一、框架组成

1、内置模块

（1）网络模块

（2）声音模块

（3）UI 模块

（4）消息模块

（5）常用工具模块

（6）日志模块

2、可拓展模块（各常用插件）

（1）二维码生成插件

（2）视频播放插件

（3）触屏插件

（4）原生相册读取插件

（5）拍照插件

（6）日志插件

（7）各种效果 Scroller 插件

（8）ShareSDK 插件

（9）剪贴板插件

3、框架内默认集成的插件：

（1）DoTween

（2）ITween

二、框架初始化（必做）

1、在编辑器内点击菜单栏 FrameWork->Create FrameWork Object，会看到场景内生成了一个名为 FrameWork 的游戏物体

2、可以自行对 FrameWork 进行 DontDestroyOnLoad 等设置

二、内置模块使用说明

1、网络模块

使用方式：单例

使用示例：

```
NetMngr.GetSingleton().各种方法();
```

说明：

详情请查看模块说明文档，文档路径：FrameWork->Doc->Server 模块说明

2、声音模块

使用方式：单例

使用示例：

```
SoundManager.GetSingleton().各种方法();
```

包含功能：

（1）背景音乐

（2）音效

（3）音乐、音效的音量设置

（4）播放音效时关闭背景音乐、播放音效时停止其他音效等效果

说明：

提供了最常用的声音控制方法和一些简单的音效播放设置

3、UI 模块

包含类：

BasePopup(弹窗基类)：

使用方式：

创建自己的弹窗类，然后继承该类

提供方法：

public virtual void ShowView(Action onComplete = null);

功能：显示弹窗

参数说明：

onComplete：动画播放完成回调

public virtual void HideView(Action onComplete = null);

功能：隐藏弹窗

参数说明：

onComplete：动画播放完成回调

提供参数（非静态）：

（1）AnimType DefaultAnimType：默认动画类型

（2）float DefaultAnimSpeed：默认动画速度，单位为时间

（3）Ease DefaultEase：默认动画 Ease，对应 DoTween Ease

（4）Color MaskColor：遮罩颜色

（5）bool IsShowMask：是否显示遮罩

（6）bool IsShowTop：弹窗时是否显示在顶层

使用说明：

（1）该类为各弹窗基类，**继承该类后需要调用 base.Init();方法进行初始化**

（2）支持在显示弹窗或隐藏弹窗之前使用链式的方式设置本次弹窗的各项属性

使用示例：

SetAnimType(AnimType.Alpha).SetEase(Ease.InBounce).SetSpeed(0.5f).ShowView();

PopupCommon(通用弹窗)：

使用方式：

（1）在编辑器内点击 FrameWork->Init Prefab->UI->PopupCommon 创建预设

（2）通过 GetSingleton()方法获取单例

注意：

每个需要用到该弹窗的场景都需要放置一个预设

提供方法：

public void ShowView(string msg, Action animCallback = null, bool showCancel =

false, Action sureCallback = null, Action cancelCallback = null, bool autoHide = true);

功能：显示弹窗

参数说明：

msg：消息内容

animCallback：动画完成回调

showCancel：是否显示取消按钮

sureCallback：确认按钮点击回调

cancelCallback：取消按钮点击回调

autoHide：点击确认后是否隐藏弹窗

public void HideView(Action onComplete = null);

功能：隐藏弹窗

参数说明：

onComplete：动画完成后回调

ScrollMessage(滚动消息):

使用方式：

（1）在编辑器内点击 FrameWork->Init Prefab->UI->ScrollMessage 创建预设

（2）获取物体上的脚本

提供方法：

public void AddMessage(string msg, int repeatCount = 1);

功能：增加一条消息

参数说明：

msg：消息内容

repeatCount：滚动次数，0 代表无限循环滚动

public void RemoveFirstMessage();

功能：移除第一条数据，即正在滚动的数据

public void ClearAllMessage();

功能：清除所有数据

提供参数：

（1）float MoveSpeed：移动速度，对应锚点坐标

（2）Int MessageCount：消息数量

说明：

当存在消息时，按设置的消息进行滚动播放。当无消息时自动隐藏

4、消息模块

使用方式：单例

使用示例：

MessageManager.GetSingleton().各种方法();

包含方法：

注册消息侦听

public void RegisterMessageListener(string message, Action<System.Object> callback)

移除消息侦听

public void UnRegisterMessageListener(string message, Action<System.Object> callback)

发送消息

public void SendMsg(string message, System.Object param = null)

