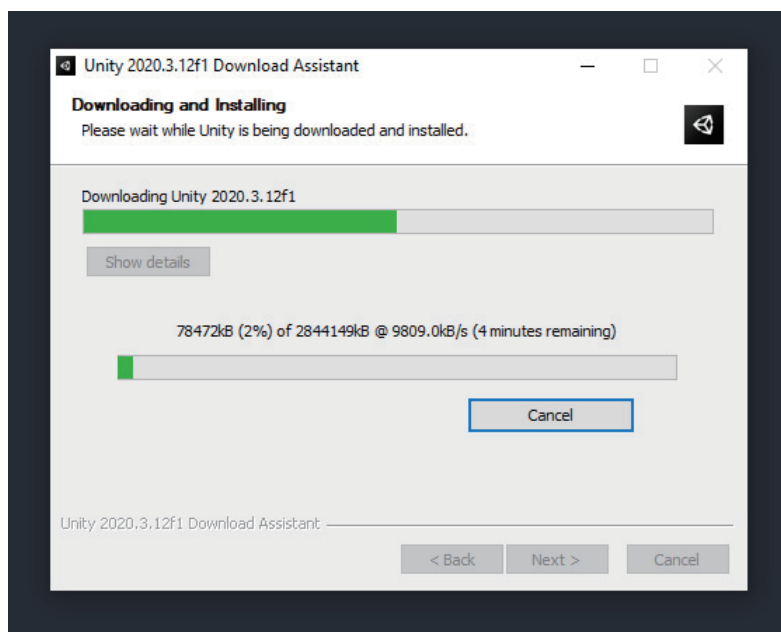


图 1.17 下载和安装 Unity 编辑器

完成下载和安装后，Unity 编辑器的图标将出现在桌面上。

1.2.4 探索 Unity 编辑器

单击 Projects 视图右上角的 New Project 按钮创建一个全新的 Unity 项目。如图 1.18 所示，可以为这个新项目选择不同的 Unity 编辑器版本。Unity 提供了一些内置的项目模板，如 2D 模板、3D 模板、HDRP 模板和 URP 模板。还可以从 Unity 上下载和安装更多的模板，

如 VR 模板和 AR 模板。在 PROJECT SETTINGS 中，可以设置项目的名称和项目的位置。

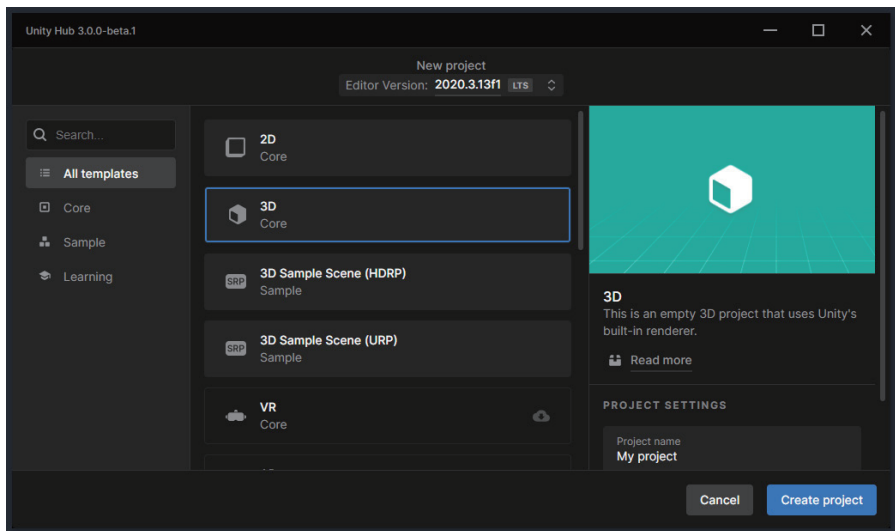


图 1.18 创建一个新项目

在此选择默认的 3D 模板，并将项目命名为 UnityBook。单击 Create Project 按钮，所选的 Unity 编辑器将启动并打开一个新项目。

Unity 编辑器界面的默认布局如图 1.19 所示，包含以下 5 个关键区域。

- (1) 工具栏。
- (2) Hierarchy 窗口。
- (3) Scene 视图和 Game 视图。
- (4) Inspector 窗口。
- (5) Project 窗口。

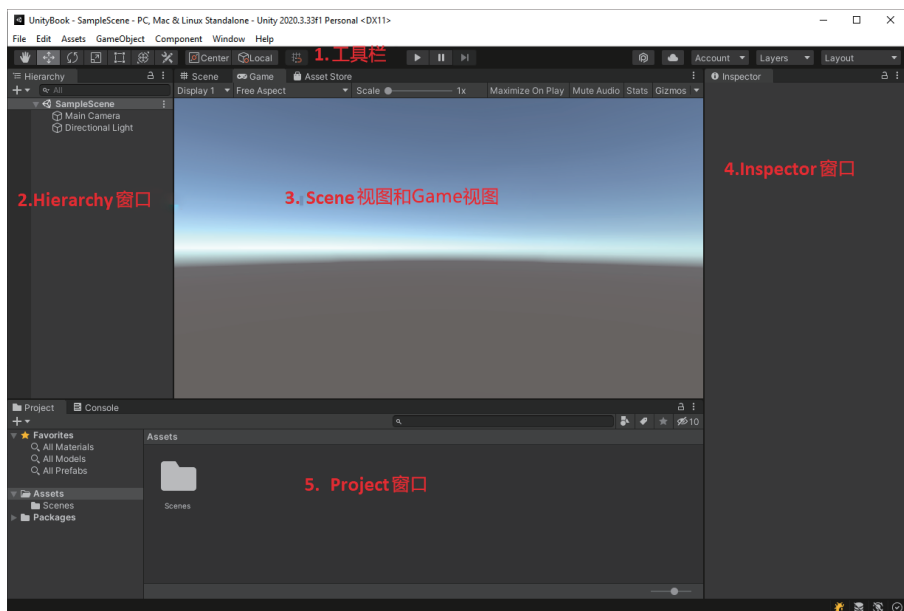


图 1.19 Unity 编辑器界面

1. 工具栏

工具栏如图 1.20 所示，位于 Unity 编辑器界面的顶部。



图 1.20 工具栏

从左到右，工具栏中的第一组工具是变换工具集，如图 1.21 所示。变换工具在 Scene 视图中使用，用于平移场景，以及移动、旋转和缩放场景中的单个游戏对象。

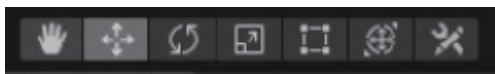


图 1.21 变换工具集

第二组工具是 Gizmo 手柄位置切换集，如图 1.22 所示，用于定义 Scene 视图中任何变换工具 Gizmo 的位置。



图 1.22 Gizmo 手柄位置切换集

然后是 Play（运行）按钮、Pause（暂停）按钮和 Step（步进）按钮，如图 1.23 所示。这三个按钮用于 Game 视图。

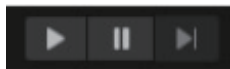


图 1.23 Play 按钮、Pause 按钮和 Step 按钮

在工具栏右侧的工具集如图 1.24 所示。首先是 Unity Plastic SCM 按钮，它允许开发者直接访问 Unity 编辑器中的 Plastic SCM 版本控制和源代码管理工具。然后，可以单击 cloud 按钮打开 Unity Services 窗口，在该窗口中可以访问 Unity 提供的许多云服务，如 Cloud Build（云构建）服务、Analytics（分析）服务和 Ads（广告）服务。还可以从 Account 下拉菜单中访问 Unity 账户。此外，还有两个下拉菜单，即 Layers 和 Layout：使用 Layers 下拉菜单可以控制在 Scene 视图中显示哪些对象；使用 Layout 下拉菜单可以更改 Unity 编辑器的布局或创建 Unity 编辑器的新布局。

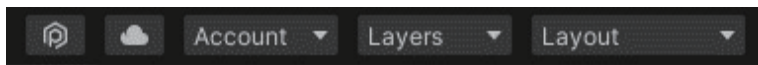


图 1.24 工具栏右侧的工具集

2. Hierarchy 窗口

Hierarchy 窗口如图 1.25 所示。可以看到，Unity 编辑器中的 Hierarchy 窗口会显示场景中的所有内容，如 Main Camera（主照相机）、Directional Light（方向光）和 Cube（立方体），这些被称为 Game Object（游戏对象）。还可以在 Hierarchy 窗口中组织游戏世界中的所有对象。

