客户端性能优化技术分享

潘宇频
2020.5.15

目录

前情提要

自我介绍

GoG项目UI卡顿分析 & 优化

- https://docs.google.com/presentation/d/16phV0YLh873CtWnBoWoq8t-UuN93uzK5O9MONB4PMOI
- U3D的加载机制介绍、性能瓶颈分析、初步的优化方案

性能优化工具文档

- https://docs.google.com/document/d/1XpihdsRERQ7d6ez8botUimH-2_27rZV1qfrCeo0yQQQ
- 工具研发的思路、详细使用说明

优化和引擎开发工作计划

- https://docs.google.com/presentation/d/1kdIwd6CY4Fv_ih4rX6G4-SSnSoJH-I1kRIIAtQsK7_I
- 优化线未来打算做的事情、希望达成的目标

性能优化的目标

可度量的指标

- 闪退率：3%
- 内存：高档PSS<=900；中档PSS<=800；低档PSS<=700
- CPU占用率：综合CPU平均占用(90%)小于60%，单核CPU峰值占用（90%）小于90%
- 帧率：核心游戏场景默认要求90%不低于25 FPS（18FPS）

消除卡顿

= 每帧时长不超过33.3ms

降低内存

= 严格控制内存中的资源数量

性能依赖项目组整体的意识、也依赖程序的个人能力

PerfTool 工具

目标：定位代码性能损耗的瓶颈，保证每帧消耗时长低于33ms

Unity Profiler：Deep开销大；必须真机连数据线；无法交由其它人测试运行

UWA：有自己独特的优势，但是不是那么细

PerTool：自研，定制性强，与业务结合紧密，方便集成自动化

项目名	起始帧	类名	函数名	时长	GO数量	Comp数量	原记录	压缩后	设备信息 deviceModel: MacBookPro16,1 deviceName: Yupin的MacBook Pro deviceType: Desktop os: Mac OS X 10.15.4 memorySize: 16384 graphicsDevice: Emulated GPU running OpenGL ES 3.0 graphicsVendor: Emulated
0.PublicHUD.publicHUD	29	PublicHUD	publicHUD	202.31	633	1031	88	50	[PublicHUD.publicHUD] 202.31ms, gc:3.9MB, unt:8.2MB, f: [29, 29] 333, - [Prefab/UI/Huds/PublicHUD : LoadAsset] `82.18ms`, gc:204.0KB, unt:4.8MB,
1.VerificationPopup.OpenPopup	38	VerificationPopup	OpenPopup	61.01	93	148	169	72	[VerificationPopup.OpenPopup] 61.01ms, gc:1.3MB, unt:2.9MB, f: [38, 38] 333, - [Prefab/UI/Popups/VerificationPopup : LoadAsset] `9.34ms`, gc:76.0KB, unt:2
2.VerificationPopup.OnCloseHandler	68	VerificationPopup	OnCloseHandler	8.48	0	0	2	2	[VerificationPopup.OnCloseHandler] 8.48ms, gc:116.0KB, unt:-10.0KB, f: [68, 6 - [EventDelegate.Execute mCachedCallback] `8.46ms`, gc:116.0KB, unt:-10.0K
3.Sign/SignPopup.OpenPopup	73	Sign/SignPopup	OpenPopup	164.08	92	130	162	69	[Sign/SignPopup.OpenPopup] 164.08ms, gc:20.4MB, unt:1.9MB, f: [73, 73] 224 - [Prefab/UI/Popups/Sign/SignPopup : LoadAsset] `9.86ms`, gc:60.0KB, unt:1.8
4.SignPopup.OnCloseButtonPressed	105	SignPopup	OnCloseButtonPressed	6.99	0	0	2	2	[SignPopup.OnCloseButtonPressed] 6.99ms, gc:184.0KB, unt:-16.5KB, f: [105, - [EventDelegate.Execute mCachedCallback] `6.98ms`, gc:184.0KB, unt:-16.5K
5.PublicHUD.OnClickHeroBtn	153	PublicHUD	OnClickHeroBtn	2.15	0	0	1	1	[PublicHUD.OnClickHeroBtn] 2.15ms, gc:8.0KB, unt:585.9KB, f: [153, 153] 249 - [EventDelegate.Execute mCachedCallback] 2.35ms, gc:12.0KB, unt:585.9KB,
6.HeroCard/HeroInventoryDlg.OpenDlg	153	HeroCard/HeroInven	OpenDlg	172.64	1089	1283	1747	637	[HeroCard/HeroInventoryDlg.OpenDlg] 172.64ms, gc:2.9MB, unt:3.9MB, f: [153 - [Prefab/UI/Dialogs/HeroCard/HeroInventoryDlg : LoadAsset] `26.30ms`, gc:0.0
7.HeroInventorySlotEx.OnHeroCardClicked	181	HeroInventorySlotEx	OnHeroCardClicked	0.29	0	0	1	1	[HeroInventorySlotEx.OnHeroCardClicked] 0.29ms, gc:8.0KB, unt:0.0B, f: [181, - [EventDelegate.Execute mCachedCallback] 0.49ms, gc:8.0KB, unt:0.0B, f: [18

0.PublicHUD.p
1.VerificationP
2.VerificationP
3.Sign_SignPo
4.SignPopup.C
5.PublicHUD.C
6.HeroCard_H
7.HeroInventor
8.HeroReform
9.HeroReform
10.HeroReform
11.HeroReform
12.HeroReform
13.HeroInvent
14.CityMapPre
15.CityMapPre
16.CityMapPre

stat.md

```
[HeroCard/HeroInventoryDlg.OpenDlg] 2786ms, gc:-6.0MB, unt:10.7MB, f: [456, 456] 333,
- [Prefab/UI/Dialogs/HeroCard/HeroInventoryDlg : LoadAsset] `122ms`, gc:236.0KB, unt:647.9KB, f: [456, 456] 333,
- [Prefab/UI/Dialogs/HeroCard/HeroInventoryDlg : Instantiate] `122ms`, gc:120.0KB, unt:129.8KB, f: [456, 456] 333,
  - [UI.UIBindView.Awake] `5ms`, gc:4.0KB, unt:0.0B, f: [456, 456] 333,
    - [UI.UIBindView.CacheTransformByUniqueName] 3ms, gc:4.0KB, unt:0.0B, f: [456, 456] 333,
      - [UI.UIBindView.CacheTransformByUniqueName] 3ms, gc:4.0KB, unt:0.0B, f: [456, 456] 333,
        - [UI.UIBindView.CacheTransformByUniqueName] 0ms, gc:4.0KB, unt:0.0B, f: [456, 456] 333,
          - [UI.UIBindView.CacheTransformByUniqueName] 0ms, gc:4.0KB, unt:0.0B, f: [456, 456] 333,
            - [UI.UIBindView.CacheTransformByUniqueName] 0ms, gc:4.0KB, unt:0.0B, f: [456, 456] 333,
        - [UI.UIBindView.CacheTransformByUniqueName] 2ms, gc:0.0B, unt:0.0B, f: [456, 456] 333,
          - [UI.UIBindView.CacheTransformByUniqueName] 1ms, gc:0.0B, unt:0.0B, f: [456, 456] 333,
            - [UI.UIBindView.CacheTransformByUniqueName] 1ms, gc:0.0B, unt:0.0B, f: [456, 456] 333,
              - [UI.UIBindView.CacheTransformByUniqueName] 1ms, gc:0.0B, unt:0.0B, f: [456, 456] 333,
                - [UI.UIBindView.CacheTransformByUniqueName] 1ms, gc:0.0B, unt:0.0B, f: [456, 456] 333,
                  - [UI.UIBindView.CacheTransformByUniqueName] 1ms, gc:0.0B, unt:0.0B, f: [456, 456] 333,
        - [UI.UIBindView.CacheTransformByUniqueName] 1ms, gc:0.0B, unt:0.0B, f: [456, 456] 333,
          - [UI.UIBindView.CacheTransformByUniqueName] 1ms, gc:0.0B, unt:0.0B, f: [456, 456] 333,
            - [UI.UIBindView.CacheTransformByUniqueName] 1ms, gc:0.0B, unt:0.0B, f: [456, 456] 333,
              - [UI.UIBindView.CacheTransformByUniqueName] 1ms, gc:0.0B, unt:0.0B, f: [456, 456] 333,
        - [UI.UIBindView.CacheTransformByUniqueName] 1ms, gc:0.0B, unt:0.0B, f: [456, 456] 333,
          - [UI.UIBindView.CacheTransformByUniqueName] 1ms, gc:0.0B, unt:0.0B, f: [456, 456] 333,
    - [UI.UIBindView.UpdateCallbackUI] 1ms, gc:0.0B, unt:0.0B, f: [456, 456] 333,
  - [FixedUIRect.OnEnable] `2613ms`, gc:5.2MB, unt:9.9MB, f: [456, 456] 333,
    - [UITextureAutoSetter.OnEnable] `74ms`, gc:60.0KB, unt:65.5KB, f: [456, 456] 333,
      - [UITextureAutoSetter.SetTexture] `74ms`, gc:60.0KB, unt:65.5KB, f: [456, 456] 333,
```

示例:看一下实际输出的结果

https://docs.google.com/spreadsheets/d/1HdICBgAhUxnP0ejv02iLFTfabI-hnfTEco2_0NIvKU4/edit#gid=799600584

文档:https://docs.google.com/document/d/1XpihdsRERQ7d6ez8botUimH-2_27rZV1qfrCeo0yQQQ/edit#heading=h.u50q59xpi25q

接下来介绍这个系统的实现细节

TimeRecorder

```
public class TimeRecorder : IDisposable
{
    private string _name;
    public long StartTimeUs { get; private set; }
    public long EndTimeUs { get; private set; }
    public long DurationUs { get; private set; }
    public int FrameIndex { get; private set; }

    public MemorySizeSnapShot startMem;
    public MemorySizeSnapShot endMem;

    private static List<TimeRecorder> _allTimeRecorderBuffer = new List<TimeRecorder>();

    public void Start();
    public void End();
    public void Dispose(){ this.End(); }
}
```

```
public struct MemorySizeSnapShot
{
    public long gcTotalMemory;
    public long monoUsed;
    public long unityTotalAllocated;

    //GC.GetTotalAllocatedBytes
    //Profiler.GetMonoUsedSizeLong
    //Profiler.GetTotalAllocatedMemoryLong
```

PerfMan & UIPerfRecord

PerfMan：性能测量的全局管理器

UIPerfRecord：数据项目。同一时刻只有一个UIPerfRecord项目起效，所有函数的调用时长汇总收集到它那里

```
public static class PerfMan
{
        //创建一个数据项目，并且把它设为当前的焦点
        public static UIPerfRecord CreatePerfRecord(string uiName, string funcName)
        //设置焦点项目
        public static void SetFocusPerfRecord(UIPerfRecord upr)
        //测量一段代码
        public static TimeRecorder Perf(string stepName, params object[] args)
        //输出所有结果到文件
        public static void Output()
}
```

示例

```
//创建一个项目
public void OpenDlg(string dialogType, Dialog.DialogParameter param = null, bool lockInput = true,
   bool hideToast = true, bool hidePublichud = true)
{
          perfRecord = PerfMan.CreatePerfRecord(dialogType, "OpenDlg");
}

.....

//测量一行代码
using (PerfMan.Perf("builder.Build"))
builder.Build(path);

.....

//测量多行代码
using (PerfMan.Perf("Process scrollViewListWrapper"))
{
      _scrollViewListWrapper.SetRefreshCallback(OnRefreshItem);
      _scrollViewListWrapper.SpawnNewList(_prefab, items.Count, 0f);
 }
```

