

# 将Unity游戏移植到Windows Phone 8.1



Elissa Tong | 微软开发技术顾问  
[elistong@microsoft.com](mailto:elistong@microsoft.com)

# 本节课程内容概述

## 课程标题

01   准备工作	系统需求   开发环境
02   创建Unity游戏	了解Unity编辑   游戏开发概念   移植到Windows Phone
03   硬件更改	多种分辨率   方向   内存用量   纹理压缩   角色动作
04   API了解	.NET API   Unity插件   Action   Global Instances
05   开发工具	

# 01 | 准备工作：系统了解



# 系统需求

Unity 引擎	<ul style="list-style-type: none"><li>• Unity 4.2+有Windows Store Apps及Windows Phone 8.0<ul style="list-style-type: none"><li>• Unity 4.5+开始支持Universal Apps及Windows Phone 8.1</li><li>• 建议用最新的Unity版本</li></ul></li></ul>
Windows PC	<ul style="list-style-type: none"><li>• Windows 8.x Professional+, 64 bit</li><li>• Microsoft Visual Studio Professional, Premium, Ultimate 2013<ul style="list-style-type: none"><li>• Update 2+支持Universal Apps</li></ul></li><li>• Microsoft Visual Studio Express 2013</li><li>• Windows Phone设备</li></ul>
Mac OS X	<ul style="list-style-type: none"><li>• Boot Camp 5+</li><li>• Windows 8.x Professional &amp; up 64-bit ISO</li><li>• Windows 7 USB/DVD download tool</li><li>• Parallels 9+</li></ul>

# 系统安装资料

## Windows PC

Unity:

- [unity3d.com/unity/download](http://unity3d.com/unity/download)

Windows OS:

- [windows.microsoft.com/zh-cn/windows/download-shop](http://windows.microsoft.com/zh-cn/windows/download-shop)

Visual Studio IDE:

- [visualstudio.com/zh-cn/downloads/download-visual-studio-vs](http://visualstudio.com/zh-cn/downloads/download-visual-studio-vs)

## Mac OS X

用 Boot Camp:

- [microsoftstore.com/store/msusa/html/pbPage.Help\\_Win7\\_usbdvd\\_dwnTool](http://microsoftstore.com/store/msusa/html/pbPage.Help_Win7_usbdvd_dwnTool)
- <http://windows.microsoft.com/zh-cn/windows-8/install-windows-on-mac>
- [help.apple.com/bootcamp/mac/5.0/help](http://help.apple.com/bootcamp/mac/5.0/help)
- [digitaltrends.com/computing/how-to-install-windows-8-on-a-mac/](http://digitrends.com/computing/how-to-install-windows-8-on-a-mac/)

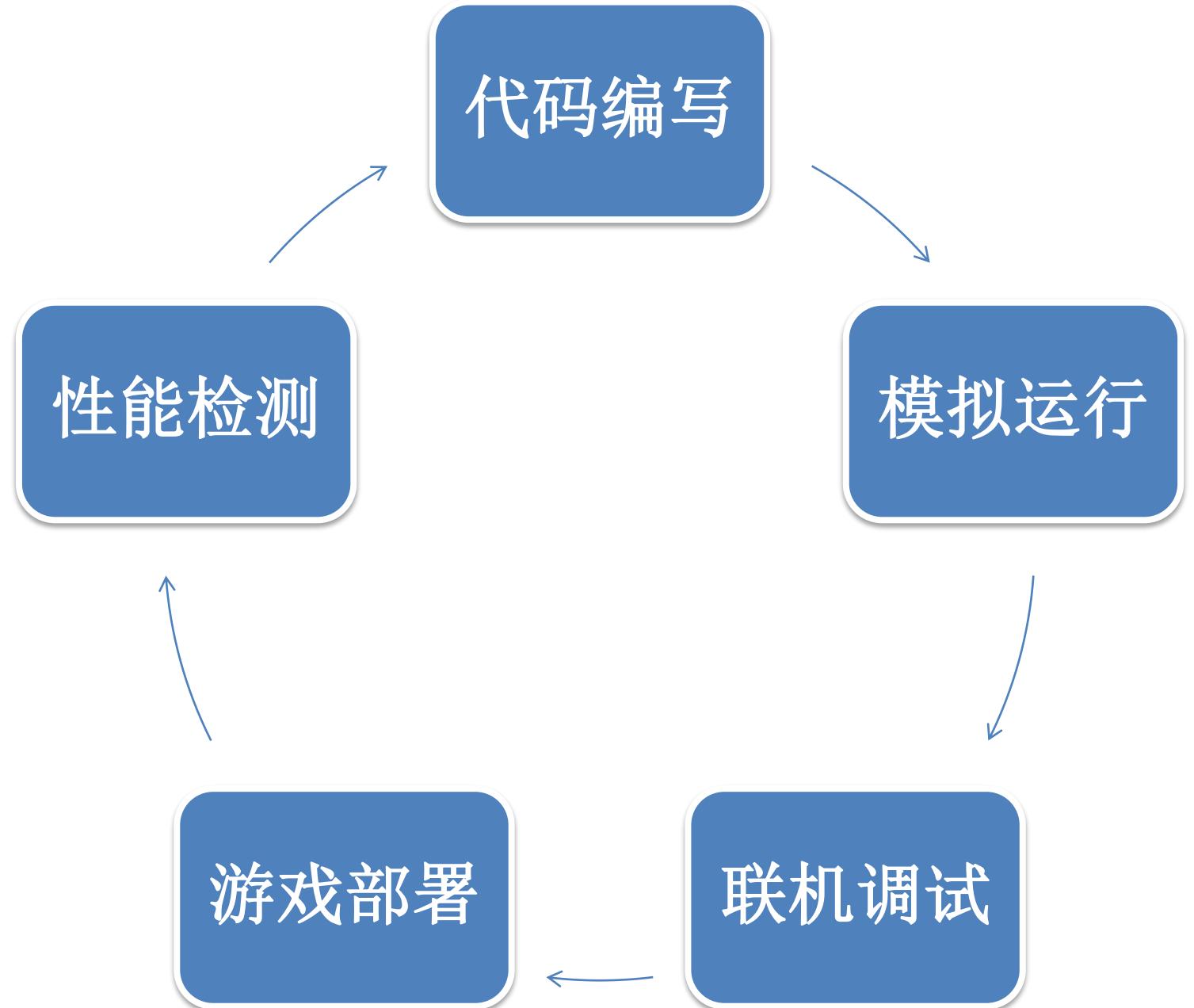
用 Parallels:

- [download.parallels.com/desktop/v9/ga/docs/zh-CN/Parallels%20Desktop%20User's%20Guide](http://download.parallels.com/desktop/v9/ga/docs/zh-CN/Parallels%20Desktop%20User's%20Guide)

# 01 | 准备工作：了解开发工具



# 手游开发过程



# 手游开发过程

## Windows Phone 工具：

Windows Phone

Windows Phone Emulator

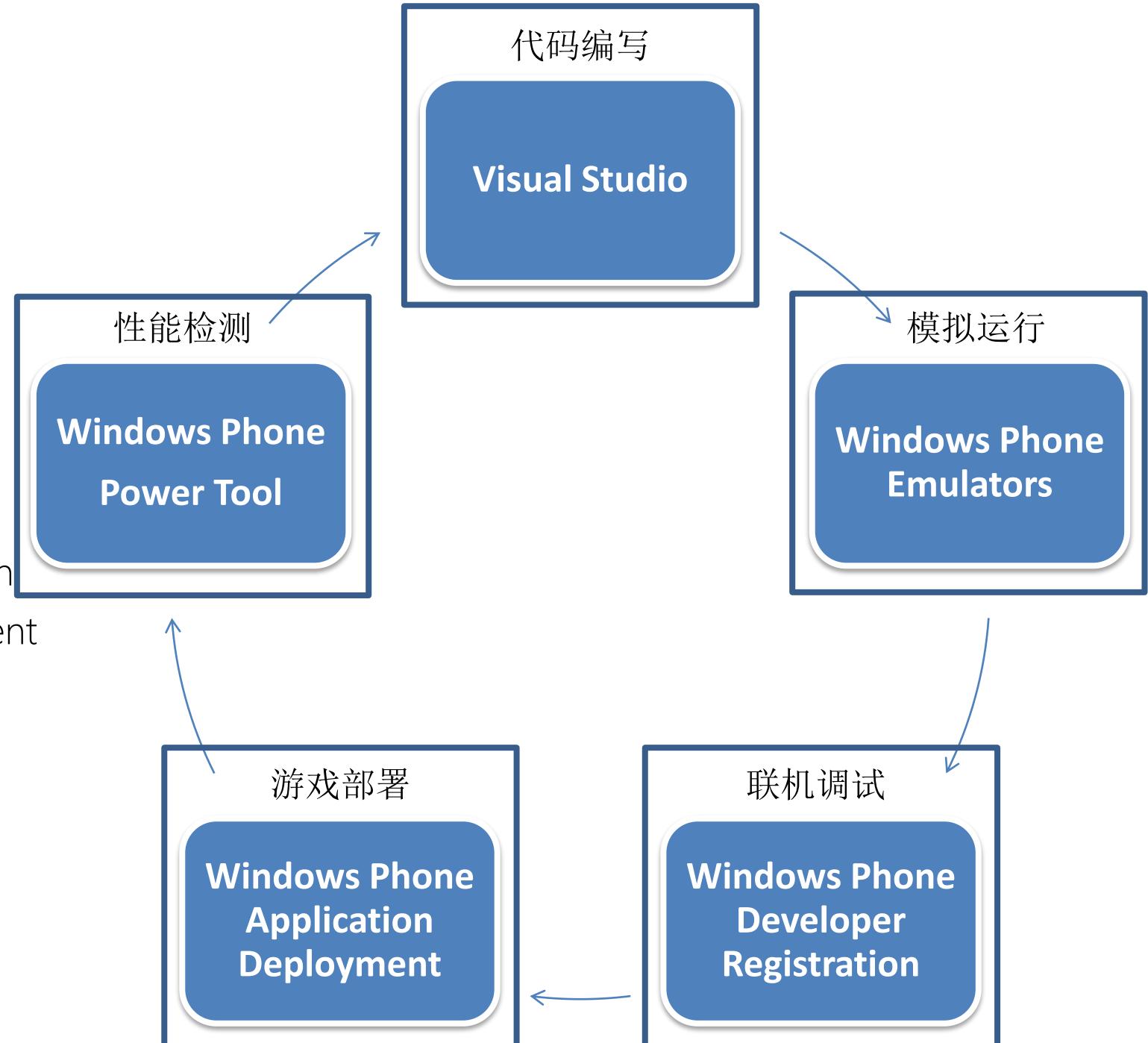
Hyper-V Manager

Dev Center App for WP8

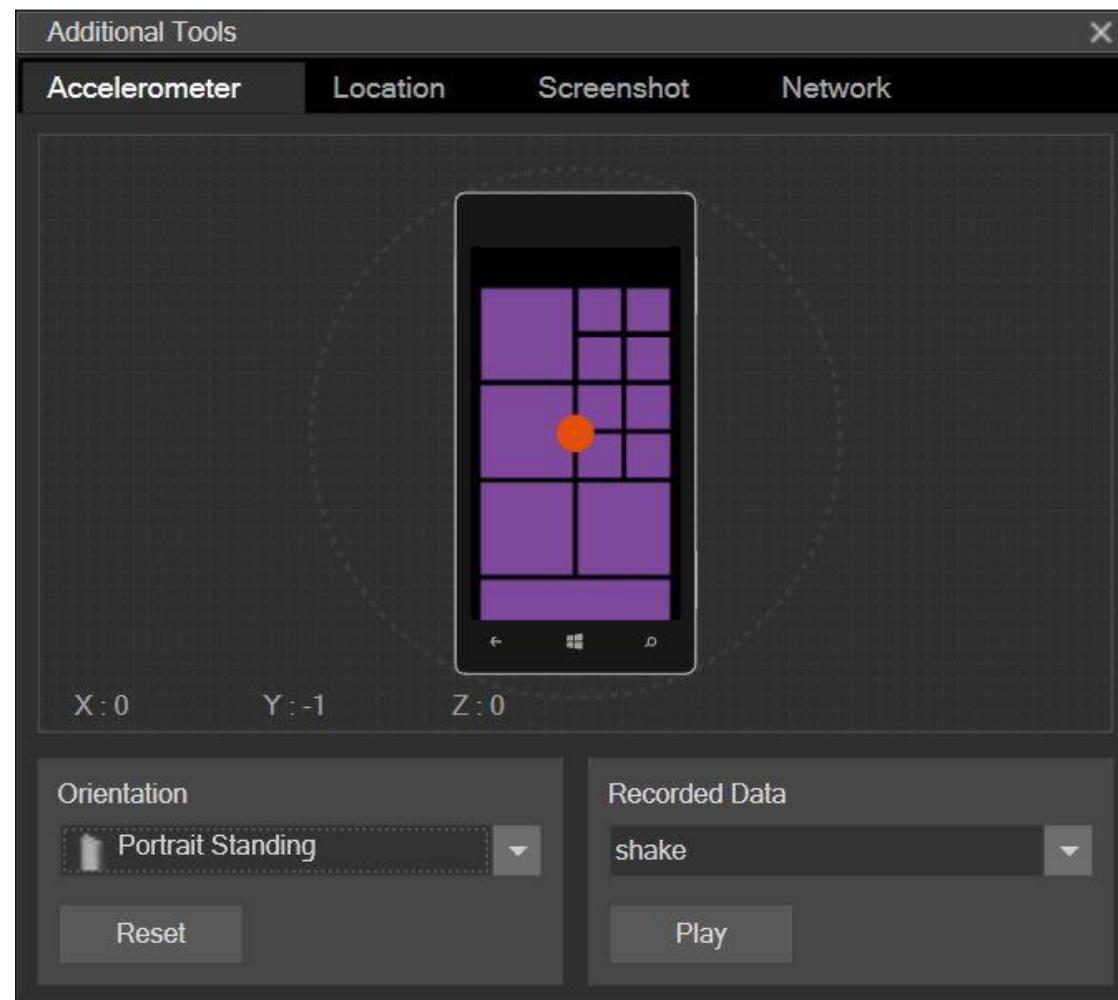
Windows Phone Power Tools

Windows Phone Developer Registration

Windows Phone Application Deployment



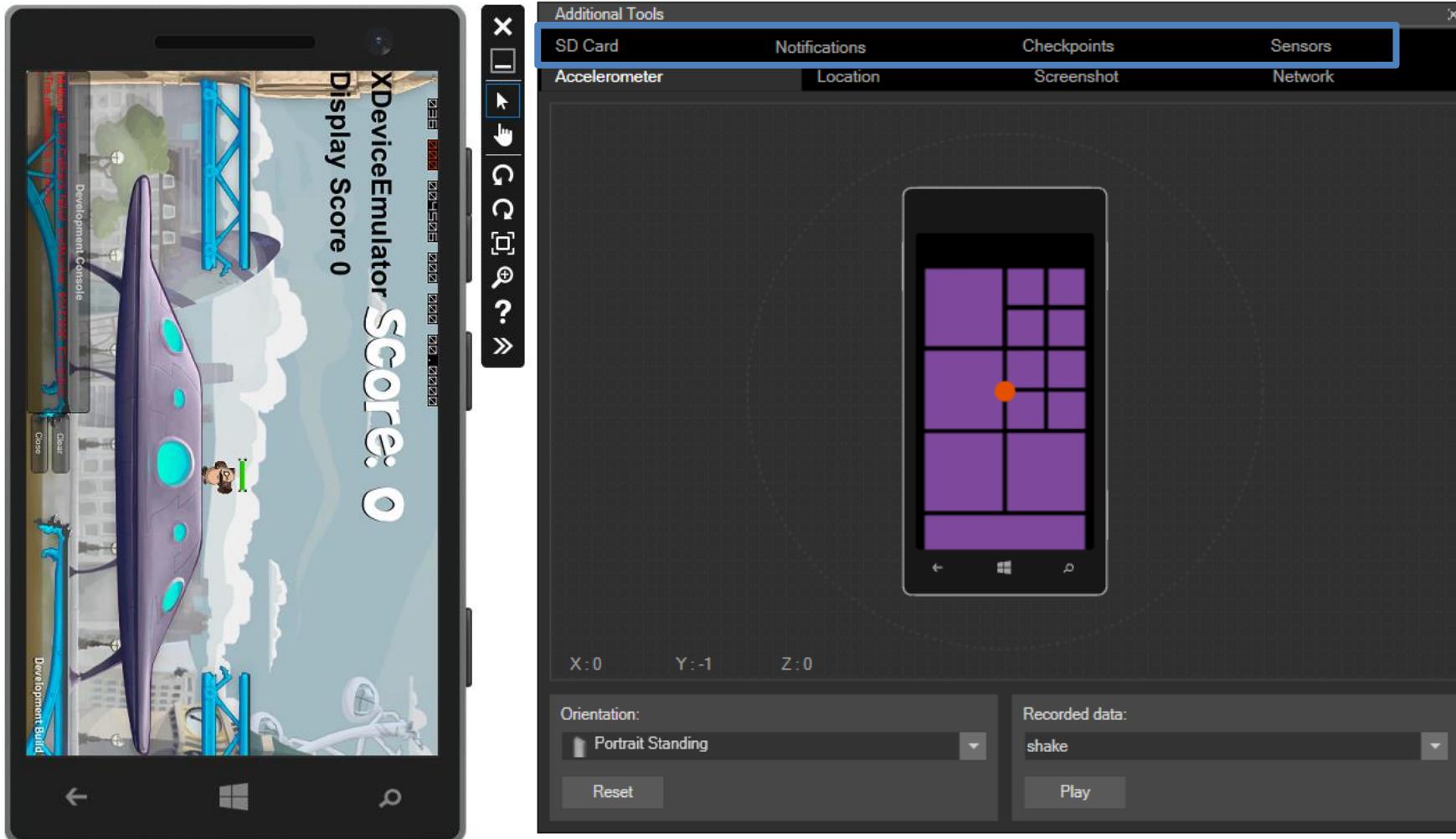
# 模拟器 (8.0)



## 模拟器功能：

- 加速机 Accelerometer
- 位置 Location
- 屏幕快照 Screenshot
- 网络 Network

# 模拟器 (8.1)



## 模拟器的提升：

- SD Card
- Notifications
- Checkpoints
- Sensors

<http://msdn.microsoft.com/zh-cn/library/windows/apps/dn629629.aspx>

# 帧速率计数器

## 代码开关

1

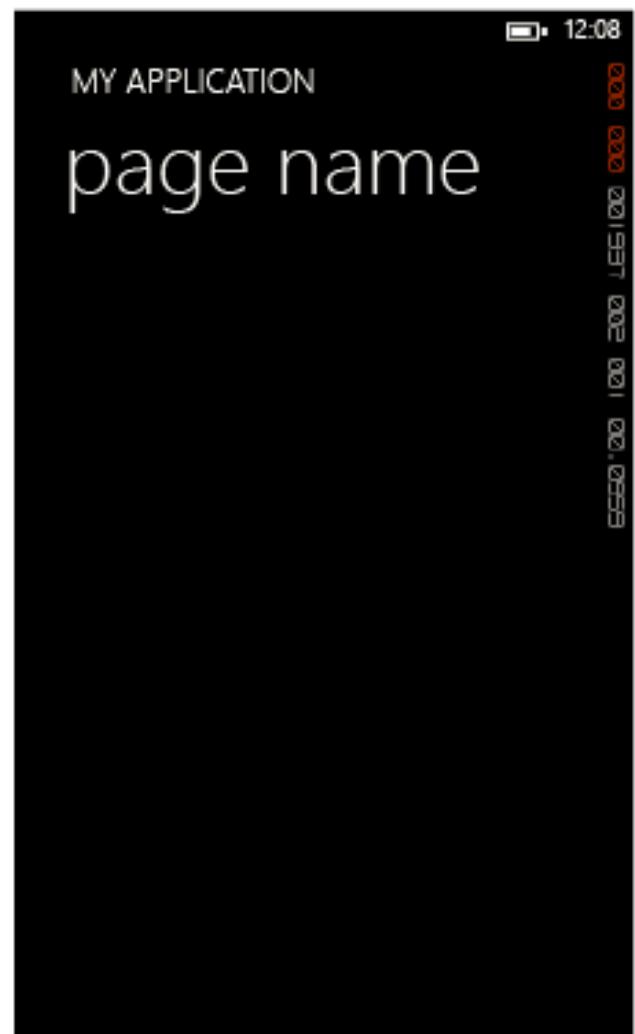
App.xaml.cs:

```
Application.Current.Host.Settings.EnableFrameRateCounter = true;
```

2

```
// ((1024 * 1024) = 1048576
string memUsage =
(Microsoft.Phone.Info.DeviceStatus.
ApplicationCurrentMemoryUsage / 1048576.0f).ToString() + " MB";

string peakMemUsage = (Microsoft.Phone.Info.DeviceStatus.ApplicationPeakMemoryUsage / 1048576.0f).ToString() + " MB";
```



