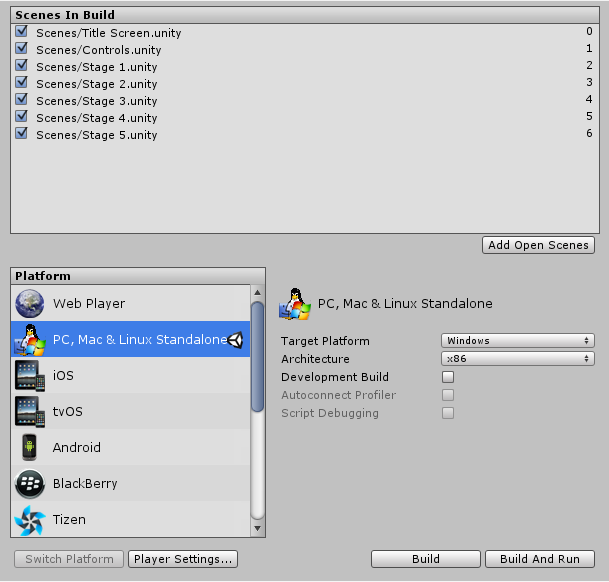
# 打包与发布

在游戏开发进行到一定阶段时，我们会想要看看游戏在编辑器外独立运行会是什么效果。本节将指导读者进行实际的发布操作，包含多种平台的发布方法。

## 打包设置

主菜单的File > Build Settings选项可以打开发布设置窗口，其中有很多发布选项。如图：



打包界面

## 发布设置菜单

Scenes In Build一栏是一个场景列表，默认是空的。如果场景列表为空时进行发布，那么只有当前正在编辑的场景会被发布出去且作为启动场景。很多时候我们只是想快速做一些测试工作，就会只发布当前编辑的场景。

正式发布的游戏往往包含多个场景，有两种方式添加场景，一是点击“Add Open Scenes”按钮，打开一个选择场景的窗口进行选择操作；二是直接将保存的场景文件从工程窗口拖拽到发布窗口中。

拖动场景到列表中以后，你会发现每个场景都被分配了一个数字，其中Scene 0就是游戏开始后第一个被自动加载的场景。当你需要加载一个新的场景时，可以在脚本中调用SceneManager.LoadScene方法。

如果你已经添加了多个场景文件并且希望重新排列它们的顺序，只需要在列表中简单地选中并拖拽它们即可，将一个场景拖拽到其它场景的下面或者下面就可以交换场景的顺序了。

要删除某个已经添加的场景，只要选中该场景并按下键盘上的Delete键即可。

“Platform”列表就是可供选择的要发布的平台，选中某个平台后点击“Build”按钮就可以发布针对这个平台的游戏包了，这时会弹出一个对话框选择文件名称和保存位置，之后就进入了打包发布的流程。整个操作非常简单，流程也是傻瓜化的。特别是发布PC、Mac等桌面系统的游戏包是非常简单的。

发布窗口中“Development Build（开发版本）”的选框用来开启性能分析器（Profiler）功能，同时也会打开相应的调试功能，方便在正式发布版本中调试问题或者分析性能。

## 发布为桌面程序

用Unity发布游戏为独立的Windows、Mac或Linux程序非常容易。如上节所说，只要添加场景并进行一些简单的设置，然后点击“Build”按钮即可。最终发布生成的程序因平台不同而有很大区别。在Windows下会生成可执行文件（后缀为.exe），同时有一个存放数据的文件夹（名为XXX\_Data）被生成，这个文件夹内包含着所有游戏需要用到的资源。对Mac系统来说，会生成一个app bundle，其中包含了所有必要的程序和资源，可以直接执行。

所以如果想将Mac版本的游戏发给其他人，只需要拷贝这个bundle即可。而传输Windows版本的游戏，就要同时提供发布的exe文件和资源包，另外注意还有发布时生成的动态链接库文件（例如UnityPlayer.dll）也是必要的。

## 发布时的内部流程

发布时，首先Unity会根据所选平台的模版建立一个空白的应用程序，然后遍历所有场景，依次打开这些场景，分析其中的资源并优化，然后将它们依次打包到应用程序中。其中，针对场景用到的所有资源，会将它们保存成一种特定的格式到独立的文件中。