说明：

1、该模块提供了本地代码对象之间传递消息的能力

2、消息间的传参是以 Object 类型进行传递，所以需要进行拆装箱操作

3、**注意：传参的类型为 C#的 Object 格式而非 Unity 的 Object 格式**

5、常用工具模块

包含类：

EventTriggerListener：

使用方式：静态

说明：可以增加侦听的方式增加 EventTrigger 的各事件

MicroPhoneInput：

使用方式：单例

使用示例：

MicroPhoneInput.GetSingleton().各种方法();

说明：提供了录音功能，详情请查看类内注释

GameTools：

使用方式：单例

使用示例：

GameTools.GetSingleton().各种方法();

包含方法：

截图：

public void PrintScreenTexture(int x, int y, int width, int height ,Action<Texture2D> callback);

public void PrintScreenTexture(int x, int y, int width, int height, Action<Sprite> callback);

参数说明：

x：开始的屏幕 x 坐标

y：开始的屏幕 y 坐标

width：图片宽度

height：图片高度

callback：完成回调

说明：提供了返回 Texture2D、Sprite 两种截图方式

修改图片宽高：

public Texture2D ScaleTexture(Texture2D source, int targetWidth, int targetHeight);

参数说明：

source：原图

targetWidth：新图片宽度

targetHeight：新图片高度

说明：

返回一张指定宽高的新的图片

保存文件到本地：

public bool SaveFileToLocal(string path, byte[] file);

参数说明：

path：路径

file：文件字节流

加载本地文件

public byte[] LoadFileFromLocal(string path);

参数说明：

path：路径

加载网络图片

public void GetTextureFromNet(string url, System.Action<Texture2D> callback);

public void GetTextureFromNet(string url, System.Action<Sprite> callback);

参数说明：

url：路径

callback：回调

说明：加载完成图片后会触发回调方法

加载本地或网络 JSON 文件

public void LoadJson(string url, Action<Hashtable> callback, bool fromLocal = false);

public void LoadJson(string url, Action<string> callback, bool fromLocal = false);

参数说明：

url：路径

callback：回调

fromLocal：是否从本地加载

保存 JSON 文件到本地

public bool SaveJsonToLocal(string path, string json);

public bool SaveJsonToLocal(string path, Hashtable jsonData);

参数说明：

path：路径

json：json 数据

jsonData：json 数据

ObjectPool<T>：

类型：基础数据结构对象池

使用方式：实例化

包含方法：

获取对象：

public T Get()

说明：从对象池内获取对象，如果对象池内无对象，则会 New 出一个新对象

放入对象：

public void Put(T obj)

说明：把对象放回到对象池内

清理对象：

public void Clear(int retainCount)

public void ClearAll()

说明：提供了清理部分对象和清理全部对象两种清理方式

GameObjectPool：

类型：Unity 游戏物体对象池

使用方式：实例化

包含方法：

获取对象：

public GameObject Get()

说明：从对象池内获取对象，如果对象池内无对象，则会实例化出一个新对象

放入对象：

public void Put(GameObject obj)

说明：把对象放回到对象池内

清理对象：

public void Clear(int retainCount)

public void ClearAll()

说明：提供了清理部分对象和清理全部对象两种清理方式

AssetsBundle 导出：

（1）在工程内对需要导出的资源进行 AssetsBundle 设置

（2）点击菜单栏->FrameWork->Export AssetsBundle->Platform->各平台

（3）目前支持 4 个常用平台导出：Windows64、Windows32、安卓、IOS

GetGPS：

使用方式：单例

使用示例：

GetGPS.GetSingleton().各种方法();

包含方法：

1、StartGPS()开启 GPS

2、StopGPS()停止 GPS

3、GetLongitude()获取经度

4、GetLatitude()获取纬度

5、Distance(float lng1, float lat1, float lng2, float lat2)获取两个经纬度的距离

说明：

1、如果手机不开启 GPS 或者设备不支持 GPS，获取到的经纬度都是 0

2、使用前需要先调用 StartGPS()开启 GPS

6、日志模块

使用方式：单例

使用方式：

LogManager.GetSingleton().各种方法();

说明：

1、模块提供了保存 Unity 内错误日志到本地的功能

2、默认模块不启用，需要调用 Init()方法启动模块

3、每条错误日志组成：错误日志前最大 30 条日志 + 错误日志 + 错误日志堆栈

3、每次启动模块时会自动清理 3 天之前的日志

4、可使用 GetAllLogPath()方法获取所有日志文件路径

5、可使用 GameTools 工具类的 LoadFileFromLocal()方法加载日志文件

三、可拓展模块说明

框架内包含了各种常用的插件但是默认不集成进工程，如有需要可以自行导入并查阅相关文档进行相应配置