自动注入

自动注入代码：C# → Assembly → 注入 → IL2CPP → so

参考资料：Unity3D研究院自动注入代码统计每个函数的执行效率以及内存分配（九十八）

VSCode插件：https://marketplace.visualstudio.com/items?itemName=icsharpcode.ilspy-vscode

```
    IL_0000: ldstr "ActivityStepRequire.<UpdatUI>m__0" /* 700084F0 */
    IL_0005: call void [kingsgroup.perftool.Runtime]ClientCore.Performance.PerfMan::StartPerfFuncOnFocus(string) /* 0A000096 */
    IL_000a: nop
    IL_000b: ldarg.1
    IL_000c: ldflda int32 ActivityStepRewardInfo::Step /* 04003F02 */
    IL_0011: ldarg.0
    IL_0012: ldfld int32 ActivityStepRewardInfo::Step /* 04003F02 */
    IL_0017: call instance int32 [mscorlib]System.Int32::CompareTo(int32) /* 0A000135 */
    IL_001c: stloc.0
    IL_001d: br IL_0022

    IL_0022: ldloc.0
    IL_0023: call void [kingsgroup.perftool.Runtime]ClientCore.Performance.PerfMan::EndPerfFuncOnFocus() /* 0A000098 */
    IL_0028: ret
} // end of method ActivityStepRequire::'<UpdatUI>m__0'
```

实现细节

- 函数开头的注入
  - `MethodReference startProfile = moduleDefinition.ImportReference(methodBegin);`
  - `MethodReference endProfile = moduleDefinition.ImportReference(methodEnd);`
  - `ILProcessor ilProcessor = methodDefinition.Body.GetILProcessor();`
  - `Instruction first = methodDefinition.Body.Instructions[0];`
  - `string methodName = typeDefinition.FullName + "." + methodDefinition.Name;`
  - `ilProcessor.InsertBefore(first, Instruction.Create(OpCodes.Ldstr, methodName));`
  - `ilProcessor.InsertBefore(first, Instruction.Create(OpCodes.Call, startProfile));`

- 函数离开的注入
  - `lastInstructionList.Add(Instruction.Create(OpCodes.Call, endProfile));`

- 采坑：处理分支跳转（if、switch）、异常处理（try）
- 偶尔注入的时候会异常，二分查找问题函数
- 目前3W+的函数注入

API:配置需要注入的类

```
//类是否需要自动注入代码(没有UIPerf
public override bool IsClassAutoProfile(TypeDefinition typeDefinition)
{
   bool isClassAutoPerf = Internal_IsClassAutoProfile(typeDefinition);
   return isClassAutoPerf;
}

//方法是否需要自动注入代码
public override bool IsMethodAutoProfile(TypeDefinition typeDefinition, MethodDefinition methodDefinition)
{
   //省略Lambda表达式。如果要测量的话，请自己加
   if (methodDefinition.Name.StartsWith("<")) return false;
   if(methodDefinition.Name.EndsWith("BuildDB") || typeDefinition.Namespace == "DB"
      || typeDefinition.FullName.StartsWith("Config"))//默认测量BuildDB方法
      return true;
   return false;
}

[UIPerf]
public class HeroReformSubAppointView : MonoBehaviour

[NoUIPerf]
static bool CheckRequirement(string idStr,int reqValue)
```

KOA的使用

分类

- 网络请求回调
- 资源加载
- Dialog和Popup开启
  - UIManager.OpenDlg
  - UIManager.OpenPopup
- NGUI所有事件响应
  - EventDelegate.Excute()
- 初始化加载
- 手动开启关闭
  - 按钮触发，测量一段时间内Update的耗时

自动注入设置

- MonoBehaviour子类
- 类位于DB命名空间
- 类名Config开头
- 方法名以BuildDB开头
- 方法名不以<开头（忽略匿名方法）

KOA UI开启的典型性能热点

- 配置表的读取
- 大面板加载
- 业务逻辑计算量大
  - Config排序、DB查询
- ScrollView初始化

TODO

QA定期出报告，对比性能数据

自动化测试

统计函数调用前后的资源增量

统计界面开关前后的资源增量

- 提问？

Asset Validator

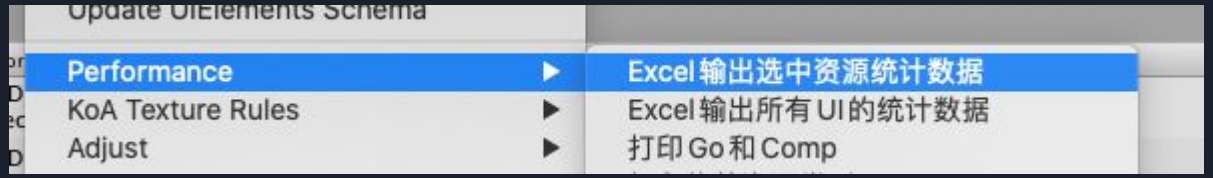
静态检查工具集。

主要功能：

- 检查Unity资源的正确性
- API
- Editor Window
- CICD集成
- 报表和通知

先介绍两个独立的小工具

小工具:资源引用分析



	A	B	C
1		值	文件大小
2	资源路径	Assets/__UIData/_Resources/Prefab/UI/Dialogs/HeroCard/HeroInventoryDlg.prefab	
3	类型	Prefab	
4	引用资源总数	154	
5	GameObject总数	924	
6	Component总数	1,505	
7	Mesh总数	0	
8	顶点总数	0	
9	三角形总数	0	
10	AnimationClip总数	1	
11	Controller总数	0	
12			
13	Particl System总数	12	
14	Mesh Renderer总数	0	
15	Skinned Renderer总数	0	
16			
17	Fbx	0	0
18	Texture	39	0
19	Material	9	7,415
20	Shader	1	1,838
21	Font	3	211,316
22	Asset	0	0
23	Prefab	26	3,267,584
24	Timeline	0	0
25			

概览 | 引用的C# | 引用文件 | Component | Fbx | Texture | Material | Shader | Mesh | AnimationClip | Timeline

160%

引用分析实现细节

https://docs.unity3d.com/ScriptReference/AssetDatabase.GetDependencies.html

```
public class AssetRefStatistics
{
        public static AssetRefStatistics Create(string path, bool isRecursion = true){}
        //路径
        public string Path { get; private set; }
        //资源类型
        public EAssetFileType AssetFileType { get; private set; }
        // 所有引用的资源文件
        public Dictionary<string, AssetFileRef> AllAssetsDict = new Dictionary<string, AssetFileRef>();
        //所有引用的Fbx
        public List<FbxFileInfo> AllFbxFiles = new List<FbxFileInfo>();
        // 所有的Mesh
        public List<MeshInfo> AllMeshes = new List<MeshInfo>();
        public int AllVertexCount = 0;
        public int AllTriangleCount = 0;
        // 所有的动画
        public List<AnimationClipInfo> AllAnimationClips = new List<AnimationClipInfo>();
        // 所有控制器
        public List<AnimatorControllerInfo> AllAnimationController = new List<AnimatorControllerInfo>();
        // 所有的纹理
        public List<TextureFileInfo> AllTextures = new List<TextureFileInfo>();
        public long AllTextureMemorySize = 0;
        // 所有材质
        public List<MaterialInfo> AllMaterials = new List<MaterialInfo>();
        ……
```

小工具:出包资源分析

	A	B	C	D	E	F	G	H
1	资源路径	资源类型	所属AB	Editor磁盘文件大小	AB中大小	纹理占用内存大小	GameObject数	Component数
3185	Assets/__RawArtData/March/Knight/Attack/Camera5/Knight_Attalk05_014.png	Texture	static+default	130,213	819	65,536	0	0
3186	Assets/__RawArtData/March/Knight/Attack/Camera5/Knight_Attalk05_015.png	Texture	static+default	129,576	716	65,536	0	0
3187	Assets/__RawArtData/March/Knight/Dead/l/Camera4.0000.png	Texture	static+default	118,560	1,024	65,536	0	0
3188	Assets/__RawArtData/March/Knight/Dead/l/Camera4.0001.png	Texture	static+default	111,381	819	65,536	0	0
3189	Assets/__RawArtData/March/Knight/Dead/l/Camera4.0002.png	Texture	static+default	98,950	819	65,536	0	0
3190	Assets/__RawArtData/March/Knight/Dead/l/Camera4.0003.png	Texture	static+default	90,175	716	65,536	0	0
3191	Assets/__RawArtData/March/Knight/Dead/l/Camera4.0004.png	Texture	static+default	102,613	1,024	65,536	0	0
3192	Assets/__RawArtData/March/Knight/Dead/l/Camera4.0005.png	Texture	static+default	108,482	1,228	65,536	0	0
3193	Assets/__RawArtData/March/Knight/Dead/lu/Camera3.0000.png	Texture	static+default	110,093	921	65,536	0	0
3194	Assets/__RawArtData/March/Knight/Dead/lu/Camera3.0001.png	Texture	static+default	101,769	819	65,536	0	0
3195	Assets/__RawArtData/March/Knight/Dead/lu/Camera3.0002.png	Texture	static+default	96,804	921	65,536	0	0
3196	Assets/__RawArtData/March/Knight/Dead/lu/Camera3.0003.png	Texture	static+default	95,918	819	65,536	0	0
3197	Assets/__RawArtData/March/Knight/Dead/lu/Camera3.0004.png	Texture	static+default	98,698	921	65,536	0	0
3198	Assets/__RawArtData/March/Knight/Dead/lu/Camera3.0005.png	Texture	static+default	107,295	1,126	65,536	0	0
3199	Assets/__RawArtData/March/Knight/Dead/u/Camera5.0000.png	Texture	static+default	82,697	819	65,536	0	0
3200	Assets/__RawArtData/March/Knight/Dead/u/Camera5.0001.png	Texture	static+default	89,973	716	65,536	0	0
3201	Assets/__RawArtData/March/Knight/Dead/u/Camera5.0002.png	Texture	static+default	103,095	921	65,536	0	0
3202	Assets/__RawArtData/March/Knight/Dead/u/Camera5.0003.png	Texture	static+default	101,732	819	65,536	0	0
3203	Assets/__RawArtData/March/Knight/Dead/u/Camera5.0004.png	Texture	static+default	97,719	1,024	65,536	0	0
3204	Assets/__RawArtData/March/Knight/Dead/u/Camera5.0005.png	Texture	static+default	104,358	1,331	65,536	0	0
3205	Assets/__RawArtData/March/Knight/Station/Knight_Station02.png	Texture	static+default	129,529	716	65,536	0	0
3206	Assets/__RawArtData/March/Knight/Walk/camera1/camera1_01.png	Texture	static+default	125,200	819	65,536	0	0
3207	Assets/__RawArtData/March/Knight/Walk/camera1/camera1_02.png	Texture	static+default	127,069	921	65,536	0	0
3208	Assets/__RawArtData/March/Knight/Walk/camera1/camera1_03.png	Texture	static+default	127,743	819	65,536	0	0
3209	Assets/__RawArtData/March/Knight/Walk/camera1/camera1_04.png	Texture	static+default	126,350	819	65,536	0	0
3210	Assets/__RawArtData/March/Knight/Walk/camera1/camera1_05.png	Texture	static+default	125,331	716	65,536	0	0
3211	Assets/__RawArtData/March/Knight/Walk/camera1/camera1_06.png	Texture	static+default	126,138	921	65,536	0	0
3212	Assets/__RawArtData/March/Knight/Walk/camera1/camera1_07.png	Texture	static+default	128,019	1,024	65,536	0	0

重点:静态资源检查框架

现状

- UWA
  - https://blog.uwa4d.com/archives/UWA_Pipeline3.html
- Unity官方
  - https://upr.unity.cn/instructions#assetchecker
- Odin Project Validator
  - https://odininspector.com/odin-project-validator
- Maintainer
  - https://codestage.net/uas/maintainer/
- 其它零散的自研工具

