项目规范

一、美术资源

所有文件都要以英语命名，不同的文件类型，会有不同的前缀，比如有一张图片是一个白色的房子，那么就命名为**T_WhiteHouse**，每个英语单词开头都要大写，这样不会区分不开不同单词

文件类型	命名方式
图片（必须用PNG格式或PSD格式）	T_NameName
模型（必须FBX格式 注意模型的几何中心一定要在坐标系原点！）	SM_NameName
视频（必须用MP4格式）	V_NameName
音频（必须用WAV格式）	W_NameName
美术人员提交图片前要注意有没有抠图	

二、游戏引擎内

Unity引擎内

文件规范

同样，所有文件都需要英文命名
文件夹名以大小写分割单词，例如"**BagSystem**",除非是像"**UI_System**"这种不用下划线分割下就不好区分的单词，都不要使用下划线。
（之所以要这样，是因为用下划线分割单词会导致文件夹的名称过长，在Unity里面显示不全）

不同功能模块的文件，需要放在不同文件夹中，比如UI功能就新建一个文件夹名叫"**UI_System**"，而所有UI的美术资源就放在文件夹"**UI_System/Resources**"
（为什么Unity中美术资源都要放在一个命名为Resources的文件夹内）

每个功能板块文件夹内的脚本逻辑尽量保持独立，不受其他功能意向，尽量保证可以单独导出后在其他Unity项目内稍加更改就
能快速复用

文件类型	命名方式
场景（scene）	S_NameName

代码规范

变量命名方式：

每个变量都以下划线“_”分割每一个单词，比如“跳跃高度”命名为**jump_height**
（为什么代码里面的变量要以下划线分割呢？因为这样能让你轻松看懂这个变量的意思，如果连在一起你可能不能很快看懂）

类的非静态成员变量要加上一个下划线“”后缀,而类的静态成员变量需要加一个"s"作为后缀。而方法和函数的命名都要以大小写分割每一个单词，这样做可以让变量名和函数名区分开，并且开头的第一个字母不用大写，例如：

```
public class PlayerMovement : MonoBehaviour
{
    private int index_;//
    public static int number_s;//类的静态成员变量需要加一个"_s"作为后缀
    public void setIndexOfPlayer(int index)//以大小写分割每一个单词，但第一个字母不要大写
    {
        index_=index;//这里体现了类的成员变量要加后缀的好处，方便在这个时候区分形式参数和类的成员变量
    }
    public int getIndexOfPlayer()//以大小写分割每一个单词，但第一个字母不要大写
    {
        return index_;
    }
}
```

（为什么函数的第一个字母不要大写呢？其实是为了方便你在Visual Studio 2019里面编写代码时候，能够在不切换大小写的情况下很快敲出函数的开头，而这个时候IDE会智能提示补全函数名，就不用反复切换大小写把函数名给敲完了）

尽量使用[header("标题")]分割不同大类的成员变量，这样方便你在Unity的编辑窗口中查看它，比如：

```
public class PlayerMovement : MonoBehaviour
{
    [Header("移动参数")]
    public float speed = 8f;
    public float crouchSpeedDivisor = 3f;        //控制减慢的速率（人物下蹲时候速度所除以的那个基数）

    [Header("跳跃参数")]
    public float jumpForce = 6.3f;               //基本的跳跃的力
    public float jumpHoldForce = 1.9f;            //长摁跳跃按键的力
    public float jumpHoldDuration = 0.1f;         //摁多长时间
    public float cruchJumpBoost = 2.5f;
}
```

此外，我们还要养成写注释的好习惯，不然别人看不懂，无法完成项目合作

虚幻引擎内

文件规范

所有文件夹以驼峰命名法命名，并且一种功能模块内的东西，都装到同一个文件夹内，比如背包系统，命名为**BagSystem**，并且把所有背包系统的功能组件和资源放到里面。

代码规范

类名

所有Actor类的子类蓝图类类名以"A__"前缀+驼峰命名法命名，例如第一人称视角玩家，命名为**A_FirstPersonCharacter**

控件蓝图以"UI__"前缀+驼峰命名法命名,例如背包界面，命名为**UI_BagMenu**

游戏模式类蓝图以"GM__"前缀+驼峰命名法命名,例如正常难度游戏模式，命名为**GM_Normal**

动画蓝图以"ABP__"前缀+驼峰命名法命名,例如人物动画，命名为**ABP_Character*

结构体蓝图以"ST__"前缀+驼峰命名法命名,例如背包元素，命名为**ST_BagItem**

C++类类名以驼峰命名法命名，如敌人类，写成**Enemy**

类的成员变量

蓝图的成员变量以下划线命名法命名，比如数组大小，写成**array__size**

C++类的成员变量以下划线命名法加上 '__'后缀，比如跳跃高度，写成**jump_height_s**

C++类的静态成员变量相比普通成员变量多加一个"s"后缀，例如**jump_height_s**

函数名与函数的形式参数名

C++和蓝图类的函数名都要以驼峰命名法命名，但第一个字母要小写，函数的形式参数以下划线命名法命名，但要加"_"前缀，例如设置血量，写成**setBlood(float __blood)**

之所以函数名的第一个字母要小写，主要是为了利用好编译器的智能代码提示，在你不用切换大写的时候，就可以快速写出函数名的开头，就能很快补全函数名。

函数内的变量名（执行完函数后就会被销毁的变量名）

用下划线命名法，无需添加前后缀，都是小写字母，例如数组长度，写成**array_size**