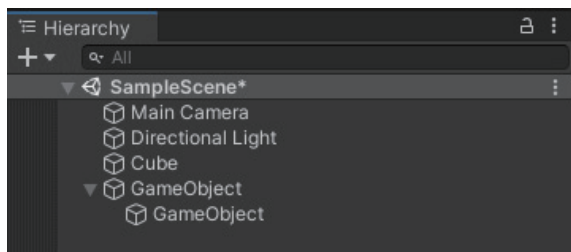


图 1.25 Hierarchy 窗口

在场景中创建游戏对象非常容易。右击 Hierarchy 窗口，在弹出的菜单中选择要创建的对象，如图 1.26 所示。

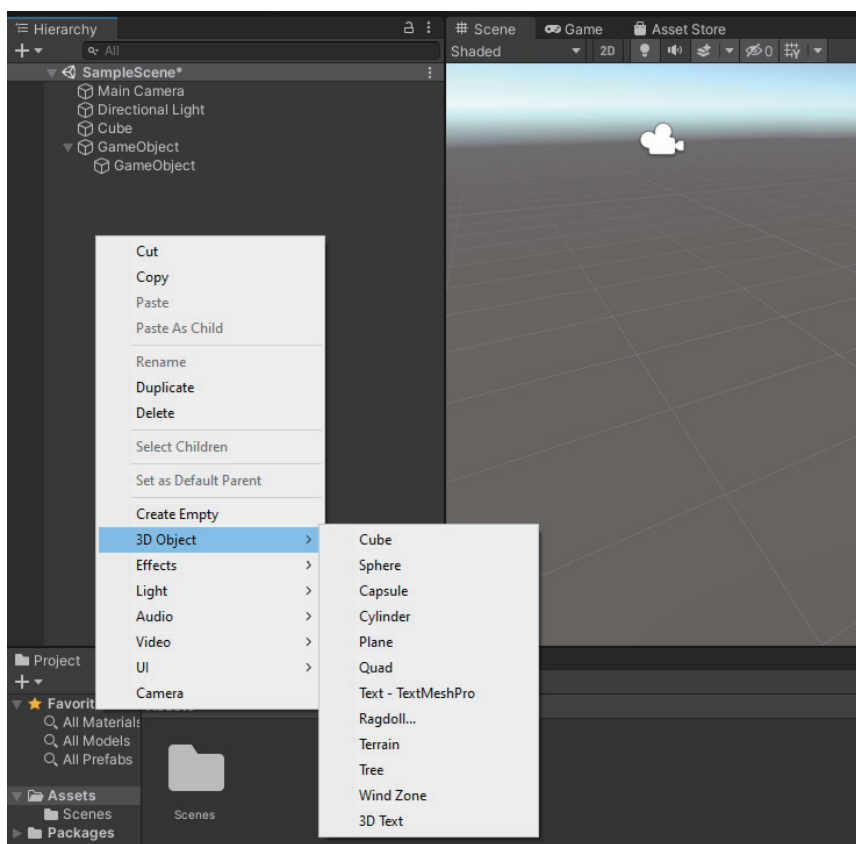


图 1.26 创建游戏对象

值得注意的是，Unity 使用父子层级结构来组织游戏对象，如图 1.27 所示。因此，可以将一个对象创建为另一个对象的子对象。如果想创建一个游戏对象作为另一个游戏对象的子对象，只需要先选择父游戏对象，然后右击它创建子游戏对象。

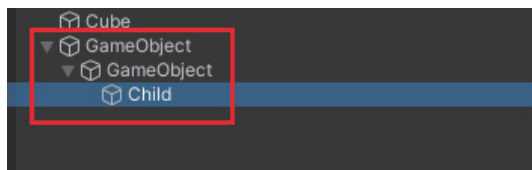


图 1.27 父子层级结构

创建父子层级结构的另一种方法是，在 Hierarchy 窗口中直接将现有的游戏对象拖到父游戏对象上。如图 1.28 所示，将名为 Cube 的游戏对象拖到名为 Child 的游戏对象上，以创建父子层级结构。

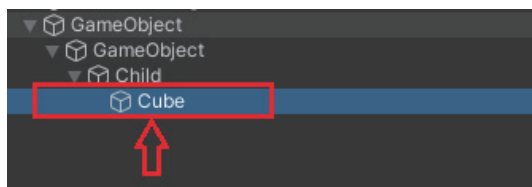


图 1.28 创建父子层级结构

Hierarchy 窗口允许隐藏和显示 Scene 视图中的游戏对象，而不改变它们在 Game 视图或最终应用程序中的可见性，如图 1.29 所示。

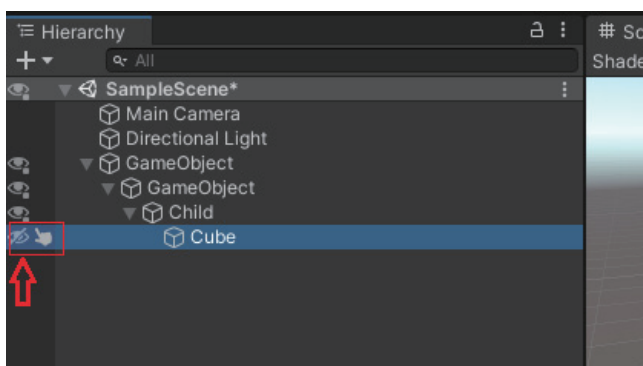


图 1.29 隐藏和显示游戏对象

3. Scene 视图和 Game 视图

默认的 Unity 编辑器布局的中心是 Scene 视图和 Game 视图，这是 Unity 编辑器中最重要的窗口。Scene 视图是正在创建的游戏世界的交互式视图，如图 1.30 所示。

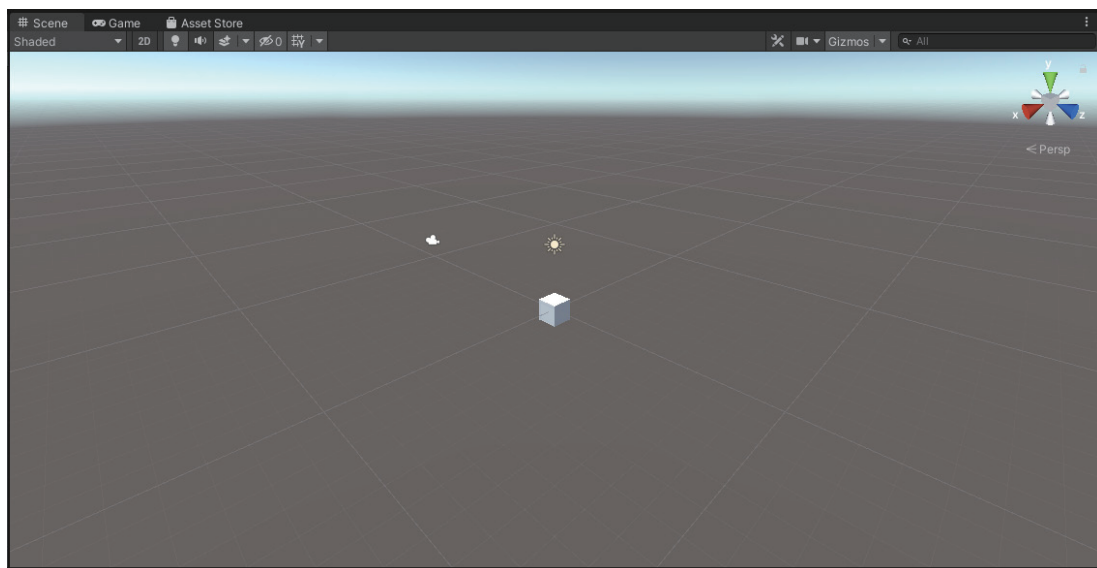


图 1.30 Scene 视图

可以使用 Scene 视图来操作游戏对象，并从不同的角度查看它们。在 Scene 视图中还有一些有用的工具，如右上角的场景小工具（Scene Gizmos tool），如图 1.31 所示。它将显示 Scene 视图相机的当前方向，并允许快速修改视图的角度和投影模式。

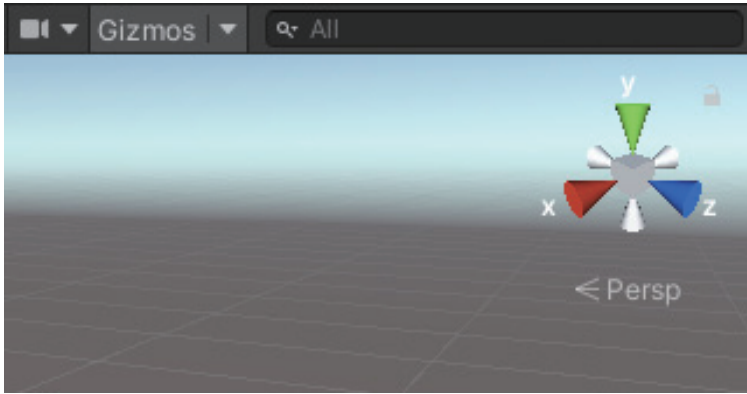


图 1.31 Scene Gizmos tool

如果要修改 Scene 视图相机的设置，可以单击 Gizmos 按钮旁边的 Camera 按钮，以打开 Scene Camera 窗口，如图 1.32 所示。在该窗口中，可以调整 Scene 视图相机的一些设置，如 Field of View（视野）和 Camera Speed（相机速度）。