构图线程FPS

用户界面线程FPS

纹理内存使用量

图面计数器

中间图面计数器

屏幕填充速率计数器

# 模拟器错误

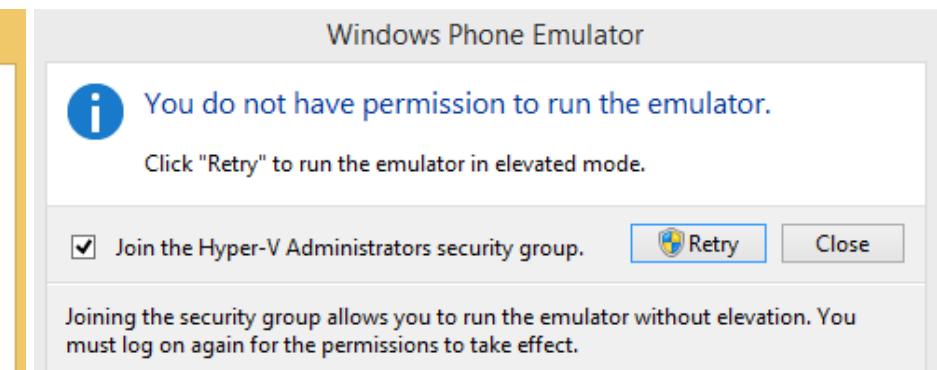
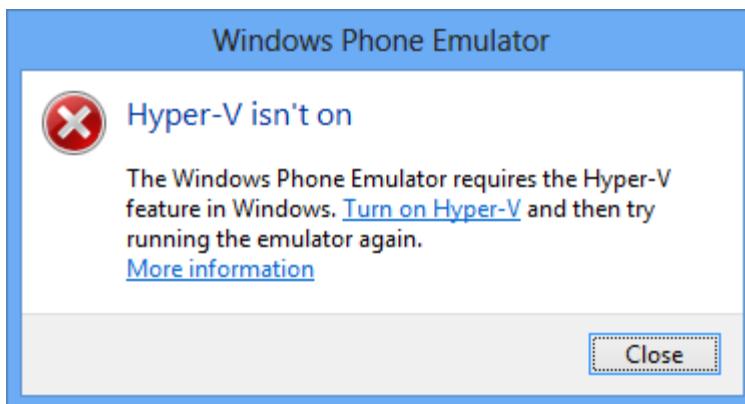
- **解决方案：**

1. 关掉和删除不用的软件
2. 电脑有8GB RAM 内存就减少这个问题
3. 改一下注册表编辑器: <http://support.microsoft.com/kb/2911380/en-us>



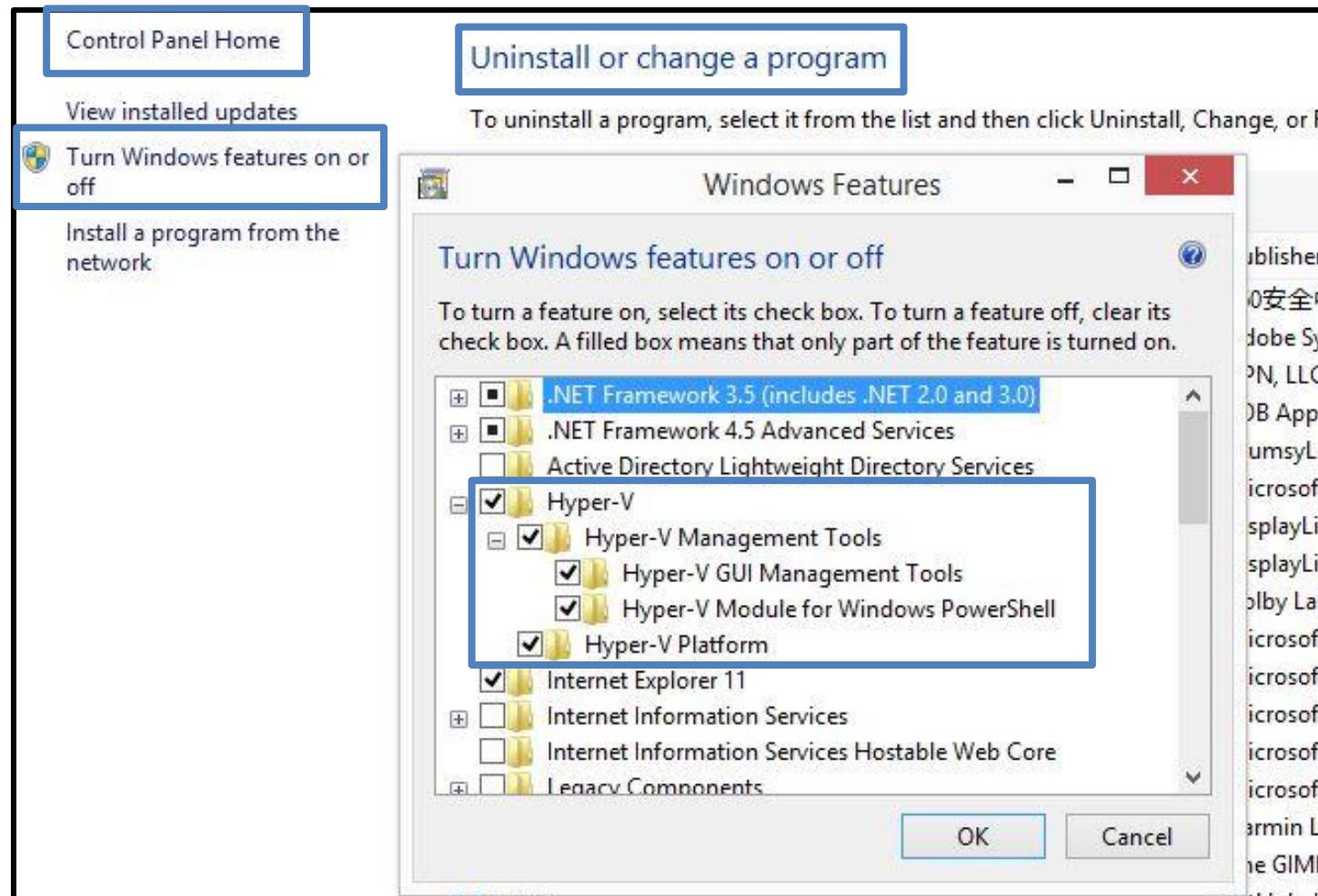
# 模拟器错误

- 关于虚拟化技术
- [http://msdn.microsoft.com/zh-cn/library/windows/apps/jj863509\(v=vs.105\).aspx](http://msdn.microsoft.com/zh-cn/library/windows/apps/jj863509(v=vs.105).aspx)



# Hyper-V

- 第1步：启用Hyper-V



Windows Phone 模拟器本质是：

使用Hyper-V虚拟化技术的虚拟机

# Hyper-V

- 第2步：查询你的电脑是否支持Virtualization？
- <http://technet.microsoft.com/enus/sysinternals/cc835722.aspx>

使用工具  
Coreinfo  
查询

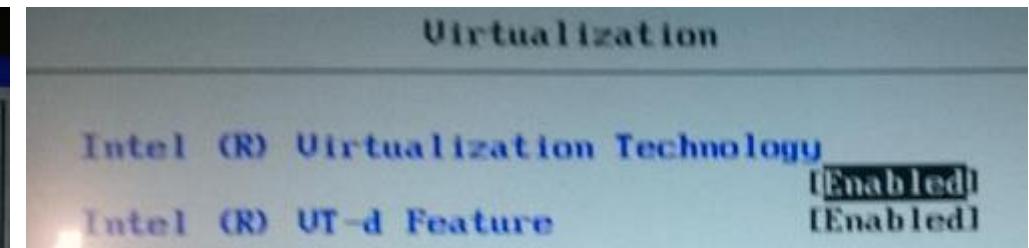
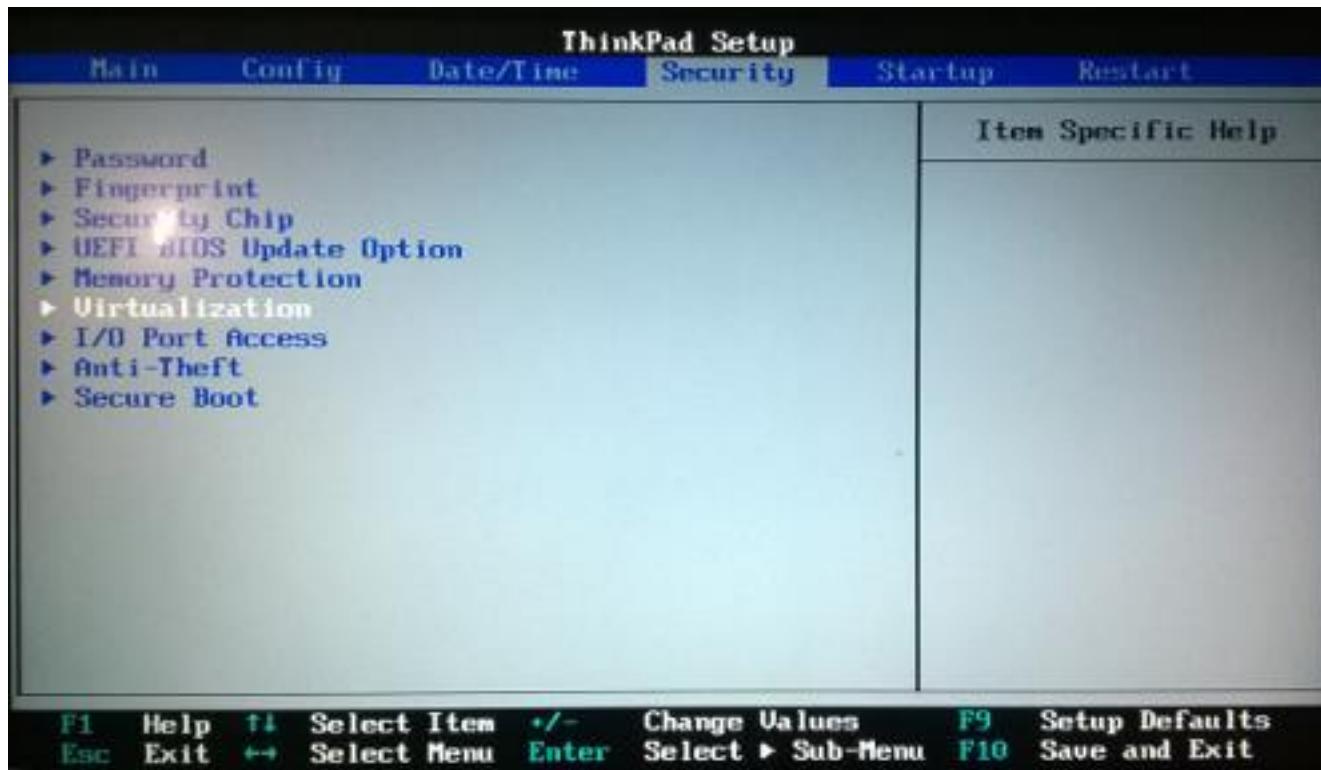
```
C:\Users\yimei\Downloads\Coreinfo>Coreinfo.exe

Coreinfo v3.21 - Dump information on system CPU and memory topology
Copyright (C) 2008-2013 Mark Russinovich
Sysinternals - www.sysinternals.com

Intel(R) Core(TM) i7-3520M CPU @ 2.90GHz
Intel64 Family 6 Model 58 Stepping 9, GenuineIntel
HTT          *      Hyperthreading enabled
HYPERVISOR   *      Hypervisor is present
VMX          -      Supports Intel hardware-assisted virtualization
SVM          -      Supports AMD hardware-assisted virtualization
EM64T        *      Supports 64-bit mode
```

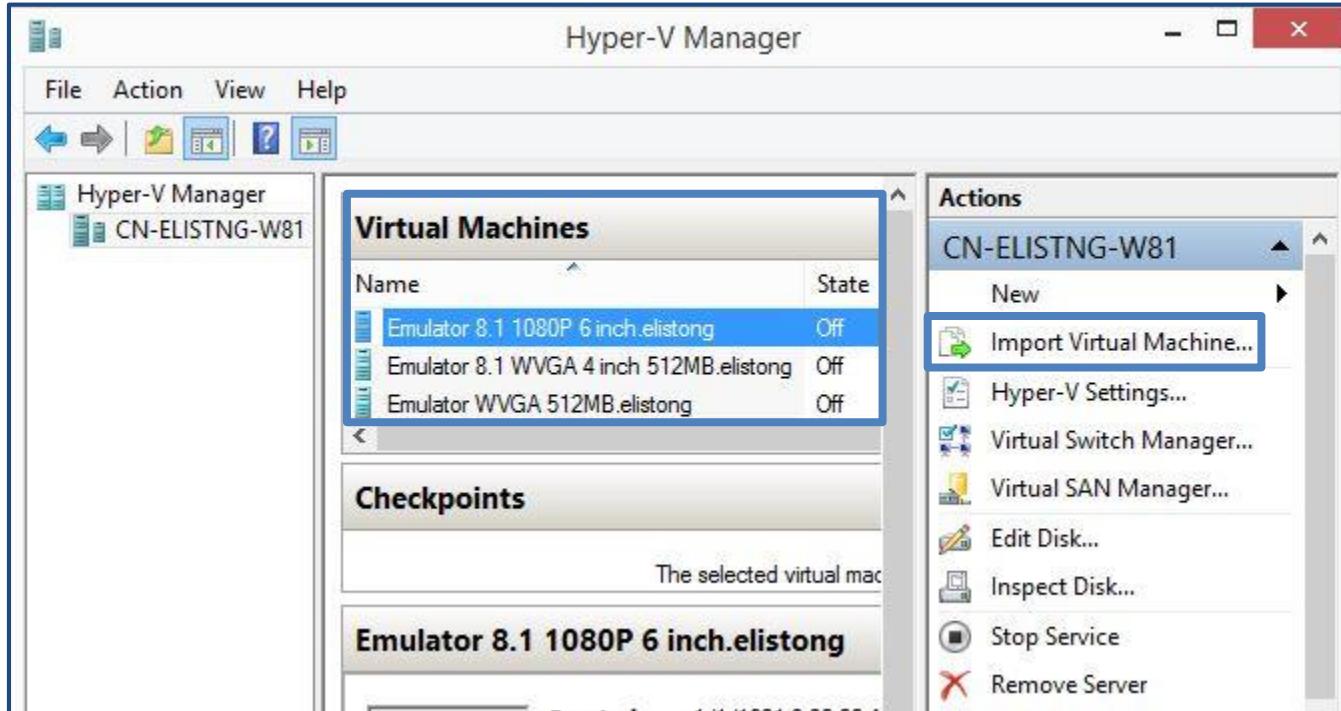
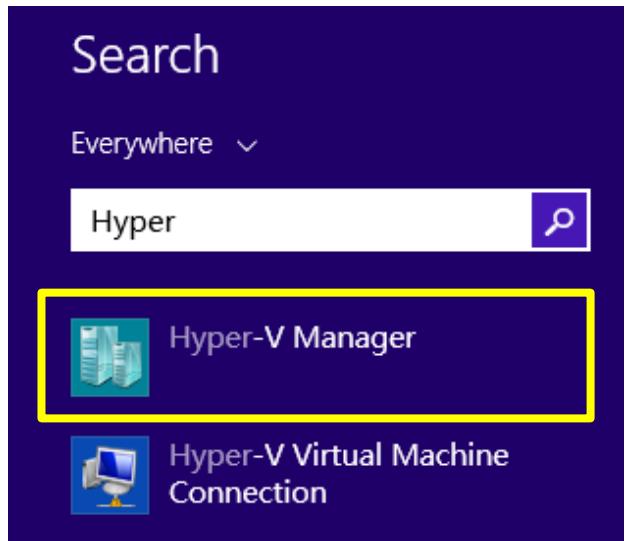
# Hyper-V

- 第3步：在BIOS中打开虚拟化: Virtualization



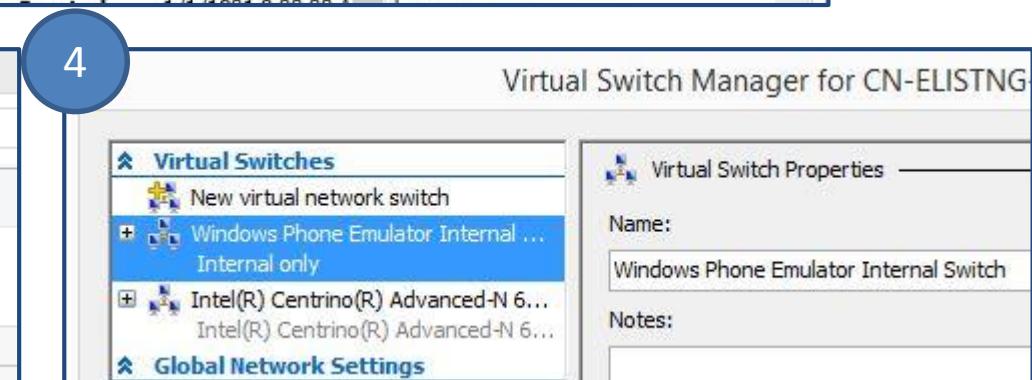
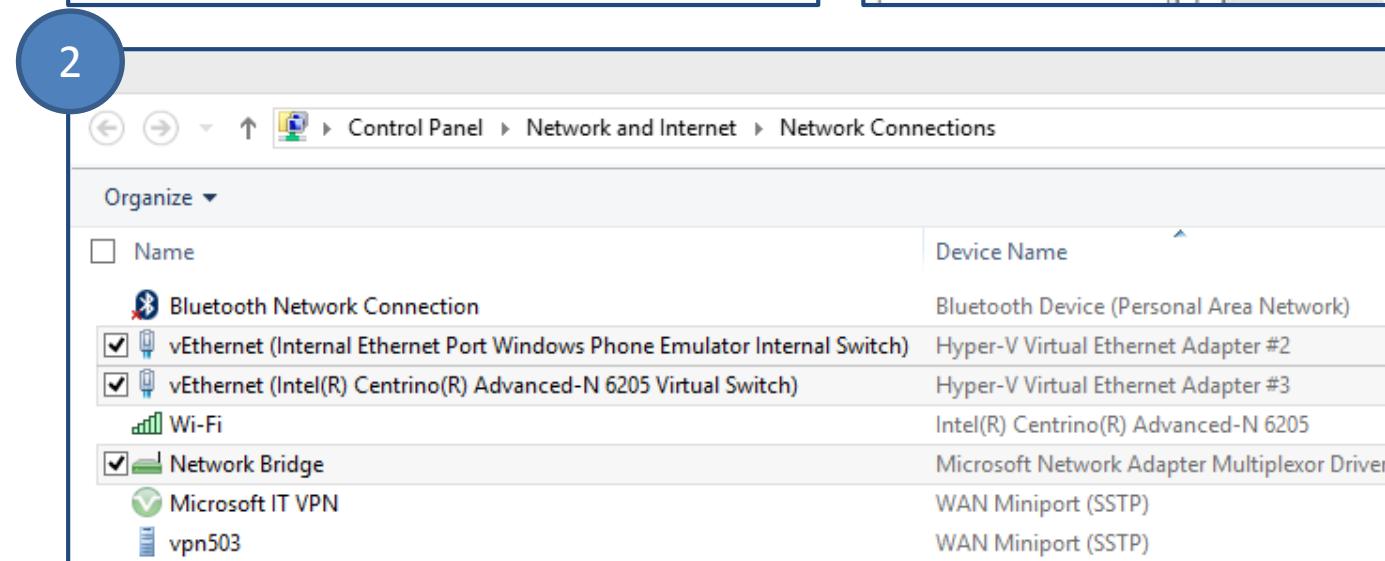
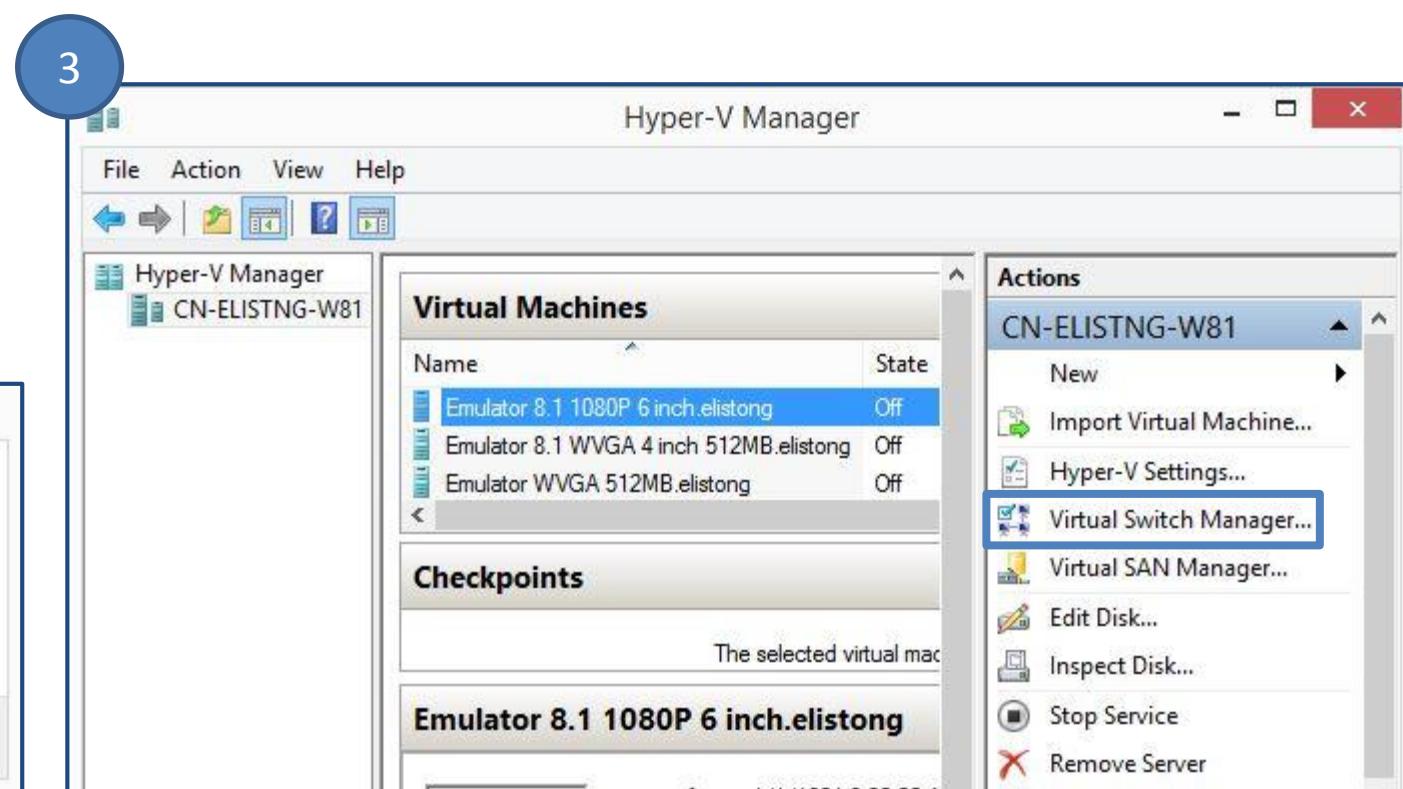
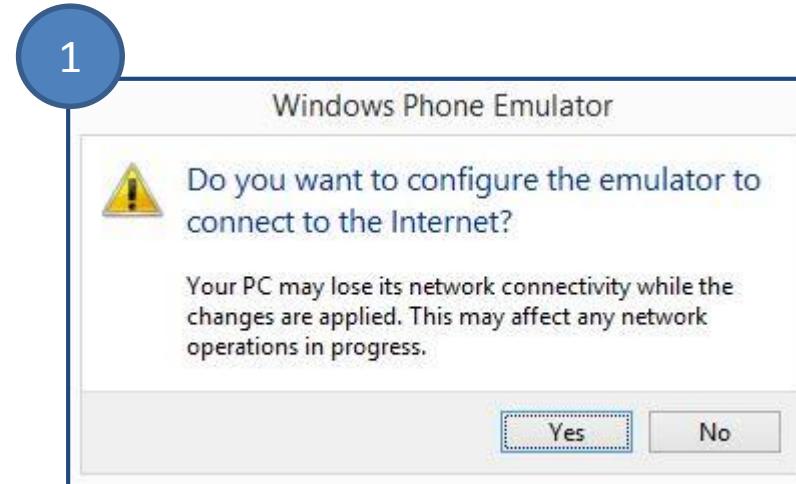
# Hyper-V