痛点

- 仅可全局的资源检查配置，不够细化，灵活性有限
- 统一的错误输出、持续集成支持

资源检查工具的需求

- 资源正确的重要性：正确& 性能
  - Prefab是否丢失关键节点？
  - 是否缺失关 键Component？
  - 美术规格是否超标？
- 交叉引用正确:Code ↔ Asset，Config ↔ Asset、Config ↔ Code
  - 代码引用的资源是否存在？
  - 配表中引用的prefab是否存在？
  - 配表中的数据代 码是否支持？
- 资源正确性：二维表格，功能点+属性点
  - 大地图图块：它的Sprite设置是否正确？引用材 质是否合法？物理碰撞是否开启？ Transfom是否正确？
  - 3D龙模型：它的Mesh顶点数量是否合 规？通道数量是否正确？材 质是否合规？材质引用的纹理是否设置正确？引用目录是否正确？ 总资源量是否合 规？1、2、3级资源是否分别合规？
- 检查工具的交互界面：人工检查、CICD
  - UE、美术、策划可以在Editor中检查自己的产出、快速定位解决 问题、减轻前端负担
  - CICD能定期出报告，及时反馈资源错误

功能点	Scene	Prefab	Fbx	Animation	Controller	particle	Material	Texture	Shader	Timeline	Excel
大地图地块	x	x	x				x	x	x		
大地图树木		x					x	x	x		
大地图城市		x	x				x	x	x		x
[redacted]		x	x	x			x	x	x		x
[redacted]示之特效						x	x	x	x		
[redacted]之Timeline			x							x	x
战斗场景	x		x				x	x	x		
战斗武将		x	x	x	x		x	x	x		x
[redacted]											x
受击特效						x	x	x	x		x
buff特效						x	x	x	x		x
技能特效						x	x	x	x		x
UI板子Panel		x						x		x	
UI板子Item		x									
UI贴图							x	x	x		
ui特效		x				x	x	x	x		

```
fileFormatVersion: 2
guid:
a961e4975c6c0f4428398fbe140a2742
timeCreated: 1544786021
licenseType: Free
TextureImporter:
 fileIDToRecycleName: {}
 serializedVersion: 4
 mipmaps:
  mipMapMode: 0
  enableMipMap: 0
  sRGBTexture: 1
  linearTexture: 0
  fadeOut: 0
  borderMipMap: 0
  mipMapFadeDistanceStart: 1
  mipMapFadeDistanceEnd: 3
 bumpmap:
  convertToNormalMap: 0
  externalNormalMap: 0
  heightScale: 0.25
  normalMapFilter: 0
 isReadable: 0
 grayScaleToAlpha: 0
 generateCubemap: 6
 cubemapConvolution: 0
 seamlessCubemap: 0
 textureFormat: 13
 maxTextureSize: 128
 textureSettings:
  filterMode: -1
  aniso: 1
  mipBias: -1
  wrapMode: 1
 nPOTScale: 0
 lightmap: 0
 compressionQuality: 100
 spriteMode: 1
 spriteExtrude: 0
 spriteMeshType: 1
 alignment: 0
 spritePivot: {x: 0.5, y: 0.5}
 spriteBorder: {x: 0, y: 0, z: 0, w: 0}
 spritePixelsToUnits: 100
 alphaUsage: 1
 alphaIsTransparency: 1
 spriteTessellationDetail: -1
 textureType: 8
 textureShape: 1
 maxTextureSizeSet: 0
 compressionQualitySet: 0
 textureFormatSet: 0
 platformSettings:
 - buildTarget: DefaultTexturePlatform
   maxTextureSize: 128
   textureFormat: -1
   textureCompression: 1
   compressionQuality: 100
   crunchedCompression: 0
   allowsAlphaSplitting: 0
   overridden: 0
 spriteSheet:
   serializedVersion: 2
   sprites: []
   outline: []
 spritePackingTag:
march_anims_archer_attack
 userData:
 assetBundleName:
 assetBundleVariant:
```

AssetValidator方案

客户端程序，用代码，约束所有的资源正确性

项目组每个人都可以用工具检查自己的产出

工具由前端程序开发，因为：

- 最终为客户端的正确性和性能负责的是前端
- 游戏运行时出了问题只有前端可以定位
- 性能相关的规范只能由前端制定
- 前端有义务开发工具，提升美术、策划的工作效率
- 前端做好这件事情可以大幅降低自己的负担，最大收益者可能是自己

既然前端开发和维护工具，那么与其配置参数，不如直接撸码

- 可以自定义的复杂检查规则；封装和复用

API：AssetRefStatistics_Requirement

```
public class AssetRefStatistics_Requirement
{
        //顶点数量上限
        public void Require_MeshVertexTotalCount(int min, int max)
        // GO总数过大
        public void Require_GameObjectTotalCount(int min, int max)
        // ParticleSystem总数过大
        public void Require_ParticleSystemTotalCount(int min, int max)
        // 允许的纹理贴图格式
        public void Allow_TextureFormat(HashSet<UnityEditor.TextureImporterFormat> formatSet)
        // 引用文件的路径不能含有关 键字
        public void Forbidden_AssetReferenceHasKeyWords(HashSet<string> keywordsSet)
        // 允许Texture的引用路径
        public void Require_TextureStartsWith(HashSet<string> startsStringSet)
        // 允许动画帧率
        public void Require_AnimationClip_FrameRate(int min, int max)
        ……

}
```

API：SerializedObject_Requirement

Unity3D可序列化的类型是有限的

https://docs.unity3d.com/ScriptReference/SerializedPropertyType.html

检查所有的UnityEngine.Object。
原理：public SerializedObject(Object obj);

```
public class SerializedObject_Requirement<T> where T : UnityEngine.Object
{
        public static SerializedObject_Requirement<T> Create(T obj, IssueReportFunc reportIssue,
        RecordSeverityEx severity)

        public void Require(Func<T, bool> checkFun, string errorInfo)

        public void ReqProp_Int(string propPath, Func<T, int, bool> checkFun, string errorInfo)
        public void ReqProp_String(string propPath, Func<T, string, bool> checkFun, string errorInfo)
        public void ReqProp_Vector3(string propPath, Func<T, Vector3, bool> checkFun, string errorInfo)
        public void ReqProp_ObjectRef(string propPath, Func<T, UnityEngine.Object, bool> checkFun,
          string errorInfo)
```

API：GameObject_Requirement

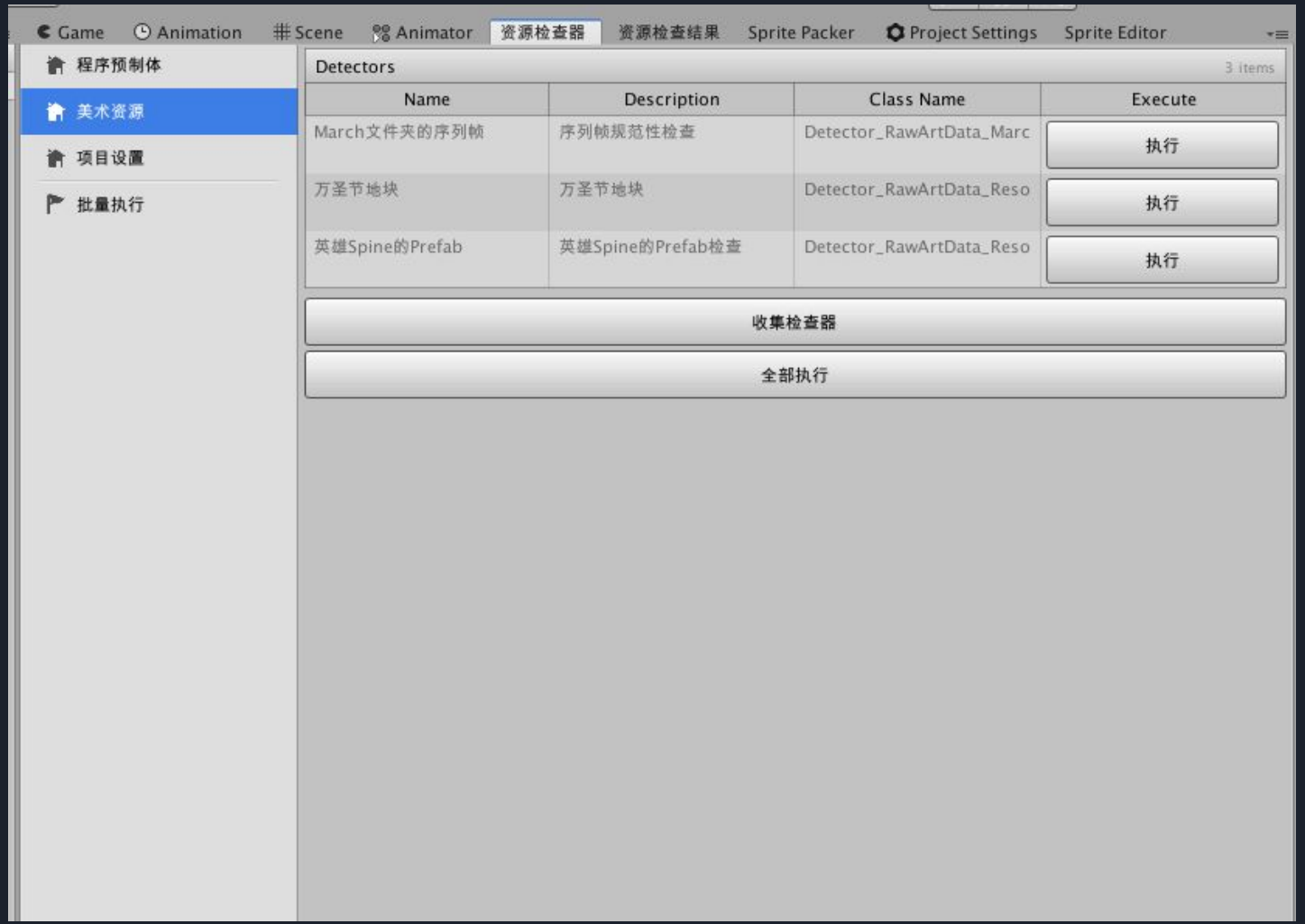
```
public class GameObject_Requirement
{
        static public GameObject_Requirement Create(GameObject gameObject, bool traceChildren,
IssueReportFunc reportIssue, RecordSeverityEx severity)

        public GameObject_Requirement RequireChildrenPath(string path, bool traceChildrenRecursion)
        public List<GameObject_Requirement> AllowChildrenByGlob(Glob glob, bool traceChildrenRecursion)
        public void AllowComponentType(HashSet<Type> compTypeSet)
        public void RequireComponent_ExactlyOne<T>() where T : Component

        public void RequireComponent<T>(Func<T, bool> checkFun, string errorInfo) where T : Component

        public void ReqCompProp<T, PropType>(string propPath, Func<T, PropType, bool> checkFun,
          string errorInfo) where T : Component
        public void ReqCompProp_ObjectRef<T>(string propPath, Func<T, UnityEngine.Object, bool> checkFun,
          string errorInfo) where T : Component
        ……

}
```