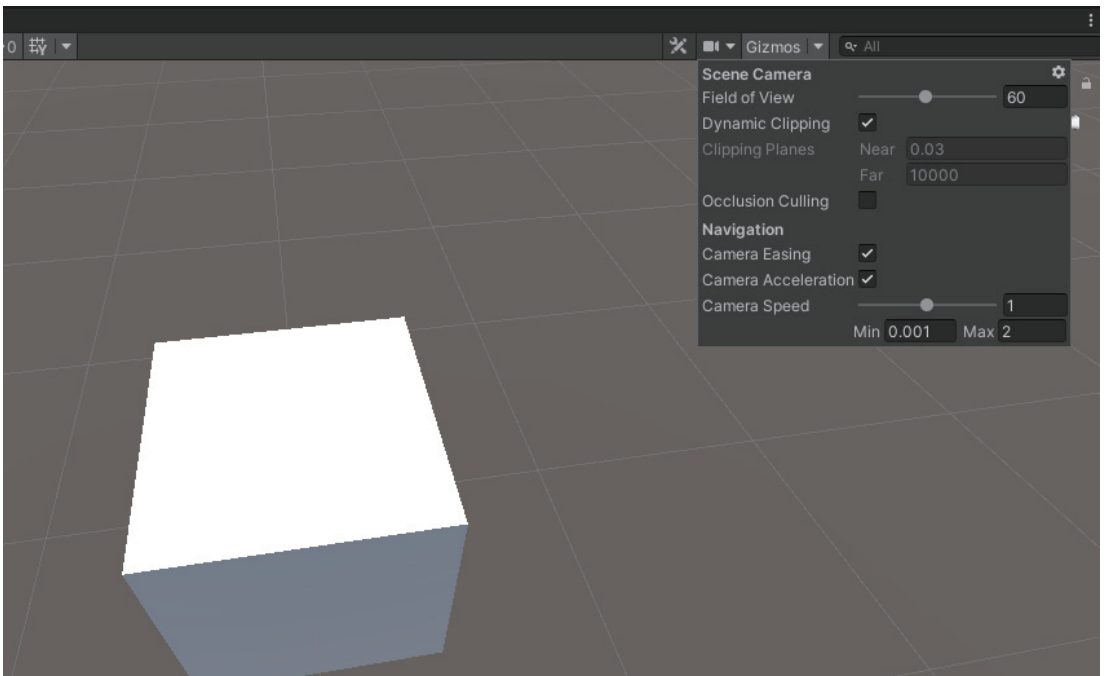


图 1.32 Scene Camera 设置

视图网格（visual grid）是一种可以在 Scene 视图中使用的有用工具。通过将游戏对象移动到最近的网格位置来对齐游戏对象。如图 1.33 所示，还可以将游戏对象移动到沿 X 轴、Y 轴或 Z 轴投影的网格。

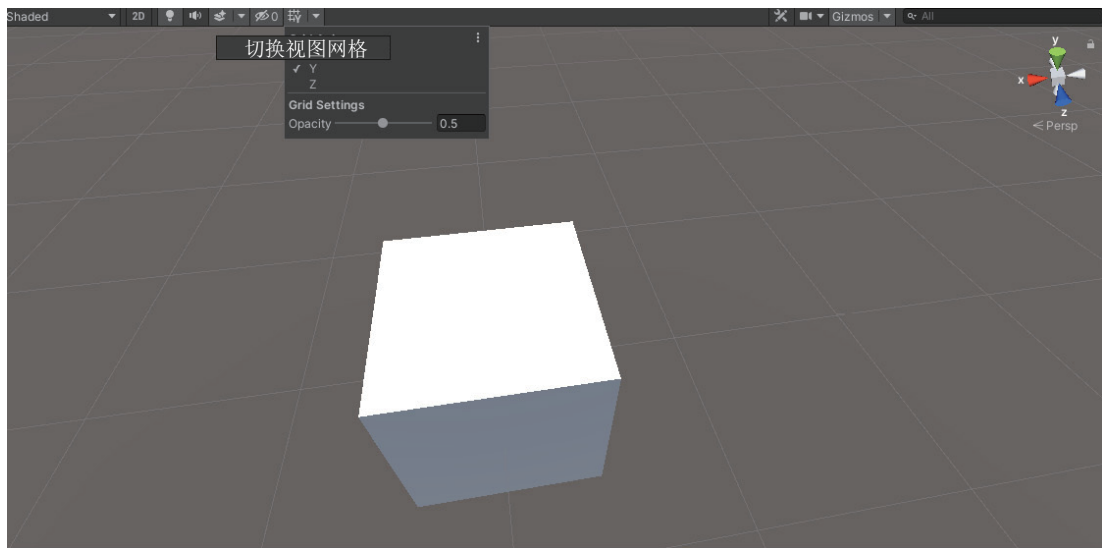


图 1.33 切换视图网格

Scene 视图中，最后一个有用的工具是在场景中使用的 Shading Mode（绘制模式），如图 1.34 所示。

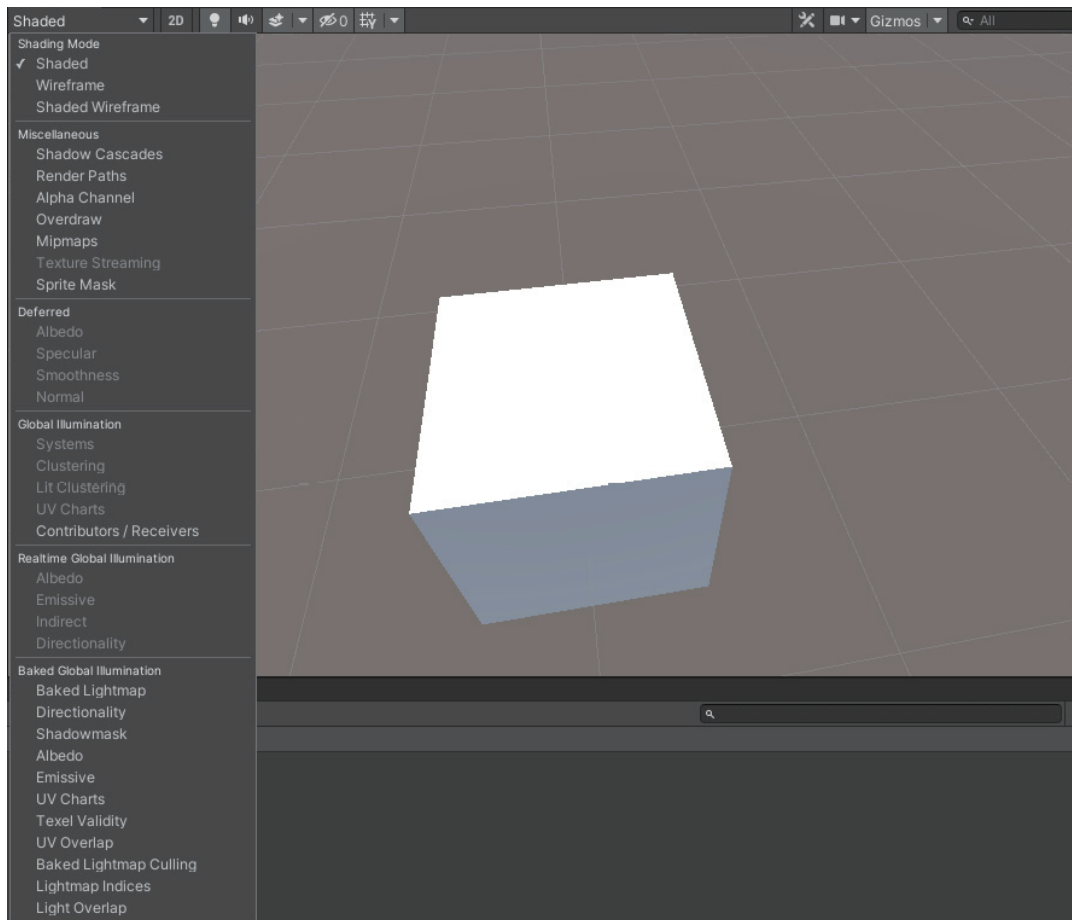


图 1.34 场景中的 Shading Mode

如果项目使用 Unity 的内置渲染管线，那么 Shading Mode 将会很有用，因为场景中不同的 Shading Mode 可以帮助理解和调试场景中的照明。

在默认布局中，Game 视图会出现在与 Scene 视图相同的区域中。单击 Game 标签从 Scene 视图切换到 Game 视图，如图 1.35 所示。

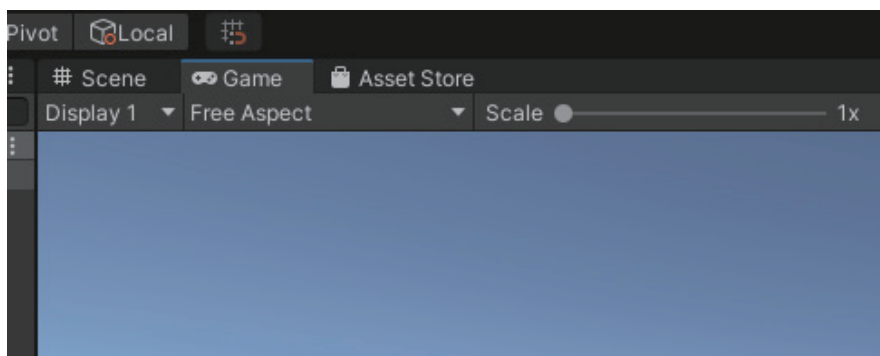


图 1.35 单击 Game 标签从 Scene 视图切换到 Game 视图

Game 视图将呈现最终发布的游戏，如图 1.36 所示。Game 视图的内容是通过相机呈现出来的，所以在 Game 视图中不能像在 Scene 视图中那样随意修改视角和投影模式，修改相机对象的设置才能实现此功能。

