Unity 小游戏 WASM 分包工具使用说明

目录

前置说明:

- **在unity侧使用美团 Unity 游戏转换插件导出时，需要开启profiling func**，这样打包时就会导出symbol文件

无需担心开启profiling func导致的包体增大，分包工具会优化函数名

- **美团客户端支持:**
  - **安卓端 12.25.400 版本及以上支持，12.25.400 以下版本执行默认加载逻辑（测试版美团 APP）:**

    aimeituan-release_12.25.400-331867-aarch64.apk
    114.03MB
  - **iOS 端12.25.400 版本及以上支持，12.25.400 以下版本执行默认加载逻辑（测试版美团 APP）**

    imeituan-12.25.400-232640.20240925190436.ipa
    135.77MB
- **请使用最新版本的美团小游戏 Unity 转换工具进行游戏转换:**

  mgc.package
  14.46MB

1、安装

请确保本地已经安装 Node.js (Node.js 版本不低于 v14.13.0)，然后运行以下命令安装最新版本（**0.0.65**及以上）的 美团小游戏 CLI 工具（mgc-cli-external）：

代码块 Shell

```
$ npm i -g mgc-cli-external@latest
```

2、创建美团小游戏配置文件

在游戏项目父目录或者其他目录（注意不要在当前待打包游戏目录或者子目录）执行以下命令 （**如果已经创建过可以忽略，但是需要增加一个新字段 localIP 用于启动本地服务进行 WASM 函数收集上报**）

代码块 Shell

```
$ mgc config
```

配置成功后会在当前目录生成配置文件 mgc.config.js，配置示例如下

代码块 Shell

```
module.exports = {
  "type": "UnityGame",                    // 包类型，unity 游戏需要指定 type 类型为 UnityGame
  "appId": '4503b3691d974573a747ed85f74f1a29',  // 游戏的appId
  "appSecret": '752d7e4a3b8b4525',        // 游戏的appSecret
  "entry": './build/wechatgame'           // 需要被打包的小游戏的本地路径
  "localIP": '192.168.1.1:5000'           // 本地设备的网络IP地址，同时需要自定义端口号(请留意该端口未被占用)
}
```

3、构建测试包

在生成 mgc.config.js 的**同级目录**执行以下命令，用于构建测试包进行 WASM 函数收集作为分包依据 （构建时间可能会很久，请耐心等待）

代码块 Shell

```
$ mgc wasm-init
```

4、开启 WASM 函数收集服务

在生成 mgc.config.js 的同级目录执行以下命令，用于开启 WASM 函数收集服务, 函数收集完成前不要关闭服务

代码块 Shell

```
$ mgc wasm-log
```

5、首次收集

新开一个终端窗口，在生成 mgc.config.js 的同级目录执行以下命令：

代码块 Shell

```
$ mgc debug
```

执行完上述命令后，如果一切正常，会出现二维码，用真机扫码运行游戏，**进入游戏后尽量覆盖游戏内的场景，特别是启动后最先进入的场景和关卡（比如新手教学，游戏最初的关卡内容）来收集信息**，WASM 函数收集服务每 5s 会轮训一次进行函数收集，终端窗口会实时显示函数收集信息

6、首次构建分包

在生成 mgc.config.js 的**同级目录**执行以下命令，根据已收集的 WASM 函数调用信息进行首次分包构建 （构建时间可能会很久，请耐心等待）

代码块 Shell

```
$ mgc wasm-split-dev
```

7、函数收集

在生成 mgc.config.js 的同级目录执行以下命令：

代码块 Shell

```
$ mgc debug
```

执行完上述命令后，如果一切正常，会出现二维码，用真机扫码运行游戏，

尽量多次扫码测试，覆盖尽可能多的场景，

有条件的话，可以尽量覆盖各种机型(主流品牌)再多跑几次收集，

当WASM 函数收集服务所在的终端页显示的收集到的函数个数相对稳定时，就可以再次执行以下命令进行**二次分包构建**

代码块 Shell

```
$ mgc wasm-split-dev
```

完成构建后，执行以下命令构建 debug 游戏包进行验证，如果 WASM 函数收集服务所在的终端窗口无函数更新显示或者游戏启动过程中无阻塞性等待耗时就基本完成函数收集了。

8、构建 Release 版分包

在生成 mgc.config.js 的同级目录执行以下命令，用于构建 Release 版分包。然后再执行打包发布命令发布 debug 包或者 release 游戏包进行测试，运行正常即可进行接下来的发布流程

代码块 Shell

```
$ mgc wasm-split-prod
```

9、函数增量收集

- 如果游戏通过上述操作构建分包成功。我们在使用转换SDK导出新包时，如果想复用之前的函数收集信息，可以执行 mgc wasm-update 命令，会更新symbol 表映射构建新的包。然后发布debug包运行测试即可。
  注意还需要重新执行mgc wasm-log 看首包函数是否有变动。
  如果要弃用之前收集的函数信息，在出新包后还是执行 mgc wasm-init 命令即可
- 代码块 C#

```
```

- 如果导出的新包 WASM 文件没有变更（文件名hash作为确认标识），之前收集的函数信息不会被清理，也不需要变更，执行 mgc wasm-split-dev 或者 mgc wasm-split-prod 命令即可进行分包。
- mgc wasm-update 只在出新包（第一次）需要复用原来收集的函数信息且WASM 文件发生变更时使用