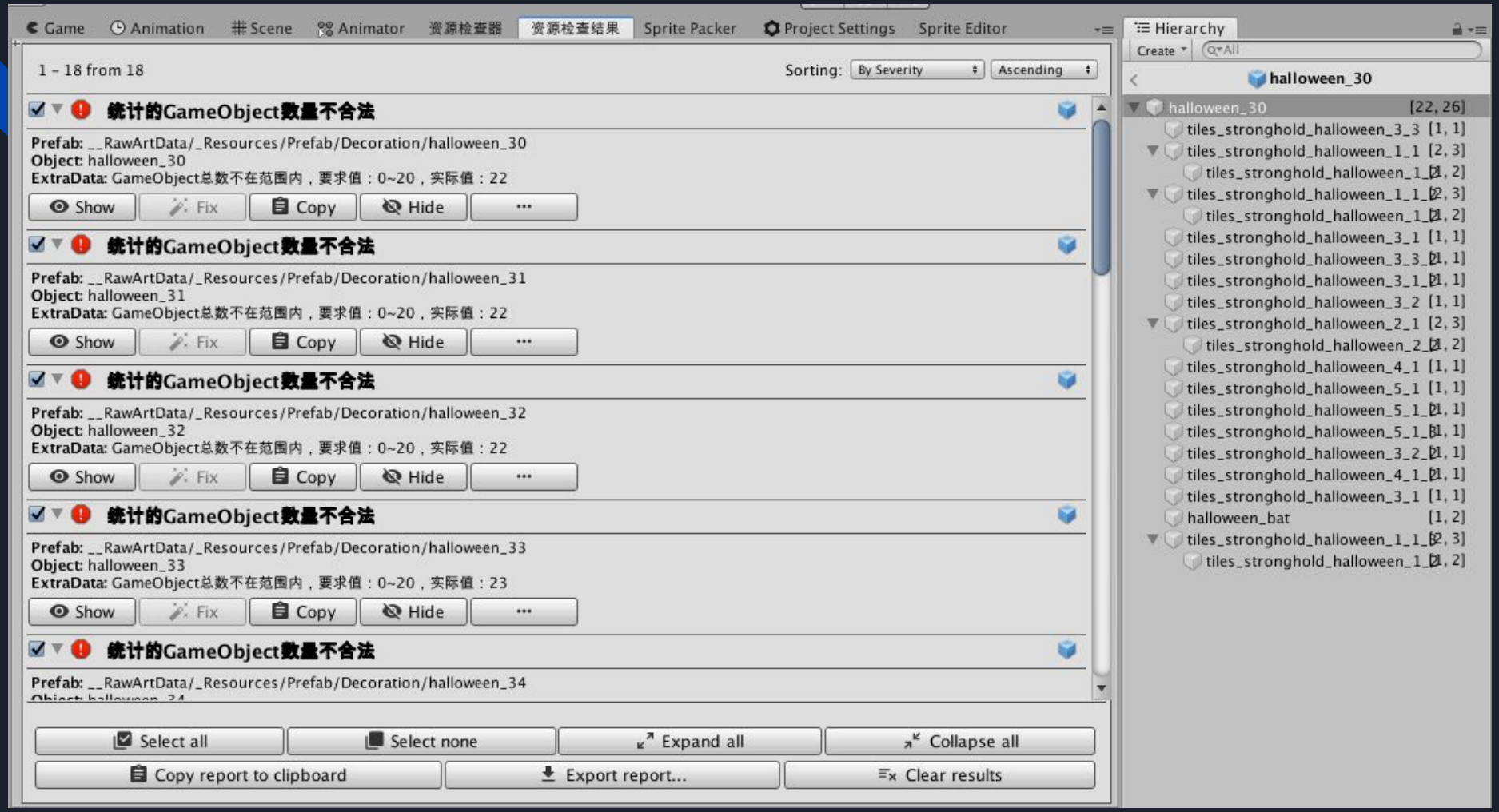
使用:设计并实现资源检查器

专门对某一类资源做检查的类

```
[Detector("美术资源")]
public class Detector_RawArtData_March : BaseIssueDetector
{
        public override string GetName() { return "March文件夹的序列帧"; }

        public override string GetDescription() { return "序列帧规范性检查"; }

        public override void BeforeDetect() {}

        public override List<string> GetTargetAssetList()
        {
                var result = EditorUtils.CollectAssetFiles("Assets/__RawArtData/March", "*.png");
                return result;
        }
```

```
public override void CheckAsset(string assetPath, UnityEngine.Object asset)
{
  //camera1_01  UnityEngine.Texture2D
  {
    var texture2d = AssetDatabase.LoadAssetAtPath<Texture2D>(assetPath);
    var sor = SerializedObject_Requirement<Texture2D>.Create(texture2d, ReportIssue, RecordSeverityEx.Error);
    sor.ReqProp_Int("m_ForcedFallbackFormat", (comp, value) => { return value == 4; }, "必须是4"); //【Forced Fallback Format】:4;
    sor.ReqProp_Bool("m_DownscaleFallback", (comp, value) => { return value == false; }, "必须是false"); //【Downscale Fallback】:false;
    sor.ReqProp_Int("m_Width", (comp, value) => { return value <= 256; }, "必须小于256"); //【Width】:128;
    sor.ReqProp_Int("m_Height", (comp, value) => { return value <= 256; }, "必须小于256"); //【Height】:128;
    //sor.ReqProp_Int("m_CompleteImageSize", (comp, value) => { return value <= 256 * 256; }, "必须是65536"); //【Complete Image Size】
:65536;
    sor.ReqProp_Int("m_MipCount", (comp, value) => { return value == 1; }, "必须是1"); //【Mip Count】:1;
    sor.ReqProp_Bool("m_IsReadable", (comp, value) => { return value == false; }, "必须是false"); //【Is Readable】:false;
    sor.ReqProp_Bool("m_StreamingMipmaps", (comp, value) => { return value == false; }, "必须是false"); //【Streaming Mipmaps】:false;
    sor.ReqProp_Int("m_StreamingMipmapsPriority", (comp, value) => { return value == 0; }, "必须是0"); //【Streaming Mipmaps Priority】:0;
    sor.ReqProp_Int("m_ImageCount", (comp, value) => { return value == 1; }, "必须是1"); //【Image Count】:1;
    sor.ReqProp_Int("m_TextureSettings.m_FilterMode", (comp, value) => { return value == 1; }, "必须是1"); //【Filter Mode】:1;
```


Game
Animation
Scene
Animator
资源检查器
资源检查结果
Sprite Packer
Project Settings
Sprite Editor
程序预制体
美术资源
项目设置
批量执行
Detectors
3 items
Name
Description
Class Name
Execute
March文件夹的序列帧
序列帧规范性检查
Detector_RawArtData_Marc
执行
万圣节地块
万圣节地块
Detector_RawArtData_Reso
执行
英雄Spine的Prefab
英雄Spine的Prefab检查
Detector_RawArtData_Reso
执行
收集检查器
全部执行


Game
Animation
Scene
Animator
资源检查器
资源检查结果
Sprite Packer
Project Settings
Sprite Editor
1 - 18 from 18
Sorting: By Severity Ascending
统计的GameObject数量不合法
Prefab: __RawArtData/_Resources/Prefab/Decoration/halloween_30
Object: halloween_30
ExtraData: GameObject总数不在范围内，要求值：0~20，实际值：22
Show
Fix
Copy
Hide
统计的GameObject数量不合法
Prefab: __RawArtData/_Resources/Prefab/Decoration/halloween_31
Object: halloween_31
ExtraData: GameObject总数不在范围内，要求值：0~20，实际值：22
Show
Fix
Copy
Hide
统计的GameObject数量不合法
Prefab: __RawArtData/_Resources/Prefab/Decoration/halloween_32
Object: halloween_32
ExtraData: GameObject总数不在范围内，要求值：0~20，实际值：22
Show
Fix
Copy
Hide
统计的GameObject数量不合法
Prefab: __RawArtData/_Resources/Prefab/Decoration/halloween_33
Object: halloween_33
ExtraData: GameObject总数不在范围内，要求值：0~20，实际值：23
Show
Fix
Copy
Hide
统计的GameObject数量不合法
Prefab: __RawArtData/_Resources/Prefab/Decoration/halloween_34
Select all
Select none
Expand all
Collapse all
Copy report to clipboard
Export report...
Clear results
Hierarchy
Create
halloween_30
halloween_30 [22, 26]
tiles_stronghold_halloween_3_3 [1, 1]
tiles_stronghold_halloween_1_1 [2, 3]
tiles_stronghold_halloween_1_1 [1, 2]
tiles_stronghold_halloween_1_1_1 [2, 3]
tiles_stronghold_halloween_1_1 [1, 2]
tiles_stronghold_halloween_3_1 [1, 1]
tiles_stronghold_halloween_3_3_1 [1, 1]
tiles_stronghold_halloween_3_1_1 [1, 1]
tiles_stronghold_halloween_3_2 [1, 1]
tiles_stronghold_halloween_2_1 [2, 3]
tiles_stronghold_halloween_2_1 [1, 2]
tiles_stronghold_halloween_4_1 [1, 1]
tiles_stronghold_halloween_5_1 [1, 1]
tiles_stronghold_halloween_5_1_1 [1, 1]
tiles_stronghold_halloween_5_1_3 [1, 1]
tiles_stronghold_halloween_3_2_1 [1, 1]
tiles_stronghold_halloween_4_1_1 [1, 1]
tiles_stronghold_halloween_3_1 [1, 1]
halloween_bat [1, 2]
tiles_stronghold_halloween_1_1_3 [2, 3]
tiles_stronghold_halloween_1_1 [1, 2]

日常.md | 2020_05_14_23_31_07_万圣节地块.md × | 445.Research_ResearchBaseDlg.OpenDlg.md

ers > yupinpan > Desktop > koa_master_pyp_read > Game > AssetValidatorResult > 2020_05_14_23_31_07_万圣

```
# 错误总数：18
# 清单
---
1. 统计的GameObject数量不合法
```
Prefab: __RawArtData/_Resources/Prefab/Decoration/halloween_30
Object: halloween_30
ExtraData: GameObject总数不在范围内，要求值：0~20，实际值：22
```
---
2. 统计的GameObject数量不合法
```
Prefab: __RawArtData/_Resources/Prefab/Decoration/halloween_31
Object: halloween_31
ExtraData: GameObject总数不在范围内，要求值：0~20，实际值：22
```
---
3. 统计的GameObject数量不合法
```
Prefab: __RawArtData/_Resources/Prefab/Decoration/halloween_32
Object: halloween_32
ExtraData: GameObject总数不在范围内，要求值：0~20，实际值：22
```
```

- Effect_Prefab_Show : 102
- Effect_UI_Material : 0

功能性资源检测

- [redacted]40
- UI面板 : 3058
- UI特效 : 338

配置表资源检测

- Announce : 0
- AtlasResConfig : 0
- BuildHarvest : 1
- BuildNeutral : 1
- BuildPlunder : 0
- Captive : 4
- NpcBuild : 0
- NpcConfig : 391
- TeamBase : 213
- TeamBattleAttr : 3007
- TeamGuanzhi : 0
- TeamLevel : 0
- TeamQuality : 261
- TeamRoleModel : 1341

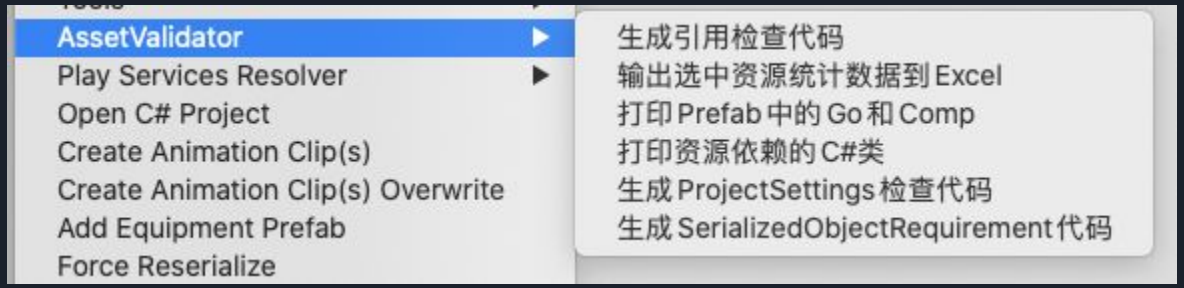
耗时统计

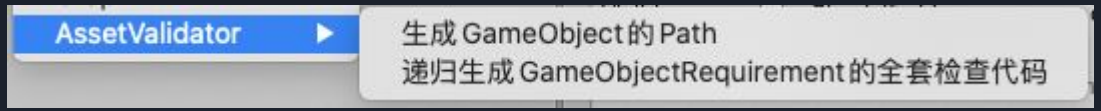
- 开始时间： 2019[redacted]
- 结束时间： 2019[redacted]
- 总时长： 00:08:19.167

[redacted]资源检查构建详情

- 触发分支: origin/mail-by-python
- 版本信息: 686969134a33728460a3f66e8f441b9330163ca3
- 构建序号: 11
- 构建详情:[redacted]
- 附件地址:[redacted]/assetst-validation-report.11.tar.gz

