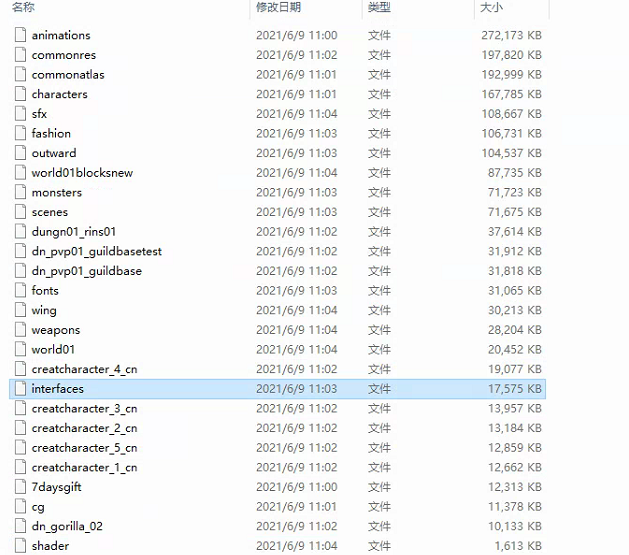
# Tera的分包机制和余烬做分包的几个问题

## Tera的包体策略

1. 玩家下载的apk可以体验10级以内的所有游戏内容，首包保证所有游戏内容都存在
2. 进入游戏后，玩家可以根据当前的等级下载对应的游戏包，比如玩家下载10-20级分包资源后可以体验所有10-20级内容
3. 首包apk大小 1.8G，所有分包大小 1.72G

## Tera首包内StreamingAsset下资源

|  |
| --- |
| AssetBundles 1.64G  Audio 147M 音频  package 103M 配置数据，lua代码  Video 40M CG视频 |



首包美术资源大小(1.64G)

## Tera首包中的美术资源AssetBundles (在所有资源中去掉10级以外的场景资源)

|  |
| --- |
| 7daysgift  cg  cganimator  characters  monsters  outward  wing  weapons  animations  commonres //场景之间共享的fbx贴图等  fonts  fashion  commonatlas //UI图集  interfaces //UI prefab  others  scenes //场景资源(天空球等)  sfx  shader  timeline  creatcharacter\_1\_cn //创建角色  creatcharacter\_2\_cn  creatcharacter\_3\_cn  creatcharacter\_4\_cn  creatcharacter\_5\_cn  dn\_gorilla\_02 //选人场景  dn\_gorilla\_selectchar\_animator  dn\_gorilla\_selectchar\_animator2  dn\_pvp01\_guildbase //公会基地  dungn01\_rins01 //10级副本  world01 //第一个大地图  world01blocksnew //地图分块 |

## Tera分包中的美术资源AssetBundles

|  |
| --- |
| city01  city01blocksnew  city02  city02blocksnew  dn\_pvp01\_guildbaselv2  dn\_pvp01\_guildbasetest  dn\_pvp02\_arena01  dn\_pvp02\_arena02  dn\_pvp03\_skyarena  dn\_pvp03\_skyarenablocksnew  dn\_pvp04\_palace01  dn\_pvp05\_fearless  dn\_env02\_mageplatform  dn\_env02\_mageplatform\_02  dn\_evn01\_dnest  dn\_evn03\_killerparty  dungn00\_empirerelicprologue  dungn00\_throne01  dungn02\_cave01  dungn03\_qxgj01  dungn04\_zuras01  dungn05\_elfarch  dungn06\_empirerelic  dungn07\_mansion01  world02  world02blocksnew  world03part1  world03part1blocksnew  world03part2  world03part2blocksnew  world04part1  world04part1blocksnew  world04part2  world04part2blocksnew  world05  world05blocksnew  world06part01  world06part01blocksnew |

**分包的目的就是划分出 10-20 20-30 30-40 40-50 50-60级用到的资源，做成一个大的分包文件, 以10-20级资源为例，分包1中包含**

AssetBundles/

"city01",

"city01blocksnew",

"dn\_pvp02\_arena02",

"dungn02\_cave01",

"dn\_pvp01\_guildbaselv2"

