

# C5 UI 资源以及制作规范

## 前言

本文档主要用于阐述 C5 项目的 UI 资源使用规范以及 UI 制作流程规范，包括不限于资源格式、资源大小、资源组织形式以及各类资源工具等介绍。

目前基于 C5 战斗外跑测数据分析：[UI 各类设备性能跑测分析](#)

## 目标

- 高端机、中高端机：60 帧，无发热（峰值小于<800mA）
- 中低端机、低端机：30 帧，无发热（峰值小于<800mA）

## UI 运行时总规范

规范项	局外阈值	局内阈值	检测方式
<b>DrawCall</b>	高：150 中：150 低：120	高：60 中：60 低：50	性能后台预警
<b>Overdraw</b>	高：10 中：10 低：8	高：6 中：6 低：5	性能后台预警

## UI Prefab 规范

注：下述规范只针对非战斗部分，战斗部分的优化目标不设上限

CI：自动化跑测。Editor：编辑器下保存时检测并弹框

规范项	局外阈值	局内阈值	检测方式
<b>DrawCall</b>	高：50 中：50 低：40	高：50 中：50 低：40	CI+Editor
<b>Overdraw</b>	高：6 中：6	高：2 中：2	CI+Editor

	低： 5	低： 2	
图集个数	高： 公用<=1， 系统<=1 中： 公用<=1， 系统<=1 低： 公用<=1， 系统<=1	同局外	CI+Editor
图集总尺寸	高： <= 4096x2048 中： <= 4096x2048 低： <= 4096x2048	同局外	CI+Editor
散图个数（待讨论）	高： 30 中： 30 低： 25	高： 10 中： 10 低： 8	CI+Editor
散图总尺寸	高： <=4096x4096 中： <=4096x4096 低： <=4096x2048	同局外	CI+Editor
非必要动效个数（待讨论） Particle+动效类嵌套 prefab+spine	高： 5 中： 5 低： 0?	高： 5 中： 5 低： 0?	CI+Editor
UIParticle 数量	高： 5 中： 3 低： 3	高： 0 中： 0 低： 0	CI+Editor
<del>Animation</del> 资源数量	高： <del>10</del> 中： <del>10</del> 低： <del>8</del>	高： <del>10</del> 中： <del>10</del> 低： <del>8</del>	CI+Editor
UIWidget 数量 （后期）	高： <= 250 中： <= 200 低： <= 150	高： <= 250 中： <= 200 低： <= 150	CI+Editor
加载耗时（高端机异步？）	高： <del>&lt;= 500ms</del> 中： <del>&lt;= 600ms</del> 低： <del>&lt;= 800ms</del>	高： <del>&lt;= 500ms</del> 中： <del>&lt;= 600ms</del> 低： <del>&lt;= 800ms</del>	CI
界面实例化耗时	低： <= 125ms（参考卡顿）		CI

贴图实际显示像素密度	待定	待定	CI+Editor
无效资源节点	$\leq 0$	$\leq 0$	CI+Editor

**特殊 ICON 图：** 这些属于特殊的贴图，可以后续作为动态图集或者特殊图集进行约束

- 道具图标
- 技能图标
- 英雄头像
- ...