代码生成器





看看示例

- 可检查单一资源
- 可检查引用依赖
- 尽可能多地覆盖属性
- 用回调函数的形式确定资源是否正确，有很强的可扩展性
- 统一的框架处理，易于维护、统计输出

TODO

- 覆盖项目的关键资源，形成规范和检查器
- CI集成、钉钉通知
- 粒子系统检查：Editor自动运行，检查Overdraw、粒子峰值

问题？

UIOptimize：UI性能优化工具集

导出检查

UI拆分和分块加载

UI控件绑定代码自动生成

UICustomContainer

UI导出检查

1. 导出前会执行错误检查，错误未解决前无法导出
2. 禁止Disable的GameObject（目前是警告，不影响导出；以后会换为Error，程序必须用代码控制）
3. 禁止Disable的Component
4. 禁止Missing的Prefab引用
5. 禁止Missing的Component
6. 禁止同一GameObject下有重复的Component（有白名单）
7. 所有Component的属性中不能有Missing的引用
8. 禁止引用其他Prefab的子节点
9. 禁止Disconnected的Prefab引用
10. 自动移除冗余的资源依赖（U3D版本升级导致）
11. 禁止引用本场景内、但是非该UI节点下的其它对象（防止在场景中编辑UI的时候胡乱拖拽）
12. 引用了非static+default AB包中的纹理会给出警告
13. GameObject数量超过200会给出警告
14. 检测除了Common图集外其它Texture引用数量、大小是否超出上限
15. 引用纹理的ReadWrite检查、压缩格式是否正确
16. Prefab必须位于Component目录下
17. 允许Component的白名单，不认识的Component会给警告
18. 禁止设置NGUI的EventDelegate
19. 与Maintainer插件集成，错误和警告会输出到列表，可快速跳转到出错的GameObject或Component。部分错误提供一键修复功能。


Game
Animation
Scene
Animator
资源检查器
资源检查结果
Sprite Packer
Project Settings
1 - 74 from 74
Sorting:
By Severity
ExtraData: 不允许设置EventDelegate，请在代码中绑定事件函数！
Show
Fix
Copy
Hide
NGUI在Inspector中绑定事件
Scene: __UIData/UIScenes/ItemStore
Object: UIManager/UI Root/UI2DCamera/ItemStoreAndShopDlg/Content/_Shop_Loader/Shop/ItemNew (1)/ItemGroip/StoreItemS
Component: UIEventTrigger
Property: On Click[0]
ExtraData: 不允许设置EventDelegate，请在代码中绑定事件函数！
Show
Fix
Copy
Hide
NGUI在Inspector中绑定事件
Scene: __UIData/UIScenes/ItemStore
Object: UIManager/UI Root/UI2DCamera/ItemStoreAndShopDlg/Content/_Shop_Loader/Shop/ItemNew (1)/ItemGroip/StoreItemS
Component: UIButton
Property: On Click[0]
ExtraData: 不允许设置EventDelegate，请在代码中绑定事件函数！
Show
Fix
Copy
Hide
NGUI在Inspector中绑定事件
Scene: __UIData/UIScenes/ItemStore
Object: UIManager/UI Root/UI2DCamera/ItemStoreAndShopDlg/Content/_Shop_Loader/Shop/ItemNew (1)/ItemGroip/StoreItemS
Component: UIEventTrigger
Property: On Click[0]
ExtraData: 不允许设置EventDelegate，请在代码中绑定事件函数！
Show
Fix
Copy
Hide
Select all
Select none
Expand all
Collapse all
Copy report to clipboard
Export report...
Clear results


UI Prefab Export Config (Script)
Script
UIPrefabExportConfig
Export Path
Assets/__UIData/_Resources/Prefab/UI/Dialogs/ItemStor
导出Prefab路径： Assets/__UIData/_Resources/Prefab/UI/Dialogs/ItemStore/ItemStor
Find
Scene路径： Assets/__UIData/UIScenes/ItemStore.unity
Open
Choose Export Path
Export
Export Opt
Check Opt (仅检查不导出)

UI拆分和分块加载

需求

- UE在场景中编辑界面，所见即所得
  - 允许界面比较复杂
- 导出时做拆分，分成独立的Prefab
- 运行时按需加载拆分后的Prefab

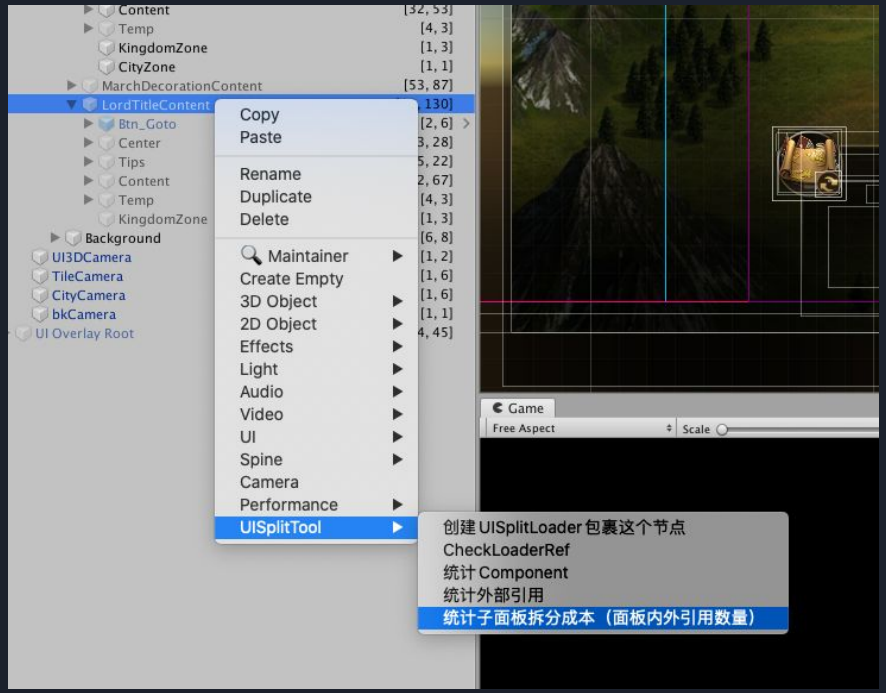
方案

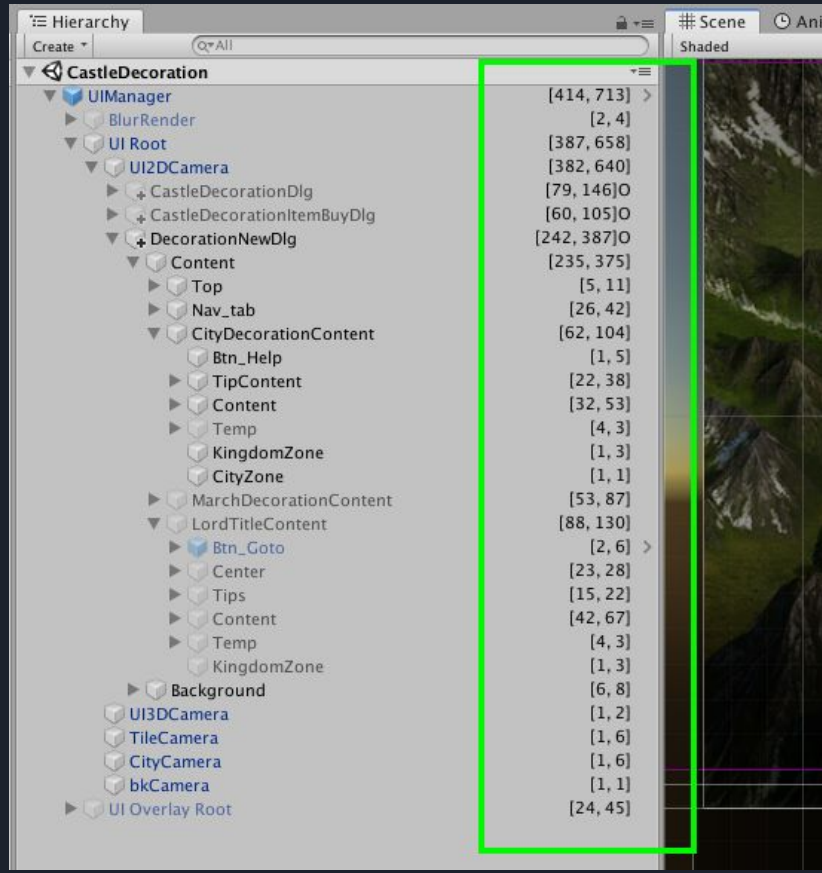
- 导出的时候拆散Prefab，记录原来位置的GameObject的Path，运行时找到并加载

