

Q3`25-Slots-优化本地工程启动速度-程序

一、解决的问题：

- build_local_app.sh 脚本中 uglifyjs 命令执行时间过长，拖慢程序本地工程单次启动的速度，影响开发节奏：

```
+ ../node_modules/.bin/uglifyjs ../res_doublehit/game.js -o -o /dev/null

real    0m19.574s
user    0m20.716s
sys     0m0.676s
```

- uglifyjs 用于检查代码语法错误，之前尝试屏蔽该命令，导致了一些语法错误带到了打包机，付出了额外的错误查询成本；

二、优化效果：

- 本地启动耗时缩短 19.5m：（缩减比例：2/3）

三、解决方案：

- 1、修改 build_local_app.sh 脚本、新增 build_lcoal_check.sh 脚本：

日常开发中启动时不做语法检查，如果出现语法错误，单独启动该脚本定位；

```
scripts/build_local_alpha.sh

@@ -8,15 +8,18 @@ resPath=$1
8 8
9 9  ulimit -n 65535
10 10 echo "build_local_alpha.sh"
11 - npm install --no-audit --no-fund
11 +
12 + set -x
13 +
14 + time npm install --no-audit --no-fund
12 15
13 16 cd ../src
14 17 # 生成 game.min.js 并提取源映射到 game.min.js.map
15 - browserify -d gameLoader.js > ../game.min.js
16 + time browserify -d gameLoader.js > ../game.min.js
17 # 生成 game.js 并提取源映射到 $resPath/game.js.map
17 - browserify -d main.js > ../$resPath/game.js
18 + time browserify -d main.js > ../$resPath/game.js
18 21 # 本地只用来做语法检查，不需要的可以关闭（保留源映射需注释或调整 uglifyjs 参数）
19 - ../node_modules/.bin/uglifyjs ../$resPath/game.js -o -o /dev/null
22 + # time ../node_modules/.bin/uglifyjs ../$resPath/game.js -o -o /dev/null
20 23
21 24 mkdir -p ../publish/
22 25 echo "{}" > ../publish/moduleDebugList.json
No newline at end of file
```

```
new file mode 100755
@@ -0,0 +1,36 @@
1 +#!/bin/zsh
2 +echo "[START] Building local check"
3 +
4 +if [ $# != 1 ]; then
5 +    echo "Usage:./build_local_check.sh resPath"
6 +    exit 1
7 +fi
8 +resPath=$1
9 +
10 +sh build_local_app.sh $resPath
11 +
12 +# 恶心想心啊! 破 AI
13 +error_handler() {
14 +    # 捕获原始错误输出
15 +    local error_output=$1
16 +    # error_output 第一行类似于"Parse error at ../res_oldvegas/game.js:25106,66",我需要取出起一行, 将其中的../res_oldvegas/game.
17 +    local first_line=$(echo "$error_output" | head -n 1)
18 +    # 提取类似 ../res_oldvegas/game.js 的路径部分
19 +    local relative_path=$(echo "$first_line" | grep -oE '\.\/[^\:]*')
20 +    # 将相对路径转换为相对于工作区根目录的路径
21 +    local workspace_relative_path=$(echo "$relative_path" | sed 's|\.\/||')
22 +    # 然后将错误日志中的路径替换为工作区相对路径
23 +    local modified_error=$(echo "$error_output" | sed "s|$relative_path|$workspace_relative_path|")
24 +
25 +    # 取出行号
26 +    local line_number=$(echo "$first_line" | grep -oE ':[0-9]*')
27 +    echo "line_number: $line_number"
28 +    code -g "$relative_path${line_number}"
29 +    echo "$modified_error"
30 +}
31 +
32 +error_output=$(./node_modules/.bin/uglifyjs "$resPath/game.js" 2>&1) || error_handler "$error_output"
33 +
34 +# ./node_modules/.bin/uglifyjs "$resPath/game.js"
35 +
36 +echo "[DONE] Builed local check"
No newline at end of file
```