- 第4步：Hyper-V Manager



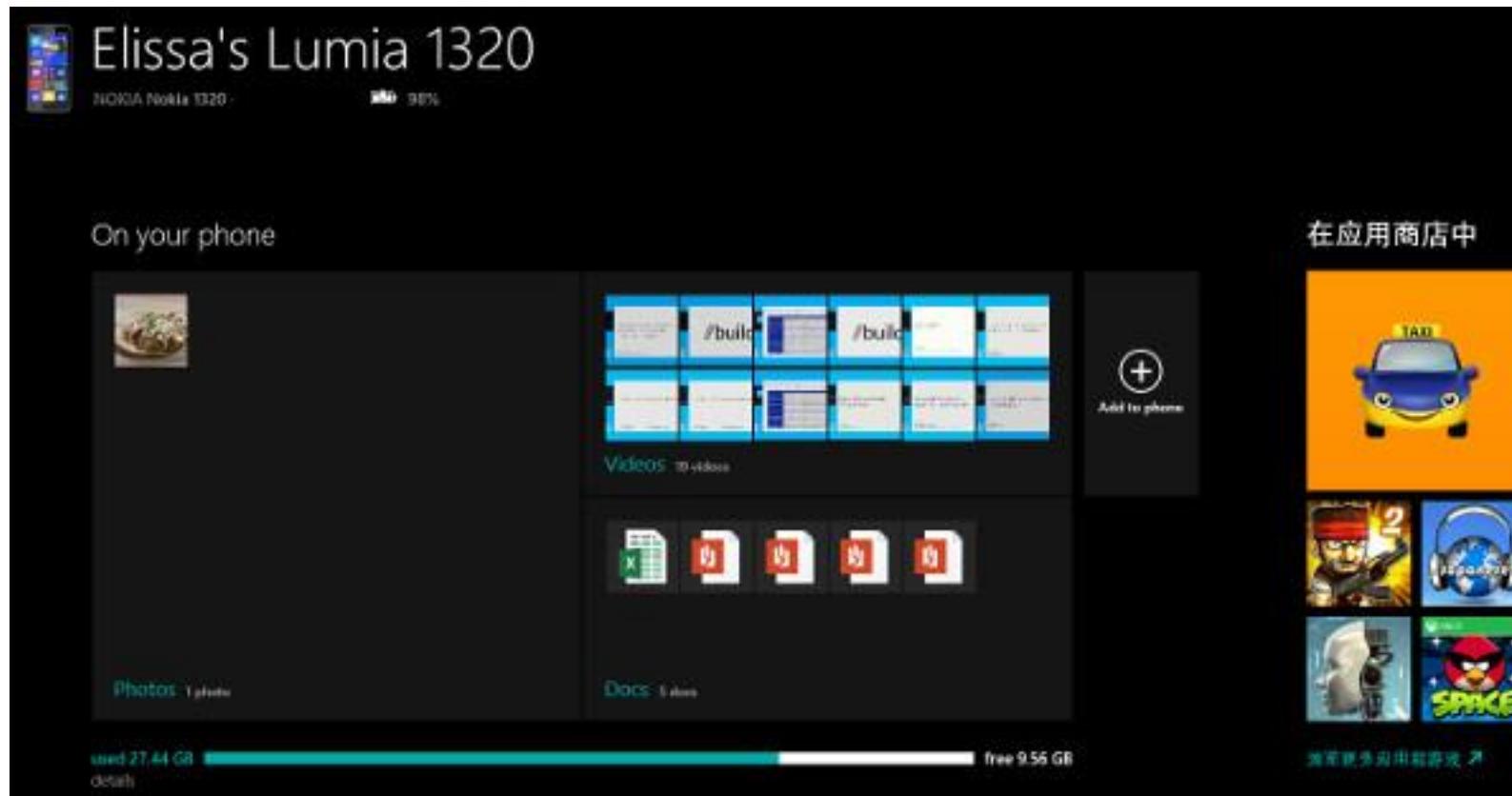
Name	Date modified	Type	Size
Flash.480x800	10/10/2012 3:58 AM	Hard Disk Image F...	39,278 KB
Flash.720x1280	10/10/2012 3:58 AM	Hard Disk Image F...	39,278 KB
Flash.768x1280	10/10/2012 3:58 AM	Hard Disk Image F...	39,278 KB

# 模拟器网络

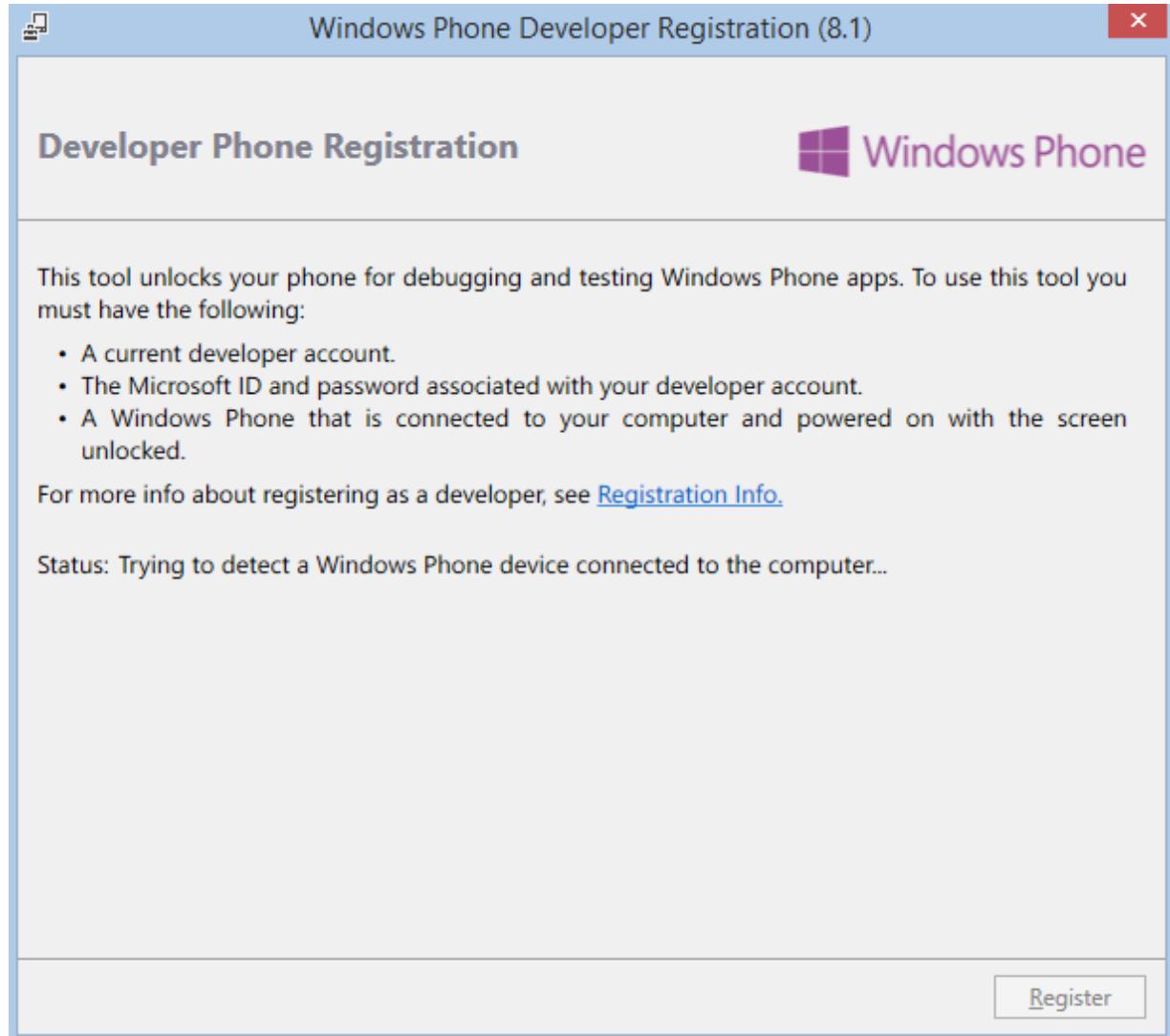


# Windows Phone工具

- 设备用USB连上电脑会打开工具
- 功能包含：照片，视频，文章，音乐



# 手机注册



- 建议用一个WP设备来测试
- 测试设备需要注册
- 参考资料:

<http://irisclasson.com/2013/10/28/problems-and-fixes-when-registering-a-windows-phone-8-device-to-deploydebug-applications>

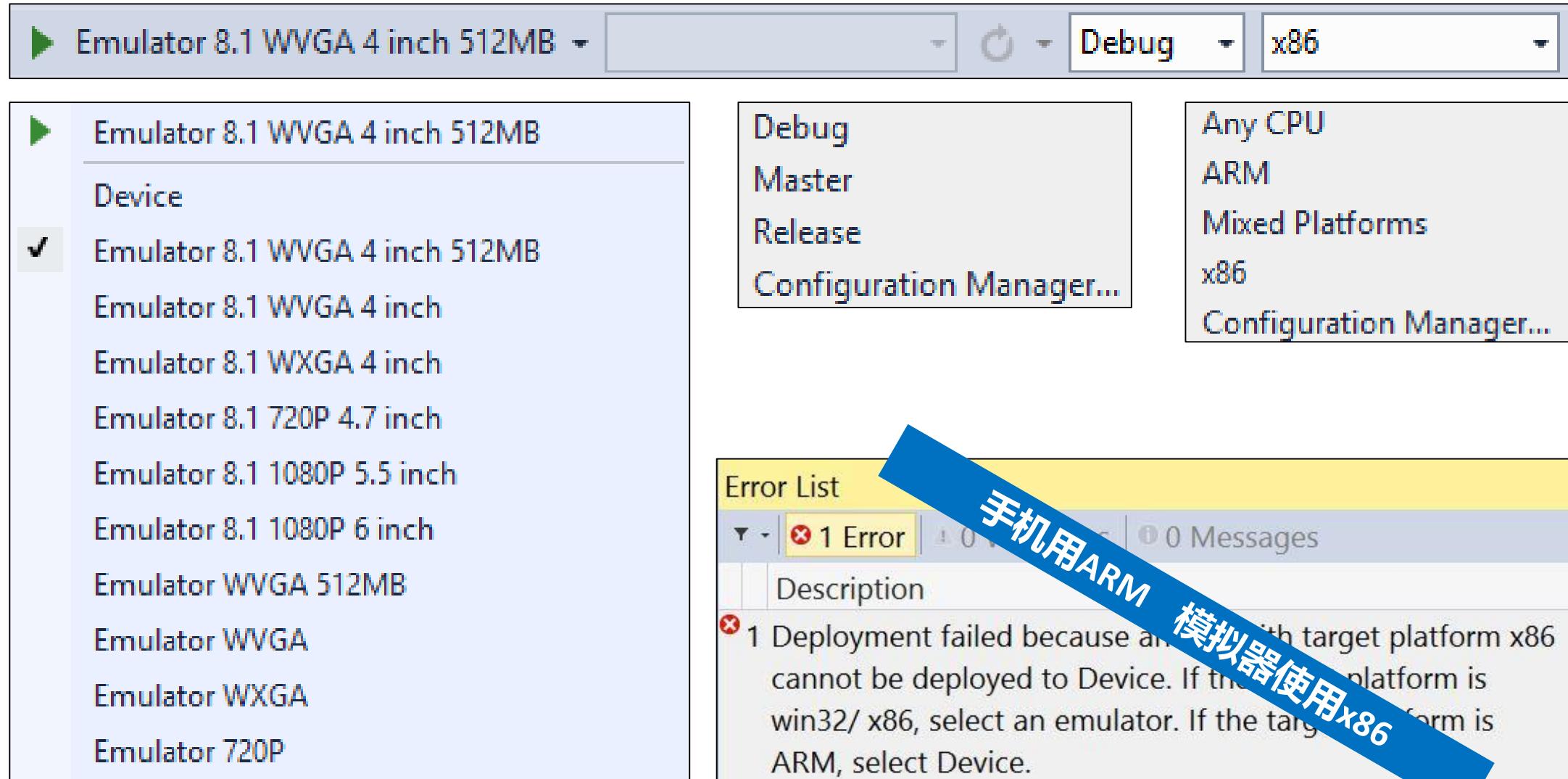
# 手机注册查询

- <https://dev.windowsphone.com/zh-cn/Account/Devices>

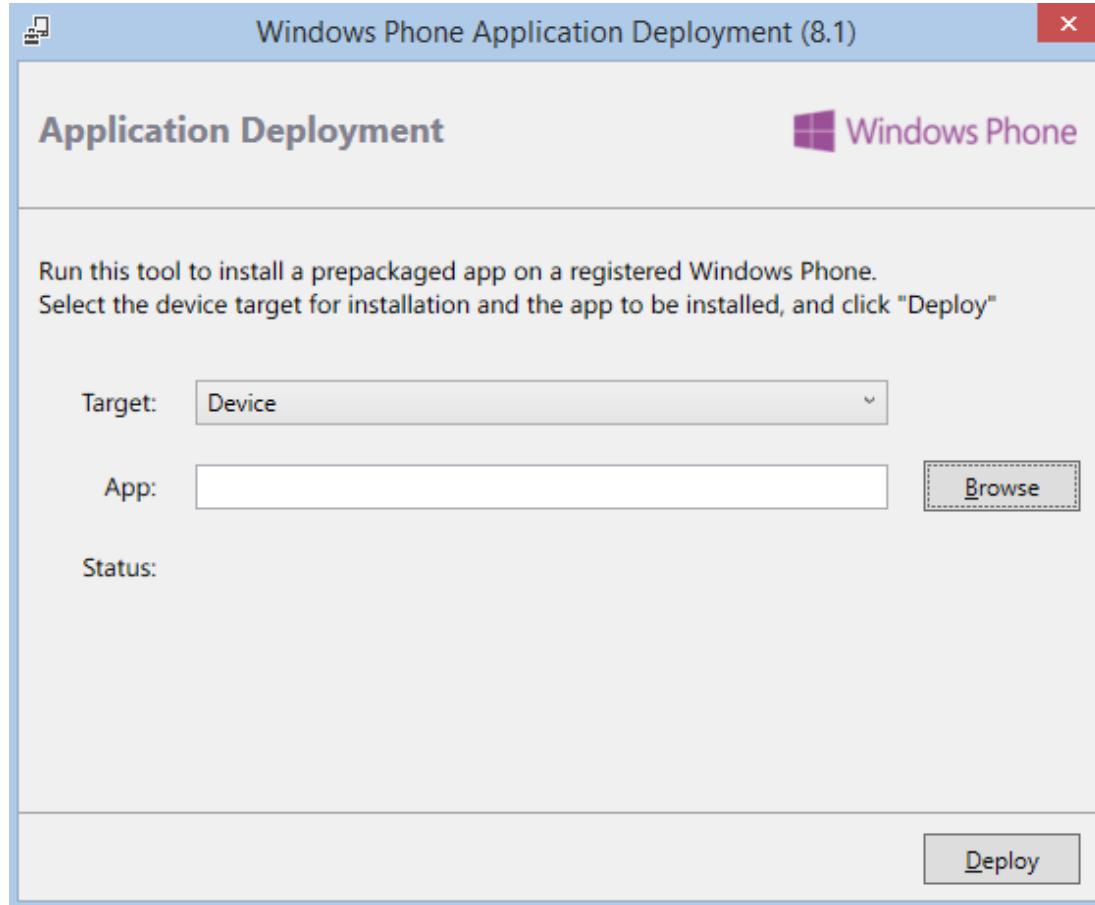
The screenshot shows the Windows Phone Developer Center interface. The top navigation bar includes links for '仪表板' (Dashboard), '入门' (Getting Started), '设计' (Design), '开发' (Development), '发布' (Deployment), and '社区' (Community). The user's email, 'elissatong@hotmail.com', is visible on the right. The main content area is titled '手机' (Devices) and displays instructions for registering phones for testing. It mentions that up to 3 phones can be registered. A table lists two registered devices: 'Elissa's Lumia 1320' and 'Elissa's Nokia Lumia 820'. Both entries include registration dates (2014/4/10 and 2014/2/18 respectively) and expiration dates (2016/4/10 and 2016/2/18 respectively), along with a '删除' (Delete) link.

手机名称	注册日期	到期日期	
Elissa's Lumia 1320	2014/4/10	2016/4/10	<a href="#">删除</a>
Elissa's Nokia Lumia 820	2014/2/18	2016/2/18	<a href="#">删除</a>

# 联机调试



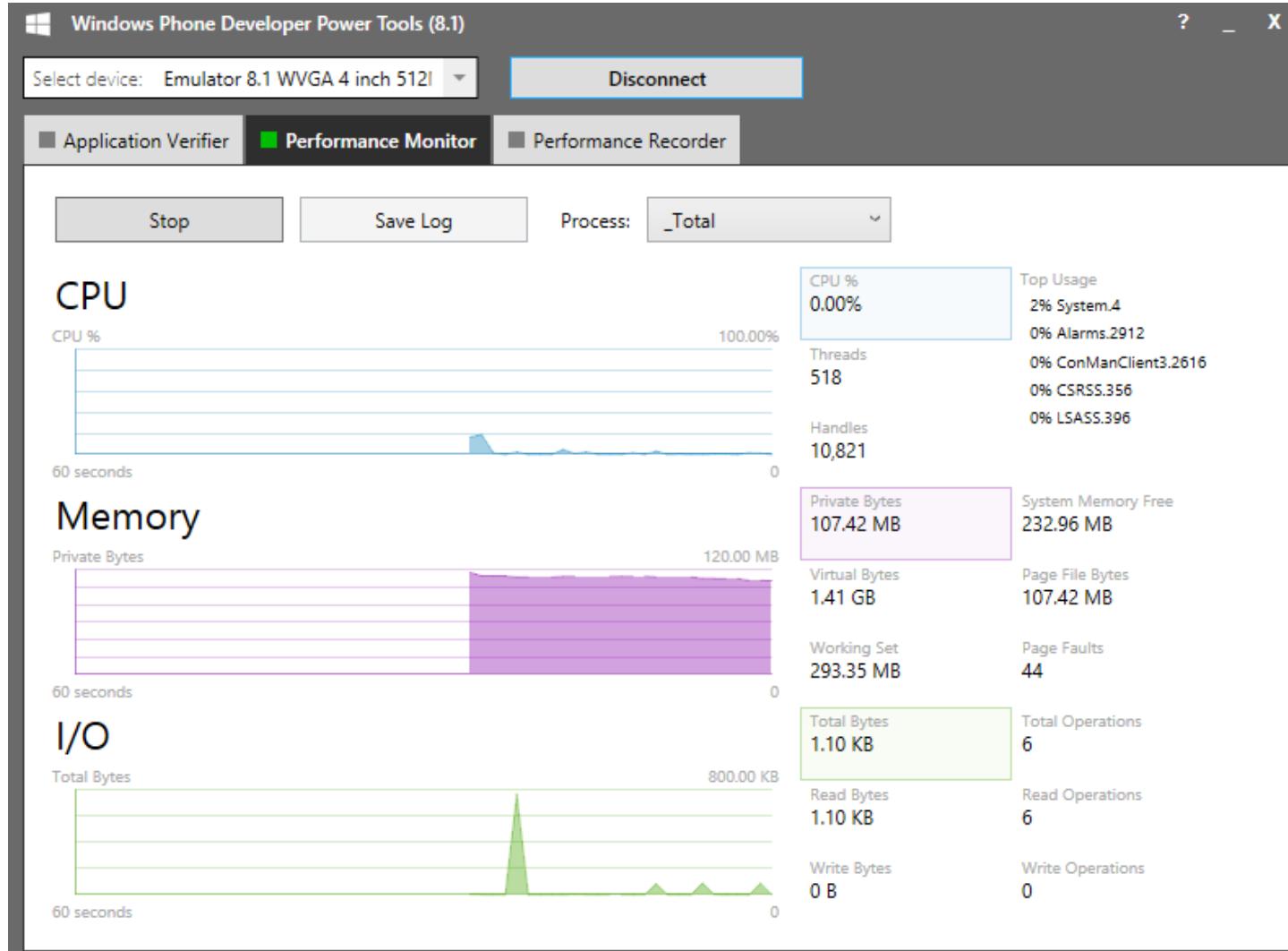
# 应用部署



## 步骤

1. 选择设备或模拟器
2. 上传Windows Phone XAP包
3. 点击Deploy

# Windows Phone Power Tools



## Application Verifier

用于发现资源泄露和程序漏洞

## Performance Monitor

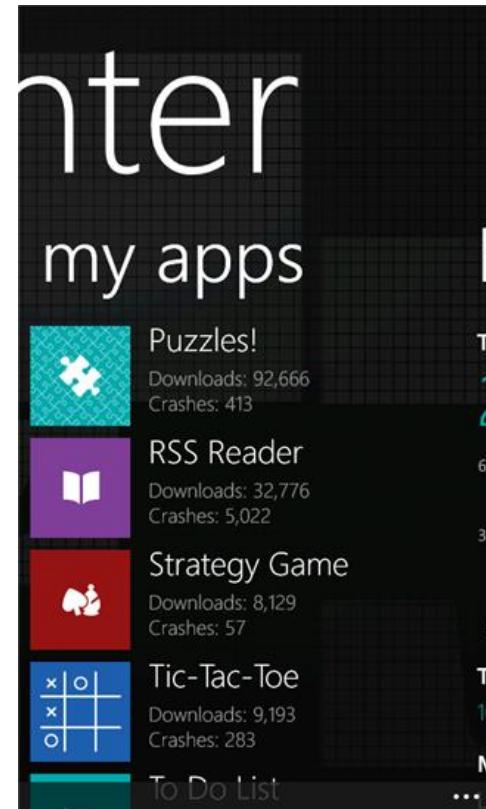
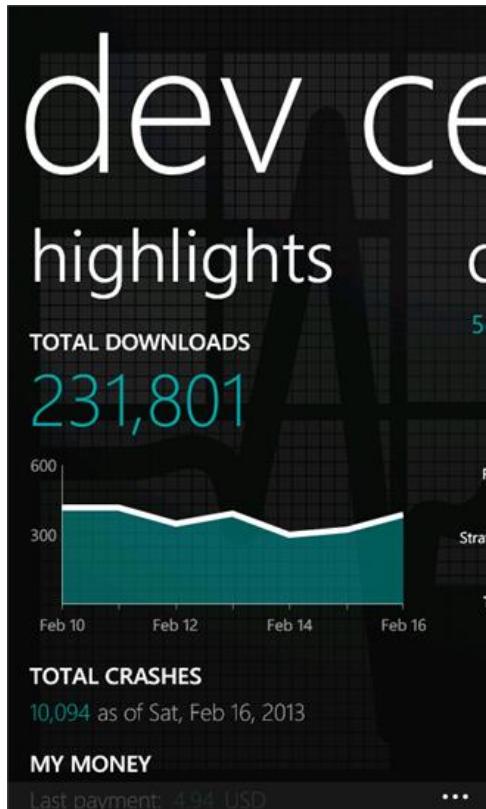
性能实时检测

## Performance Recorder

性能记录

# 开发人员中心应用 Dev Center

- 在Windows Phone商店下载应用
- 功能包含：App资料，下载量，用户评论，崩溃资料



# prime[31] Unity 插件

- 有 Windows Store 及 Windows Phone 8 插件

- 应用内购物 IAP
- 试用版 Trial
- 社区支持 Social
- 微软广告 Ads
- 动态测贴 Live Tiles
- 推送通知 Push Notifications
- Flurry 数据分析 Flurry Analytics
- 等等 & more!