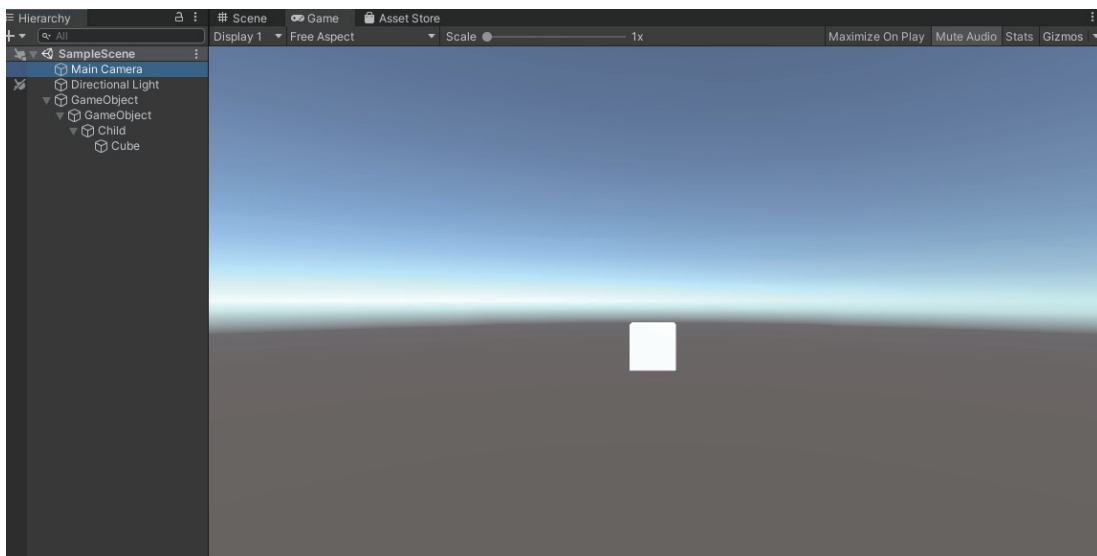



图 1.36 Game 视图

单击工具栏中的 Play 按钮可以直接在 Game 视图中运行游戏。需要注意的是，在 Play 模式下，所做的任何更改都是暂时的，当退出游戏时将被重置，因此不建议在 Play 模式下做大量更改。

接下来，介绍 Game 视图中的三个工具。

第一个工具是 Free Aspect。当为具有不同长宽比的屏幕开发游戏时，Free Aspect 下拉菜单非常有用，如图 1.37 所示。可以选择不同的值来测试游戏在这些屏幕上的外观，还可以单击菜单底部的  按钮来添加自定义值。

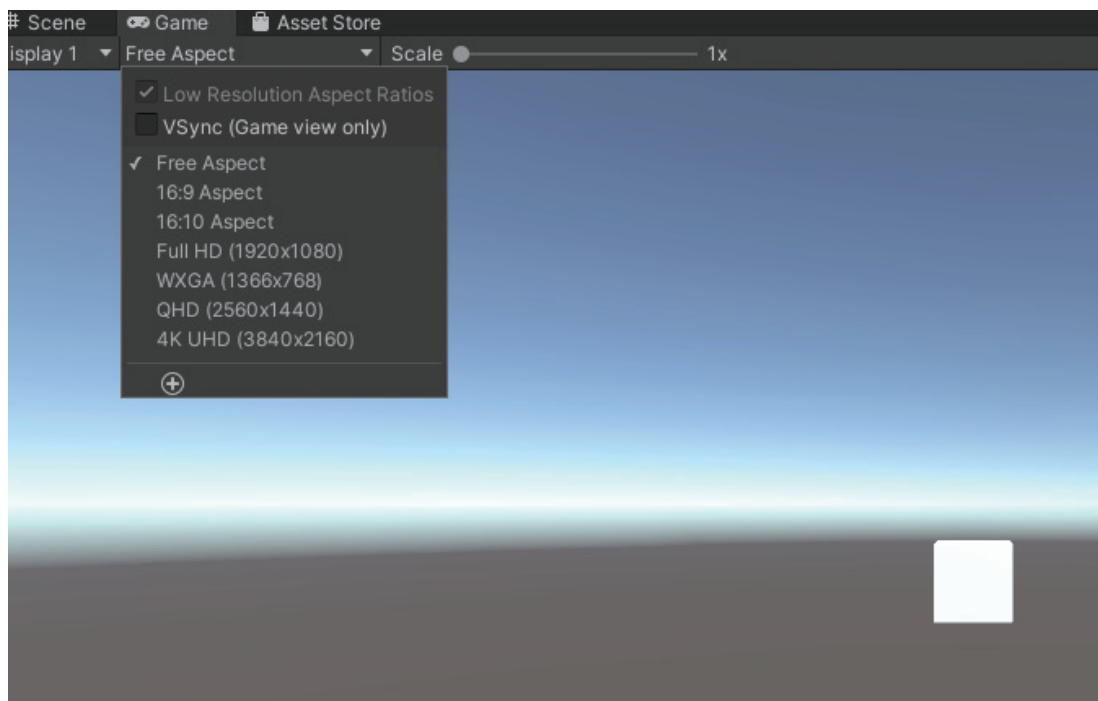


图 1.37 Free Aspect 下拉菜单

第二个工具为 **Maximize On Play**，如图 1.38 所示。当进入 **Play** 模式时，可以最大化 **Game** 视图以进行全屏预览。

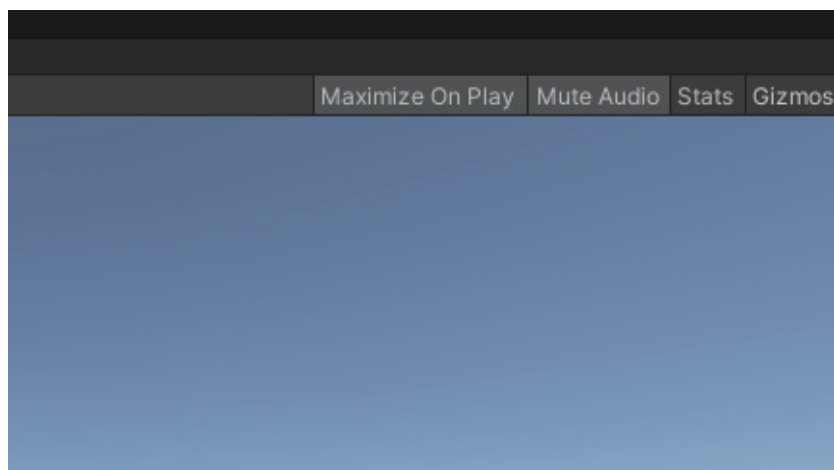


图 1.38 Maximize On Play 标签

第三个工具为 **Stats**。这个工具很有用，因为它可以显示关于游戏的音频（**Audio**）和图形（**Graphics**）的渲染统计信息，如图 1.39 所示。因此可以使用它来监控 **Play** 模式下的游戏性能。

在 **Scene** 视图中，可以查看和调整正在创建的游戏世界。在 **Game** 视图中，可以看到最终的游戏。所以，这个区域在编辑器中非常重要。

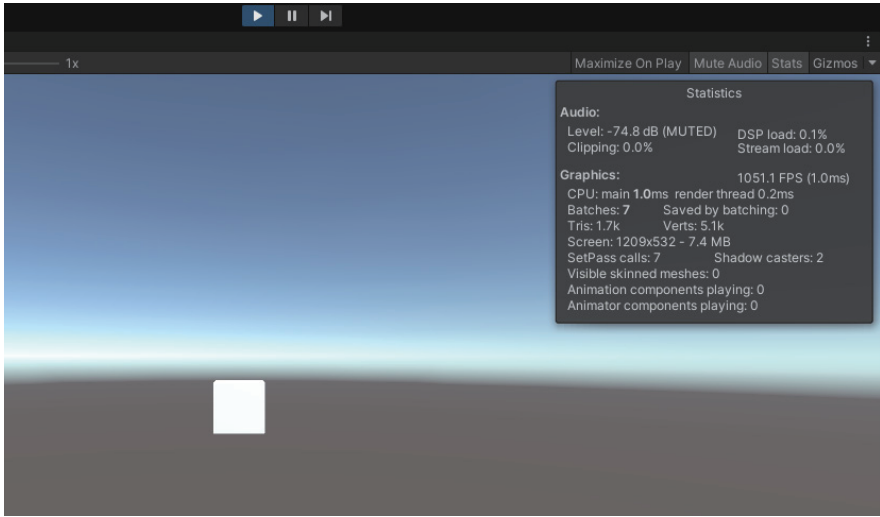


图 1.39 Stats 标签

4. Inspector 窗口

如果要修改游戏对象或游戏对象上的组件的属性，则需要使用 Inspector 窗口，如图 1.40 所示。

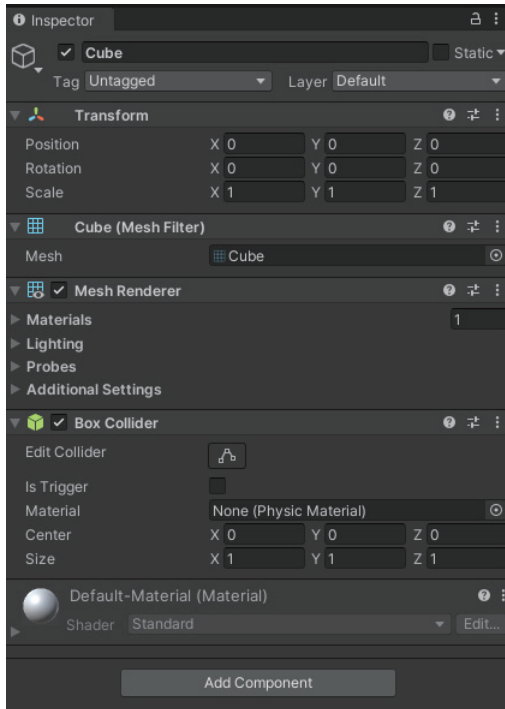


图 1.40 游戏对象的 Inspector 窗口

在 Scene 视图或 Hierarchy 窗口中选择游戏对象，然后在 Inspector 窗口中查看或修改这些组件及其属性。Inspector 窗口提供了一些有用的工具，用来帮助修改游戏对象。例如，如果要复制游戏对象的组件值，可以右击该组件，然后在弹出的菜单中选择 Copy Component 选项，如图 1.41 所示。