任何具有“EditorOnly”标签的游戏物体，不会被打包到发布版中。这个标签表示这些物体只在开发过程中用到，一般是用来扩展编辑器功能，或是显示一些辅助信息的。

当加载一个新的场景或关卡的时候，前一个场景的所有物体都会被销毁。也有办法防止这一点，只需要在脚本中对不想销毁的物体使用DontDestroyOnLoad方法即可。这个方法通常用于在切换场景时保持音乐播放，或者让控制游戏流程的脚本保持状态而不会被重置。

|  |
| --- |
| // 例如不希望某个物体gameObjA在切换场景时被销毁  DontDestroyOnLoad(gameObjA); |

使用SceneManager.sceneLoaded注册事件通知，可以让物体在场景加载完毕后得到通知。

|  |
| --- |
| public class TestScene : MonoBehaviour {  void Start () {  SceneManager.sceneLoaded += OnSceneLoaded;  }  void OnSceneLoaded(Scene scene, LoadSceneMode mode) {  Debug.Log("场景加载完毕 " + scene.name + " 加载模式 " + mode);  }  } |

多场景管理的更多问题请参考关于多场景管理的章节。

## 发布为安卓应用程序

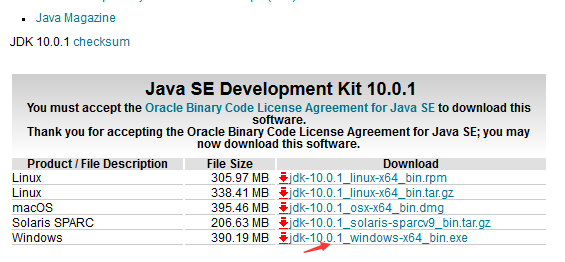
在将你的游戏发布成安卓应用程序之前，我们首先需要配置好打包需要的环境变量，以及相关开发工具包才能够进行打包。

### JDK概述

在下载安装之前，我们先了解一下什么是JDK，还有为什么打包会需要JDK。JDK(Java Development Kit)是 Java语言的软件开发工具包，主要用于移动设备、嵌入式设备上的java应用程序。由于安卓应用程序开发是以Java语言为基础的，所以要想我们的游戏能够发布到安卓平台上，就必须需要Java的支持。

### JDK的下载安装

由SUN公司开发的Java语言，后续由甲骨文公司—Oracle进行收购。所以我们现在需要在甲骨文公司的官网：https://www.oracle.com/cn/index.html。找到我们需要的当前最新的JDK，点击下载即可。如图：



JDK下载

选择系统对应的JDK包下载完成后，运行jdk-10.0.1\_windows-x64\_bin.exe程序，如图：



运行JDK安装程序

选择下一步后，进行安装组件的选择与安装目录的设置，如图：



设置JDK安装路径

设置好安装目录，点击下一步进行安装，然后安装jre，如图：

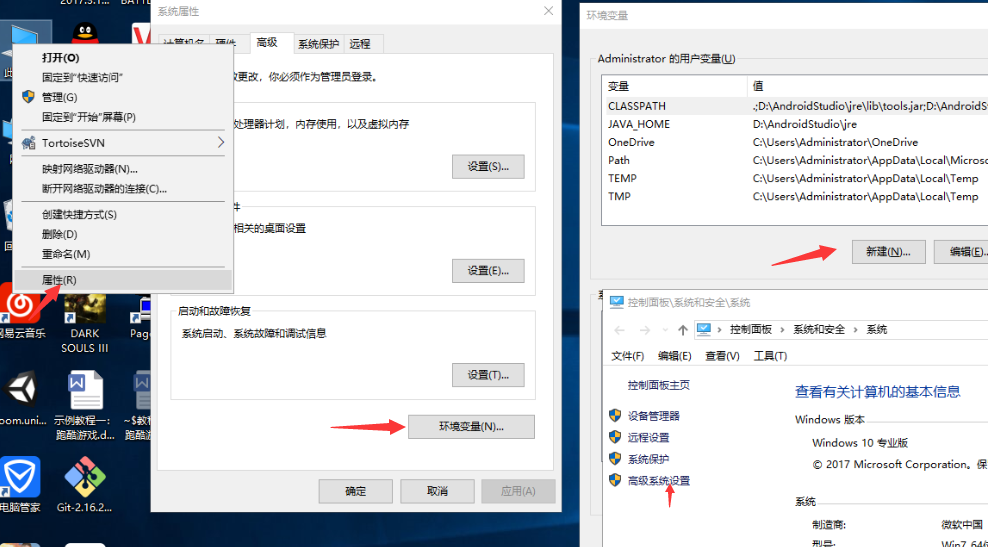


设置JRE安装路径

点击下一步后，等待程序完成安装即可。

### 环境变量配置

完成JDK安装后，我们还需要设置JAVA程序需要的环境变量，才能够进行使用。我们以Windows10系统举例。右键【此电脑】--单击【属性】--单击【高级系统设置】--单击【环境变量】，如图：