# 其他工具

## Visual Studio Productivity Power Tools 2013

<http://visualstudiogallery.msdn.microsoft.com/dbcb8670-889e-4a54-a226-a48a15e4cace>

## Visual Studio 2013 Tools for Unity

<http://visualstudiogallery.msdn.microsoft.com/20b80b8c-659b-45ef-96c1-437828fe7cf2>

## Prime31 Unity Plugins

<https://prime31.com/>

## Dev Center App

<http://www.windowsphone.com/zh-cn/store/app/dev-center/2d3063c2-4b29-4e69-9c03-50b67b0e6aec>

## Protobuf-net for WP8 .NET

<http://code.google.com/p/protobuf-net/>

# 准备工作总结

Windows  
Store Apps

Windows  
Phone 8.0

Windows  
Phone 8.1

Universal  
Apps

Windows 8  
开发系统

支持  
Mac OS X

Boot Camp 或  
者Parallels

Visual Studio

Windows  
Phone 模拟器

虚拟化技术  
Hyper-V工具

手机注册  
工具及网站

Windows  
Phone Power  
Tools

Dev Center  
App

Prime31插件

Visual Studio  
Tools for Unity

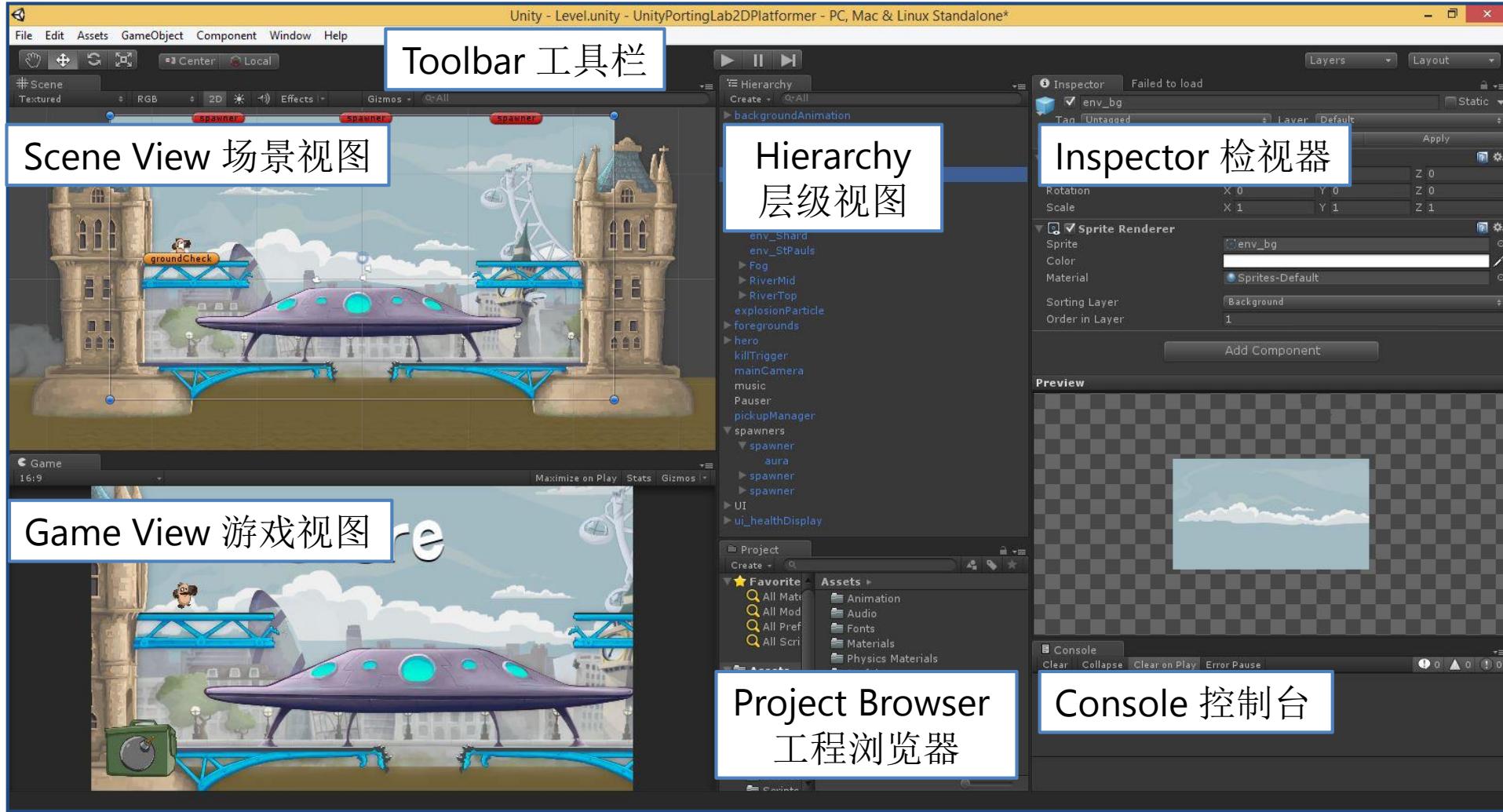
and more...

## 02 | 创建Unity游戏



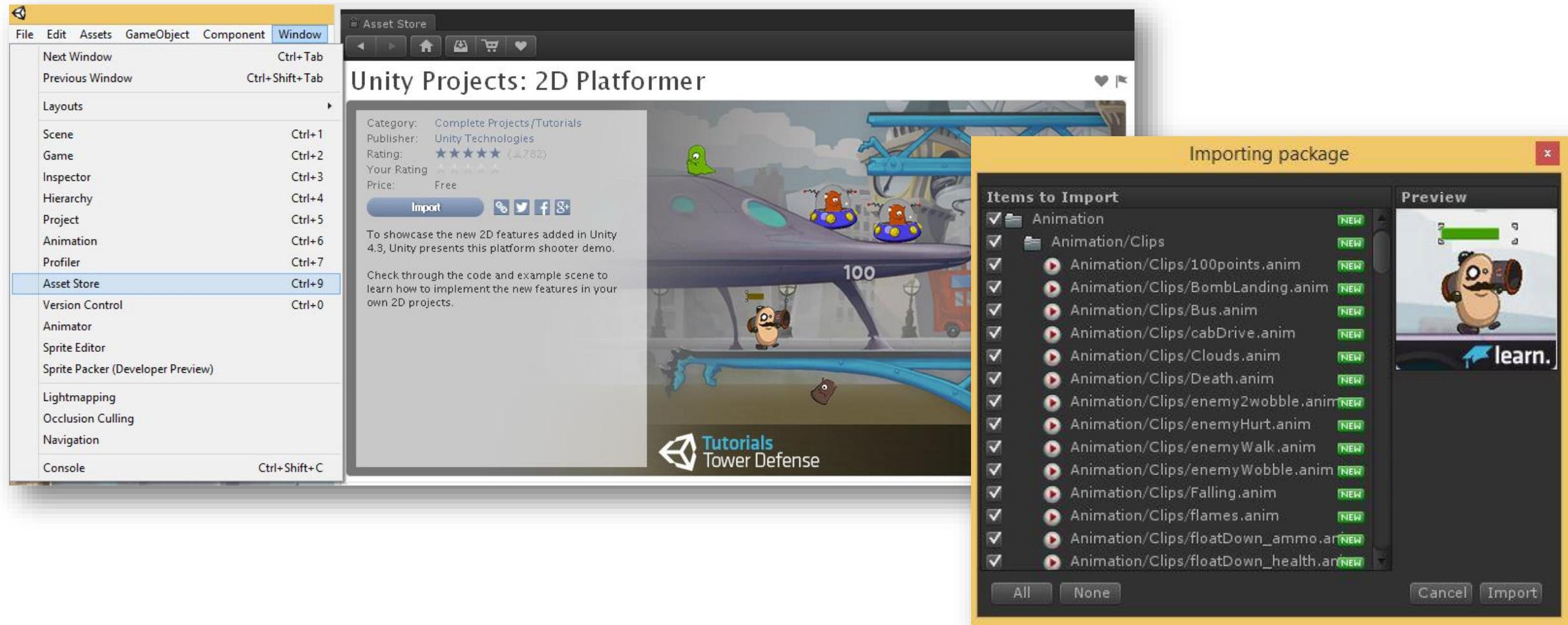
# Unity3D 编辑

- [http://edu.china.unity3d.com/learning\\_document/getData?file=/Manual/LearningtheInterface.html](http://edu.china.unity3d.com/learning_document/getData?file=/Manual/LearningtheInterface.html)



# Asset Store 资源商店

## 下载学习课程

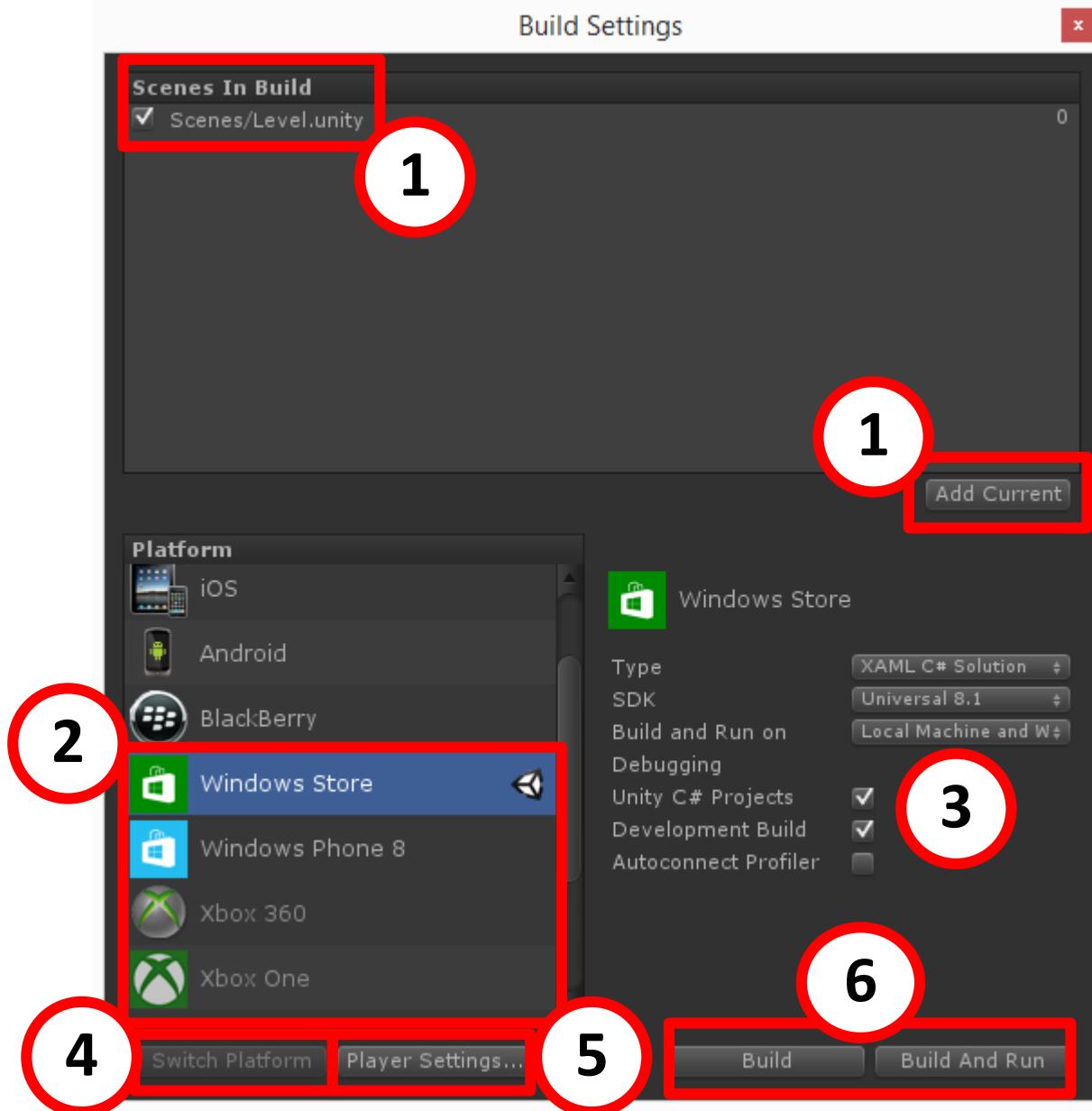


# DEMO

---

创建Unity游戏

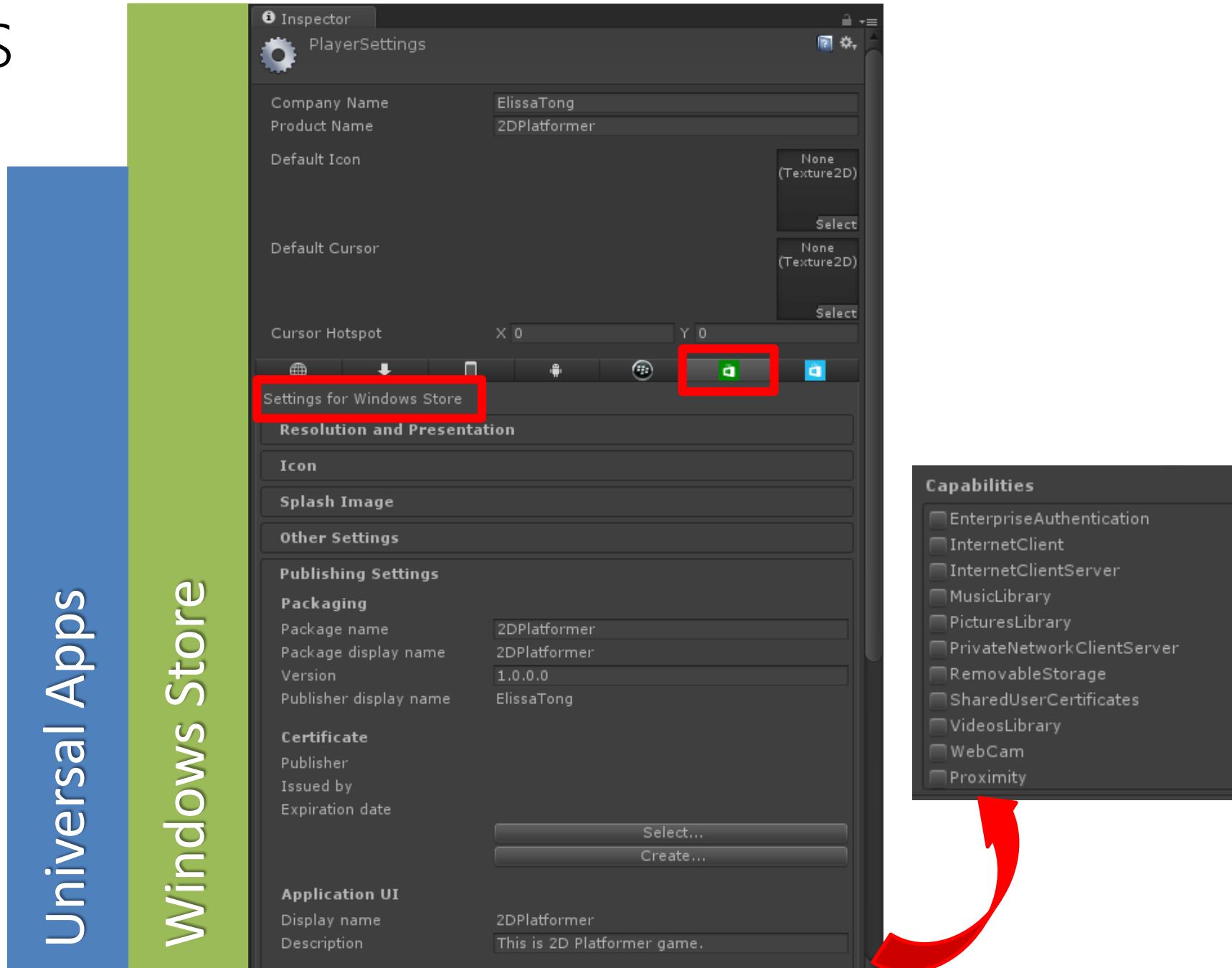
# 构建设备



1. 选择你构建的Scene (关卡 )
2. 选择平台
3. 选择平台设置
4. 点击Switch Platform
5. (Optional)点击Player Settings
6. 点击Build或者Build And Run

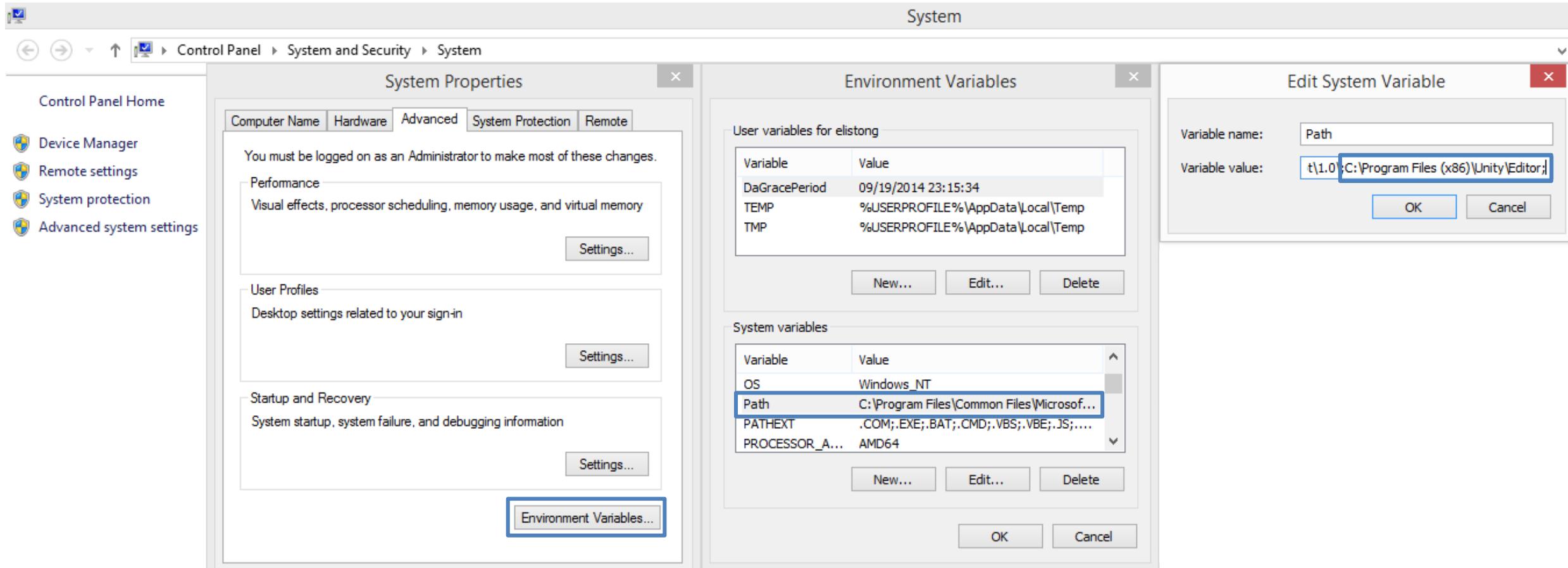
# Player Settings

Player Settings的设置在  
Visual Studio Solution的  
Package.appxmanifest



# 自动构建

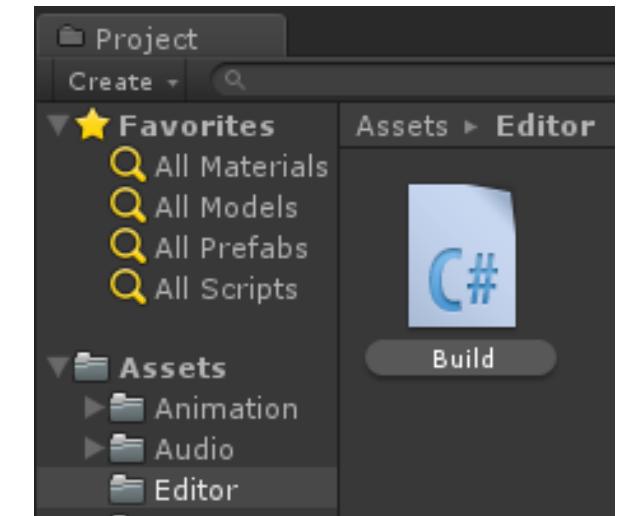
## 第1步：在环境变量的路径，加Unity.exe



# Unity的构建编辑脚本

## 第2步：加一下的脚本到Unity的Assets/Editor文件夹

```
public class CommandBuild
{
    [MenuItem("Build/Universal Apps 8.1 %U")]
    public static void BuildUniversalApps()
    {
        ArrayList sceneList = new ArrayList();
        foreach (var scene in EditorBuildSettings.scenes)
        {
            sceneList.Add(scene.path);
        }
        string[] levels = (string[])sceneList.ToArray(typeof(string));
        BuildPipeline.BuildPlayer(    levels,
                                    "Builds/UniversalApps",
                                    BuildTarget.MetroPlayer,
                                    BuildOptions.ShowBuiltPlayer);
    }
}
```