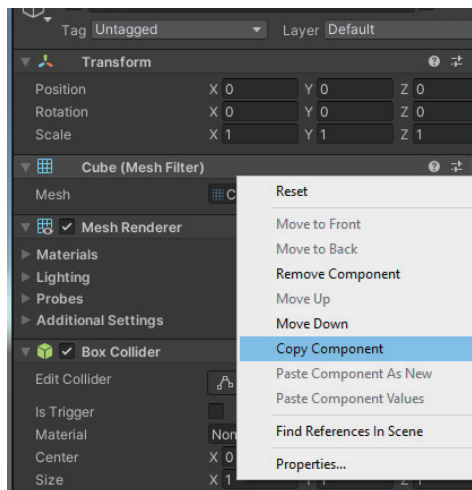


图 1.41 Copy Component 选项

该窗口不仅可以检查 Scene 视图中的游戏对象，还可以检查 Project 窗口中的数字资源。在 Project 窗口中选择一个数字资源，Inspector 窗口将显示与控制 Unity 在运行时如何导入和使用该资源相关的设置，如图 1.42 所示。

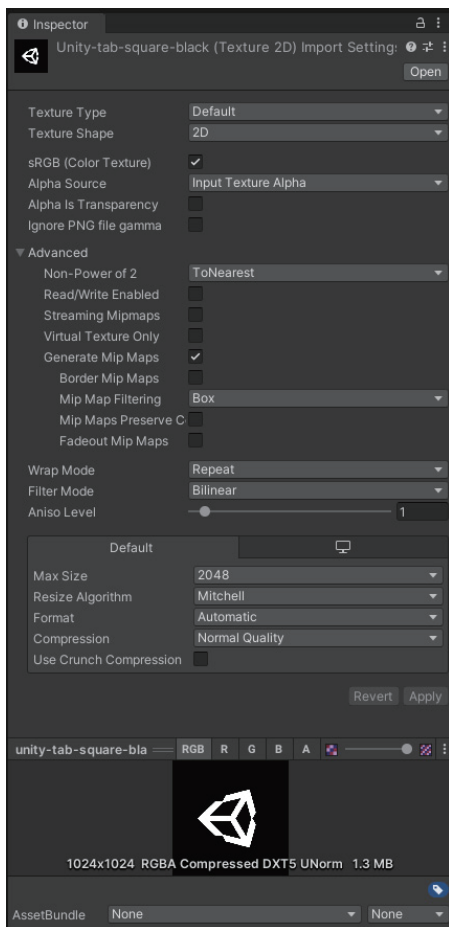


图 1.42 资源的 Inspector 窗口

5. Project 窗口

在 Project 窗口中，可以找到项目的所有数字资源。Project 窗口的工作方式就像一个文件浏览器，在文件夹中组织资源文件，如图 1.43 所示。

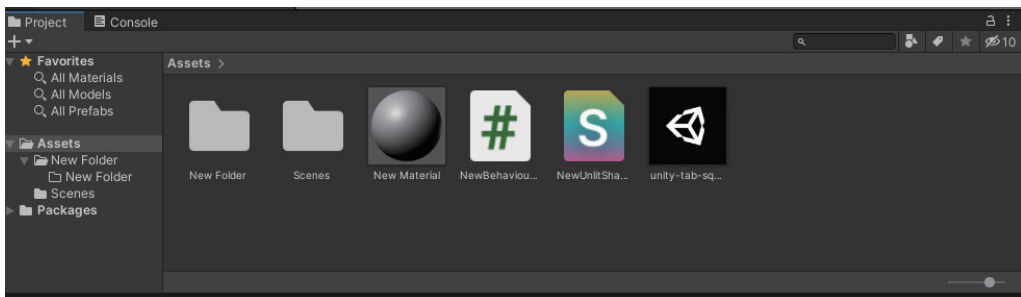


图 1.43 Project 窗口

Project 窗口是导航和查找游戏中资源的主要方式。它提供了两种方法搜索资源：按类型搜索（Search by Type）或按标签搜索（Search by Label）。如图 1.44 所示为按类型搜索资源。

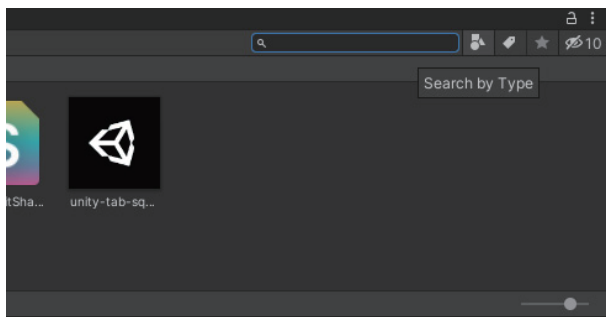


图 1.44 按类型搜索资源

导入外部数字资源或直接在 Unity 编辑器中创建资源非常容易。右击 Project 窗口，在弹出的菜单中可以选择创建资源或导入现有资源，如图 1.45 所示。

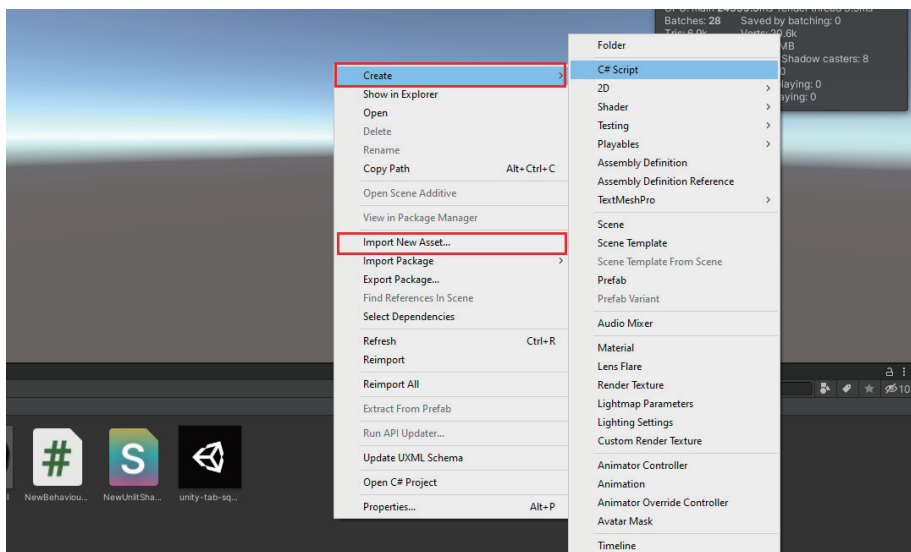


图 1.45 创建资源或导入现有资源

希望通过阅读本节能够对 Unity 编辑器有充分的理解。接下来，将介绍什么是游戏引擎，以及 Unity 作为游戏引擎提供了哪些重要功能。

1.3 在 Unity 中使用不同的功能

如今，Unity 不再只是一个游戏引擎，而是一个广泛应用于各种行业的创意工具。然而，Unity 仍然保留着其游戏引擎的根源，并且它仍然是最受欢迎的游戏引擎之一。要学习如何使用 Unity 来开发游戏，必须先了解 Unity 作为游戏引擎为游戏开发者提供了什么功能。

事实上，几乎所有的游戏引擎都为游戏开发者提供了与 Unity 相似的功能模块。所以，第一个问题是，什么是游戏引擎？

1.3.1 什么是游戏引擎

术语“游戏引擎”在游戏行业中被广泛使用，但并不是每个人都知道这个术语的含义，尤其是新的游戏开发者，因此需要解释什么是游戏引擎并介绍 Unity 中相应的功能。

游戏引擎不仅是计算机图形渲染器。当然，渲染是游戏引擎的一个重要功能，但创建游戏的过程要比渲染复杂得多。

作为一名游戏开发者，需要导入不同类型的数字资源，如 3D 模型、2D 纹理和音频，而这些数字资源大多不是在游戏引擎中创建的。因此，游戏引擎应该提供管理数字资源的功能。除了数字资源之外，还需要使用脚本来添加游戏逻辑，以指导这些资源执行正确的行为，如角色交互。

UI 是游戏的另一个组成部分，甚至有些游戏玩法是基于 UI 的。因此，一个好的游戏引擎应该提供一个易于使用且功能强大的 UI 工具包为游戏开发 UI。

可以使用其他软件来开发动画文件并将其导入游戏引擎中，但是为了在游戏中正确地播放和控制动画文件，游戏引擎需要提供动画功能。

物理系统在现代游戏中也是一种常见的功能，所以一个强大的游戏引擎应该提供物理系统，这样游戏开发者就不需要从头开始实现物理系统了。

在游戏中添加视频和音频可以让游戏更生动、有趣。特别是音频，合适的背景音乐和一些适当的音效可以让游戏感觉完全不同。即使它只是一个原型，背景音乐和音效也可以让游戏更完整、专业。因此，虽然很多人在谈论游戏引擎时经常忽略视频和音频的功能，但一个好的游戏引擎是不能没有视频和音频功能的。