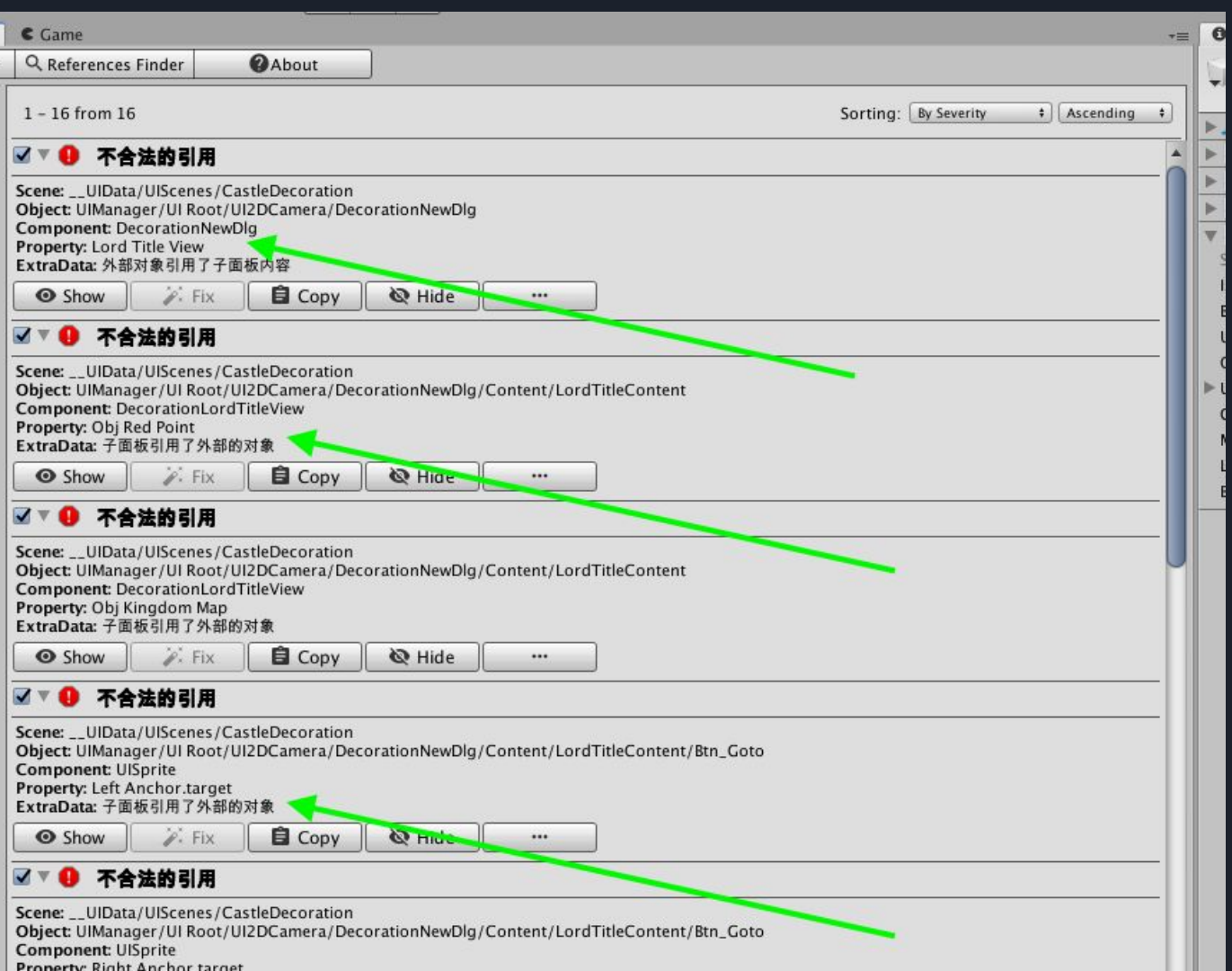
难点

- 自动拆分的逻辑实现
- 正确性保证、错误提示

场景结构复杂度显示 & 拆分成本分析



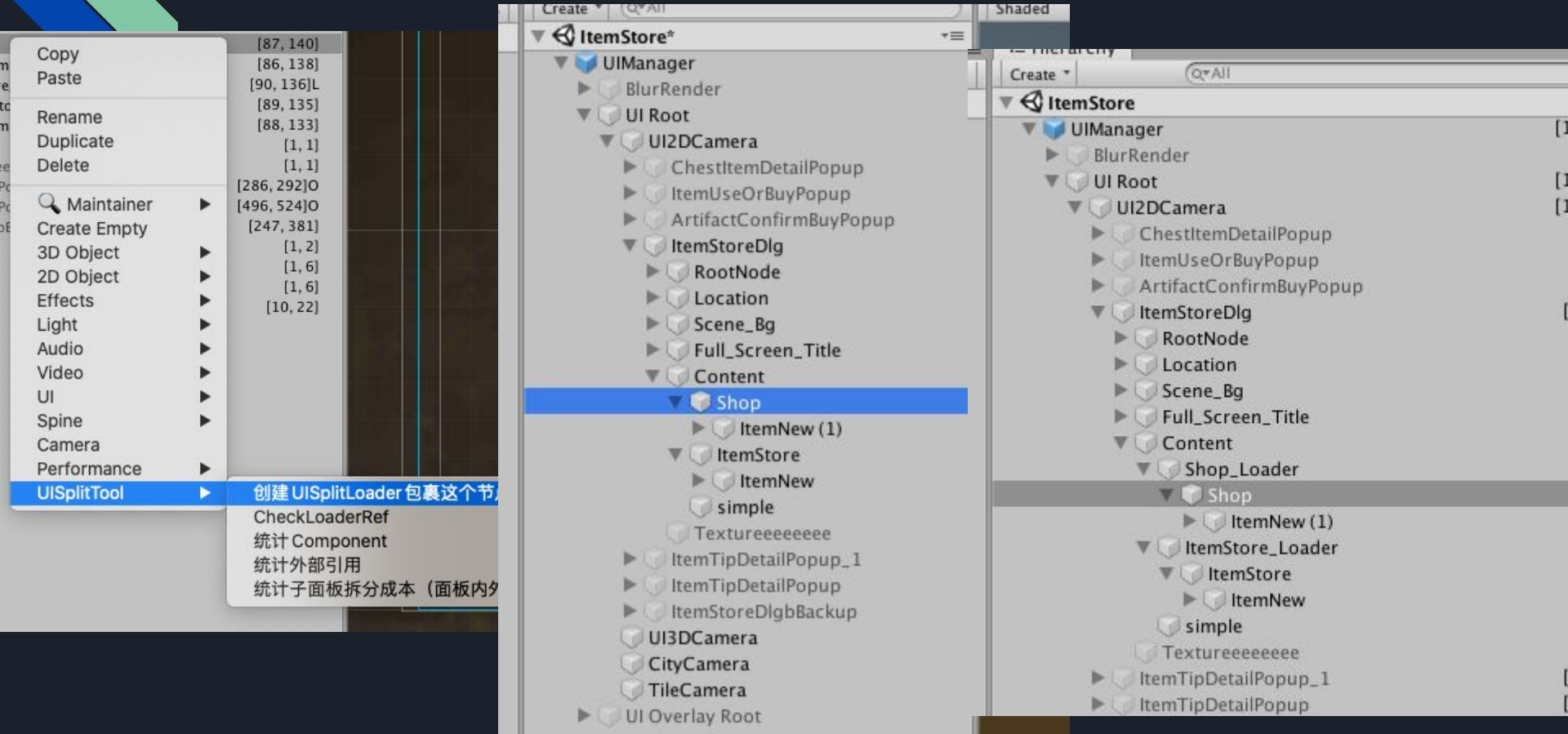


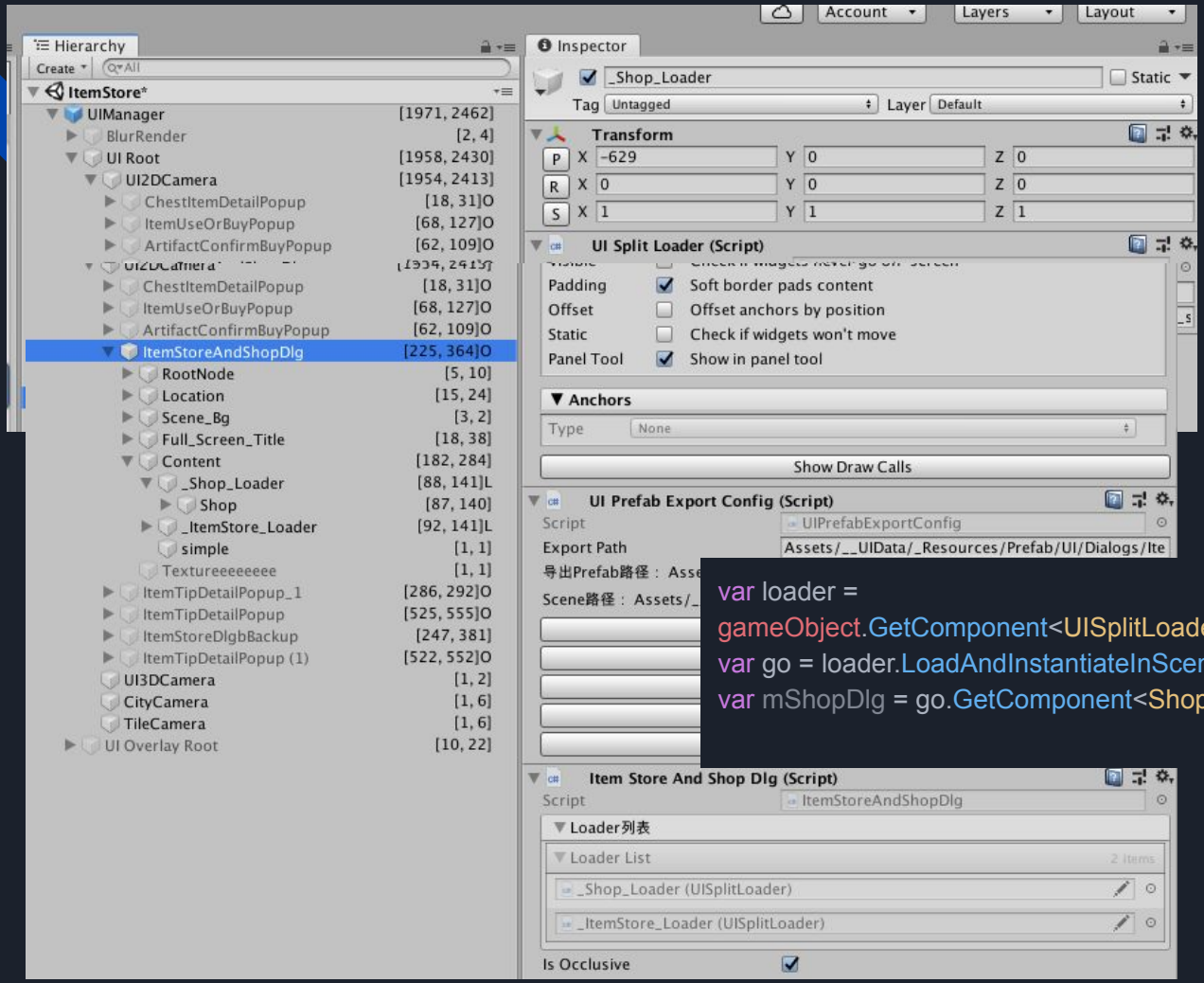


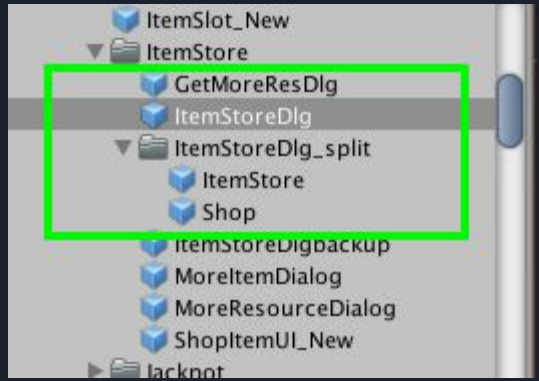
典型的内外引用

- 根面板有属性指向了子面板中的对象，这是因为子面板的对象要在Dlg、Popup的MonoBehaviour类中用到。
- 子面板的布局Anchor属性引用了外部的UIPanel等对象
- 子面板有一个父面板的引用。

拆分面板

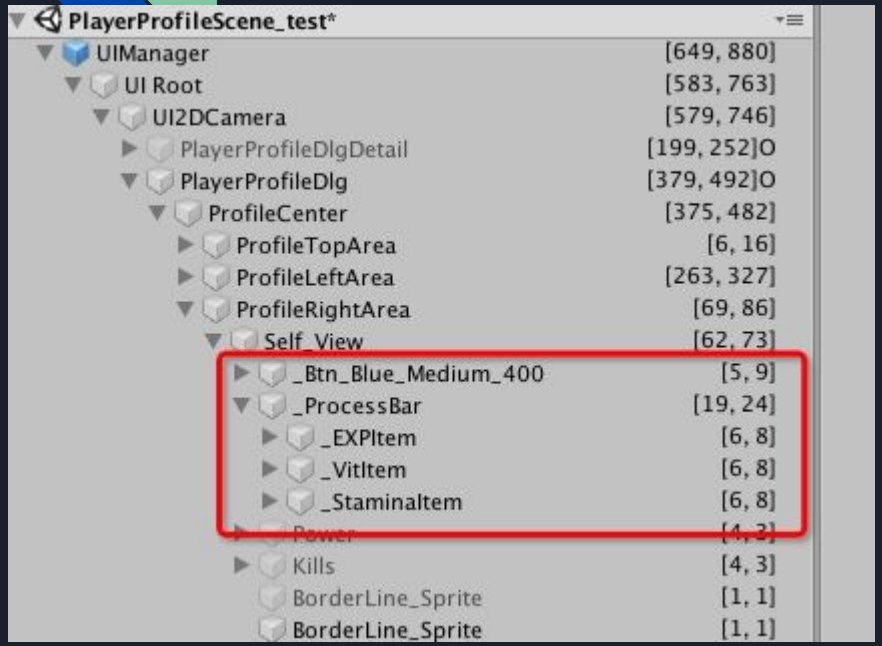

Copy
Paste
Rename
Duplicate
Delete
Maintainer
Create Empty
3D Object
2D Object
Effects
Light
Audio
Video
UI
Spine
Camera
Performance
UISplitTool
创建UISplitLoader包裹这个节
CheckLoaderRef
统计 Component
统计外部引用
统计子面板拆分成本（面板内
[87, 140]
[86, 138]
[90, 136]L
[89, 135]
[88, 133]
[1, 1]
[1, 1]
[286, 292]O
[496, 524]O
[247, 381]
[1, 2]
[1, 6]
[1, 6]
[10, 22]
Create
ItemStore*
UIManager
BlurRender
UI Root
UI2DCamera
ChestItemDetailPopup
ItemUseOrBuyPopup
ArtifactConfirmBuyPopup
ItemStoreDlg
RootNode
Location
Scene_Bg
Full_Screen_Title
Content
Shop
ItemNew (1)
ItemStore
ItemNew
simple
Textureeeeeeee
ItemTipDetailPopup_1
ItemTipDetailPopup
ItemStoreDlgbBackup
UI3DCamera
CityCamera
TileCamera
UI Overlay Root
Shaded
Create
All
ItemStore
UIManager
BlurRender
UI Root
UI2DCamera
ChestItemDetailPopup
ItemUseOrBuyPopup
ArtifactConfirmBuyPopup
ItemStoreDlg
RootNode
Location
Scene_Bg
Full_Screen_Title
Content
Shop_Loader
Shop
ItemNew (1)
ItemStore_Loader
ItemStore
ItemNew
simple
Textureeeeeeee
ItemTipDetailPopup_1
ItemTipDetailPopup