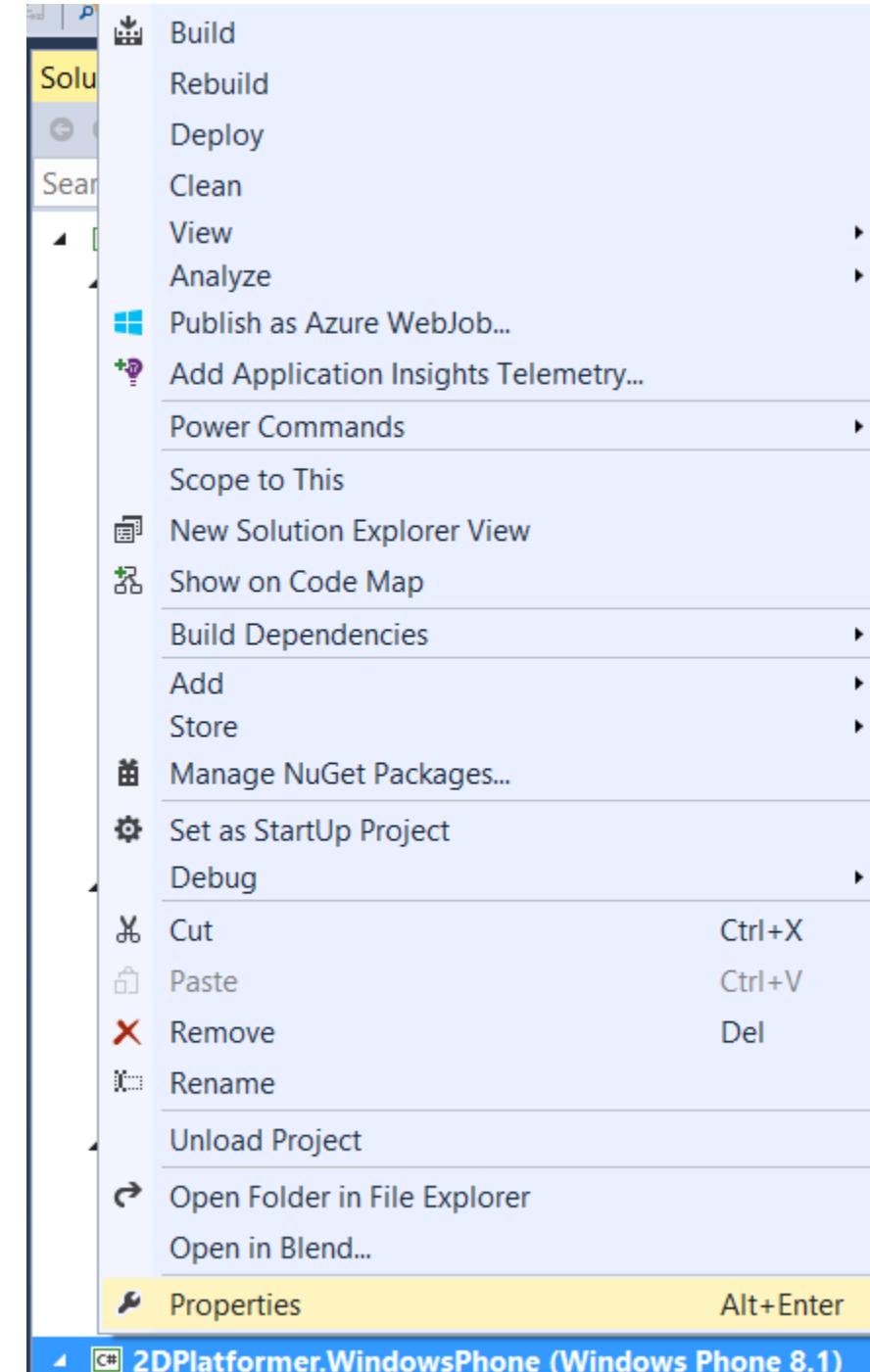
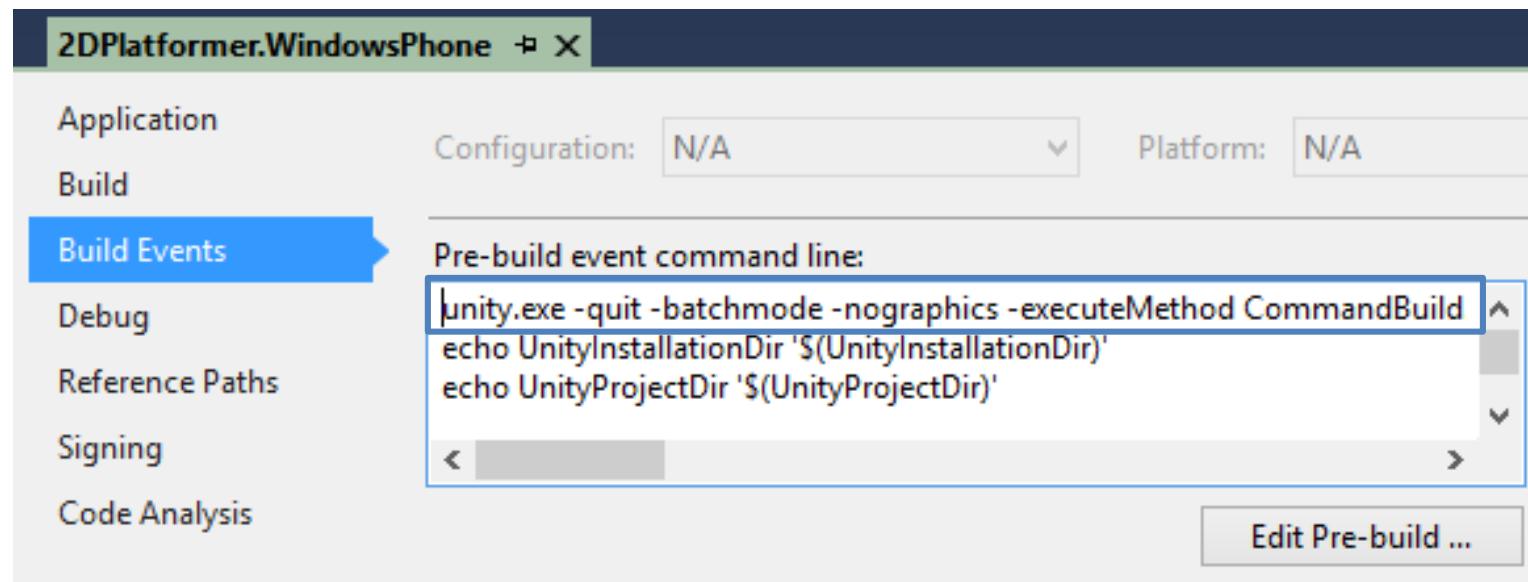


# Visual Studio构建设置

## 第3步：加构建的指令到

### Windows Phone Solution Properties

```
unity.exe -quit -batchmode -nographics -executeMethod  
CommandBuild.BuildUniversalApps -projectPath  
C:\Users\mycomputername\Documents\Development\Unity\P  
rojects\2DPlatformer
```



# 创建Unity游戏

Unity编辑

Asset Store

GameObject

Component

Prefabs

Build  
Settings

Player  
Settings

Visual Studio  
构建及部署

Loading  
Symbols

自动构建

and more...

## 03 | 硬件更改

# 多种分辨率

- 用模拟器测试每一个分辨率：WVGA, WXGA, 720p, 1080p
- <http://msdn.microsoft.com/zh-cn/library/windows/apps/xaml/hh465349.aspx>
- [http://msdn.microsoft.com/zh-cn/library/windows/apps/jj206974\(v=vs.105\).aspx](http://msdn.microsoft.com/zh-cn/library/windows/apps/jj206974(v=vs.105).aspx)

WVGA	WXGA	720p	1080p	全屏屏幕大小有效像素分辨率	设备描述
				384x640	4.5" 手机 15:9 纵横比
480 x 800 15:9	768 x 1280 15:9	720 x 1280 16:9	1080 x 1920 16:9	400x711	4.7" 手机 16:9 纵横比
450x800				491x873	5.5" 手机 16:9 纵横比
					6" 手机 16:9 纵横比

# 多种分辨率

## 手机宽度和高度是什么？

<http://blogs.windows.com/buildingapps/2013/11/22/taking-advantage-of-large-screen-windows-phones/>

```
string GetBasicScreenInfo()
{
    var width = App.Current.Host.Content.ActualWidth;
    var height = App.Current.Host.Content.ActualHeight;
    var scaleFactor = (double)App.Current.Host.Content.ScaleFactor / 100d;
    return String.Format("{0} x {1}; {2:0.0} scale factor", width, height, scaleFactor);
}

string GetExtendedScreenInfo()
{
    object temp;
    if (!DeviceExtendedProperties.TryGetValue("PhysicalScreenResolution", out temp)) return "not available";

    var resolution = (Size)temp;
    if (!DeviceExtendedProperties.TryGetValue("RawDpiX", out temp) || (double)temp == 0d) return "not available";

    var dpi = (double)temp;
    var screenDiagonal = Math.Sqrt(Math.Pow(resolution.Width / dpi, 2) + Math.Pow(resolution.Height / dpi, 2));
    var width = App.Current.Host.Content.ActualWidth;

    return String.Format("{0} x {1}; {2:0.0#} raw scale; {3:0.0}",
        resolution.Width, resolution.Height, resolution.Width / width, screenDiagonal);
}
```



# Windows 8.x: 注意屏幕以缩窄

- Windows 8.x游戏和应用可以改屏幕的大小
- 缩窄的屏幕从500 pixels开始

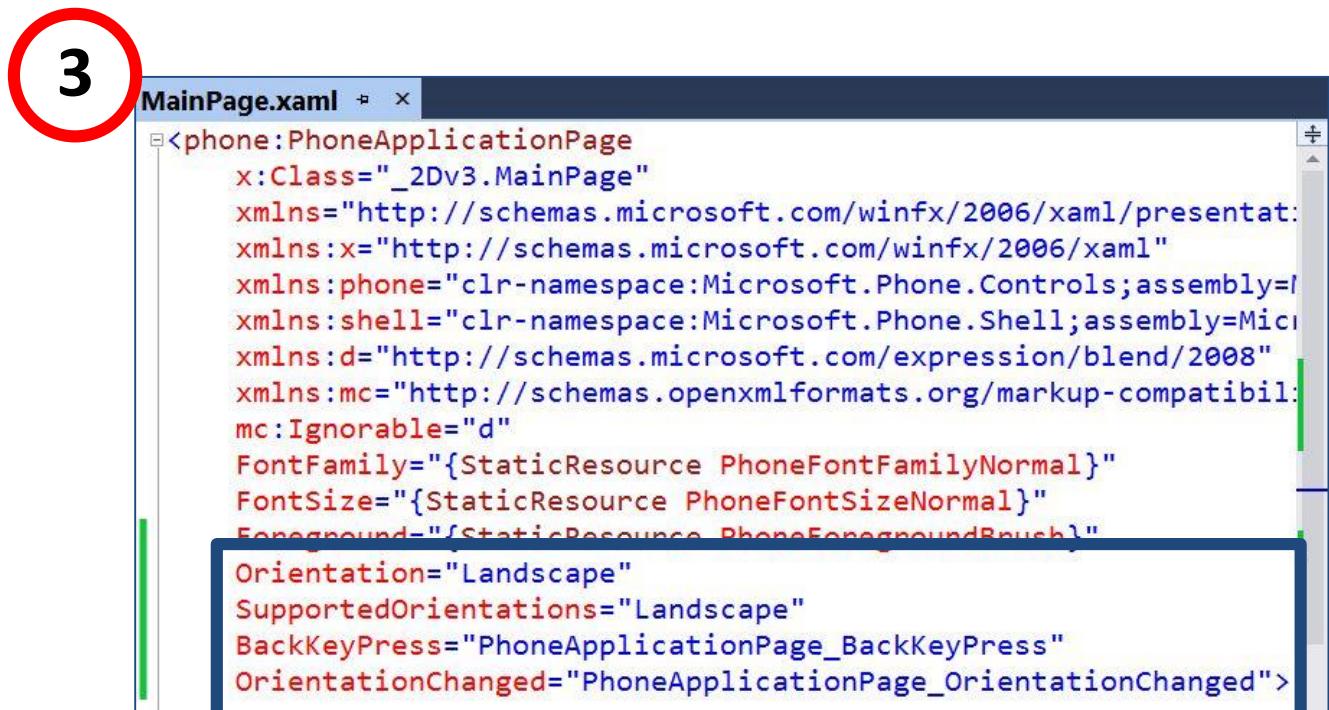
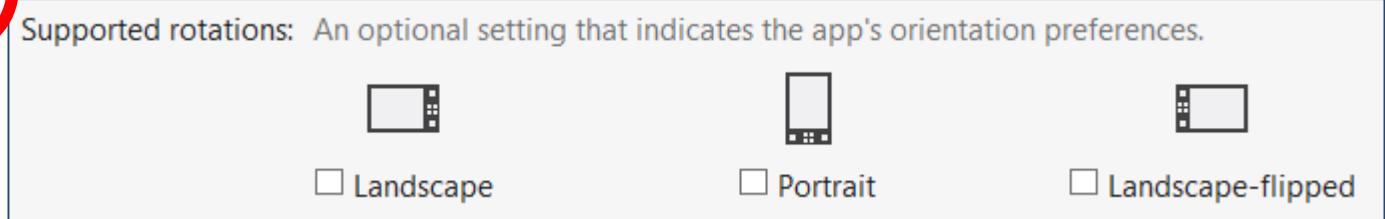
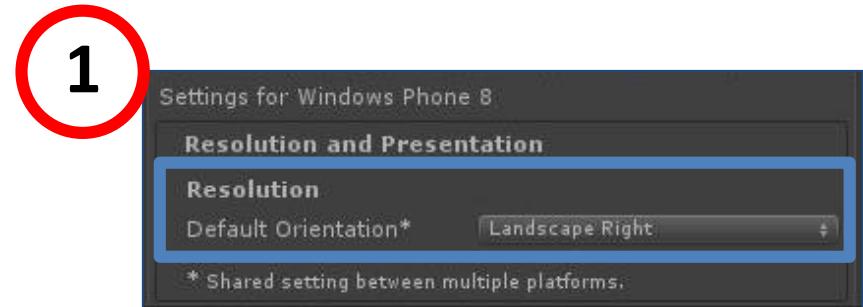
```
UnityEngine.WSA.Application.windowSizeChanged += WindowSizeChanged;  
  
public static void WindowSizeChanged(int width, int height)  
{  
    if(width <= 500)  
    {  
        GameController.SP.paused();  
    }  
    else  
    {  
        GameController.SP.unpaused();  
    }  
}
```

<http://msdn.microsoft.com/zh-cn/library/windows/apps/xaml/hh465371.aspx>

# 方向 Orientation

1. Windows Phone 8 Unity Player Settings
2. Package.appxmanifest
3. MainPage.xaml
4. Unity API 脚本

```
Screen.orientation =  
ScreenOrientation.AutoRotation;
```



# 内存

设备	内存	Windows Phone 8.0 * Native最低限制 / Silverlight最低限制 / 最大限制	Windows Phone 8.1
低内存设备WVGA	512 MB	150 MB / 150 MB / 180 MB	185 MB
720p设备	1 GB	150 MB / 300 MB / 380 MB	390 MB
新4.5" WXGA设备	2 GB	150 MB / 450 MB / 780 MB	825 MB

## Memory-related manifest entries

1

WP8.0 Capabilities ID\_FUNC\_EXTENDED\_MEM 为设备保留更高的内存分配

2

WP8.0 Requirements ID\_REQ\_MEMORY\_300 低内存的手机，则选择完全退出。应用不会出现在低内存手机的WP商店中。

3

WP8.1 Prerequisites xmlns:m3="http://schemas.microsoft.com/appx/2014/manifest" m3:MinDeviceMemory 限制最低内存的手机 (可选512 MB或者1 GB)

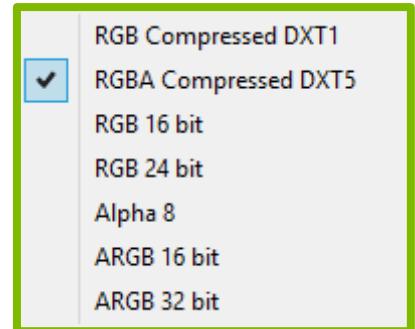
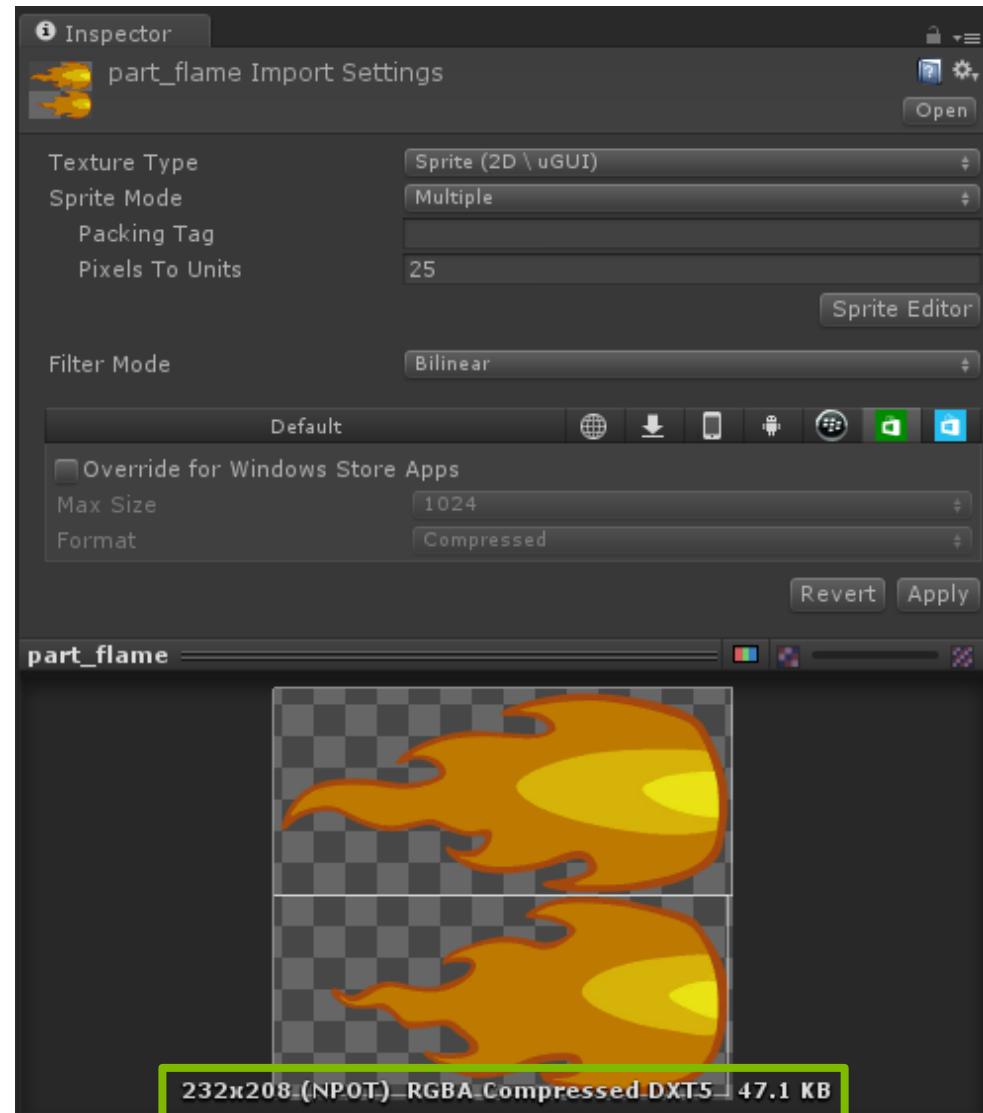
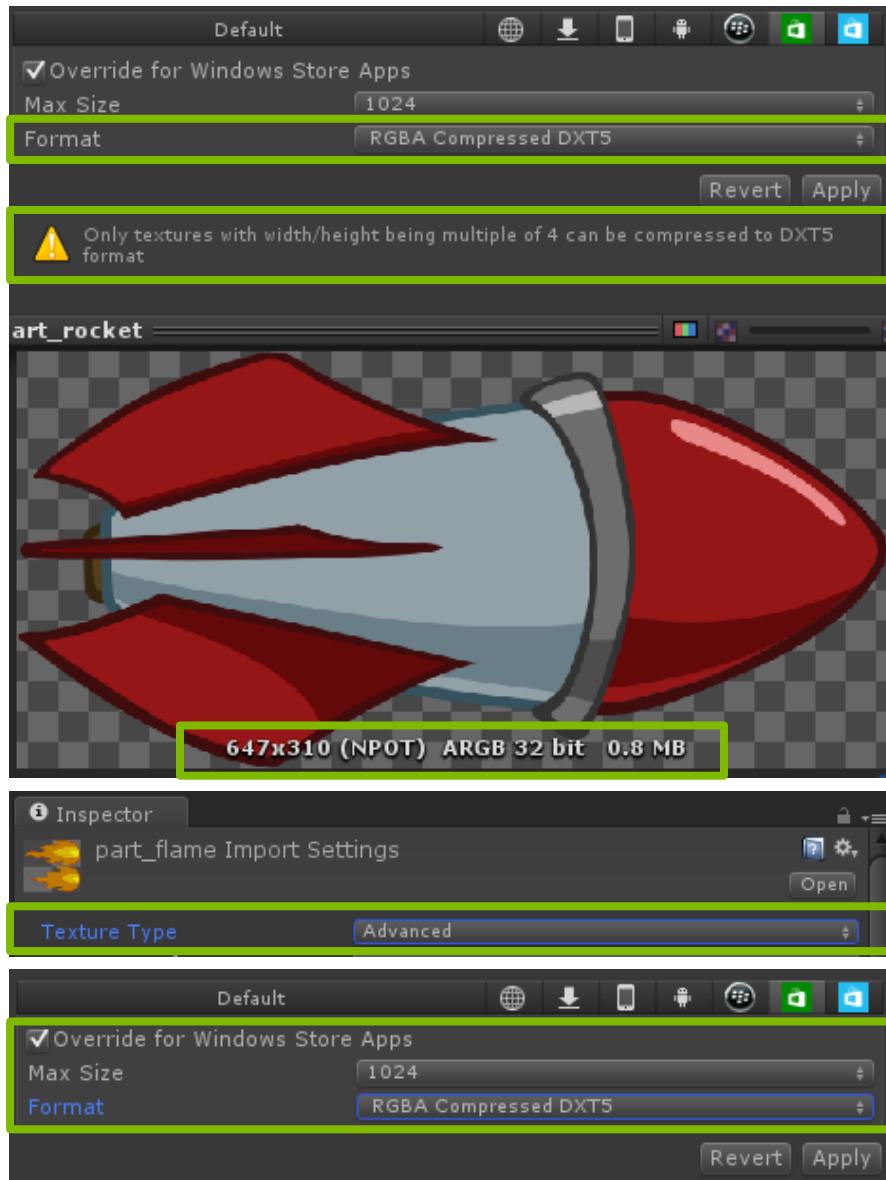
App memory limits for Windows Phone 8