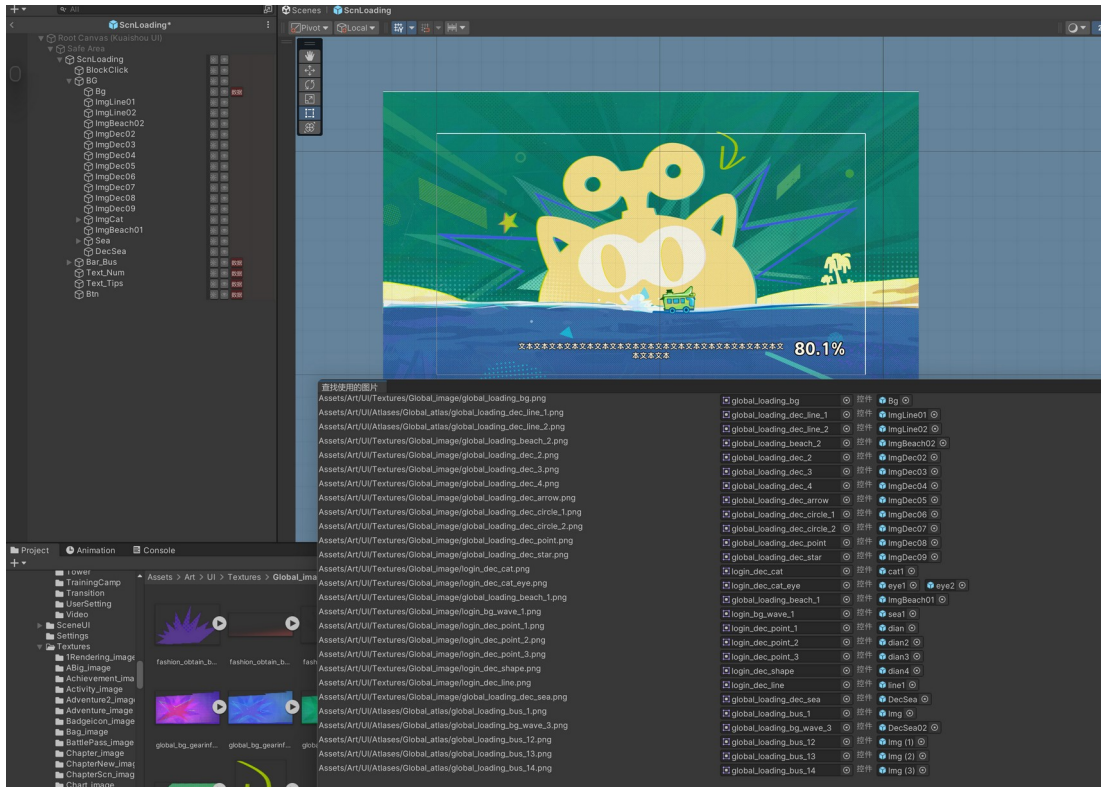
## UI 贴图规范

规范项	阈值	检测方式
贴图利用率	$\geq 65\%$ （待讨论）	CI
图集利用率	$\geq 70\%$	CI
贴图相似率阈值	$\geq 80\%$	CI
相似散图转图集阈值	$\geq 3$	CI
最大重复像素率	$\geq 80\%$	CI
散图尺寸规则	4 的倍数	CI（已有）
散图最大尺寸	$\leq 1920 \times 1080$ (按面积算就行)	CI
图集最大尺寸	$\leq 2048 \times 2048$ （common） $\leq 2048 \times 1024$ （其他）	CI
图集 <b>sprite</b> 最大尺寸	$\leq 256 \times 256$	CI
<b>UI 图片 PSNR 阈值</b>	待定	强制

