# 一：资源分类

Texture：纹理

Sprites：精灵

Prefabs：预设

Shaders：着色器

Materials：材质

AnimationClip：动画片段

Animator：动画控制器

Skeleton：骨骼

Font：字体

AudioClip：音效

Config：配置文件

Models：模型 fbx

Scenes：场景

Scripts：代码脚本

# 二：文件夹

Texture：存放贴图类图片，模型贴图、特效图片资源等

Texture/Character/ ：角色贴图总目录

Texture/Character/xxx：具体角色的目录

Texture/Effect/：特效贴图总目录

Texture/Effect/xxx：具体特效的贴图，也可以没有这么小的目录

Texture/Scene/：场景编辑，地图编辑用到的贴图

Texture/Scene/Common/：多个场景共同用到的放这里

Texture/Scene/Scene\_main/：主城用到的贴图

UISprites：存放UI 精灵相关

UISprites/Atlas：存放图集

UISprites/Sprites：存放UI 精灵图片

UISprites/Sprites/xxx：根据具体模块的资源更小粒度的路径

Prefabs：存放预设

根据不同资源建立不同的目录

Prefabs/UI：存放UI 界面预设总目录

Prefabs/UI/xxx：根据具体功能建立更小粒度的目录

Prefabs/Character：存放角色预设总目录

Prefabs/Character/LvJuRen：存放绿巨人预设的目录

如果绿巨人有不同的等级如 lv\_ju\_ren1、lv\_ju\_ren2... 将所有绿巨人的预设一同放入Prefabs/Character/LvJuRen

Prefabs/Effects/：存放特效预设

Prefabs/Scene/：存放场景地编的一些预设

Shaders：存放着色器代码

根据情况决定是否需要创建子目录

Materials：材质

Materials/Character/ 角色材质球总目录

Materials/Character/xxx 角色材质球目录

Materials/Model/ 不好确认怎么划分的，尽量要先确认

AnimationClip：存放动画的目录

AnimationClip/UI：UI动画的目录

AnimationClip/UI/xxx：根据情况创建子目录

AnimationClip/Model/模型动画

AnimationClip/Model/xxx：根据情况创建子目录

AnimationClip/Character/ 角色模型动画

AnimationClip/Character/xxx 角色模型动画

Animator：存放动画控制器目录

Animator/xxx 看情况创建子目录

Skeleton：存放骨骼

Skeleton/xxx 看情况创建子目录

Font：字体目录，一般就这一个目录就够了

AudioClip：存放音效

AudioClip/SoundEffect/：比较短的音效

AudioClip/Background/：较长的背景音乐

Config：存放配置文件

Config/BehaviorTree/ 行为树的配置文件

Config/xxx/ 具体功能的配置文件

Models：存放模型 fbx

Models/xxx：具体模型的目录

Scenes：存放场景文件

一般场景不会太多，一个目录可能就够了

Scripts：代码脚本

根据不同功能划分模块目录

# 三：命名规范

1. 资源文件名均采用驼峰式命名，每一个单词的首字母小写，中间可添加下划线，且使用能够描述其功能或意义的英文单词或词组
2. 根据功能属性和资源类型添加不同前/后缀

没有严格意义上的前/后缀区分

(1)如果只需要使用一个标示，该标示可以放在前缀，也可以放在后缀

如：绿巨人icon：spt\_ui\_icon\_lv\_ju\_ren、ui\_icon\_lv\_ju\_ren\_spt

spt 表示这是一个Sprite

(2)如果需要同时用到功能属性和资源类型，可以将一个加在前边，一个加在后边

如：绿巨人的碰撞盒：ch\_lv\_ju\_ren\_collision

ch 表示这是一个角色，collision 表示这是一个碰撞盒

主界面关闭按钮图标：spt\_main\_panel\_clone\_btn

spt 表示这是一个 Sprite，btn 表示这是一个按钮的图标

|  |  |  |  |
| --- | --- | --- | --- |
| 功能属性/资源类型 | 前缀 | 后缀 | 例子 |
| 角色：属性 | ch |  |  |
| 按钮图片 |  | btn |  |
| UI背景图 |  | bg |  |
| Collision：碰撞盒 |  | collision |  |
| Texture：纹理 |  | tex | ch\_lv\_ju\_ren\_model\_tex |
| Sprites：精灵 |  | spt | ui\_icon\_lv\_ju\_ren\_spt |
| Materials：材质 |  | mat | stone\_model\_mat |
| AnimationClip：动画片段 |  | ani | ch\_lv\_ju\_ren\_idle\_ani |
| Animator：动画控制器 |  | ac | ch\_lv\_ju\_ren\_ac |
| 特效 |  | fx | ui\_xxx\_panel\_open\_fx |
| Font：字体 |  | font | xxx\_font\_cn，xxx\_font\_en |
| 模型 |  | mod/fbx | ch\_lv\_ju\_ren\_fbx |
| AudioClip：音效 | sound | music | sound短音效，music长音乐 |
| Models：模型 |  | fbx | ch\_lv\_ju\_ren\_fbx |
| Scenes：场景 | scene |  | scene\_main |

如上 lv\_ju\_ren 也可以写成 lvjuren，LvJuRen

关于热更新部分资源依赖的要求，如果一个角色相关的 fbx、贴图、材质球、动画等是只被这一个资源使用，也可以简历一个类似 Common 的目录

Common/LvJuRen

Common/LvJuRen/Texture：纹理

Common/LvJuRen/Prefabs：预设

Common/LvJuRen/Shaders：着色器

Common/LvJuRen/Materials：材质

Common/LvJuRen/AnimationClip：动画片段

Common/LvJuRen/Animator：动画控制器

Common/LvJuRen/Models：模型 fbx

打 AB 包的时候将 Common/LvJuRen 目录内的所有资源打包为一个 ab资源