[http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj681682\(v=vs.105\).aspx/html](http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj681682(v=vs.105).aspx/html)

# 内存测试代码

```
// WP8.0.  
ulong committedLimit = Windows.Phone.System.Memory.MemoryManager.ProcessCommittedLimit;  
ulong committedBytes = Windows.Phone.System.Memory.MemoryManager.ProcessCommittedBytes;  
// WP8.1.  
ulong usageLimit = Windows.System.MemoryManager.AppMemoryUsageLimit;  
ulong currentUsage = Windows.System.MemoryManager.AppMemoryUsage;  
MemoryManager.AppMemoryUsageIncreased += OnAppMemoryUsageIncreased;  
MemoryManager.AppMemoryUsageDecreased += OnAppMemoryUsageDecreased;  
  
private void OnAppMemoryUsageIncreased(object sender, object e)  
{  
    switch (MemoryManager.AppMemoryUsageLevel)  
    {  
        case AppMemoryUsageLevel.High:  
            break;  
        case AppMemoryUsageLevel.Medium:  
            break;  
        case AppMemoryUsageLevel.Low:  
            break;  
    }  
}
```

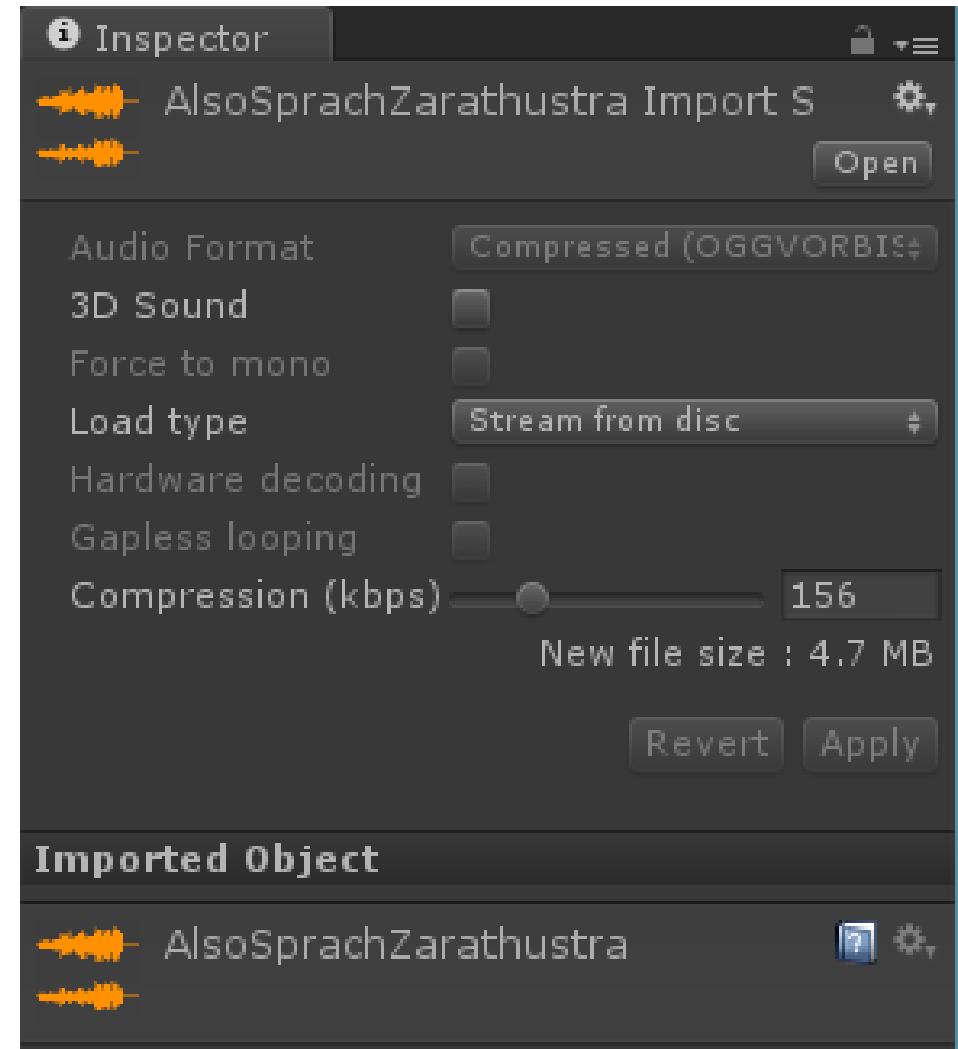
# 内存：纹理压缩



<http://docs.unity3d.com/Manual/class-TextureImporter.html>

# 内存：音效

- Load type = Stream from disc
- 不要用Compressed in memory



# 内存资料

How to disable features in apps for lower-memory phones for WP8

[http://msdn.microsoft.com/en-US/library/windows/apps/hh855083\(v=vs.105\).aspx/html](http://msdn.microsoft.com/en-US/library/windows/apps/hh855083(v=vs.105).aspx/html)

Managing Resource Constraints on Windows Phone

<http://channel9.msdn.com/Events/Build/2014/3-542>

开发面向低内存手机的 Windows Phone 8 应用

[http://msdn.microsoft.com/zh-cn/library/windows/apps/hh855081\(v=vs.105\).aspx/html](http://msdn.microsoft.com/zh-cn/library/windows/apps/hh855081(v=vs.105).aspx/html)

Windows Phone 8的应用内存限制

[http://msdn.microsoft.com/zh-cn/library/windows/apps/jj681682\(v=vs.105\).aspx/html](http://msdn.microsoft.com/zh-cn/library/windows/apps/jj681682(v=vs.105).aspx/html)

Make sure to profile your phone app's memory usage

<http://forum.unity3d.com/threads/202952-tips-and-tricks-make-sure-to-profile-your-phone-app-s-memory-usage!>

# 回退按钮

## Windows Phone的技术认证要求

[http://msdn.microsoft.com/zh-cn/library/windows/apps/hh184840\(v=vs.105\).aspx](http://msdn.microsoft.com/zh-cn/library/windows/apps/hh184840(v=vs.105).aspx)

## WP8.0: MainPage.xaml.cs

```
private void PhoneApplicationPage_BackKeyPress(object s, CancelEventArgs e)
{
    e.Cancel = UnityApp.BackButtonPressed();
    // TO DO. .
}
```

## Unity 脚本

```
if (Input.GetKeyDown(KeyCode.Escape)) {
{
    Debug.Log("ESCAPE Input key down");
    Application.Quit();
}
```

# 角色动作

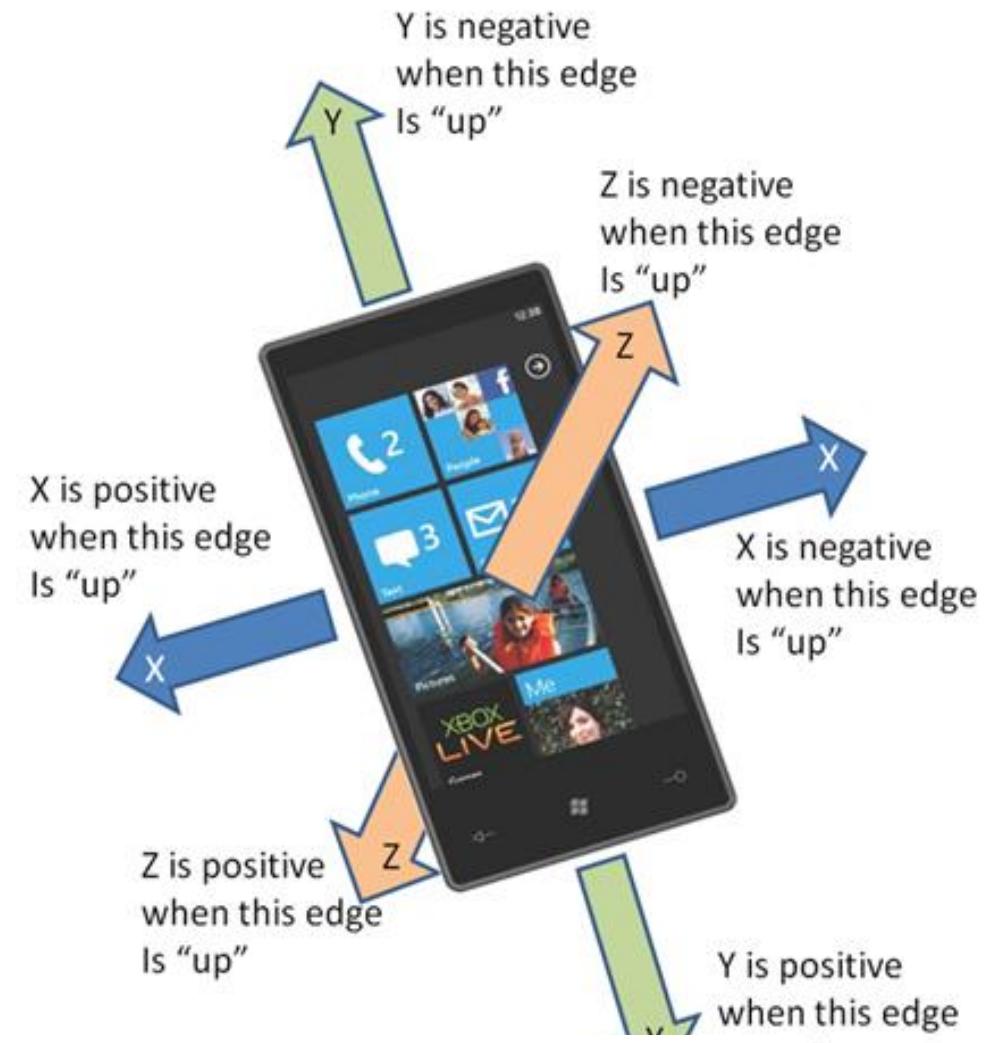
## Unity 脚本

### Accelerometer 加速机

```
if(Application.platform == RuntimePlatform.WP8Player)
{
    MoveAmount = (PlayerSpeed * Input.acceleration.x) * Time.deltaTime;
    transform.Translate(Vector3.right * MoveAmount);
    MoveAmount = (PlayerSpeed * ((-Input.acceleration.z + 0.5f)*2)) * Time.deltaTime;
    transform.Translate(Vector3.forward * MoveAmount);
}
```

### Touch 触摸

```
if (Application.platform == RuntimePlatform.WP8Player)
{
    if (grounded && Input.touchCount == 1 && Input.GetTouch(0).phase == TouchPhase.Ended)
        jump = true;
}
```



# 键盘

- OnGUI()

- 控制键盘什么时候打开及关掉

```
mKeyboard = TouchScreenKeyboard.Open("", TouchScreenKeyboardType.Default,  
true, false, false, false);
```

- TextField UI能支持键盘

```
mInputUserName = GUI.TextField(  
    new Rect(xTextField, INPUT_OFFSET, TEXTFIELD_WIDTH, INPUT_HEIGHT),  
    mInputUserName, 16, mGUILayoutTextField);
```

- 不能用: mKeyboard.text, mKeyboard.area

```
void Update ():  
if (mKeyboard.active && !string.IsNullOrEmpty(Input.inputString))  
    mInputUserName += Input.inputString;
```

```
void OnGUI ():  
    GUI.Label(mRectTextField, mInputUserName, mGUILayoutTextField);
```

# DEMO

---

硬件更改

# 硬件更改

多种分辨率

方向

内存

纹理压缩

回退按钮

加速机

触摸

and more...

## 04 | API了解



# .NET Framework

Mono .NET (Unity 3.5 .NET)

<https://github.com/mono/mono/tree/master/mcs/class/System/System.Net>

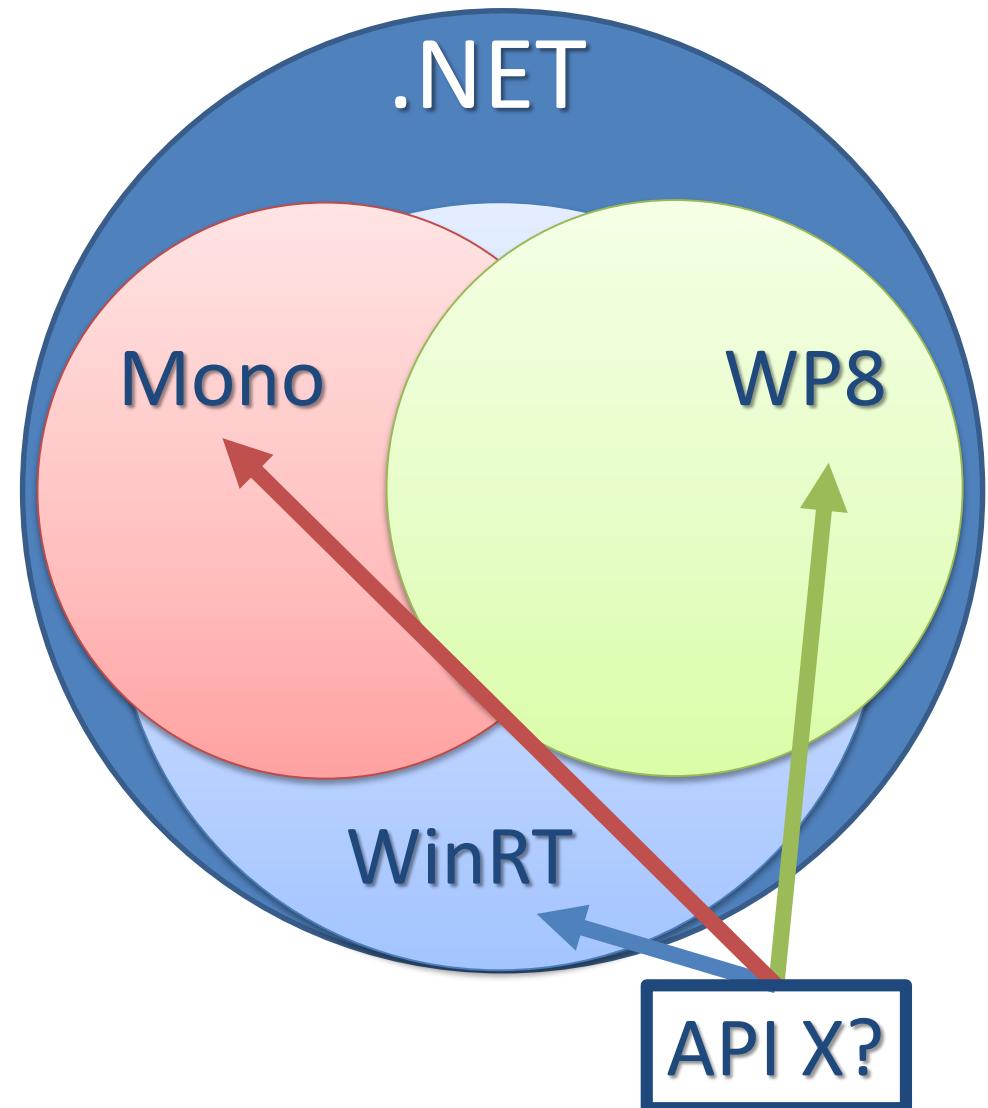
.NET API for Windows Phone 8.0

[http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj207211\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj207211(v=vs.105).aspx)

.NET API for Windows Runtime

(Windows 8.1 & Windows Phone 8.1)

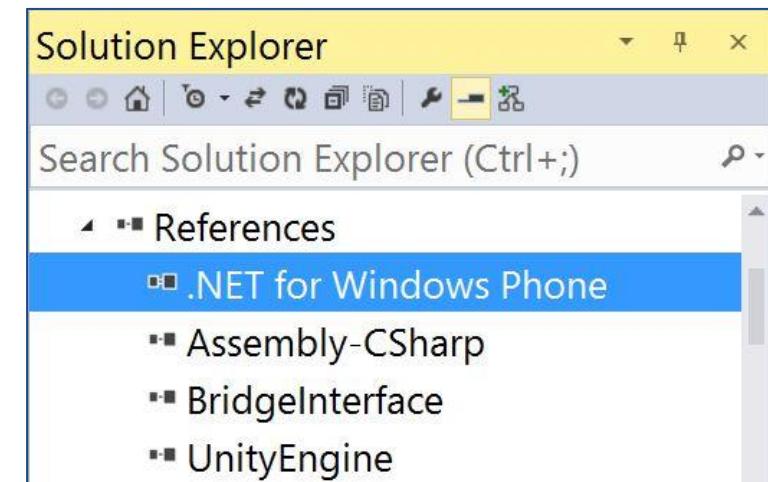
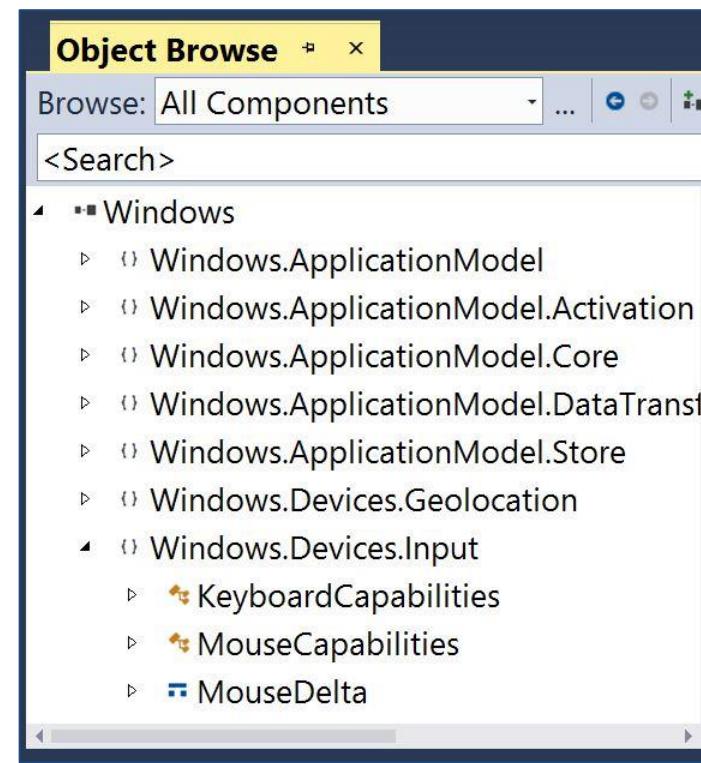
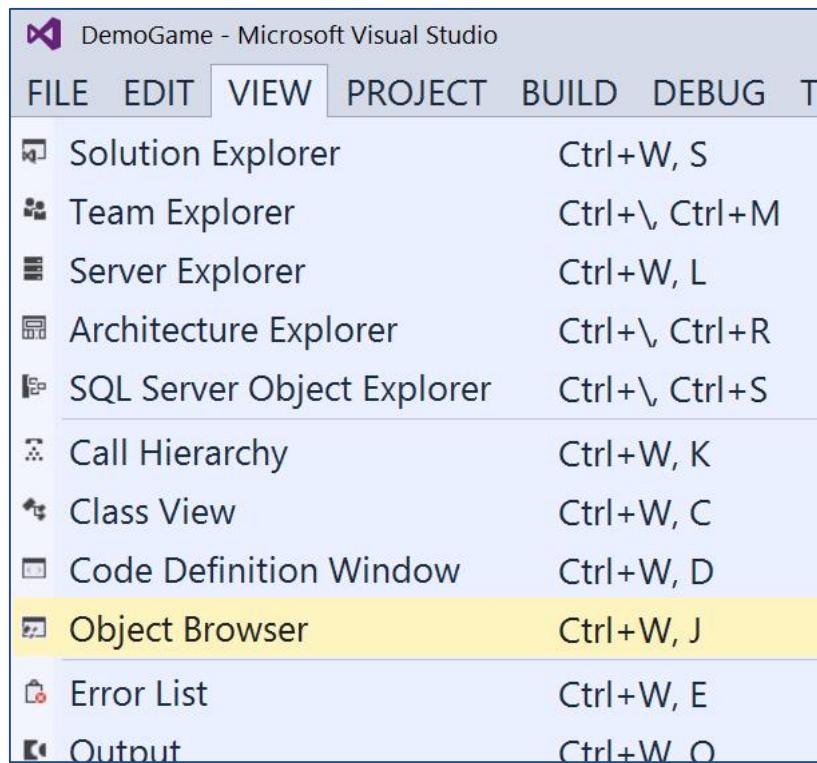
<http://msdn.microsoft.com/en-us/library/windows/apps/bb211369.aspx>



