# Stage3D.js的打包方式

使用第三方工具grunt进行零散的ts打包输出为带.t.ds、.min.js的文件。

具体步骤：

1. 全局安装Grunt CLI

npm install -g grunt-cli

这个模块添加了全局的grunt命令行调用；

1. 初始化nodejs项目，新建文件夹Stage3D，执行下面的代码：

npm init

1. 安装nodejs的模块

//TypeScript模块

npm install typescript --save-dev

//Grunt模块

npm install grunt --save-dev

//Grunt处理ts的模块

npm install grunt-ts --save-dev

//Grunt生成xxx.min.js的模块

npm install grunt-contrib-uglify --save-dev

安装中我遇到报错：Please try running this command again as root/Administrator.

通过到system32下右键选择管理员打开CMD.exe后，跳到项目目录执行命令即可成功。

1. 新建src把Stage3D.js项目的ts源码都拷贝进去。
2. 项目根目录创建Gruntfile.js文件

官网上有文件的详细说明，文件如下：

module.exports = function(grunt) {

// 项目配置

grunt.initConfig({

pkg: grunt.file.readJSON('package.json'), // 读取package.json，读取后可以获取里面的数据，比如可以把编译和输出的地址配置到里面

// 定义 TypeScript 的任务

ts: {

default : {

tsconfig: './tsconfig.json'

}

},

uglify:

{

min:

{

files: {'bin/stage3d.min.js': ['bin/stage3d.js']}

}

},

});

// 加载除grunt外额外添加的模块

grunt.loadNpmTasks("grunt-ts");

grunt.loadNpmTasks("grunt-contrib-uglify");

// 命令行输入 grunt 后会被逐个执行的任务

grunt.registerTask('default', ['ts', 'uglify:min']);

// 命令行输入 grunt publish 后会被逐个执行的任务

// grunt.registerTask('publish', ['ts', 'uglify:min']);

};

1. 项目根目录生成tsconfig.json

tsc –init

编辑如下：

{

"compilerOptions": {

"target": "es5",

"module": "amd",

"declaration": true, /\* 生成 .d.ts \*/

"sourceMap": true, /\* 生成 .js.map \*/

"outFile": "./bin/stage3d.js", /\* 生成单一文件的地址 \*/

"rootDir": "./",

"allowUnreachableCode": true, /\* 允许不能被执行到的代码存在 \*/

"sourceRoot": "./src", /\* 源文件地址 \*/

"inlineSourceMap": false /\* .js.map不合并到js中单独作为文件输出 \*/

}

}

1. 遇到的代码编译报错，需要注释掉严格模式

// "strict": true,

Stage3D.js里，有两个文件是报错的，因为传递的类型和接收的类型对不上，估计是目前的ts版本比当时的版本更高导致的，去掉类型转换即可。

1. 使用命令行执行即可

grunt

可以编译出需要的代码