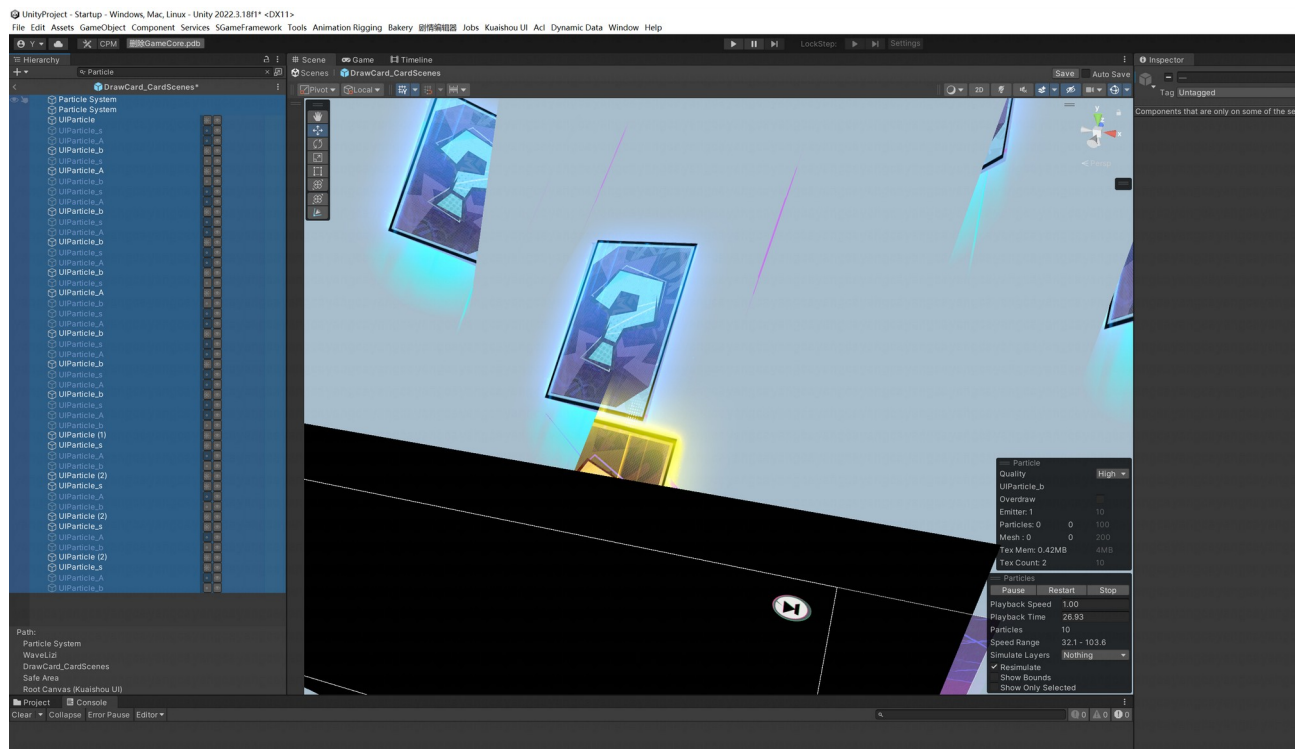
## 错误示例

- 1.多图集、尺寸过多：[C5 UI 资源以及制作规范](#)
- 2.多散图、尺寸过多：[C5 UI 资源以及制作规范](#)

散图个数超过 20 个:

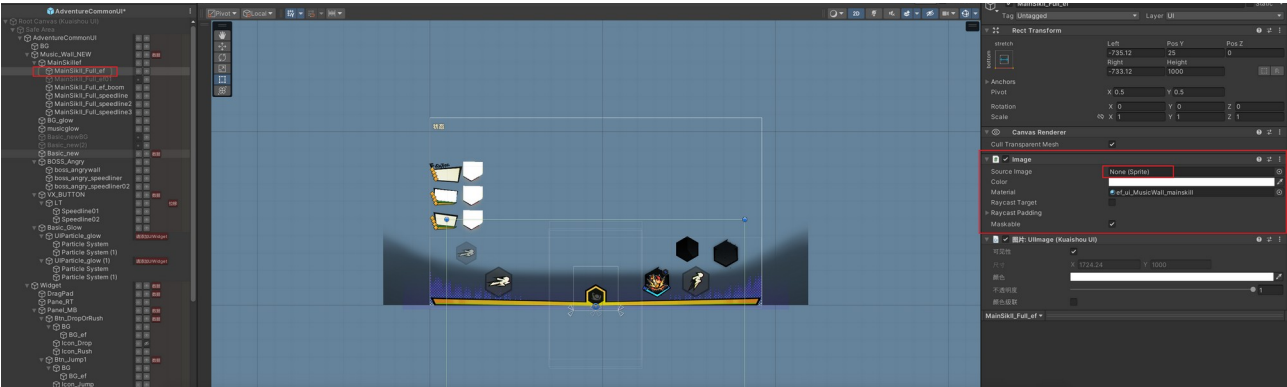


- 3.动画过多: C5 UI 资源以及制作规范
- 4.动效多

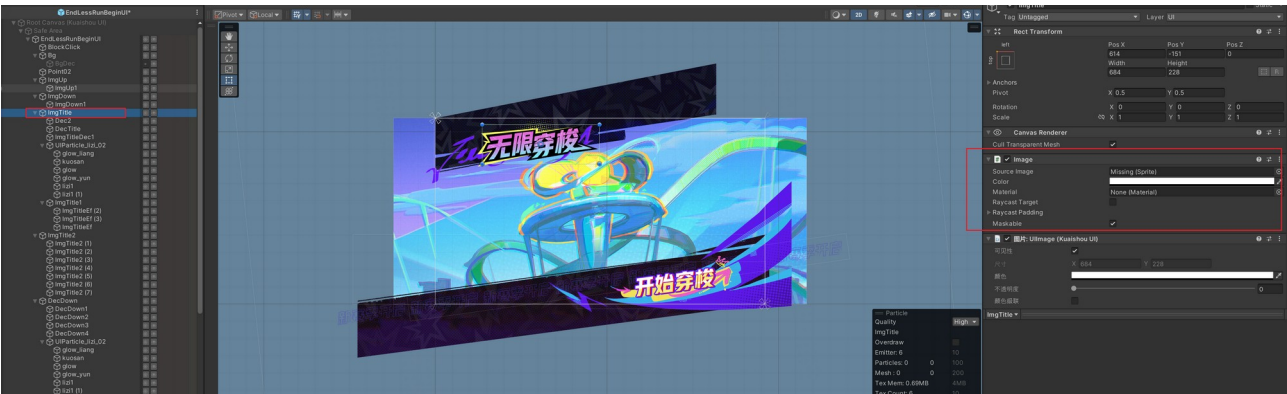


5.无效 UIWidget

