

快速入门



本章要点：

- 了解 C# 的特点
- 了解 .NET 框架
- 掌握 Visual Studio 2017 的安装与设置

C# 是一门现代的、通用的、面向对象的编程语言，它是由微软公司开发的，由 Ecma 和 ISO 核准认可。它专为公共语言基础结构 (CLI) 设计。CLI 由可执行代码和运行时环境组成，允许在不同的计算机平台和体系结构上使用各种高级语言。C# 之所以能够成为一种广泛应用的专业语言，主要有以下原因：

- 它是一种现代的、通用的结构化编程语言；
- 具有面向对象、面向组件、容易学习的特点；
- 可以产生高效率的程序；
- 可以在多种计算机平台上编译；
- 它是 .NET 框架的一部分；
- 它是 Unity 引擎开发默认支持的开发语言。

虽然 C# 的构架十分接近于传统的高级语言 C 和 C++，是一门面向对象的编程语言，但是它与 Java 非常相似，有许多强大的编程功能，再加上国内外游戏市场的火爆，尤其是手游（手游市场大部分使用 Unity 作为开发引擎，而 Unity 引擎所使用的开发语言为 C#），因此，C# 得到了广大程序员的青睐。

1.1 开发环境安装与设置

C# 是 .NET 框架的一部分，用于编写 .NET 应用程序。因此，在讨论运行 C# 程序的可用工具之前，先了解一下 C# 与 .NET 框架之间的关系。

.NET 框架是一个创新的平台，可以编写出以下类型的应用程序：

- Windows 应用程序；
- Web 应用程序；
- Web 服务。

.NET 框架应用程序是多平台的应用程序。框架的设计方式使它适用于 C#、C++、Visual Basic、JScript、COBOL 等各种语言。所有这些语言可以访问框架,彼此之间也可以互相交互。它是由一个巨大的代码库组成的,用于 C# 等客户端语言。下面列出一些 .NET 框架的组件。

- 公共语言运行库(common language runtime,CLR)。
- .NET 框架类库(.NET framework class library)。
- 公共语言规范(common language specification)。
- 通用类型系统(common type system)。
- 元数据(metadata)和组件(assemblies)。
- Windows 窗体(Windows forms)。
- ASP.NET 和 ASP.NET AJAX。
- ADO.NET。
- Windows 工作流基础(Windows workflow foundation,WF)。
- Windows 显示基础(Windows presentation foundation)。
- Windows 通信基础(Windows communication foundation,WCF)。
- LINQ。

由于本书讲解的 C# 语言主要是针对 Unity 部分开发的,因此使用微软公司的 Visual Studio 2017 社区版(免费)作为默认的开发环境。下面介绍如何下载、安装、配置 Visual Studio。

从微软公司官网下载 Visual Studio Installer(安装器),双击打开后,选择安装 Visual Studio 社区版,将弹出如图 1-1 和图 1-2 所示的界面,选中 .NET desktop development 和 Game development with Unity 两个选项,“单击”右下角的 Install while downloading 按