Audio/

"Cut\_Maincastle\_1.bnk",

"Main\_Castle\_1.bnk",

Video/

"Dn01\_Arena02.mp4",

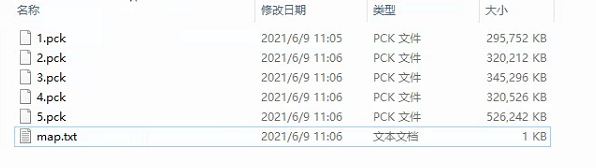
"Mall\_CG01.mp4",

"Mall\_CG01\_Loop.mp4",

"Mall\_CG02.mp4",

"Mall\_CG02\_Loop.mp4",

"Boss\*.mp4",



最终的分包大小(1.72G)

## Tera下载器的一些实现

* 1. 使用curl openssl，支持ipv6和https
  2. 由于下载文件比较大，支持断点续传(大文件下载中断，重启app后继续下载)
  3. 版本号控制，下载到临时文件夹，完成后进行md5校验，然后拷贝到游戏目录
  4. 下载使用单线程而不是协程，避免对游戏造成卡顿

## Tera分包下载的几个问题

* 1. 在现有的打包机制下，如何按游戏等级划分分包内容，同时保证首包资源不能太大
  2. 场景资源之间有公共引用资源，如何合适的提取公共引用资源，放在首包中且使首包不能过大

## 余烬的资源目录

Assets/SubAssets/Raw/ //原始资源(非prefab)

Sprite //图集资源

Animation //Animation, Controller

Character/FaceShaping //捏脸

Permanent //常驻资源

Shader

Scene //场景资源

ToGame //分块场景 .unity

Final //光照图资源 LightingData.asset

Assets/SubAssets/Res/

Prefabs //prefab目录

Atlas //新图集 .spriteatlas

Font //字体

RecastInfo //导航信息

SimpleMeshes //角色低模

## 余烬的首包资源

* 1. Assets/SubAssets/Res/Character
  2. Assets/SubAssets/Raw/Character/FaceShaping
  3. Assets/SubAssets/Res/SimpleMeshes
  4. 其他由乐变统计的大资源列表

## 余烬的打包规则

* + 1. 收集Res下的 Prefabs, Atlas, Font, RecastInfo中的prefab列表

**在 Raw/Scene/ToGame 下收集分块.unity场景**

**在 Raw/Scene/Final 下收集光照图**

2) 对1)中的每个资源每个资源(.prefab .unity)建立assetbundle

3) 统计1)中资源(.prefab .unity)的所有依赖项

**对所有被依赖的资源，如果被依赖大于1，则每个资源打成assetbundle**

4) **场景每个.unity生成一个bundle**

5) 每个 Raw/Sprite 下的sprite打成一个assetbundle

6) 捏脸 Raw/Character/FaceShaping 下的贴图和mat打成一个assetbundle

7) Raw/Shader 下的资源打成shader assetbundle

Raw/Permanent 下的资源打成 permanent assetbundle

8) Raw/Animation 下的Controller文件 打成一个assetbundle (Animator打包)

9) Raw/Animation 下的anim文件 打成一个assetbundle (Animation打包)

10) Res/Atlas 下的spriteatlas文件 打成一个assetbundle

11) PostProcess资源打成 assetbundle

12) SimpleMeshes 打成一个 assetbundle

## 余烬做分包下载的方法

* + 1. 每个场景.unity.bundle + lightmap.bundle + navmesh 做为一组, 场景之间提取公共引用资源避免场景资源过大
    2. 按游戏等级划分为几个大包包含1中的场景资源，根据游戏等级下载相应的资源
    3. 假如首包到10级，确定首包的场景资源，再加上其他资源，大小不能超过2G，否则需要把其他资源继续按照等级拆分，直到首包足够小
    4. 下载器要支持游戏内下载,下载过程中保持游戏流畅(和在游戏前下载资源不同)

## 余烬现在win包下的资源情况

所有bundle资源大小5.06G

去掉所有Raw/Scene整个目录(即去掉场景)的bundle资源大小2.33G

如果首包加上首包场景资源，大小是不能控制在2G以内的 (对比Tera 1.64G)

## 结论

余烬要做分包下载，**首先要按游戏等级划分分包资源，在现在的打包机制下按场景划分，同时要提取场景之间的公共资源做assetbundle避免单个场景过大；其次由于现在除去场景外的资源大小就有2.3G，因此需要进一步按游戏内容划分，工作量比较大；最后要有一个按分包为下载单位的游戏内下载系统，可能和现有的按assetbundle为单位的下载系统有区别，且要在游戏内下载并测试，短期内难度较大。**