环境变量配置

然后点击新建按钮，**一共需要新建3个环境变量，如下**  
**1.** 变量名：JAVA\_HOME  
变量值：C:\Program Files\Java\jdk-10.0.1 注：此处为你的JDK安装路径  
**2.** 变量名：PATH  
变量值：%JAVA\_HOME%/bin  
3. 变量名：CLASSPATH  
变量值：.;%JAVA\_HOME%/lib/tools.jar;%JAVA\_HOME%/lib/dt.jar

### SDK概述

SDK(software development kit)软件工具开发包。被软件开发工程师用于为特定的软件包，软件框架、硬件平台、操作系统等建立应用软件的开发工具的集合。而我们的游戏想要发布为安卓平台的应用程序，还需要下载安卓的SDK才行。

### 安卓SDK下载

由于之前我们的Java环境已经搭建好了，现在下载安卓的SDK包不用安装，只需要下载解压后，就可以使用了。通常下载安卓的SDK主要有以下3种方式：

1. https://developer.android.com/studio/ 官网下载 Android Studio软件安装。
2. 网上找到别人下载好的SDK压缩包，直接下载解压。
3. <http://www.androiddevtools.cn/>国内开发网站，下载SDK ToolS，进行SDK的下载。

我们这以第三种下载方式为例。

#### SDK Tools下载

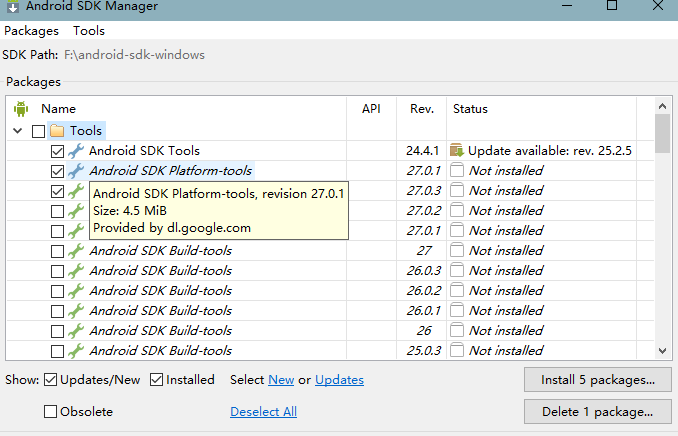
首先访问国内的开发网站<http://www.androiddevtools.cn/>，找到SDK Tools的下载选项，选择对应系统的版本，点击下载。如图：