如前所述，在游戏引擎中有许多功能供游戏开发者进行游戏开发。游戏引擎集成了创建一个游戏的所有方面，以创造完整的游戏用户体验。因此，在游戏开发中将处理不同的功能。例如，可能需要正确管理数字资源，并为游戏引擎创建适当的数字资源以优化性能，或者可能需要知道如何使用游戏引擎提供的脚本功能来为游戏开发逻辑。

作为最受欢迎的游戏引擎之一，Unity 提供了上述功能。下面将介绍 Unity 中的这些功能。

1.3.2 Unity 的功能

和其他优秀的游戏引擎一样，Unity 也为游戏开发者提供了许多功能，下面将进行介绍。

1. 图形

可以使用 Unity 的图形功能在各种平台上创建美丽的优化的图形。

渲染管线执行一系列渲染屏幕上场景内容的操作。Unity 中有以下 3 种渲染管线。

- (1) 内置渲染管线，它是 Unity 中的默认渲染管线，无法修改。
- (2) 通用渲染管线（URP），它允许开发人员为不同的平台定制和创建优化的图形。
- (3) 高清渲染管线（HDRP），它专注于高端平台上的尖端、高保真图形，如图 1.46 所示。



图 1.46 Unity HDRP 模板场景

此外，可以通过在 Unity 中使用可编程渲染管线 API 来创建自己的渲染管线，第 8 章将进行详细介绍。

2. 脚本

脚本是 Unity 的另一个基本特性，需要通过脚本来实现游戏逻辑。

Unity 引擎内部使用本地 C/C++ 构建，但是它提供了 C# 脚本 API，所以不必学习 C/C++ 来创建游戏。

3. UI（用户界面）

UI 对于游戏来说非常重要，Unity 为游戏开发者提供了以下 3 种不同的 UI 解决方案。

- (1) IMGUI（即时模式图形用户界面）。
- (2) Unity UI（uGUI）包。
- (3) UI 工具包。

IMGUI 是 Unity 中一个相对较老的 UI 解决方案，不建议用于构建运行时 UI。UI 工具包是最新的 UI 解决方案。然而它仍然缺少一些 uGUI 包和 IMGUI 所具有的特性。uGUI

软件包是 Unity 中成熟的 UI 解决方案，在游戏行业中被广泛使用，第 3 章中将详细介绍 uGUI 包。

4. 动画

动画可以使游戏更加生动。Unity 提供了一个称为 Mecanim 的强大动画系统，该系统允许重新定位动画，在运行时控制动画权重，并在动画播放中调用事件。在第 4 章中将介绍 Unity 的动画系统。

5. 物理

物理模拟是某些类型的游戏中不可或缺的功能，有些游戏玩法甚至完全基于物理模拟。Unity 中有不同的物理系统实现，可以根据游戏需要来选择，在第 5 章中将介绍 Unity 的物理系统实现。

6. 视频和音频

好的背景音乐、音效和视频可以让游戏脱颖而出，这是不可忽视的功能。Unity 提供视频和音频功能，允许游戏在不同的平台上播放视频，并支持实时混音和全 3D 空间音效。在第 6 章中将进一步讨论视频和音频。

7. 资源

可以将数字资源文件导入 Unity 编辑器中，如 3D 模型和 2D 纹理。Unity 提供了一个资源导入管线来处理这些导入的资源。还可以自定义导入设置，以控制 Unity 在运行时导入和使用资源的方式。在第 10 章中将介绍序列化系统和资源管理。

至此已经简要介绍了游戏引擎需要提供的功能，以及 Unity 提供的功能。接下来，将介绍 Unity 中的 .NET/C# 和脚本。

1.4 Unity 中的 .NET/C# 和脚本

Unity 是一个用 C/C++ 编写的游戏引擎，但为了让游戏开发者更容易开发游戏，Unity 提供了 C# 作为脚本编程语言，以在 Unity 中编写游戏逻辑。这是因为与 C/C++ 相比，C# 更容易学习。此外，它是一种“托管语言”，这意味着它将自动管理内存——分配释放内存、覆盖内存泄漏等。

本节将介绍 Unity 中的 .NET/C# 和脚本。

1.4.1 Unity 中的 .NET 配置文件

Unity 游戏引擎使用了 Mono。Mono 是一种基于 C# 和 .NET 进行脚本开发的开源的 ECMA CLI，可以在 <https://github.com/Unity-Technologies/mono/tree/unitymaster-new-unitychanges> 上跟踪 Unity 的 Mono 分支的发展。

Unity 提供了不同的 .NET 配置文件。Unity 2018 之前的版本在 Player 设置面板（选择 Editor → Project Settings → Player → Other Settings 选项）中提供了两个 API 兼容性级别，即 .NET 2.0 Subset 和 .NET 2.0。如果使用的是早期版本的 Unity，那么强烈建议进行 Unity 版本的更新。Unity 中 .NET 2.0 Subset 和 .NET 2.0 的配置文件与微软的 .NET 2.0 配置文件紧密一致。

如果使用的是 Unity 2019 或更高版本，会发现另外两个 .NET 配置文件，它们是 .NET Standard 2.0 和 .NET 4.x，如图 1.47 所示。

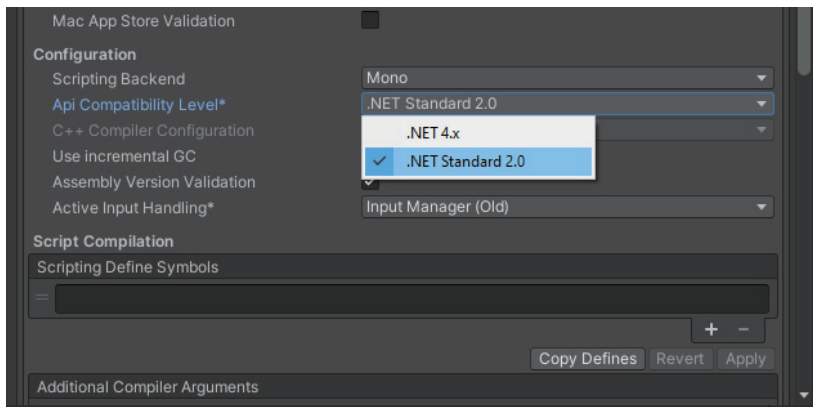


图 1.47 API 兼容性级别设置

注意

.NET Standard 2.0 配置文件的名称可能会有一点误导性，因为它与 Unity 早期版本中的 .NET 2.0 和 .NET 2.0 Subset 配置文件并没有关系。

.NET Standard 是一种正式的规范，所有 .NET 平台上的 .NET API 都必须实现该标准。这些 .NET 平台包括 .NET Framework、.NET Core、Xamarin 和 Mono。可以在 <https://github.com/dotnet/standard> 上找到 .NET Standard 存储库。

Unity 的 .NET 4.x 配置文件与 .NET Framework 的 .NET 4 系列（.NET 4.5、.NET 4.6、.NET 4.7 等）的配置文件相匹配。

因此，建议在 Unity 中使用 .NET Standard 2.0 配置文件，如果仅从兼容性上考虑，就应该选择 .NET 4.x 配置文件。

1.4.2 Unity 的脚本后端

除了 .NET 配置文件之外，Unity 还提供了两种不同的脚本后端，分别是 Mono 和 IL2CPP（Intermediate Language To C++），如图 1.48 所示。

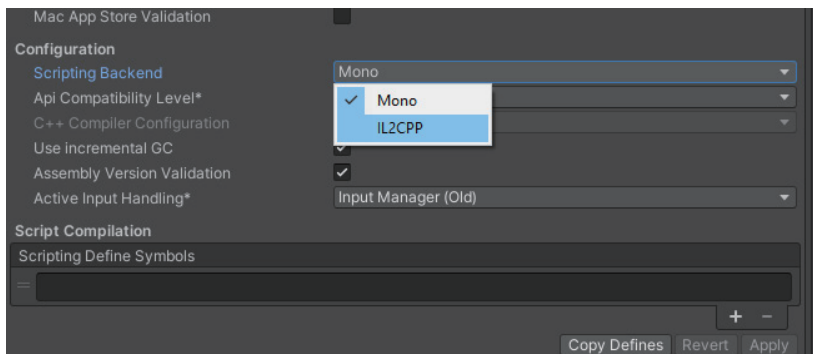


图 1.48 两种脚本后端

选择 Edit → Project Settings → Player → Other Settings 选项，可以在 Player 设置面板

中更改项目的脚本后端。

这两种脚本后端的关键区别在于它们编译 Unity 脚本 API 代码 (C# 代码) 的方式不同。

(1) Mono 脚本后端使用即时 (JIT) 编译, 并在运行时按需编译代码。它将 Unity 脚本 API 代码编译为常规 .NET DLL。Unity 使用标准的 Mono 运行时的实现来编写本机支持的 C# 脚本。

(2) IL2CPP 脚本后端使用提前 (AOT) 编译, 在运行之前编译整个应用程序。它不仅将 Unity 脚本 API 代码编译成 .NET DLL, 还将所有托管程序集转换为标准的 C++ 代码。此外, IL2CPP 运行时是由 Unity 开发的, 它是 Mono 运行时的替代品。

如图 1.49 所示, IL2CPP 不仅将 C# 代码编译为托管程序集, 还进一步将托管程序集转换为 C++ 代码, 并将 C++ 代码编译为本机二进制代码。

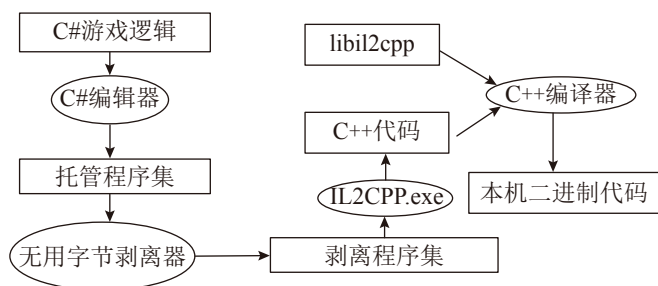


图 1.49 IL2CPP 脚本后端

显然, 与 Mono 相比, IL2CPP 需要更多的时间来编译代码, 那么为什么仍然需要 IL2CPP 呢?