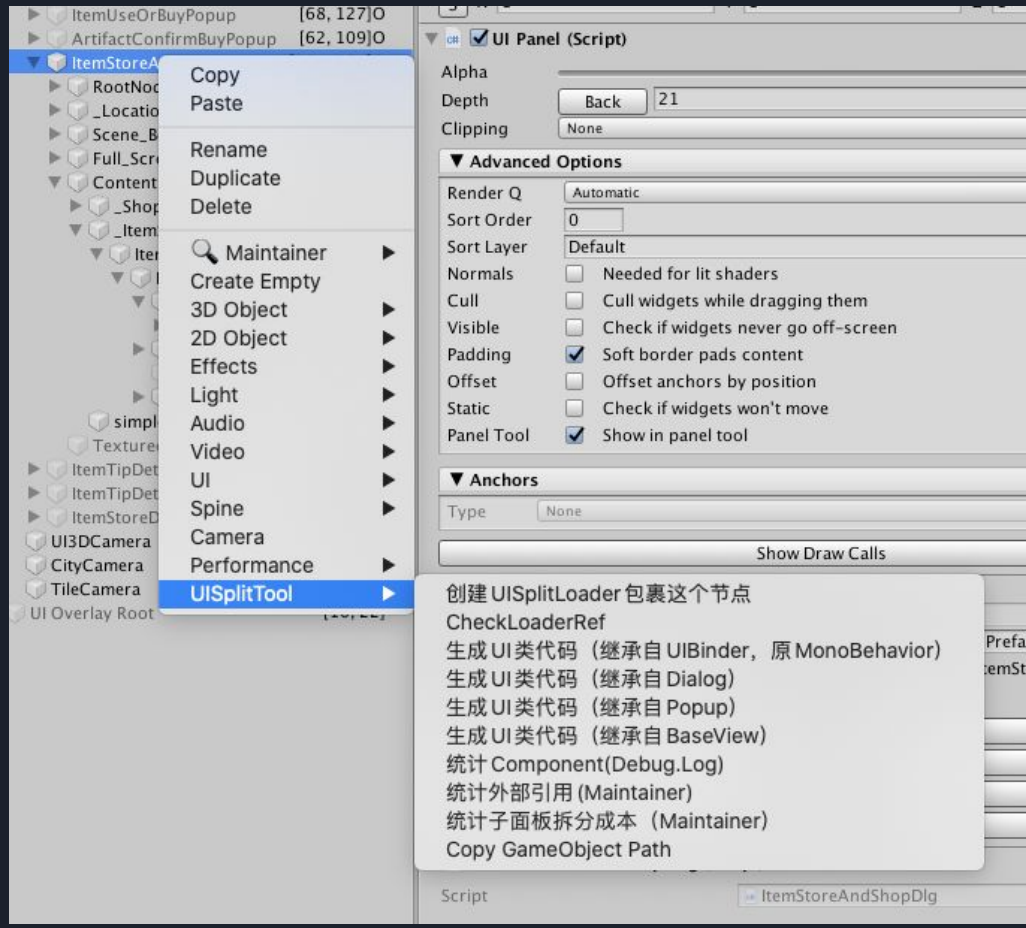


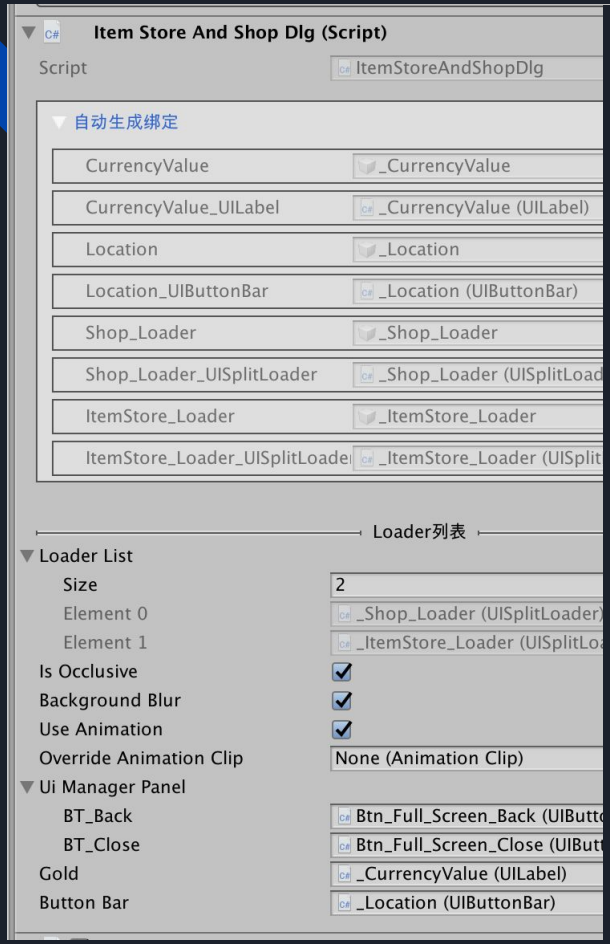



```
var loader =
gameObject.GetComponent<UISplitLoaderContainer>().GetLoader("Shop");
var go = loader.LoadAndInstantiateInScene();
var mShopDlg = go.GetComponent<ShopDlg>();
```

UI控件绑定代码生成


PlayerProfileScene_test*
UIManager [649, 880]
UI Root [583, 763]
UI2DCamera [579, 746]
PlayerProfileDlgDetail [199, 252]O
PlayerProfileDlg [379, 492]O
ProfileCenter [375, 482]
ProfileTopArea [6, 16]
ProfileLeftArea [263, 327]
ProfileRightArea [69, 86]
Self_View [62, 73]
_Btn_Blue_Medium_400 [5, 9]
_ProcessBar [19, 24]
_EXPItem [6, 8]
_VitItem [6, 8]
_StaminaItem [6, 8]
Kills [4, 3]
BorderLine_Sprite [1, 1]
BorderLine_Sprite [1, 1]


Copy
Paste
Rename
Duplicate
Delete
Maintainer
Create Empty
3D Object
2D Object
Effects
Light
Audio
Video
UI
Spine
Camera
Performance
UISplitTool
UI Panel (Script)
Alpha
Depth
Back
21
Clipping
None
Advanced Options
Render Q
Automatic
Sort Order
0
Sort Layer
Default
Normals
Needed for lit shaders
Cull
Cull widgets while dragging them
Visible
Check if widgets never go off-screen
Padding
Soft border pads content
Offset
Offset anchors by position
Static
Check if widgets won't move
Panel Tool
Show in panel tool
Anchors
Type
None
Show Draw Calls
创建UISplitLoader包裹这个节点
CheckLoaderRef
生成UI类代码（继承自UIBinder，原MonoBehavior)
生成UI类代码（继承自Dialog)
生成UI类代码（继承自Popup)
生成UI类代码（继承自BaseView)
统计Component(Debug.Log)
统计外部引用(Maintainer)
统计子面板拆分成本（Maintainer)
Copy GameObject Path
Script
ItemStoreAndShopDlg




```
namespace UIOptimize.ItemStore.ItemStoreAndShopDlg
{
  public partial class ItemStoreAndShopDlg : UI.Dialog
  {
      [Sirenix.OdinInspector.ReadOnly]
      [Sirenix.OdinInspector.FoldoutGroup("自动生成绑定")]
      [SerializeField]
      private GameObject currencyValue;
      [Sirenix.OdinInspector.ReadOnly]
      [Sirenix.OdinInspector.FoldoutGroup("自动生成绑定")]
      [SerializeField]
      private UILabel currencyValue_UILabel;
#if UNITY_EDITOR
      public override void BindReference()
      {
          currencyValue =
transform.Find("RootNode/PremiumCurrencyBtn/_CurrencyValue").gameObject;
          currencyValue_UILabel =
transform.Find("RootNode/PremiumCurrencyBtn/_CurrencyValue").GetComponents<UILabel>()[0];


.....
```

嵌套处理

- ItemStoreAndShopDlg [C#]
  - _BackButton
  - _ShopPanelLoader
    - ShopPanel [C#]
      - _curGoldCountLabel
      - _scrollView
      - _shopItemLoader
        - ShopItem[C#]
          - _nameLabel
          - _iconImage
  - _SalesmanComponent [C#]
    - _AvatarImage

UI重构验证

界面发生更改后，可以重新生成代 码，如果能够编译通过，则认为是没问题的

代码编译出错会自动回滚，可以给UE使用

UICustomContainer

```
public class UICustomContainer : UIWidgetContainer
{
        public delegate void RefreshCellFunc(int cellIndex, GameObject cell, object userData);
        public RefreshCellFunc OnRefreshCell;
        public CellData AddCell(int templateIndex, object userData = null)

        private bool UpdateCells()
        {
                FrameTimer.Reset();
                foreach (var cellData in _allCellData)
                {
                        CreateCellInstance(cellData);
                        //如果运算超时，直接return false
                        if (EnableFraming && FrameTimer.IsExceedMs(10))
                        {
                                return false;
                        }
                }

        }
}
```

TODO

- 通用化，可能不仅仅用于UI中
  - 拆分和加载
  - 代码生成和控件绑定
- KOA的Hud、Component的编辑和检查流程完善
- UI中所有的Particle异步加载
- UI相关Particle目录约定和规范检查
- UI Scene与UI Prefab的双向查找
- 参考AssetReference，实现一个统一的资源引用

问题？

Packages包管理机制

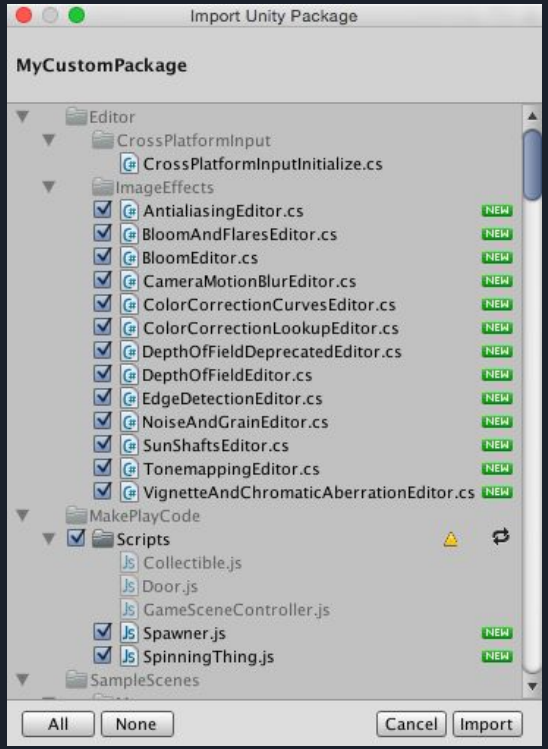
老的包管理的弊端:

- 同一个程序集
- 分散在各个目录
- 升级不方便
- 依赖管理混乱

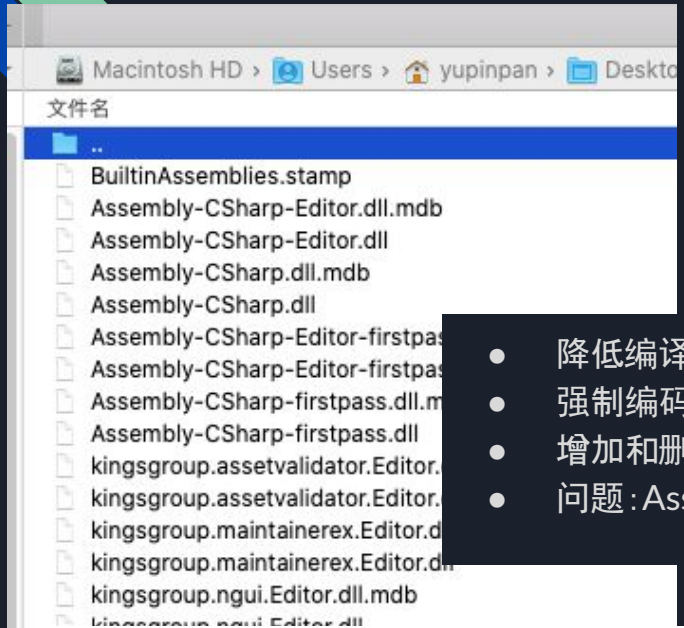
U3D在2017.2引入, 2018逐步完善

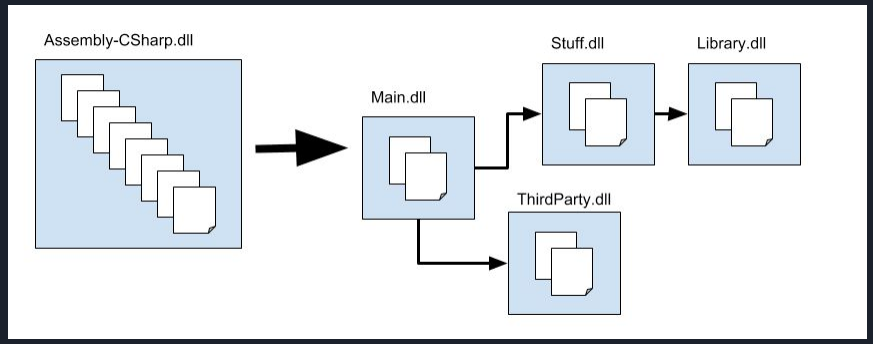
- 管理第三方库, 自动下载更新库版本
- 重点:强制、显式地划分程序集、并且定义程序集之间的依赖

文档:https://docs.google.com/document/d/1l-PgmOC4spiKH3jktqM26uGFwVBKnzNY2vXftYe-Tdg/edit#heading=h.j4gbcfjhxeuj

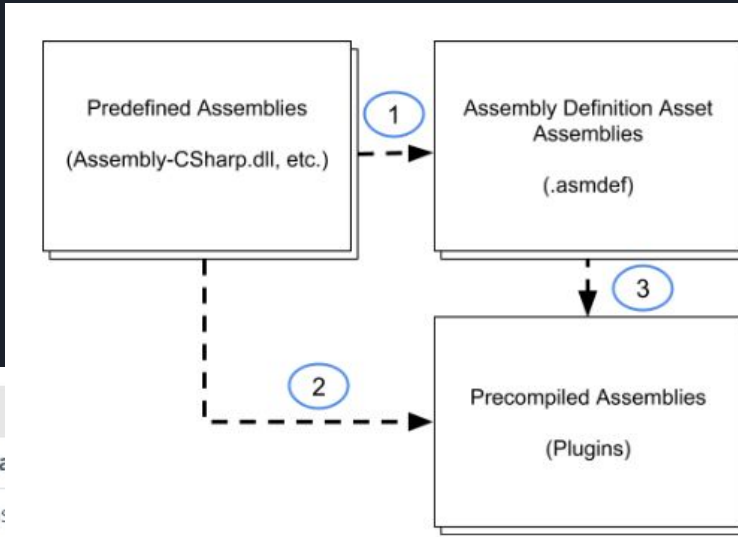


Assembly Definitions





- 降低编译时间
- 强制编码遵循依赖规则
- 增加和删除组件更加容易
- 问题：Assets目录下的不支持

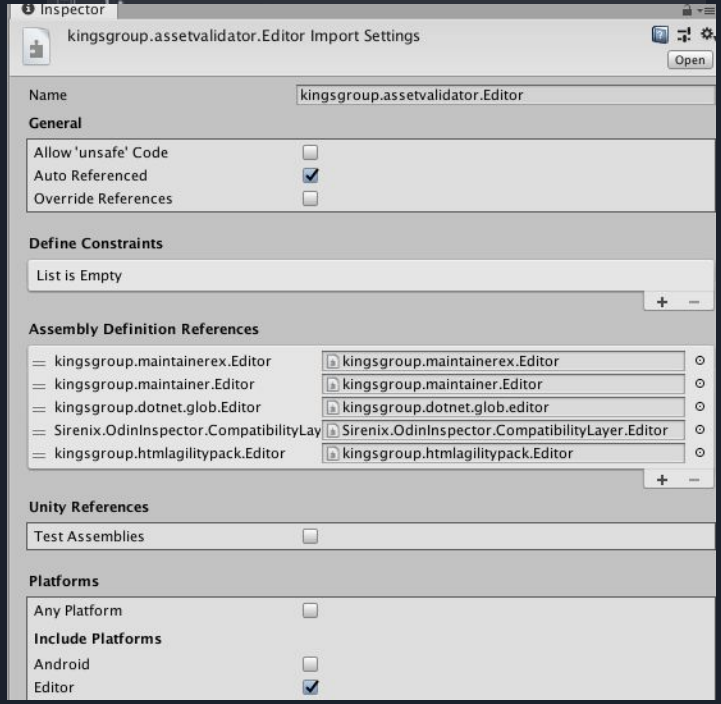


Phase	Assembly name	Script files
1	Assembly-CSharp-firstpass	Runtime scripts in folders called **Standard Assets**, Pro **Sta**
2	Assembly-CSharp-Editor-firstpass	Editor scripts in folders called Editor that are anywhere ins
3	Assembly-CSharp	All other scripts that are not inside a folder called Editor.
4	Assembly-CSharp-Editor	All remaining scripts (those that are inside a folder called Editor.

移动代码和资源到Packages目录

- 拟定包名 com.kingsgroup.xxxx
  - 包名必须唯一
- 在Packages目录下创建包的目录
  - 包名与目录名必须一致
- 编写Package.json
- 重启Editor
- 创建asmdef
  - 区分Runtime和Editor
  - 拖拽设置依赖
- 从Assets中移动.cs和.meta到包的目录下
  - 其它资源同理
- 正确填写Package.json中的依赖关系
  - 可用工具自动处理

看看示例



程序集之间依赖传递的方式

正向：直接调用类和函数、继承基类

反向：

- 提供回调函数给包
- 用Attribute反射创建对象

```
foreach (var assembly in AppDomain.CurrentDomain.GetAssemblies())
{
    var detectorTypes = assembly.GetTypes()
        .Where(x => typeof(AbstractPerfCodeInjectConfig).IsAssignableFrom(x) && !x.IsAbstract);

    foreach (var t in detectorTypes)
    {
        perfCodeInjectConfig = (AbstractPerfCodeInjectConfig)Activator.CreateInstance(t);
        Debug.Log("UIPerfCodeInject找到了Config类 " + t.FullName);
    }
}
```

安利一波KOA的包

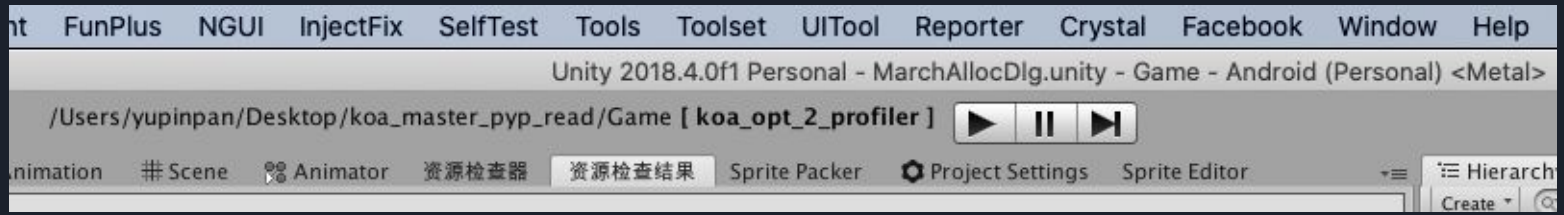
第三方

- unity-toolbar-extender：扩展标题栏
- unity-editor-spotlight：聚焦搜索
- Cecil：程序集解析和修改
- Odin Inspector：编辑器扩展
- Maintainer：资源检查工具
- UniExcel：输出Excel文件

自研

- RemoveEmptyDirectories：移除空目录
- ToolBarShowProject：显示分支和目录
- ScriptReloadTool：代码编译完成后的回调
- PerfTool：性能检测
- Asset Validator
- UIOptimize：UI相关工具集（建设中）

unity-toolbar-extender

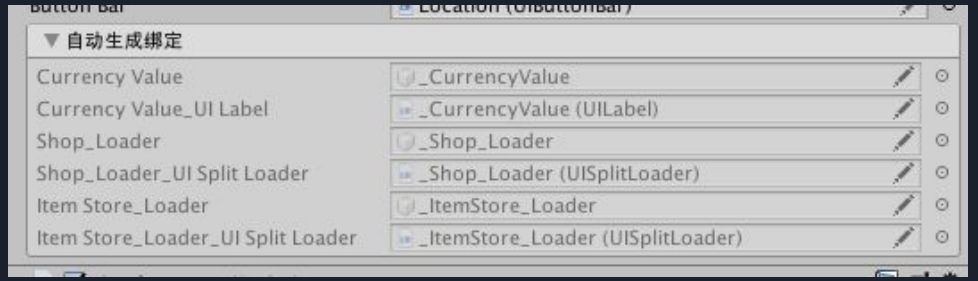


ScriptReloadTool

```
//设置这次编译成功后的回调
ScriptReloadTool.SetScriptReloadCallBackKey(Manually_Gen_AfterReloadBinderCode,
  new List<string>()
  {
      uniqId.ToString(),
      param.csFilePath,
  });

//设置这次编译失败后的回调
ScriptReloadTool.SetScriptCompileRecoveredCallBackKey(Manually_Check_GeneratedCodeCompileError);
//变更一组C#文件的内容
ScriptReloadTool.TryToChangeCSharpFileListContent(new List<CSFileChangeItem> {item}, force);
```

Odin Inspector

- https://odininspector.com/editor-windows
- 功能
  - 属性只读
  - 属性展开和收缩
  - Editor Window



MaintainerEx

- 功能
  - 遍历的工具库
    - GameObject
    - Component
    - Property
  - Issue Window
  - Find & Auto Fix

MaintainerEx API

- public static GameObjectIssueRecordEx Create_Go(string type, RecordSeverityEx severity, GameObject gameObject, string extra, GameObjectIssueFixFunction fixCb = null)
- public static GameObjectIssueRecordEx Create_Comp(string type, RecordSeverityEx severity, Component comp, string extra, GameObjectIssueFixFunction fixCb=null)
- public static AssetObjectIssueRecordEx CreateEx_Asset(string type, RecordSeverityEx severity, UnityEngine.Object asset, string extraInfo, AssetObjectIssueFixFunction fixCb = null)
- public static void TraversalGameObjectListEx(List<GameObject> goRootList, bool includeTransform, bool onlyVisibleProperty, CSTraverseTools.GameObjectTraverseCallback goCB, Action<Component> compCB, Action<SerializedProperty> propCB)
- public static void ShwoIssuesInWindow(List<IssueRecord> allIssues)

Uni-Excel

Export Excel

https://github.com/tonyqus/npoi

```
public static void ExportWorkSheet_Simple(string saveFilePath, List<object> title, int rowCount,
Func<int, List<object>> fillLineCb)
```

Mac安装：

https://stackoverflow.com/questions/20820139/unity-and-system-drawing-on-os-x

TODO

- 继续提取底层模块到Package（目前提取出了1/3的cs文件）
  - UIManager、AssetManager、Network、配置表读取……
- 开源
  - Package的源码放到GitLab
  - Test、Document
  - 用开源软件的开发方式进行管理
    - Issue、Merge Request
- 把玩法、渲染等相对通用性的模块也拆入Package
  - 大地图显示和加载
  - 寻路
  - 后期处理
- 美术资源拆出Package
  - 限定MonoBehaviour的使用
- 尝试所有业务逻辑也用Package拆分
  - 严格限定依赖
  - Assets里面只有美术资源？

问题？

Thanks

问题？