SDK Tools下载

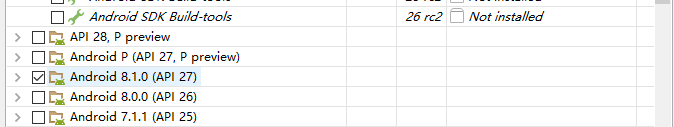
#### 运行SDK Manager，下载对应组件包

下载完成后，解压运行其中的SDK Manager文件，进行下载页面，如图：



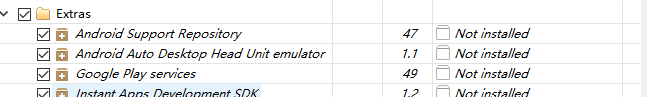
SDK组件下载

Tools选项的我们只需要下载上图所勾选的3项就行，API我们就选择当前时间安卓的最新版本进行下载，如图：



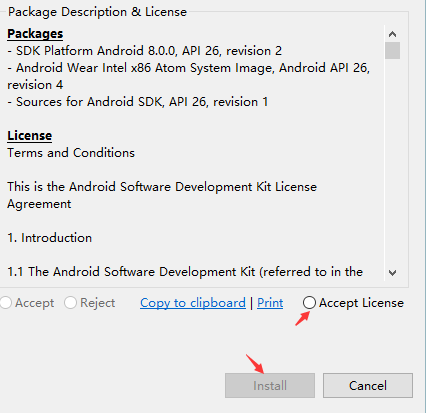
Android API下载

然后我们在附加选项—Extras选择全部勾选，如图：



Extras下载

点击右下角的Install选项进行下载，在后续的许可界面—图，选择接受许可—Accept License，然乎点击安装按钮—Install，等待安装完成即可。

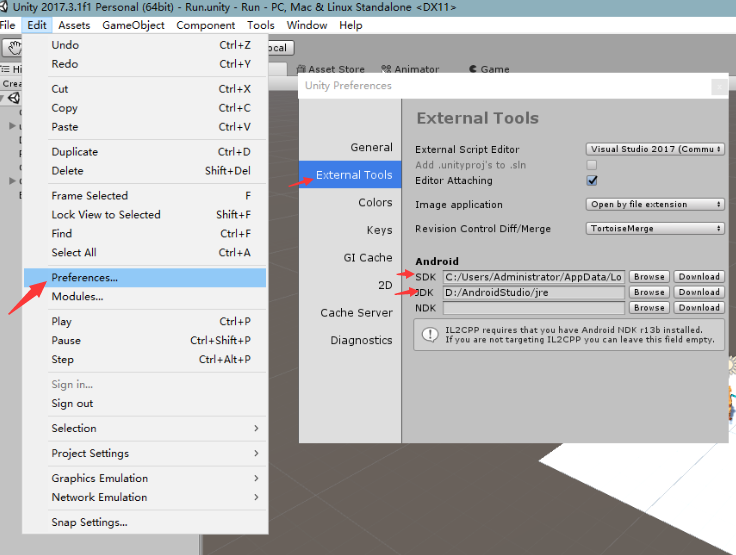


安装许可

### 导出设置

#### 路径设置

配置完打包需要的环境后，我们回到Unity进行JDK与SDK的路径设置。打开Edit—Preferences—External Tools界面，如图：

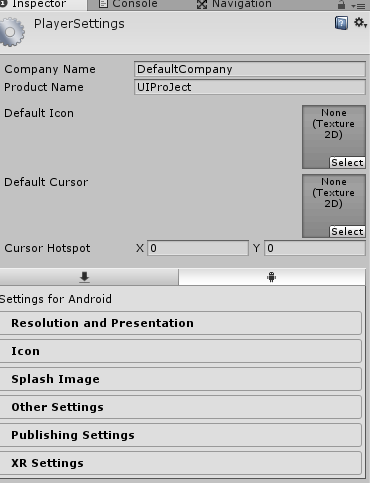


路径配置

在其中的SDK与JDK选项分别填上SDK的路径以及JDK的路径。

#### 导出设置

路径设置完成后。我们回到本章节一开始的界面。选中安卓平台，点击Switch Platform切换平台，切换完成后点击Player Settings按钮，进行导出设置。如图：



导出设置

#### 简单介绍

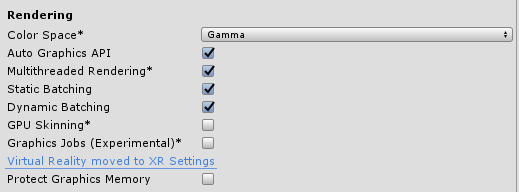
导出设置的大部分选项与本次打包联系不太，在这我们只需要简单了解一下即可。

|  |  |
| --- | --- |
| Company Name | 公司名称 |
| Product Name | 产品名称 |
| Default Icon | 默认图标 |
| Default Cursor | 默认光标-鼠标样式 |
| Cursor Hotspot | 光标点—鼠标的作用点 |
| Resolution and Presentation | 分辨率与描述菜单 |
| Icon | 图标 |
| Splash Image | 启动动画 |
| Other Settings | 其他设置 |
| Publishing Settings | 发布设置 |
| XR Settings | XR设置—应用于虚拟现实游戏 |

#### Other Settings详解

其中Other Settings中的选项与我们的打包息息相关，需要读者详细了解一下，我们以当前版本为例进行详细介绍。

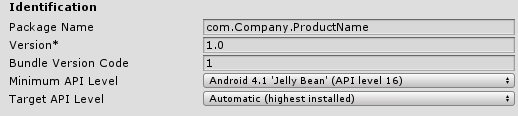
第一部分，渲染相关如图：



Rendering

|  |  |
| --- | --- |
| Rendering |  |
| Color Space | 色彩空间—用于渲染的色彩空间 |
| Gamma | 伽马空间渲染 |
| Linear | 线性渲染 |
| Auto Graphics API | 自动图形接口选择 |
| Multithreaded Rendering | 多线程渲染 |
| Static Batching | 静态批处理-降低Drawcall，优化性能 |
| **Dynamic Batching** | 动态批处理 |
| Gpu Skinning | 加速骨骼动画-启用DX11/ES3的一种蒙皮技术，仅支持VR应用程序 |
| Graphics Jobs(Experimental) | 将图形渲染循环运行至其他CPU核心，只有Vulkan图形API才支持 |
| Protect Graphics Memory | 强制图形缓冲显示经过硬件的保护路径，设备不支持则该选项无效。 |