首先, IL2CPP 使用 AOT 编译, 编译需要更长的时间, 但是当为特定平台发布游戏时, 二进制文件是完全指定的, 这意味着与 Mono 相比, 其代码生成得到了很大的提高。

其次, 在为 iOS 和 WebGL 构建项目时, IL2CPP 是唯一可用的脚本后端。除了 iOS 和 WebGL, Unity 还在 Unity 2018.2 中增加了对 Android 64 位系统的支持, 以遵守 Google Play 商店的政策, 该政策要求从 2019 年 8 月 1 日开始, 发布在 Google Play 上的应用程序需要支持 64 位架构。

如图 1.50 所示, Mono 脚本后端不支持 Android 64 位 ARM 架构。在这种情况下, 必须选择 IL2CPP 脚本后端。

无论使用 IL2CPP 进行更好的代码生成, 还是使用一些特定的平台或架构, 花费更多的编译时间仍然是 IL2CPP 的一个缺点。那么, 如何优化 IL2CPP 的编译时间呢? 在此给出以下 3 点建议。

(1) 不要删除以前的 build 文件夹, 在与该文件夹相同的位置使用 IL2CPP 脚本后端构建项目。这是因为可以使用增量构建, 这意味着 C++ 编译器只重新编译自上次构建后发生更改的文件。

(2) 将项目和目标构建文件夹存储在固态硬盘 (Solid-State Drive, SSD) 上。这是因为当选择 IL2CPP 时, 编译程序会先将该 IL 代码转换为 C++, 再对 C++ 进行编译, 这涉及大量的读 / 写操作。更快的存储设备将加快这一过程。

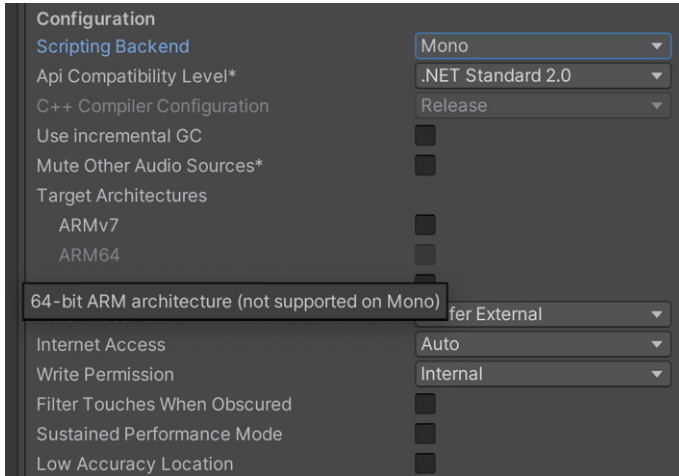


图 1.50 Mono 脚本后端不支持 Android 64 位 ARM 架构

(3) 在构建项目之前，禁用反恶意软件。当然，这取决于使用的安全策略。

至此，应该对 Unity 的脚本系统有了一个大致的了解，如 Unity 中的 .NET 配置文件、两个脚本后端及针对 IL2CPP 的一些优化技巧。

1.5 节将学习如何设置开发环境，并使用 Visual Studio 在 Unity 中开发游戏。

1.5 使用 Visual Studio 构建 Unity 游戏

在开始编写代码之前，选择合适的开发工具很重要。微软的 Visual Studio 不仅是一个广泛使用的 IDE，还是在 Windows 或 macOS 上安装 Unity 时默认安装的开发环境，如图 1.51 所示。

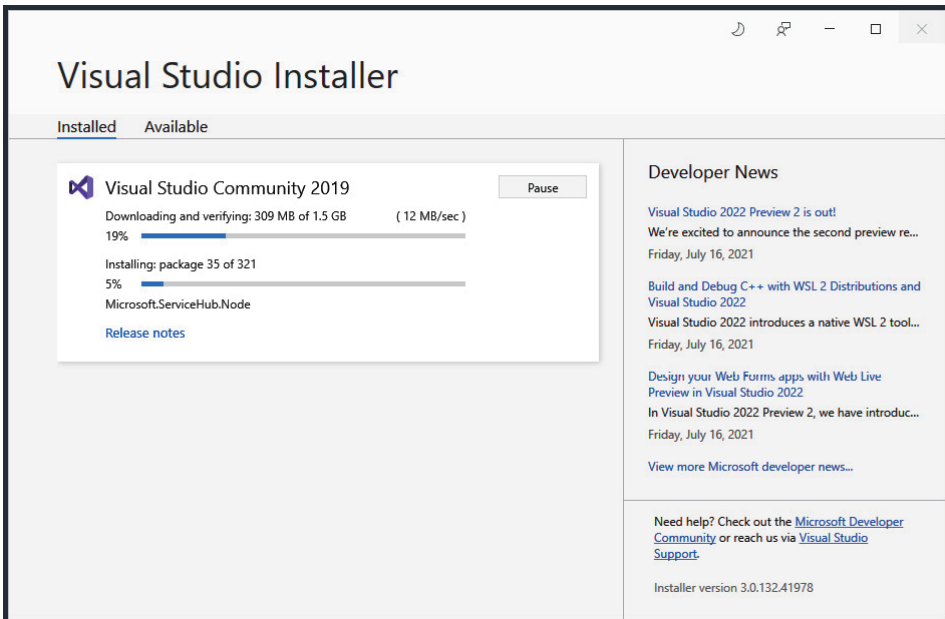


图 1.51 Visual Studio Installer

在安装 Visual Studio 的同时，还将安装 Visual Studio Tools for Unity。它是一个免费的扩展，为在 Unity 中编写和调试 C# 提供支持。

如果没有通过 Unity Hub 安装 Visual Studio，请确保安装了此扩展。可以在 Visual Studio Installer 中进行检查，如图 1.52 所示。

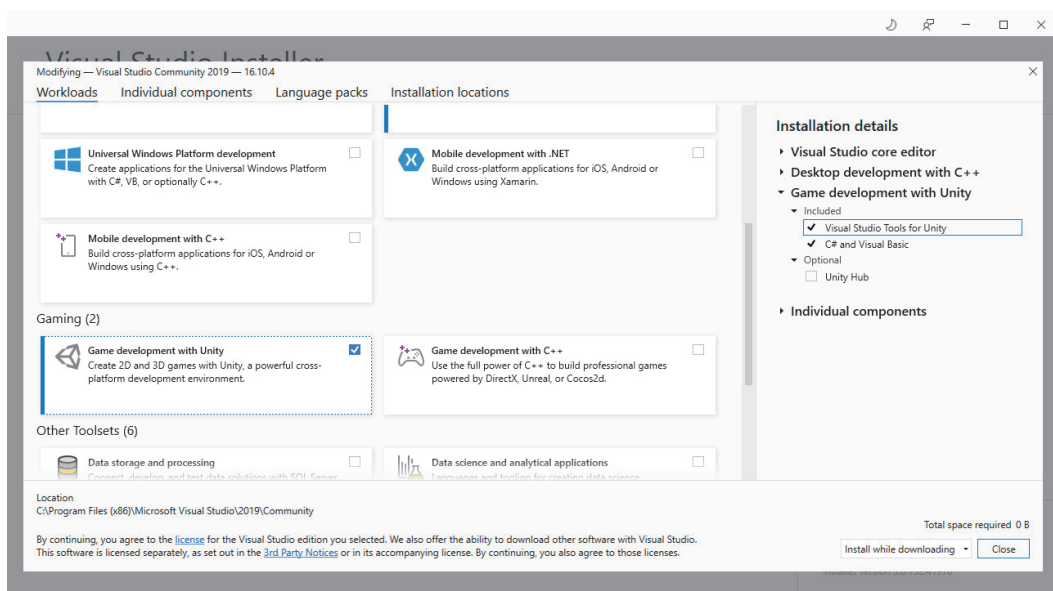


图 1.52 安装 Visual Studio Tools for Unity

安装 Unity 编辑器和 Visual Studio Community 2019 后，可以在 Unity 编辑器的 Preferences 窗口中检查 External Script Editor 设置，如图 1.53 所示。