● 这个是动效挂了材质但没挂图的节点



● 这个是当父节点的空节点，也是空图片节点

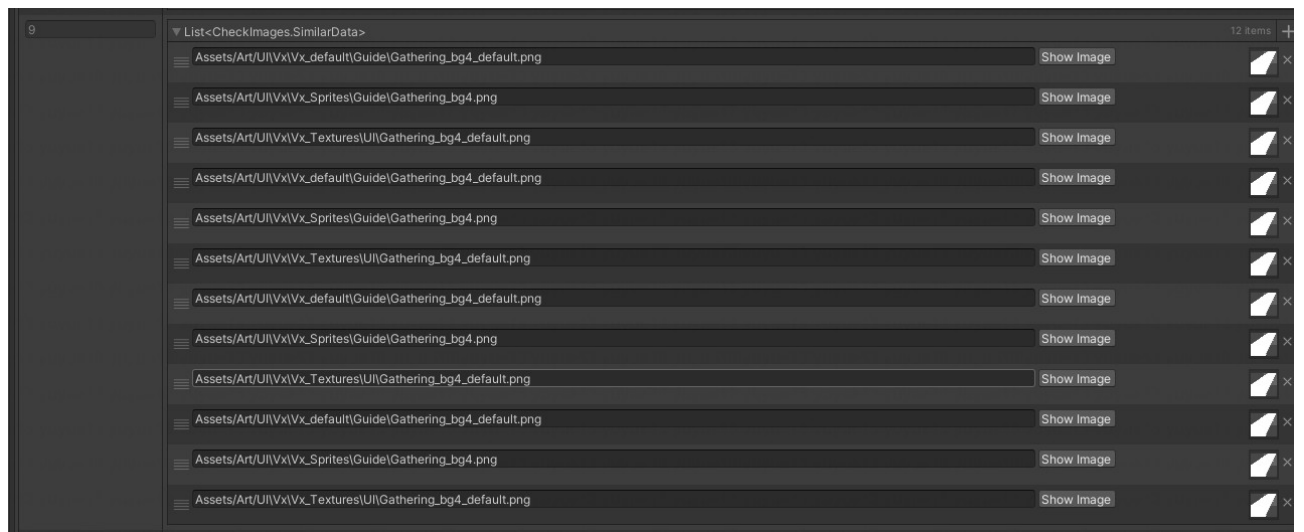
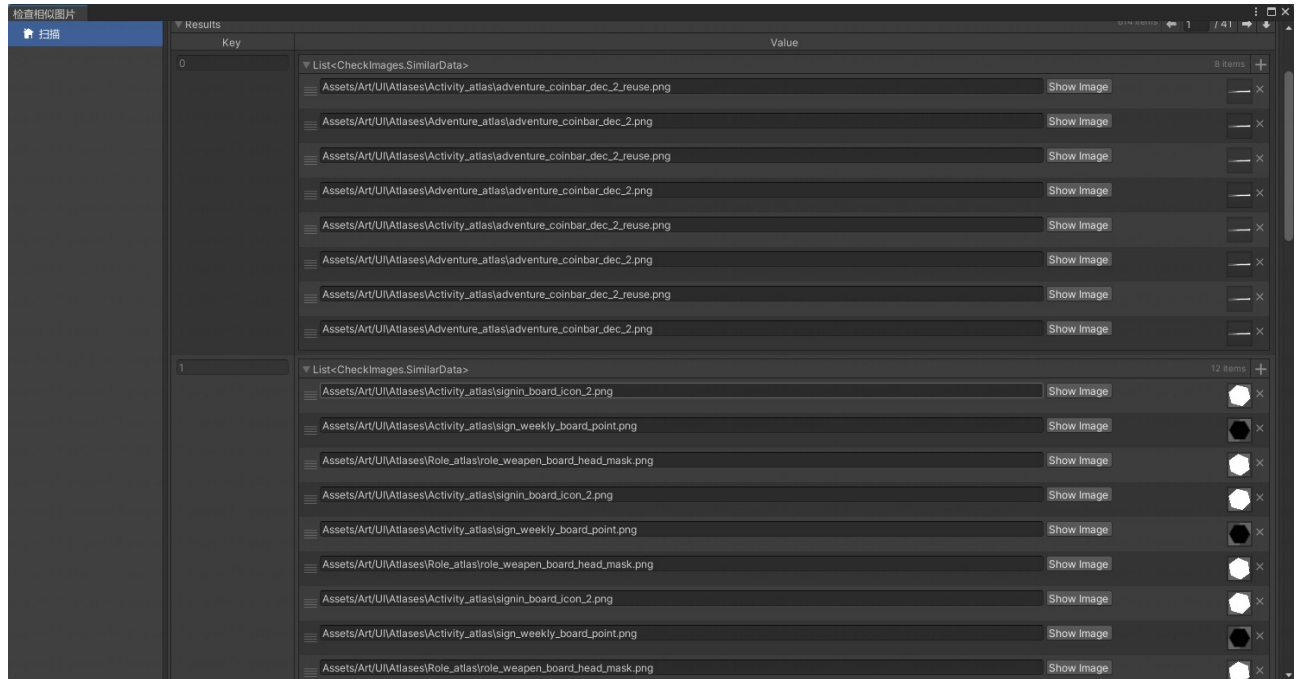