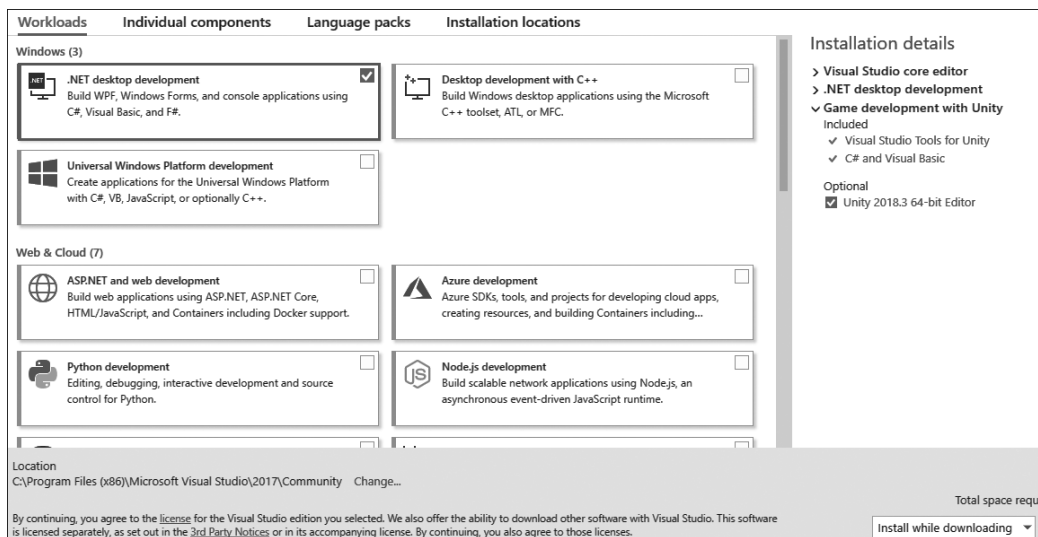


图 1-1

钮,程序将自动联网从服务器下载所需安装包,并进行安装与配置,整个过程非常简单。今后如果想添加其他模块,也可以单击图 1-3 中的 Modify 按钮,重新进入这个界面(比如选中 Game development with C++,针对 UnrealEngine4)加装模块。启动已经安装完成的 Visual Studio 2017,注册一个微软公司的账号(如果已有,可直接用账号登录),然后登录,获取社区版的授权(免费),至此,完成 IDE 的安装配置。