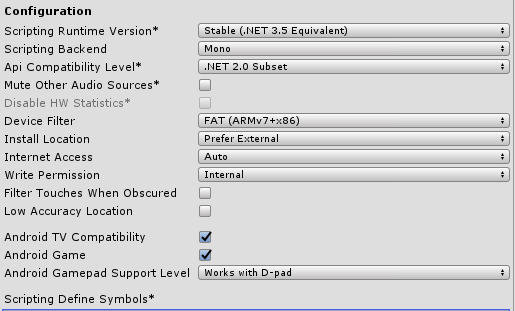
1. 识别相关如图：



Identification

|  |  |
| --- | --- |
| Identification |  |
| Package Name | 包名—打包需要按照对应格式进行修改 |
| Version | 版本号 |
| Bundle Version Code | 内部版本号 |
| Minimum API Level | 向下兼容最低的API版本 |
| Target API Level | 打包的目标API版本 |

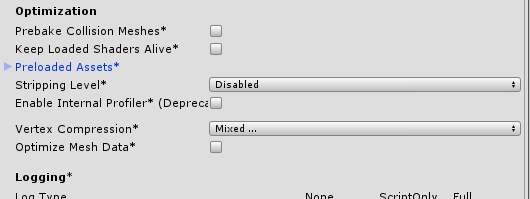
1. 配置相关如图：



Configuration

|  |  |
| --- | --- |
| Configuration |  |
| Scripting Runtime Version | 运行时的脚本版本—指当前开发语言的版本 |
| Scripting Backend | 脚本引擎 |
| Mono | 使用默认的Mono虚拟机 |
| IL2CPP | 使用IL2CPP |
| API Compatibility Level | 脚本API的版本 |
| Mute Other Audio Sources | 其他音频来源 |
| Disable HW Statistics | **禁用HW统计** |
| **Device Filter** | **设备筛选器-**选择应用程序在指定CPU上运行：ARMv7、x86； |
| **Install Location** | 安装位置 |
| Automatic | 自动选择 |
| Prefer External | 外部存储—SD卡 |
| Force Internal | 内部存储 |
| Internet Access | 网络访问权限 |
| Auto | 自动 |
| Require | 根据需求获取 |
| Write Permission | 写入权限 |
| Internal | 内部存储 |
| External(SDCard) | 外部存储—SD卡 |
| Filter Touches When Obscured | 所在窗口被其它可见窗口遮住时,是否过滤触摸事件 |
| Low Accuracy Location | 低精度定位—GPS |
| Android TV Compatibility | 安卓TV兼容选项 |
| Android Game | 应用程序标记为游戏 |
| Android Gamepad Support Level | 游戏手柄支持等级 |
| Scripting Define Symbols | 脚本编译时的标志设置 |

1. 优化相关如图：



Optimization

|  |  |
| --- | --- |
| Optimization |  |
| Prebake Collision Meshes | 是否只在创建游戏时添加碰撞盒 |
| Keep Loaded Shaders Alive | 保证Shader不会被卸载 |
| Preloader Assets | 预先读取的Assets列表 |
| Stripping Level | 代码剥离等级，只有脚本后端选择IL2CPP才能使用 |
| Disabled | 没有减少 |
| **Strip Assemblies** | 一级尺寸减少 |
| **Strip ByteCode (iOS only)** | 二级减少包含一级 |
| **Use micro mscorlib** | 三级减少包含一级二级 |
| Enable Internal profiler | 启用内部分析器，允许在安卓的logcat获取设备分析的数据。 |
| Vertex Compression | 选择顶点进行压缩，压缩可以节省内存和带宽，但精度会更低。 |
| Optimize Mesh Data | 优化Mesh数据，勾选该选项会移除材质所不需要的Mesh数据 |
| Logging | 更改Log类型 |

#### 更改包名与打包

上一小节中，我们详细的介绍了Other Settings中的选项的作用，现在我们工程打包，只需要按照格式更改包名—Package Name，以及产品名称—Product Name就行了。然后点击Build进行打包，等待完成即可。