● 战斗内交互状态复杂度高

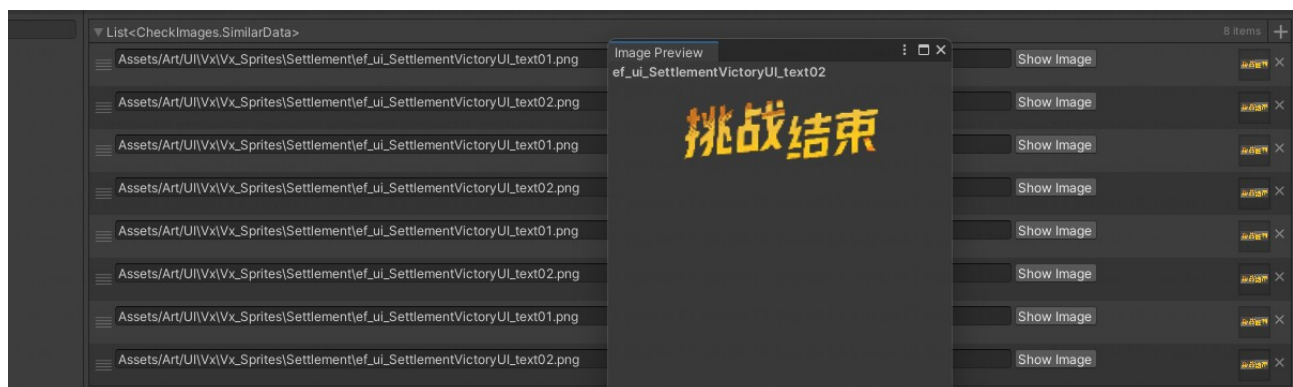
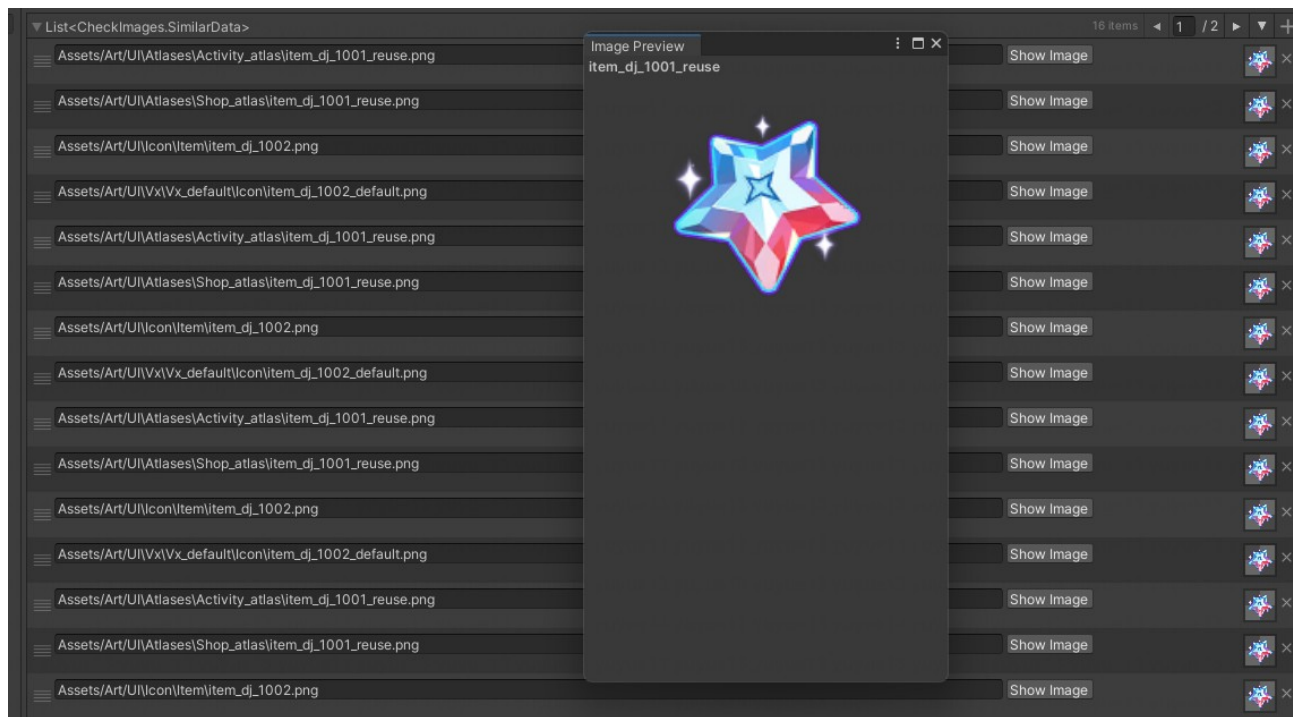
战斗内我们交互状态控制搞的好复杂，有 3 个切了状态，但是状态控制调用了 102 次，说明 1 个状态大概控制了 34 个节点

UIWidget.OnDiscreteStateChanged	3.10	3.10	43.14%	0.10	1.38%	3
> DiscreteStateController.Emit	3.00	3.00	41.77%	0.14	1.99%	102

