# 对象设计文档

1. 导言
   1. 对象设计习俗

**命名**：

文件夹：每个单词首字母大写

例如：Prefabs，AnimatorControllers

所有文件：每个单词首字母大写

例如：BoardManager.cs，Main.unity

类名：每个单词首字母大写,为名词

例如：BoardManager，Enemy

对象，变量名：第一个单词首字母小写，其余单词首字母大写，

例如：foodTiles，exit

函数名：每个单词首字母大写, 第一个单词必须为动词

例如：InitialiseList()，Start()

参数名：每个单词首字母大写，第一个字母小写

例如： Random (GameObject[] tileArray, int minimum)

**注释**：

1. 统一用中文注释
2. 每个.cs文件最上方写上日期与作者

例如：

/\*\* @author Administrator

\* @Time 2012-11-2014:49:01

\*/

1. 接口注释写于接口上方

对于每一个接口：

第1行：接口名

第2行：接口简介

第3行~第n行：接口参数简介(每一个参数依次对应一行)

第n+1行：接口返回类型简介(若为void则省略)

第n+2行：接口的其他备注

例如：

/\*\*DisplayPlayerAttr(int paramA, int paramB)

\*@brief: 将玩家的生命值和攻击力展示到屏幕上

\*@param : paramA int 攻击力

\*@param : paramB int 生命值

\*@return : bool 该函数是否执行成功

\*@remark:玩家的攻击力不能为负

\*/

1. 必要的成员变量注释写于本行后方

例如：private int columns = 8; //游戏面板的列数

1. 必要的内部函数注释写于函数上方

格式：/\*任意说明文字\*/

例如：/\*搭建游戏面板\*/

void SetupBoard (int columns, int minimum);

1. 必要的函数内每行代码注释写于上方

例如：

//移除列表中位于randomIndex处的元素

gridPositions.RemoveAt (randomIndex);

5 . 注释中的关键词

//TODO 表示当前块需要完成而未完成

//NEED 表示当前块得不到其他模块的支持而无法运行

**API文档编写：**

在完成一个类的每个接口后需要按照规定的接口文档编写规则修改API文档。

* 1. 编辑接口文档的规则

对于接口文档：

最上方写上修改日期和作者。

例如：

\* @author Administrator

\* @Time 2012-11-2014:49:01

对于每一个接口：

第1行：接口名

第2行：接口简介

第3行~第n行：接口参数简介(每一个参数依次对应一行)

第n+1行：接口返回类型简介(若为void则省略)

第n+2行：接口的其他备注

例如：

\*DisplayPlayerAttr(int paramA, int paramB)

\*@brief: 将玩家的生命值和攻击力展示到屏幕上

\*@param : paramA int 攻击力

\*@param : paramB int 生命值

\*@return : bool 该函数是否执行成功

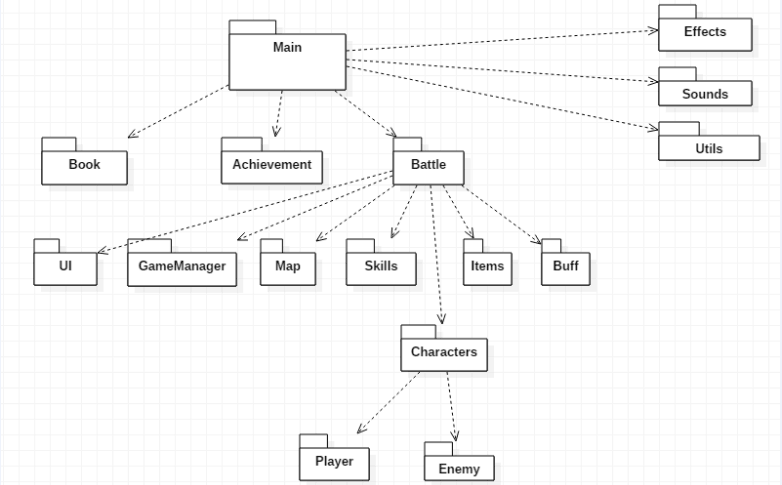
\*@remark:玩家的攻击力不能为负

* 1. 定义、首字母缩写以及简写

|  |  |  |
| --- | --- | --- |
| 缩写词 | 全称 | 含义 |
| UI | User Interface | 用户界面 |
| Rand | Random | 随机 |
| Btn | Button | 按钮 |

* 1. 参考

1. 包



1. 类的接口

术语表