

mod 文件夹架构

modname.json

```
1 {
2     "path": "testmod\\testmod.dll",
3     "name": [
4         "test_fun",
5         "test_fun2"
6     ]
7 }
```

name为mod中的函数名，path为mod编译出的动态链接库的路径

modname.cpp

模组主文件

其中有数个function，每个function对应一个modname.json中的name

function的参数为指向当前chunk的指针，chunk的结构如下

```
1 class chunk
2 {
3 public:
4     int x;
5     chunk_type type;
6     block *blocks;
7     int block_count;
8     ~chunk();
9 };
```

block结构如下

```
1 class block
2 {
3 public:
4     block_type type;
5     int x;
6     int y;
7     void *data;
8     int data_size;
9     ~block();
10 };
```

block_type和chunk_type为枚举类型，分别表示方块类型和区块类型，定义可以在map.h中找到

data为指向数据的指针，data_size为数据大小

mod 的主要功能是对chunk进行操作，增加、删除、修改block来生成特定的自定义的结构

请保证不要出现内存错误，否则会导致游戏崩溃

chunk的大小也可以在map.h中找到

build

编译环境

vs2017以上+cmake

使用cmake配置项目

```
1 cmake_minimum_required(VERSION 3.0)
2 project("testmod")
3 set(WIN32 TRUE)
4 add_library(testmod SHARED testmod.cpp)
```

其中testmod.cpp为modname.cpp的一个示例

运行build.bat即可编译出动态链接库

dll可以在build/Debug中找到

比如说testmod.dll

打包

将dll按照modname.json中的路径放入对应文件夹或者根目录

注意一个mod只能有一个dll

然后将mod文件夹打包成zip文件