## ● 相似图

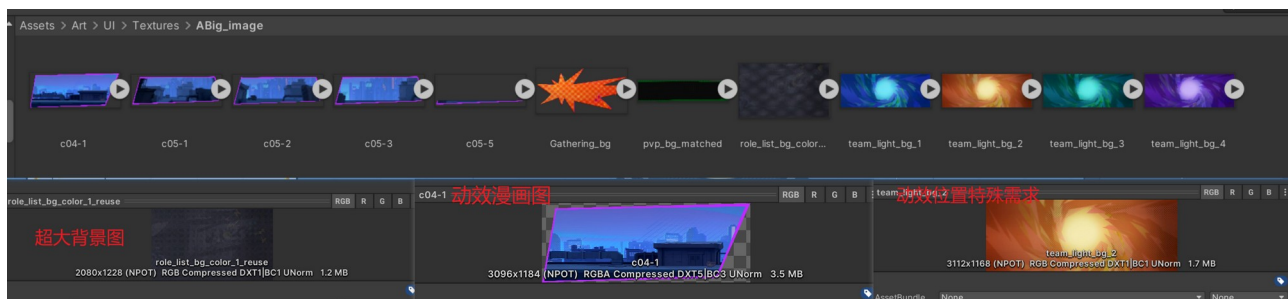






相似图片资源整理：C5 图集管理

超大散图（超过 2048 示例，现在统一放在 Assets/Art/UI/Textures/ABig\_image）



## 制作规范

### 界面拆散实践规范

参看跑测数据存在大量实例化卡顿，几乎每个界面都非常卡顿，界面加载以及实例化耗时太大，需要逐个界面做拆分。目前动态加载将会改变界面呈现方式，需要讨论

- 交互、UI 动效侧：按模块、按系统尽量拆细
- 程序侧：全部走动态加载，减轻峰值，必要实现

## 动静分离实践规范

原理：Unity 是基于 Canvas 去构建显示的网格，如果 Canvas 内的组件绝大部分属性的变更都会导致网格重建，耗费 CPU。

### 1. 尽量弃用 UIParticle（待讨论）

a. 战斗内耗时分析：bake 耗时 1~2ms 左右

b. <https://github.com/mob-sakai/ParticleEffectForUGUI/issues/193>

NOTE: BELOW ARE FPS AND STAT IN EDITOR (MAC OS X). SIMILAR FPS WAS SHOWN ON MY IPHONE OS.

Approach	FPS on Editor	FPS on iPhone6	FPS on Xperia XZ
Particle System	43	57	22
UIParticleSystem	4	3	0 (unmeasurable)
Sorting By Canvas	43	44	18
UIParticle	17	12	4
UIParticle with MeshSharing	44	45	30