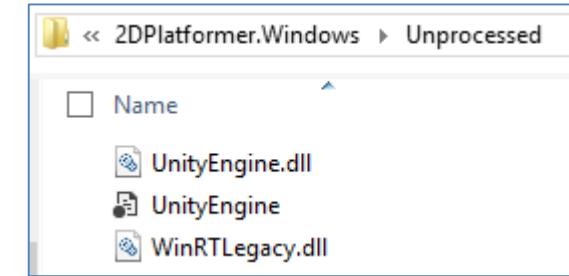
# .NET Framework

## 怎么知道哪一个.NET版本能支持什么API？

- Visual Studio IDE → View → Object Browser
- 或者打开Solution Explorer里面的References



# 缺少哪些API？



命名空间	类例子	解决方案
System.Collections	Hashtable, ArrayList, List	Use WinRTLegacy (from Unity)
System.IO	File, StreamReader, TextReader	Write using Windows.Storage
System.Xml	XmlDocument,XmlElement	Use WinRTLegacy for basic coverage
System.Reflection		Write using equivalent WinRT APIs
System.Security.Cryptography	SHA1, TripleDES	Use WinRTLegacy for basic coverage
System.Net System.Net.Sockets	Socket, NetworkStream, WebClient, HttpWebRequest	Use WinRT networking APIs: <a href="#">Windows.Networking.Sockets</a>
System.Threading	Thread	WinRT is async, use Task, use coroutines

# DEMO

---

了解.NET API

# 解决 API 错误

- 方案1: Unity插件

例子: Elissa's GitHub Unity Plugins:

<https://github.com/elissatong/UnityPortingLabs>

例子: 3rd Party GitHub Platformer Plugins:

<https://github.com/windowsgamesamples/UnityPorting>

- 方案2: Action (void delegate)

例子: WindowsGateway.cs

- 方案3: Global instances

例子: GameManager.cs

- 方案4: Proxy Interface

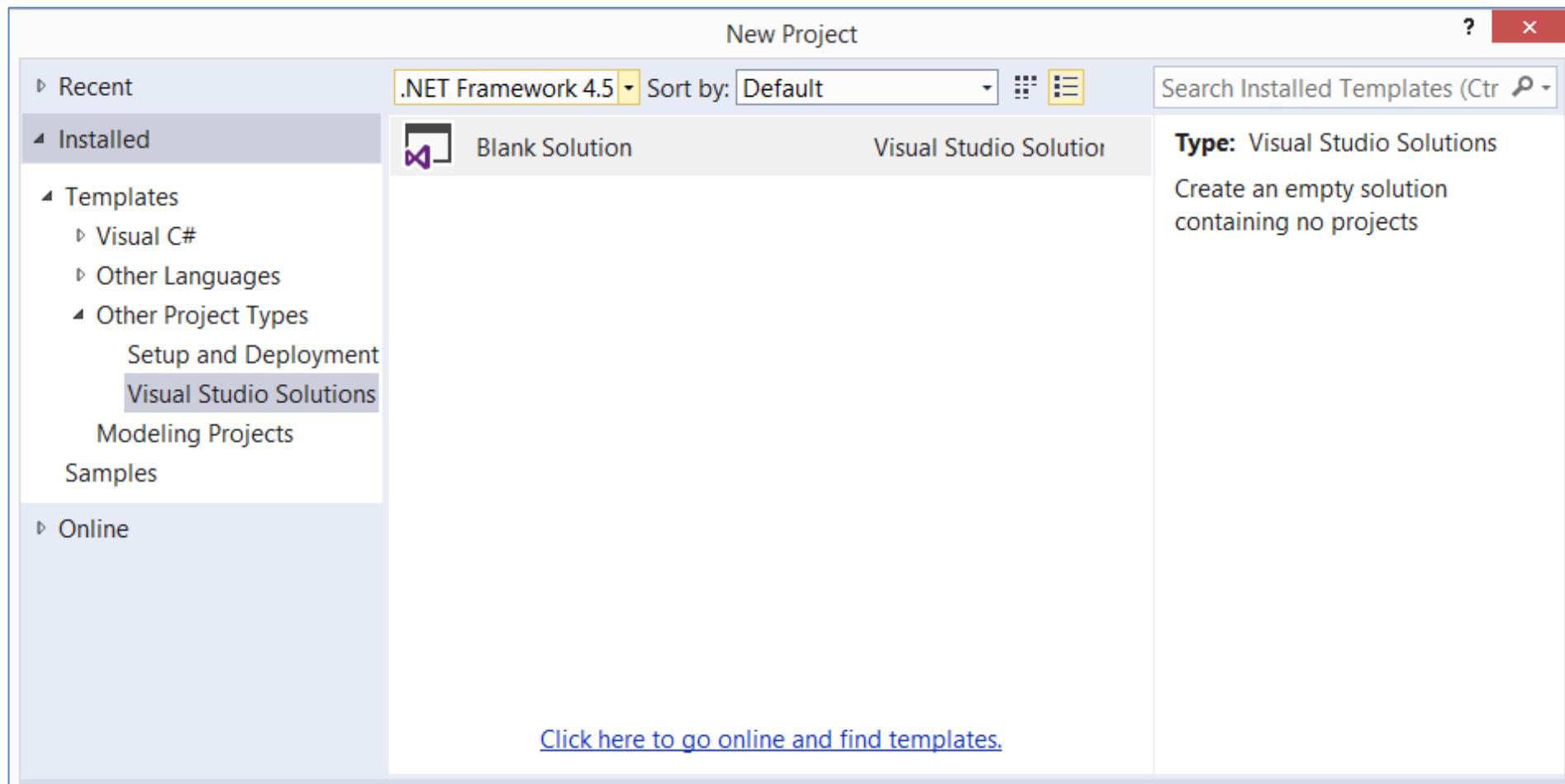
例子: Json.cs

# 平台#defines

#defines	平台支持
UNITY_WP8	Windows Phone 8.0 apps
UNITY_WP_8_1	Windows Phone 8.1 and Universal 8.1 apps on Windows Phone devices
UNITY_METRO	Windows Phone 8.1, Windows Store 8.0, Windows Store 8.1 and Universal 8.1 apps
UNITY_METRO_8_0	Windows Store 8.0 apps
UNITY_METRO_8_1	Windows Store 8.1 and Universal 8.1 apps running on Windows devices
UNITY_WINRT	Both Windows Phone and Windows Store apps, regardless of version
UNITY_WINRT_8_0	Windows Phone 8.0 and Windows Store 8.0 apps;
UNITY_WINRT_8_1	Windows Phone 8.1, Windows Store 8.1 and Universal apps for Windows and Windows Phone devices
NETFX_CORE	Windows Store 8.0, Windows Store 8.1, Windows Phone 8.1 and Universal 8.1 scripts that are compiled using Microsoft C# compiler.

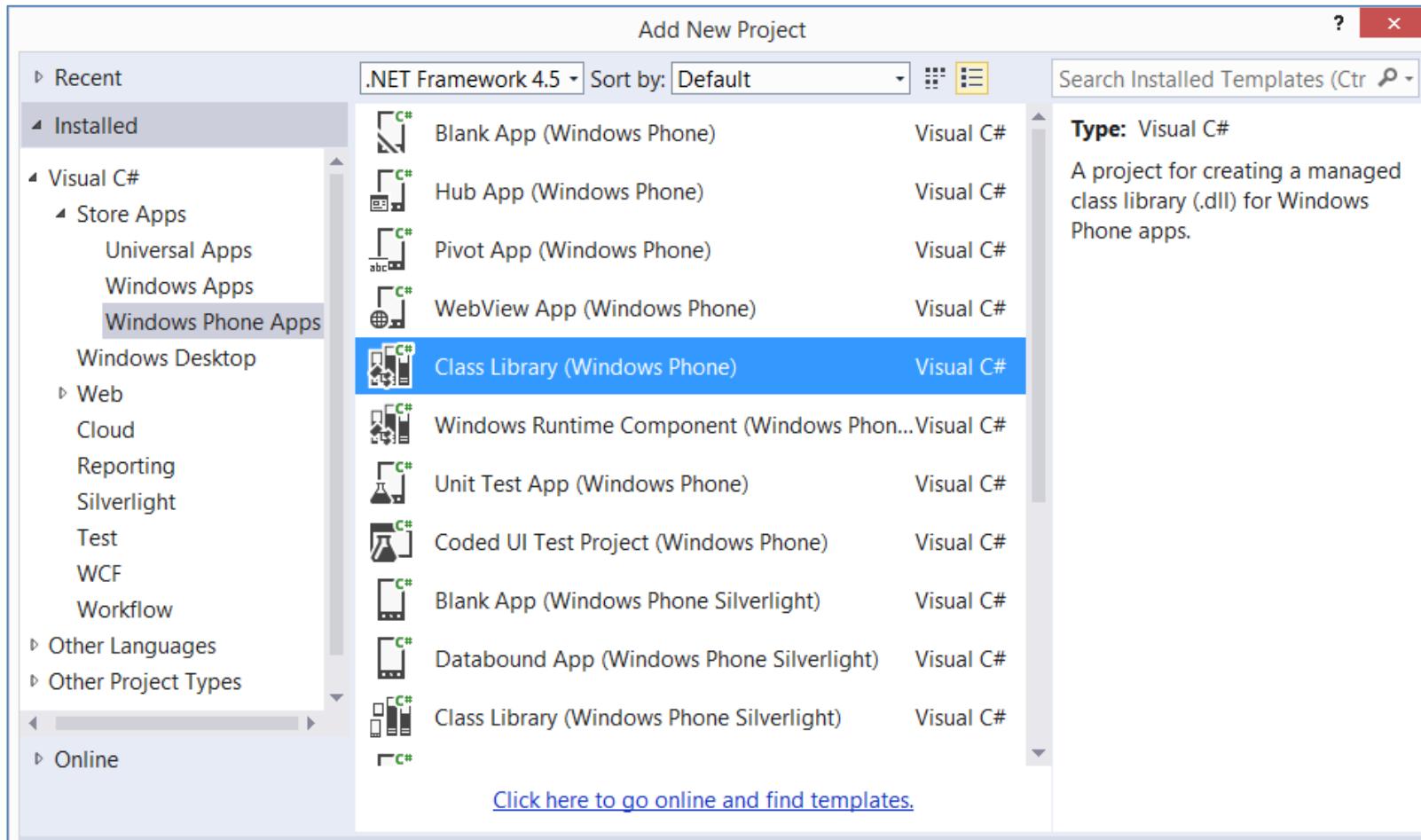
# Unity插件

## 第1步：创建Visual Studio blank solution



# Unity插件

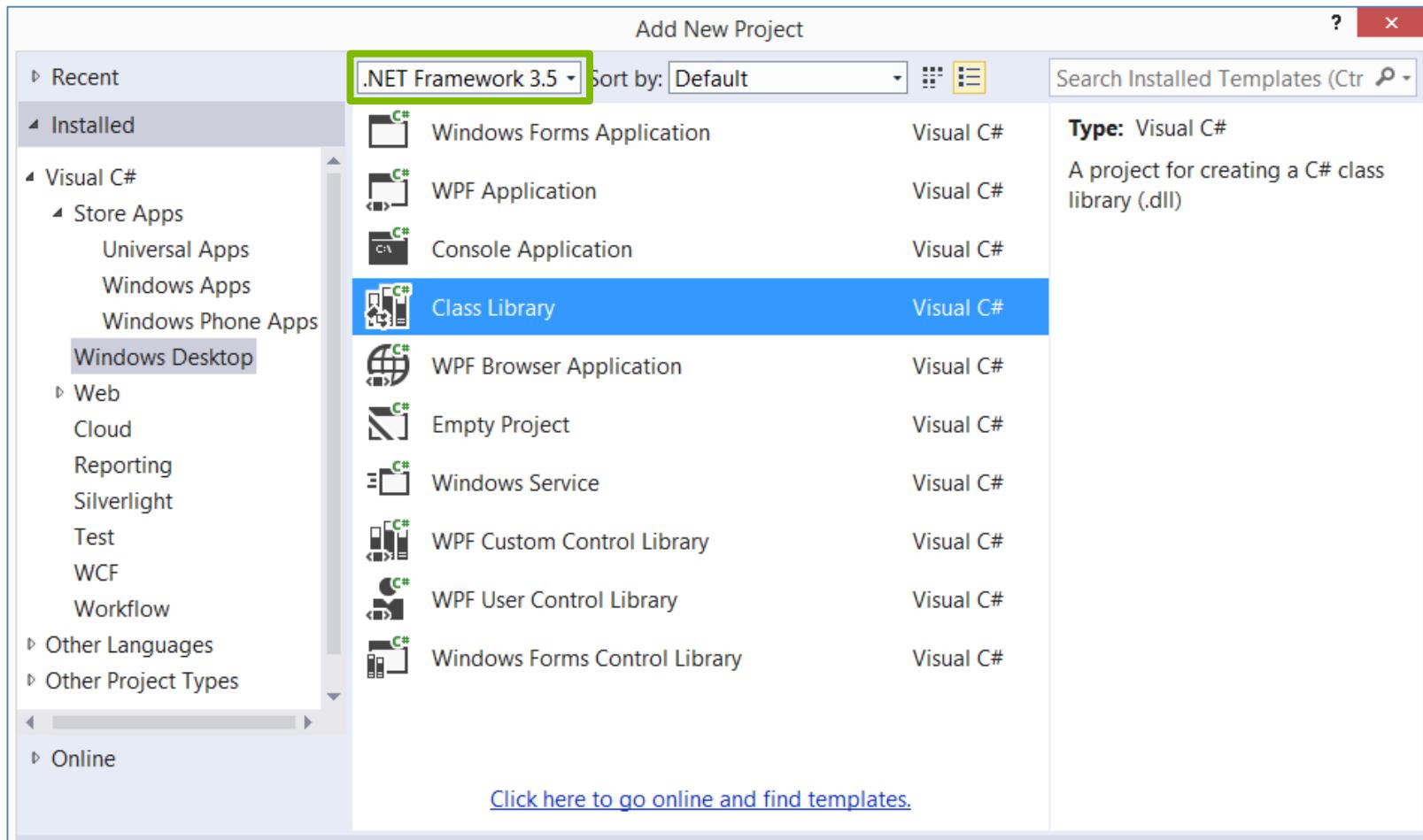
## 第2步：在Solution加一个新项目Class Library (Windows Phone)



# Unity插件

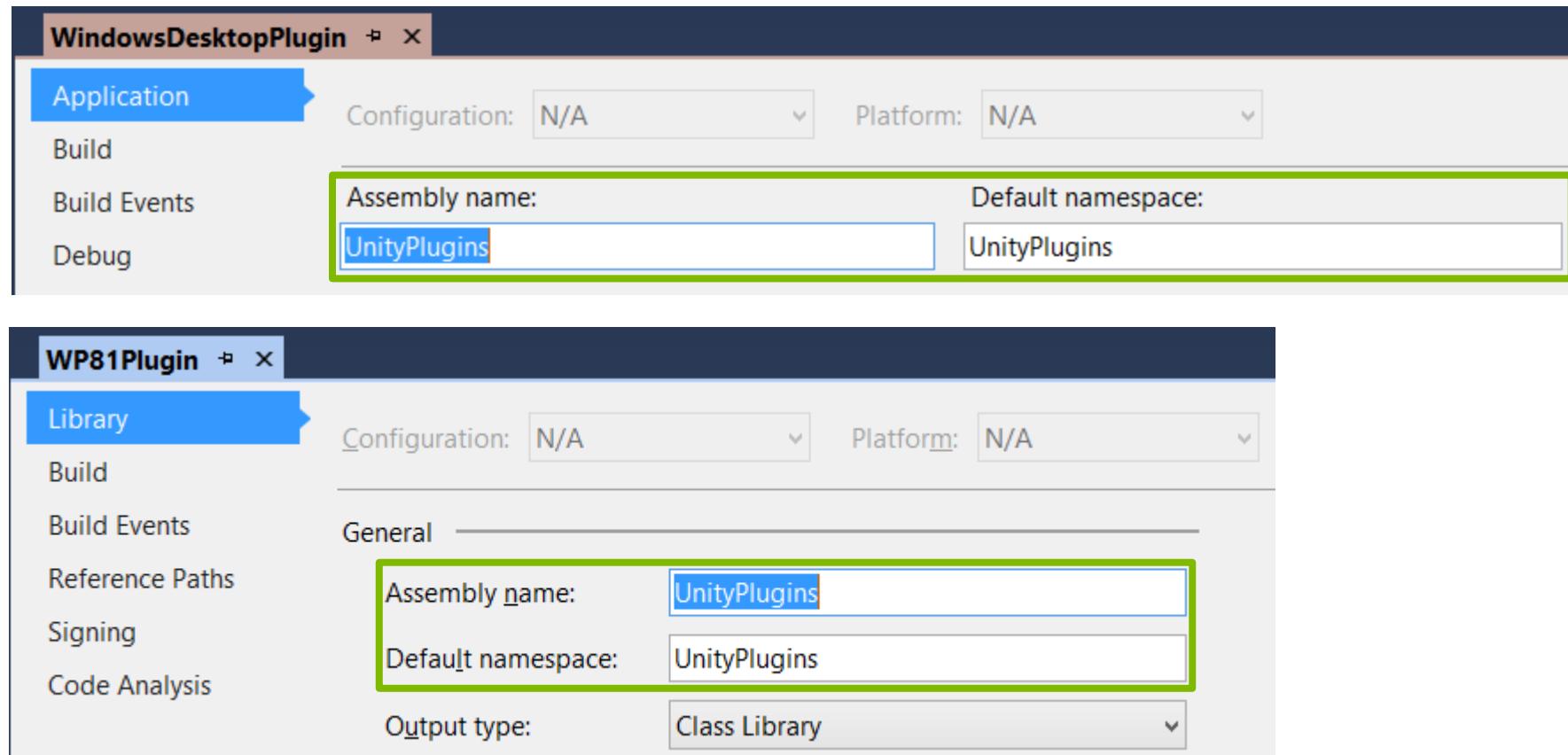
## 第3步：在Solution加一个新项目Windows Desktop: Class Library

Unity使用.NET 3.5  
所以一定改到  
.NET Framework 3.5



# Unity插件

第4步：在每一个新创建的平台Class Library打开Properties  
改Assembly name及Default namespace为同一个名字



# Unity插件

第5步：类里的代码要改Namespace。类,函数,属性的名字需要同名字。

```
namespace UnityPlugins
{
    public class Class1
    {
        public static string GetMemoryUsageLimit
        {
            get
            {
                return "Not supported";
            }
        }

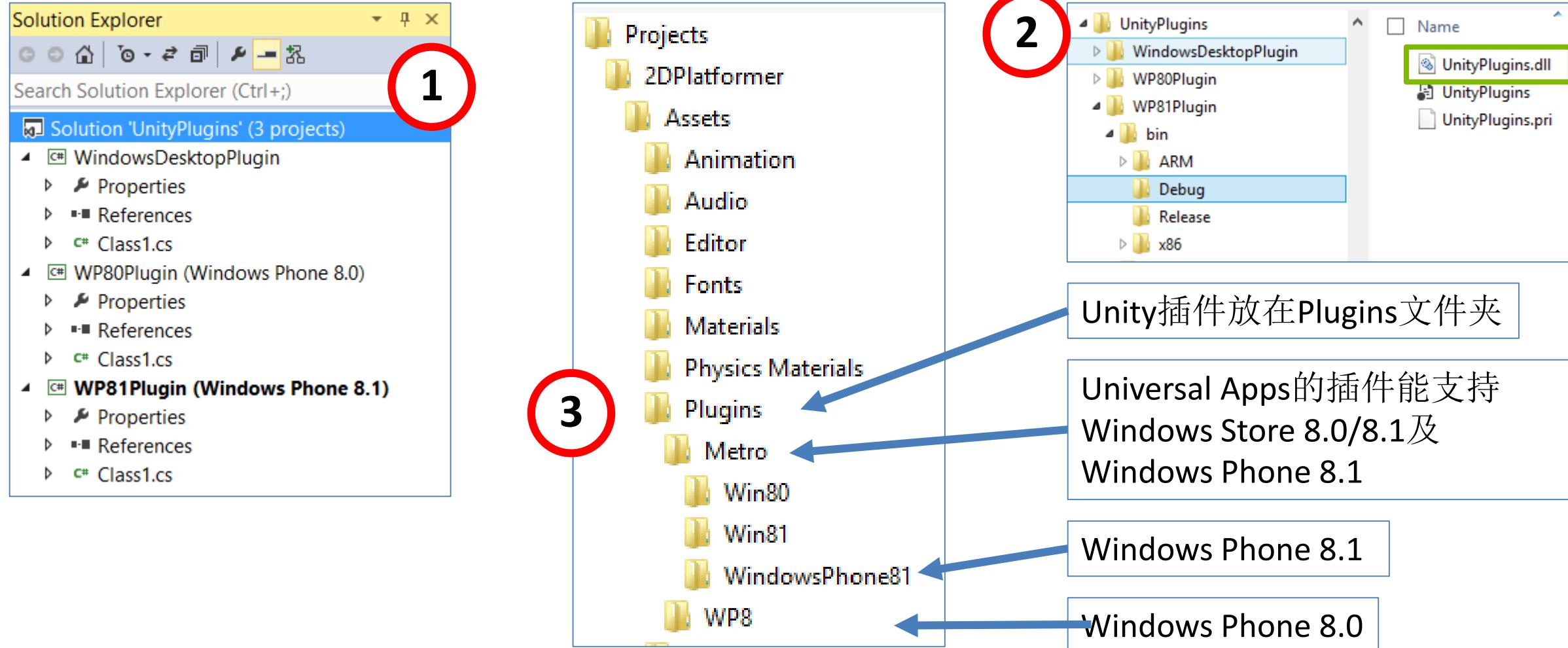
        public static string GetMemoryCurrentUsage
        {
            get
            {
                return "Not supported";
            }
        }
    }
}
```

```
namespace UnityPlugins
{
    public class Class1
    {
        public static string GetMemoryUsageLimit
        {
            get
            {
                // WP8.1.
                ulong usageLimit = Windows.System.MemoryManager.AppMemoryUsageLimit;
                ulong bytesToMB = usageLimit / (1024 * 1024);
                return bytesToMB.ToString() + " MB";
            }
        }

        public static string GetMemoryCurrentUsage
        {
            get
            {
                // WP8.1.
                ulong currentUsage = Windows.System.MemoryManager.AppMemoryUsage;
                ulong bytesToMB = currentUsage / (1024 * 1024);
                return bytesToMB.ToString() + " MB";
            }
        }
    }
}
```

