

1, 为什么要封装? 方便别人使用 不在去关心过程。 开发效率高 代码可以重复利用

UI:

1, 美术给一个 效果图。

2, 美术会将效果图 切成碎图。

3, 将碎图 打成 大图 。

4, 将制作成的 大图 导入 工程里面。 导入一个插件 Texture importer .

5, 拼界面规则:

1. 调整画布的大小 。

2. 调整 Game 窗口 大小 。

3. 以panel 为单位 拼界面。 效果图 在panel 上 碎图 就是界面上的小控件 。



1, 以 **panel** 为单位 : 为了方便管理

存入 **UIManager** :

查找自己的子控件 :

1, `GameObject.Find` 深度优先 效率最低

2, 树形关系 : `transform.Find ()` 按照路径去查找 。

让 子控件 自动存入 **UIManager** :

UI 重点思想:

1, **panel** 之间通信 通过 **UIManager** 中介 降低耦合

2, 同一个**panel** 身上控件 通过主动报告 而不是查找 。