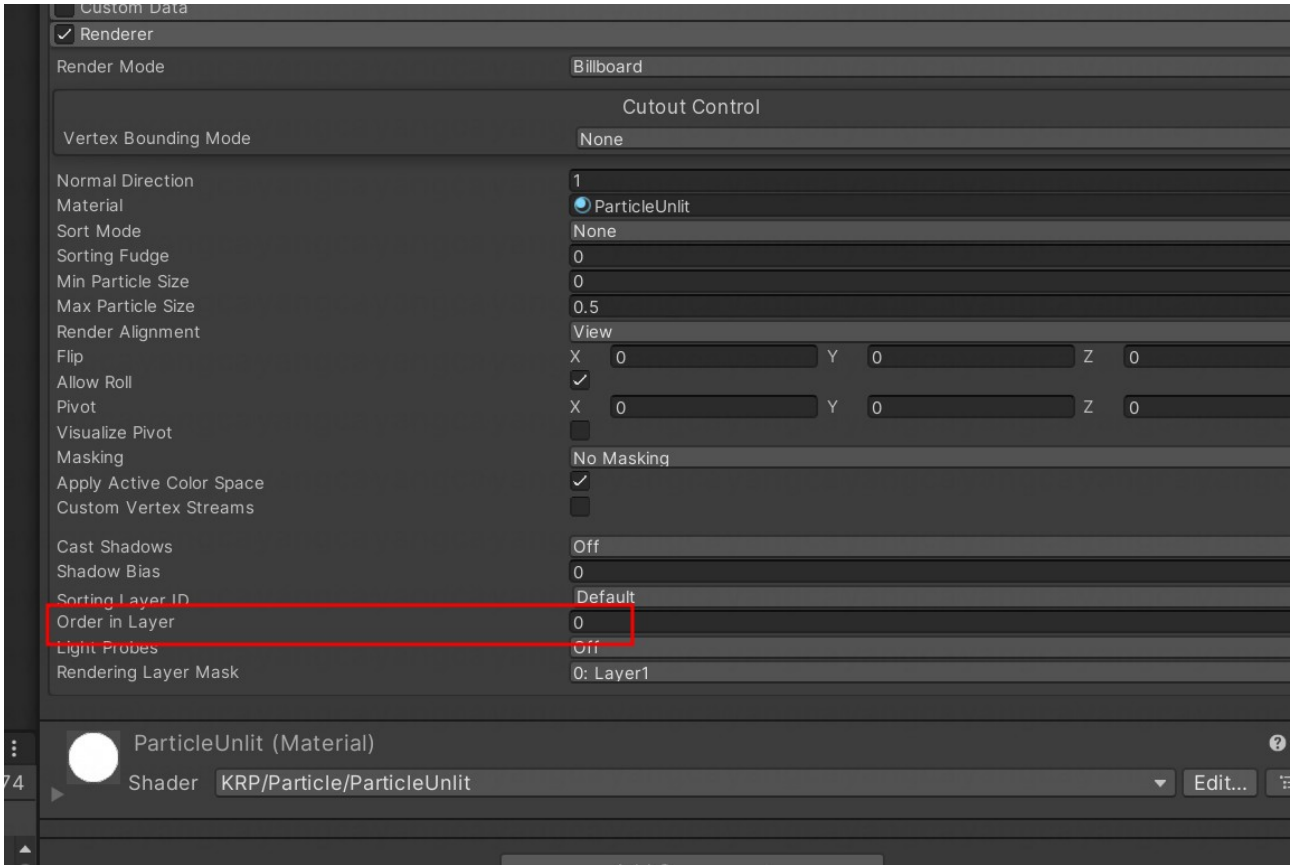
Particle System (default)

### 2. Canvas 工作流（待讨论）

- a. 带 Animation、Animator、Spine 以及其他动画的节点需要包一个 Canvas
- b. 列表等带滚动的组件需要强制加 Canvas
- c. 把一些可能经常程序修改的组件放一块：例如技能的 cd 和百分比
- d. 把一些静态的不太修改的对象放一起
- e. 动效类的 Prefab 强制独立一个 Canvas



f. 粒子系统可以设置 order in layer 控制层级



性能分级实践规范

- 1. 状态组实践
- 2. 动态加载的 LOD 组件
  - a. 自动拆分 prefab（类似动效）
  - b. 动态的自加载

主界面+时间海特殊优化

- 时间海场景需要重度优化
- 能跟时间海同时显示的界面需要重点排查性能（界面目前看不多礼包、聊天、小弹窗等，尽可能做成抓帧的形式减少同时显示）

todo 项

项	类别	人员
---	----	----

各类自动化检测工具等	工具	程序
拆分当前不合规的 prefab	优化	交互、动效
程序动态加载	优化	程序
优化列表使用	优化	程序

## Reference

逆水寒 UI 抓帧：[逆水寒手游 UI RenderDoc 抓帧分析](#)

天刀 UI 抓帧：[天涯明月刀手游 UI 抓帧切图分析](#)

忍三动效分析：[忍 3 大招特效制作方式分析](#)

C5 UIOverdraw 标准：[C5UI 参考游戏制作方式及 Overdraw 情况](#)

老 C5 规范：[UI 界面规范与待处理问题](#)

C1 规范：见评论，[C1 UI 渲染性能 M13 Review](#) 和 [M14 展望](#)

C5 资源规范：[C5 资源规范](#)

基于 2024/9/11 项目资源现状：

图集：  
[UIPrefabAtlasChecker.xlsx](#)

贴图：  
[UIPrefabTexChecker.xlsx](#)

动画：  
[UIPrefabAnimChecker.xlsx](#)