# Unity插件

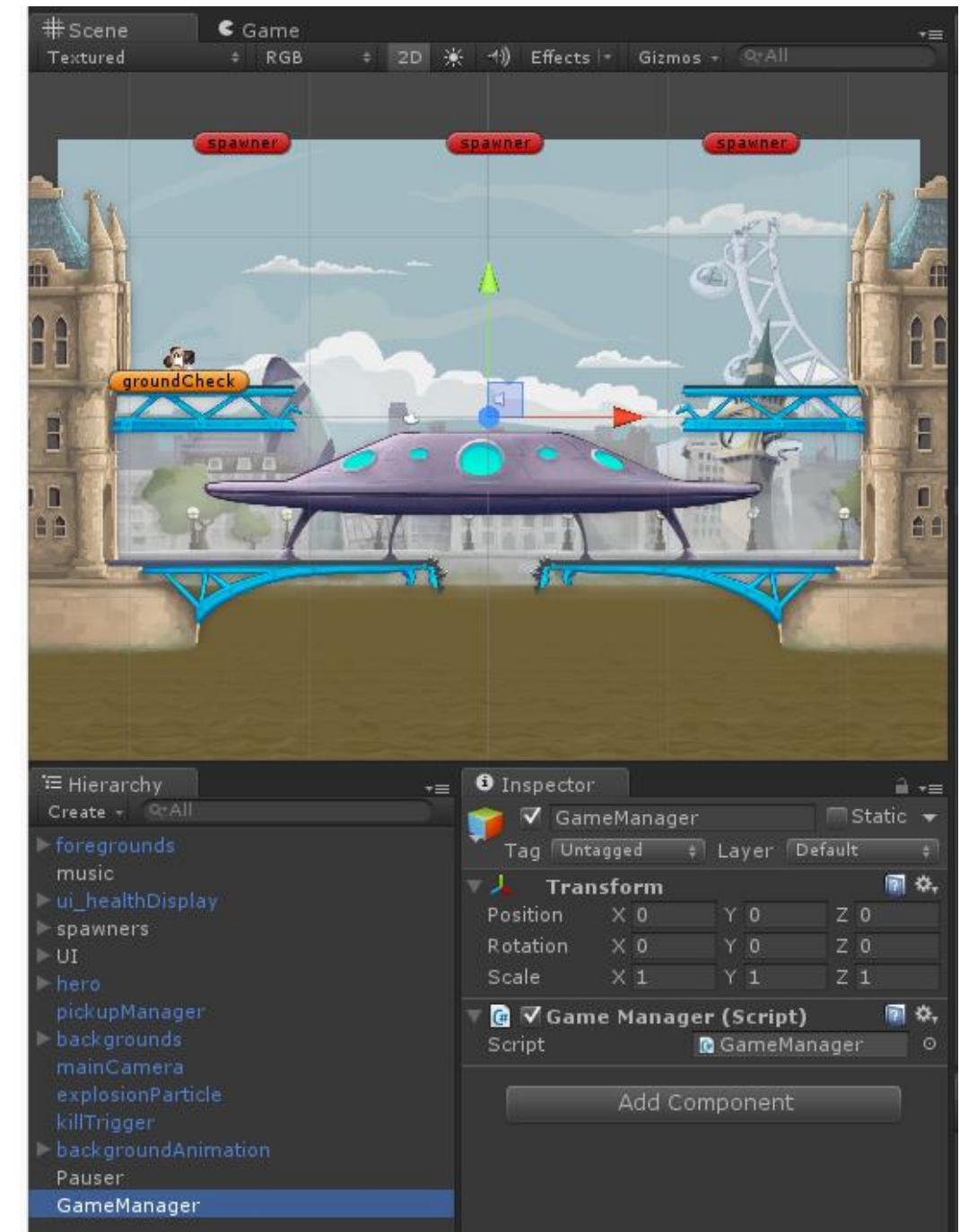
## 第6步：构建Solution后，拷贝.dll到Unity的Plugins文件夹



# Unity插件

第7步：要测试插件在Unity：

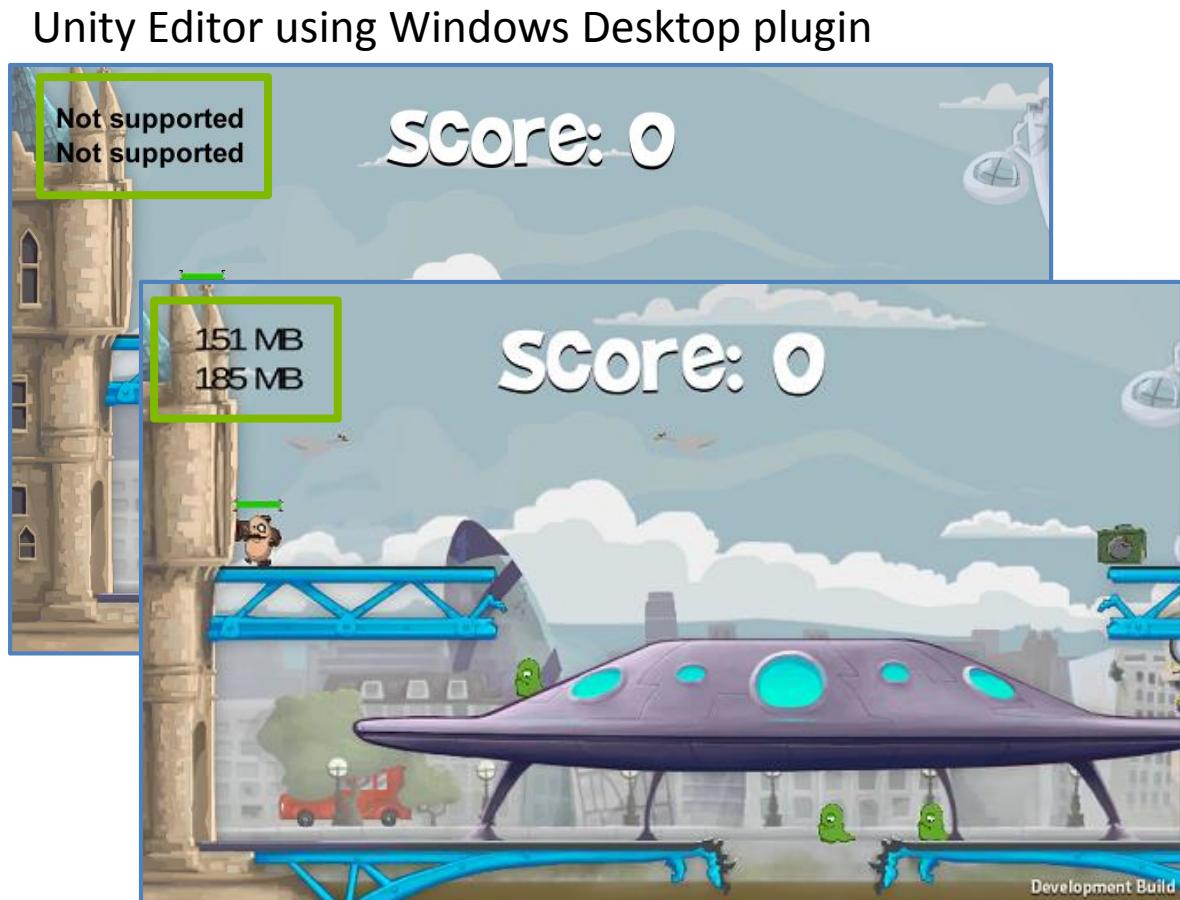
- 加一个GameObject到一个Scene里
- 加一个脚本到GameObject里
- 构建一个Universal Apps build



# Unity插件

第7步：打开Visual Studio，写Unity脚本就能有插件里的类，函数，等等。

```
public class GameManager : MonoBehaviour
{
    private GUIStyle mGUILayoutLabel;
    private void SetGuiLabelStyles()
    {
        mGUILayoutLabel = new GUIStyle();
        mGUILayoutLabel.fontSize = 24;
        mGUILayoutLabel.fontStyle = FontStyle.Bold;
        mGUILayoutLabel.alignment = TextAnchor.MiddleLeft;
    }
    void OnGUI()
    {
        GUI.Label(new Rect(40, 20, 250, 50),
            UnityPlugins.Class1.GetMemoryCurrentUsage(mGUILayoutLabel));
        GUI.Label(new Rect(40, 50, 250, 50),
            UnityPlugins.Class1.GetMemoryUsageLimit(mGUILayoutLabel));
    }
    void Start()
    {
        SetGuiLabelStyles();
    }
}
```



Windows Phone 8.1 Emulator

# 解决异步编程问题：C# async

- 例子：File IO的Unity插件

<http://msdn.microsoft.com/zh-cn/library/windows/apps/xaml/hh464924.aspx>

```
public static async Task<string> CreateFile(StorageFolder folder, string name, string path, string contents)
{
    folder = await navigateFrom(folder, path);
    StorageFile file = await folder.CreateFileAsync(name, CreationCollisionOption.ReplaceExisting);

    using(IRandomAccessStream stream = await file.OpenAsync(FileAccessMode.ReadWrite))
    {
        using (DataWriter writer = new DataWriter(stream))
        {
            writer.WriteString(contents);
            await writer.StoreAsync();
        }
    }
    return file.Path;
}
public static void CreateFile(string filename, string content, string path = "")
{
    var task = FileHandling.CreateFile(ApplicationData.Current.LocalFolder, filename, path, content);
    task.Wait();
}
```

# 测试插件

- How mobile is your .NET?
  - <http://scan.xamarin.com/>

Calls	Platform-specific method name	Namespace	Suggestions	Compatibility
+ 322	IList CollectionBase.List { get; }	System.Collections		   
+ 272	string Control.Name { set; }	System.Windows.Forms		   

Percent of code that can run unmodified on each platform



99%



99%



77%



73%



# 常用的插件

- JSON
  - JSON for .NET: \$20 <http://u3d.as/5q2>
- Protobuf
  - Protobuf-net: <http://code.google.com/p/protobuf-net/>
  - <http://purdyjotut.blogspot.com/2013/10/using-protobuf-in-unity3d.html>

# 其他资料

Windows Phone Plugins step by step guide (using C#)

<http://docs.unity3d.com/Manual/wp8-plugins-guide-csharp.html>

Windows Phone Plugins step by step guide (using C++)

<http://docs.unity3d.com/Manual/wp8-plugins-guide-cplusplus.html>

Unity Script Reference: Windows File API

<http://docs.unity3d.com/ScriptReference/Windows.File.html>

# 方案2: Action

在Unity脚本直接切换到Windows Phone Solution的代码和逻辑

第1步: 创建Unity脚本, WindowsGateway.cs

```
/// Windows specific and interop between Unity and Windows Store or Windows Phone 8
using UnityEngine; using System.Collections; using System;
public static class WindowsGateway
{
    static WindowsGateway()
    {
        UnityLoaded = delegate { };
        OnClickPlay = delegate { };
        OnScoreUpdate = delegate { };
        OnClickBuy = delegate { };
    }
    public static Action UnityLoaded;
    public static Action OnClickPlay;
    public static Action OnScoreUpdate;
    public static Action OnClickBuy;
}
```

# 方案2: Action

第2步：在Unity脚本，需要切换到Windows Phone Solution代码就写下来WindowsGateway里面的Action函数

MainManager.cs:

```
void OnGUI()
{
    if (GUI.Button(new Rect(x, yP, BUTTON_SIZE, BUTTON_SIZE), "", mGUILayout))
    {
        WindowsGateway.OnClickPlay();
        Application.LoadLevel("Level");
    }
}
```

# 方案2: Action

第3步：在Windows Phone Solution, 创建新函数，写好需要的逻辑，然后连接到Unity里的static WindowsGateway类

MainPage.xaml.cs:

```
public MainPage(SplashScreen splashScreen)
{    this.InitializeComponent();
    splash = splashScreen;
    GetSplashBackgroundColor();
    OnResize();
    onResizeHandler = new WindowSizeChangedEventHandler((o, e) => OnResize());
    Window.Current.SizeChanged += onResizeHandler;
    WindowsGateway.OnClickPlay = OnClickPlay;
}

private void OnClickPlay()
{
    FlurryAnalytics.LogEvent(Constants.EVENT_PLAY_GAME);
}
```

# 方案3: Global Instance

在Windows Phone Solution直接用Unity脚本的类，函数，属性

第1步：Unity脚本里加个static readonly 类.

GameManager.cs:

```
private static readonly GameManager instance = new GameManager();  
  
private GameManager() { }  
  
public static GameManager Instance  
{  
    get  
    {  
        return instance;  
    }  
}  
  
public bool IsMainMenu = true;  
public static bool IsPaused = false;  
public void EnableMusic() { playMusic = true; }
```

# 方案3: Global Instance

第2步: 在Windows Phone Solution, 直接用Unity脚本的static instances

```
public MainPage(SplashScreen splashScreen)
{
    this.InitializeComponent();
    splash = splashScreen;
    GetSplashBackgroundColor();
    OnResize();
    onResizeHandler = new WindowSizeChangedEventHandler((o, e) => OnResize());
    Window.Current.SizeChanged += onResizeHandler;

    bool isMainMenu = GameManager.Instance.IsMainMenu;
    bool isGamePaused = GameManager.Instance.IsPaused;
    if (isGamePaused)
    {
        // Do something
        GameManager.Instance.EnableMusic();
    }
    elseif (isMainMenu) { // Do something}
}
```

# 方案4: Proxy Interface

```
namespace Common {
    public class Json {
        public interface IProxy
        {
            T Deserialize<T>(string value);
            string Serialize(object value);
        }
        public static IProxy Proxy { get; set; }
        static Json()
        {
#if !UNITY_WP_8_1 || UNITY_EDITOR
            Proxy = new JsonJsonFx();
#else
            Proxy = new JsonJsonNet();
#endif
        }

        public static T Deserialize<T>(string value)
        { return Proxy.Deserialize<T>(value); }
        public static string Serialize(object value)
        { return Proxy.Serialize(value); }
    }
}
```

# 方案4: Proxy Interface

```
#if !UNITY_WP_8_1 || UNITY_EDITOR
using UnityEngine;
using System.Collections;

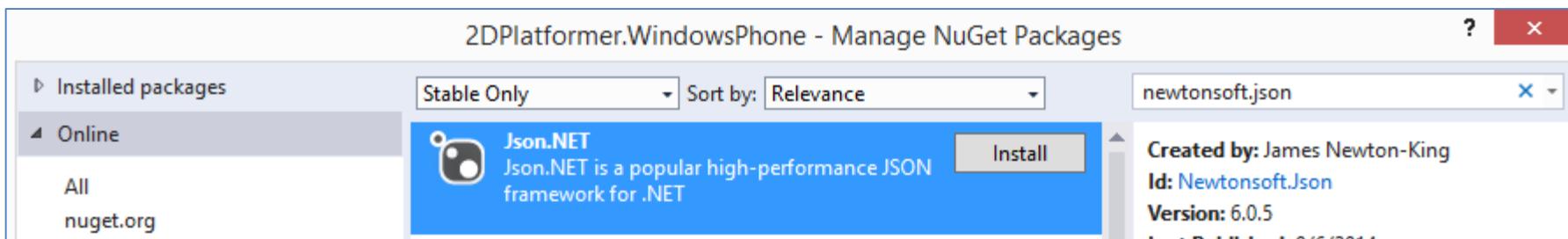
namespace Common
{
    class JsonJsonFx : Json.IProxy
    {
        public T Deserialize<T>(string value)
        {
            return JsonFx.Json.JsonReader.Deserialize<T>(value);
        }

        public string Serialize(object value)
        {
            return JsonFx.Json.JsonWriter.Serialize(value);
        }
    }
#endif
```

```
#if UNITY_WP_8_1
using UnityEngine;
using System.Collections;

namespace Common
{
    class JsonJsonNET : Json.Iproxy
    {
        public T Deserialize<T>(string v)
        {
            return Newtonsoft.Json.JsonConvert.DeserializeObject<T>(v);
        }

        public string Serialize(object v)
        {
            return Newtonsoft.Json.JsonConvert.SerializeObject(v);
        }
    }
#endif
```



# DEMO

---

解决API错误

# API了解

- API错误
  - XmlDocument
  - Reflection
  - File IO
  - Networking
- Windows Phone功能
  - IAP
  - Trial Mode
  - Maps
  - Speech
  - Live Tiles

The screenshot shows a Microsoft MSDN page titled "What's new in Windows Phone 8.1". The main content area is titled "Windows Phone 8.1 features". It includes sections for "Geolocation", "Geofencing", "Maps", and "Background transfers". Each section contains a brief description and a link to more information. The page has a standard header with navigation links like Dashboard, Get started, Design, Develop, Publish, and Community.

**Windows Phone 8.1 features**

**Geolocation**  
The Windows Runtime geolocation APIs available in Windows Phone 8 continue to be supported in Windows Phone 8.1. Some new classes have been introduced to support better interoperability with other features, such as the new geofencing feature.  
For more info, see [Detecting Geolocation](#).

**Geofencing**  
Windows Phone 8.1 introduces a set of geofencing APIs through which your app can receive notifications when the device enters or leaves an app-specified geographic region. This makes possible app scenarios such as displaying a reminder when the user arrives at home or their workplace, or notifying the user of deals and specials at stores in their area. For more information, see [Quickstart: Setting up a geofence](#).

**Maps**  
Windows Phone 8.1 maps add more features, support for offline tiles, better support for more complex gestures, and improvements in routing. For more info, see [Maps and directions](#).

**Background transfers**  
Windows Phone 8.1 supports the **Windows.Networking.BackgroundTransfer** namespace, through which you can queue uploads and downloads that will be completed by the system in the background even when your app is no longer running. These APIs replace the background transfer service APIs in Windows Phone 8.

**Media**

# API了解

Mono .NET

.NET for  
Windows  
Phone

.NET for  
Windows  
Runtime

Object  
Browser

平台#define

方案1:  
Unity插件

方案2  
Action

方案3  
Global

and more...

# 其他资料

<http://unity3d.com/cn/pages/windows/porting>

## 让您的游戏玩家数量达到数百万。 就是那样。

把您现有的Unity作品带到Windows Store和Windows Phone很容易。利用此页面上的资源将您的内容快速、高效地移植给全新的全球受众。

### 从基础开始

我们的入门指南涵盖了您需要知道的一切：配置、软件开发工具包、一般考虑和开始步骤



[Getting started on Windows Phone \(663 KB\)](#)



[Getting started on Windows Store \(720 KB\)](#)

### 然后了解移植技巧

一旦您掌握了基本知识，请阅读我们的移植技巧，精确掌握在调整您的代码用于Windows Store和Windows Phone时要考虑的因素



[Porting tips for Windows Phone \(819 KB\)](#)



[Porting tips for Windows Store \(813 KB\)](#)

# 其他资料

MVA: Porting Unity Games to Windows Store & Windows Phone

[microsoftvirtualacademy.com/training-courses/porting-unity-games-to-windows-store-and-windows-phone](https://microsoftvirtualacademy.com/training-courses/porting-unity-games-to-windows-store-and-windows-phone)

MVA: Developing 2D & 3D Games with Unity for Windows

[microsoftvirtualacademy.com/training-courses/developing-2d-3d-games-with-unity-for-windows-jump-start](https://microsoftvirtualacademy.com/training-courses/developing-2d-3d-games-with-unity-for-windows-jump-start)

2014 Building Your First Windows Game with Unity

<http://channel9.msdn.com/Events/Build/2014/2-503>

MVA: Porting Unity Games to Windows 8.1 & Windows Phone 8

[microsoftvirtualacademy.com/training-courses/porting-unity-games-to-windows-8-1-windows-phone](https://microsoftvirtualacademy.com/training-courses/porting-unity-games-to-windows-8-1-windows-phone)

Code Kwondo: Closer Look at Unity

<http://channel9.msdn.com/Events/Developer-Movement/UnityAndGaming/Part1>

Building Windows Games with Unity

<http://channel9.msdn.com/Events/Windows-Camp/Building-Windows-Games-with-Unity>

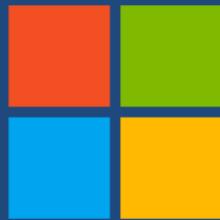
# Thank you

Elissa Tong  
[elistong@microsoft.com](mailto:elistong@microsoft.com)

<https://github.com/elissatong/UnityPortingLabs>

<http://www.microsoftvirtualacademy.com/training-courses/867>





# Microsoft

©2014 Microsoft Corporation. All rights reserved. Microsoft, Windows, Office, Azure, System Center, Dynamics and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.