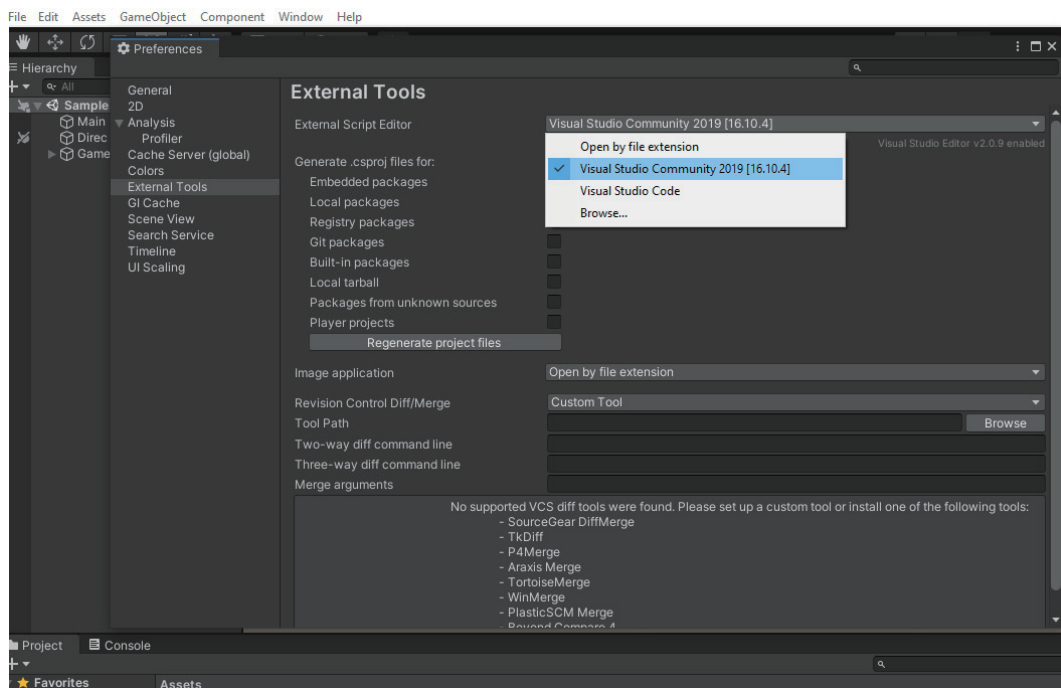


图 1.53 External Script Editor 设置

还可以通过修改此设置来选择其他脚本编辑器，如 Visual Studio Code 和 JetBrains Rider。

在 Unity 编辑器中新建一个名为 NewBehaviourScript.cs 的 C# 脚本文件，双击以在 Visual Studio 中打开它。如图 1.54 所示，默认情况下，脚本文件中有两个内置方法，即 Start 和 Update。Visual Studio 支持 Unity API 的智能感知，可以提高代码编写的速度。

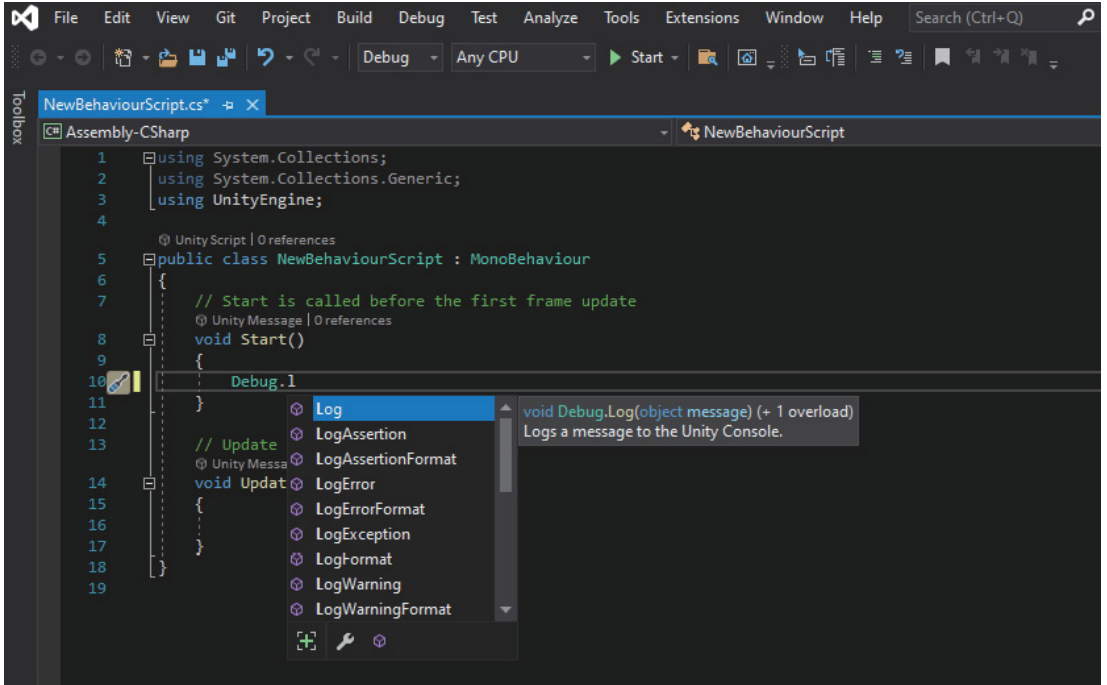


图 1.54 对 Unity API 的智能感知

在 Visual Studio 中调试代码非常容易。首先在 Start 方法中设置一个断点，如图 1.55 所示。然后在 Visual Studio 中单击 Attach to Unity 按钮，如图 1.56 所示。

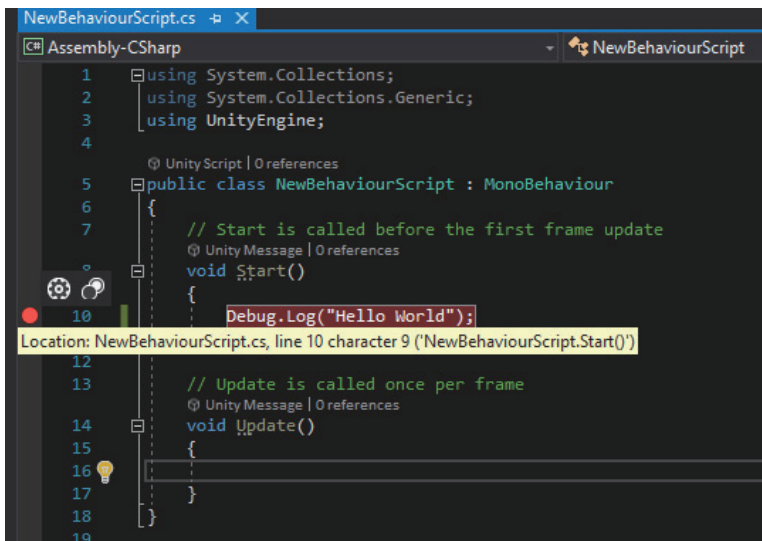


图 1.55 调试代码

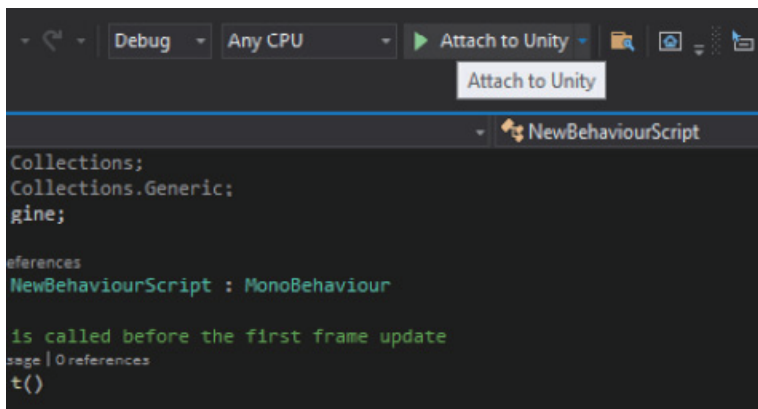


图 1.56 单击 Attach to Unity 按钮

为了运行这段代码，将该脚本附加到场景中的一个游戏对象上，并单击 Unity 编辑器中的 Play 按钮，在 Game 视图中运行游戏。调试器将在断点处停止，以便查看游戏的当前状态，如图 1.57 所示。

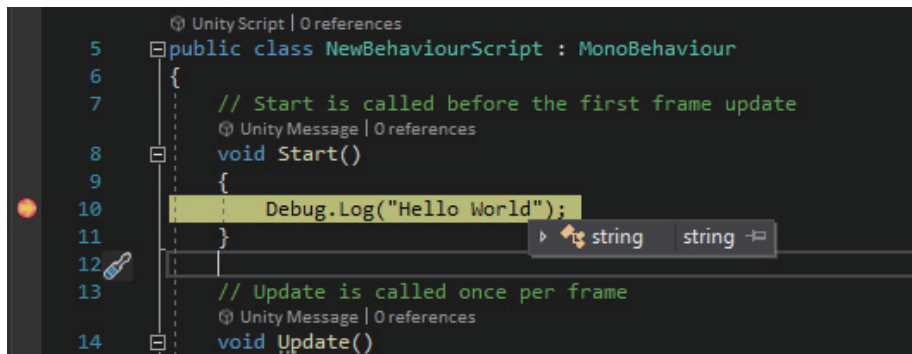


图 1.57 调试器在断点处停止

1.6 本章小结

本章首先选择满足需要的 Unity 版本和订阅计划。然后，学习如何使用 Unity Hub 安装和管理 Unity 编辑器，并探索了 Unity 编辑器的五个重要区域——工具栏、Hierarchy 窗口、Scene 视图和 Game 视图、Inspector 窗口及 Project 窗口。接着，讨论了什么是游戏引擎，并探索了 Unity 为开发者开发游戏提供的不同功能。还介绍了 Unity 中的 .NET 配置文件和 Unity 提供的脚本后端：Mono 脚本后端和 IL2CPP 脚本后端。最后，演示了如何为 Unity 编辑器设置 Visual Studio 来编写代码。