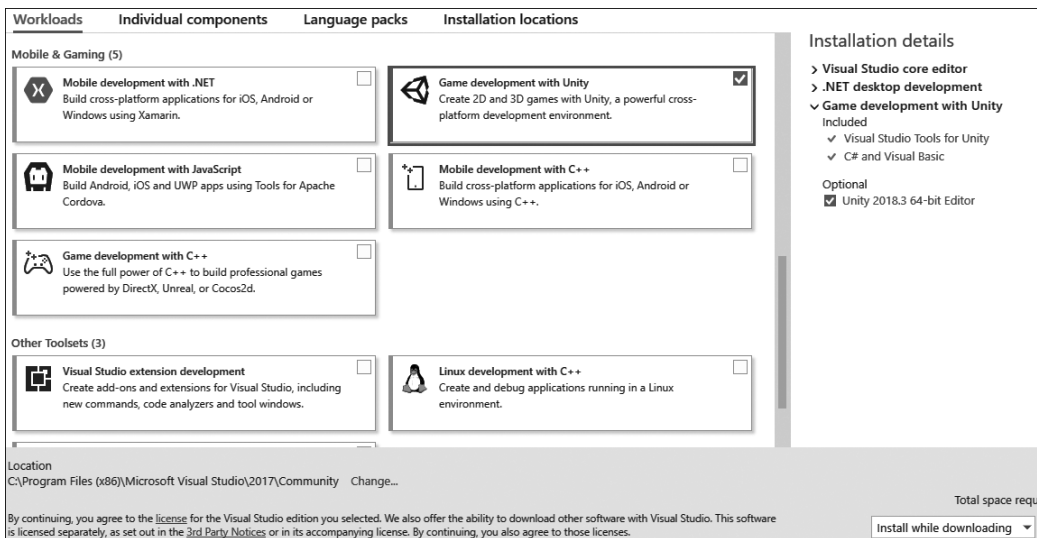


图 1-2

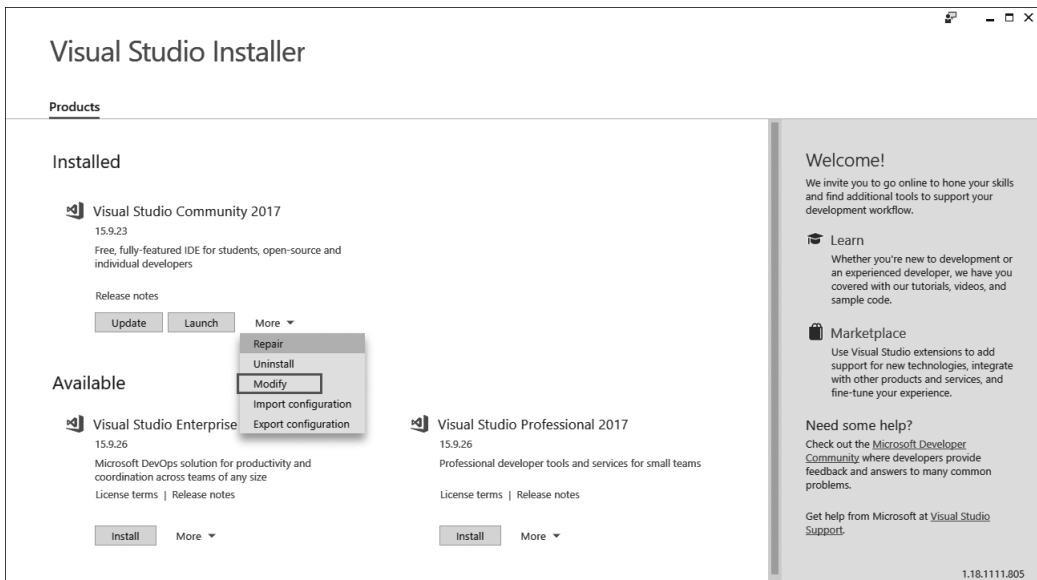


图 1-3

1.2 创建第一段 C# 代码

(1) 双击打开 Visual Studio 2017,在菜单栏上选择 File→New→Project 命令,打开如图 1-4 所示的窗口,从模板中选择 Visual C#,再选择 Console App(.NET Framework)选项,并在 Name 栏里填写项目名称“1.创建第一段代码”,单击 OK 按钮完成新建操作。

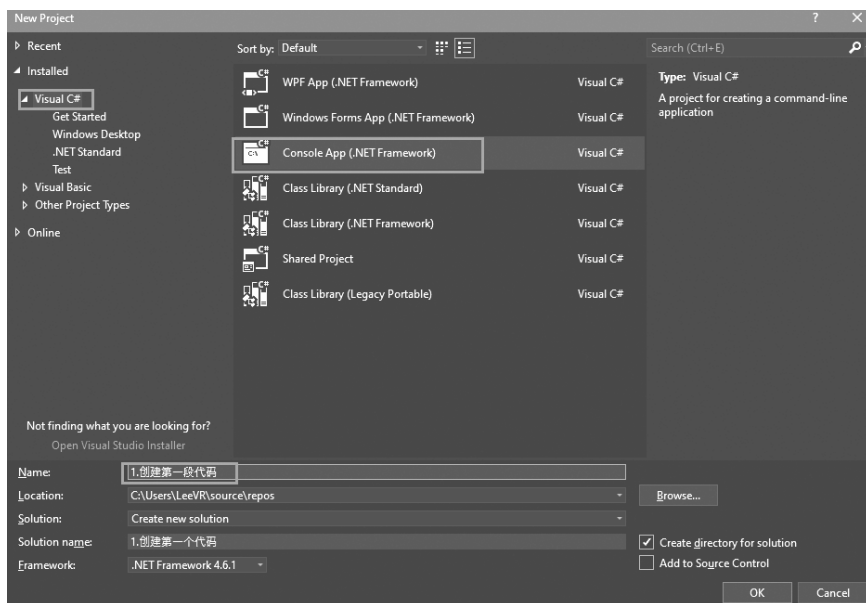


图 1-4

(2) 在 Main 方法里编写代码:“Console.WriteLine(“Hello, Unity with C#!”);”,如图 1-5 所示。按 Ctrl+F5 组合键,得到如图 1-6 所示的运行结果,由此,就得到了第一段代码的运行结果。

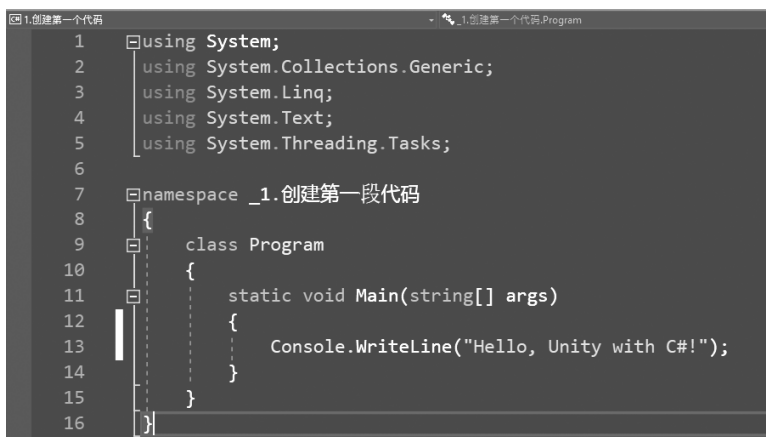


图 1-5

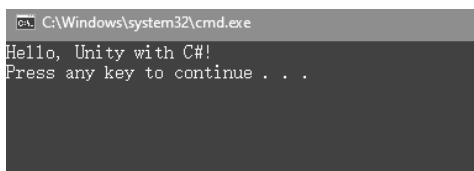


图 1-6

1.3 课后练习

请使用 Visual Studio 2017 新建一个项目,并命名为“我的第一个程序”,然后向控制台输出内容:“这是我的第一个 C# 控制台程序!”