- 2、新增 vs-code:task:

本次优化针对 vs-code:task 做了自动定位报错行的功能:

受限于 vscode eslint 未配置完成, 不能直接使用 vscode 的问题面板, 暂时采用了通过自动打开game.js 文件, 自动定位到报错行的方式;

```

29 129 ..... "options": {
32 132 ..... "group": {
33 133 ..... "kind": "build"
34 134 ..... }
135 + ..... },
136 + ..... {
137 + ..... "label": "run_local_check",
138 + ..... "type": "shell",
139 + ..... "command": "./build_local_check.sh",
140 + ..... "args": [
141 + ..... "${input:res_select}"
142 + ..... ],
143 + ..... "options": {
144 + ..... "cwd": "${workspaceFolder}/scripts"
145 + ..... },
146 + ..... "group": {
147 + ..... "kind": "build"
148 + ..... },
149 + ..... "isBackground": true,
150 + ..... "promptOnClose": true,
151 + ..... "problemMatcher": {
152 + ..... "owner": "javascript",
153 + ..... "fileLocation": "relative",
154 + ..... "pattern": {
155 + ..... "regexp": "^(UglifyError|Parse error at) (\\d+):(\\d+)$",
156 + ..... "file": 2,
157 + ..... "line": 3,
158 + ..... "column": 4
159 + ..... },
160 + ..... "background": { // 后台任务（如监听模式）的起止检测规则
161 + ..... // 任务开始激活 problemMatcher
162 + ..... "activeOnStart": true,
163 + ..... "beginsPattern": "^[START\\].*",
164 + ..... "endsPattern": "^[DONE\\].*"
165 + ..... }
166 + ..... }
35 167 ..... }
36 168 ..... ],

```

- 3、新增 vs-code:luanch:

```

32 32 ..... "preLaunchTask": "run_adb",
33 33 ..... "timeout": 100
34 34 ..... },
35 - ..... {
36 + ..... {
37 + ..... "name": "LOCAL_CHECK",
38 + ..... "type": "node",
39 + ..... "request": "attach",
40 + ..... "preLaunchTask": "run_local_check",
41 + ..... "timeout": 100
42 + ..... },
42 + ..... {
36 43 ..... // 除非local_server 意外停止或未启动，否则不用手动启动
37 - ..... "name": "LOCALSERVER",
44 + ..... "name": "LOCAL_SERVER",
38 45 ..... "type": "node",
39 46 ..... "request": "launch",
40 47 ..... "skipFiles": [

```

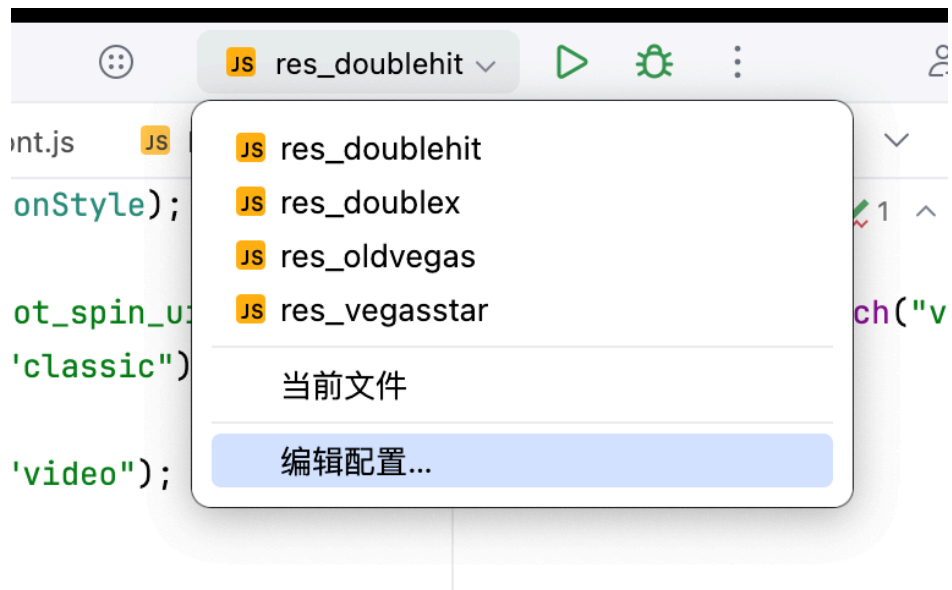
- 4、开启 vscode:task 自动检查:

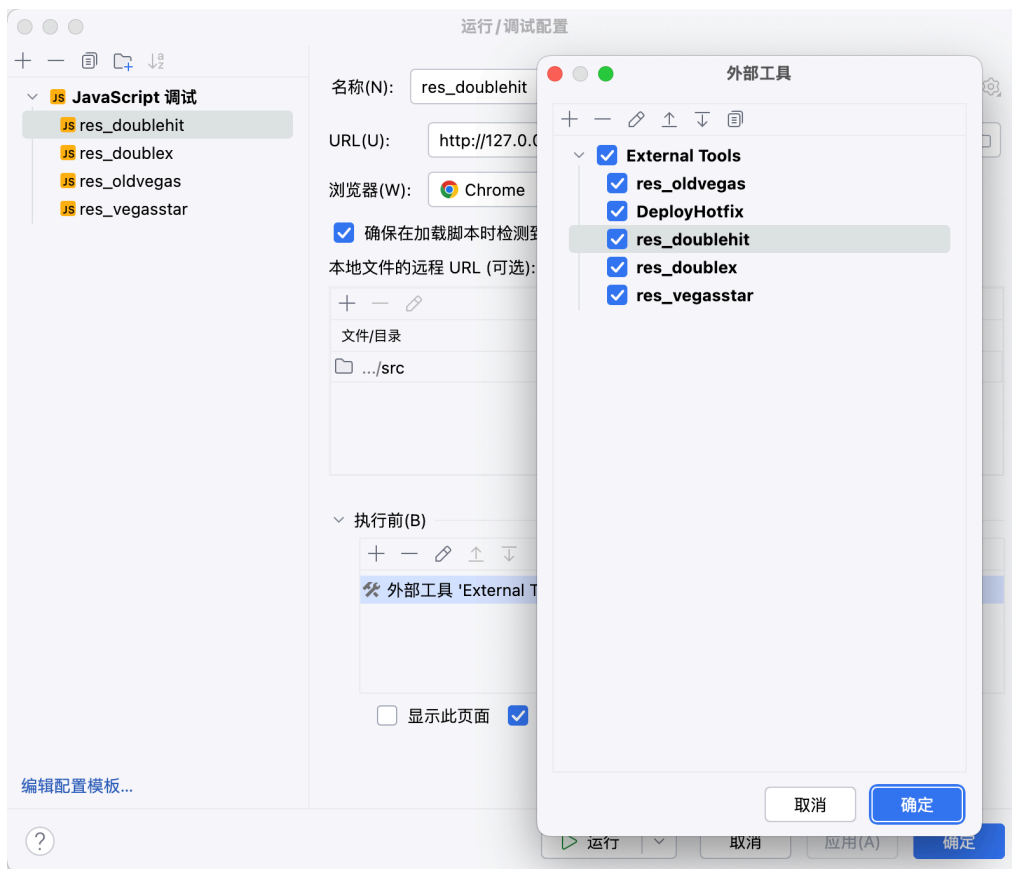
```
.vscode/settings.json
@@ -13,5 +13,8 @@
13 13  ....// 允许自动运行任务，如打开工程目录后启动 local_server 任务
14 14  ...."task.allowAutomaticTasks": "on",
15 15  ....// 启用代码片段导航
16 - ...."editor.snippetSuggestions": "top"
16 + ...."editor.snippetSuggestions": "top",
17 + ...."task.autoDetect": "on",
18 + ...."problems.decorations.enabled": true,
19 + ...."javascript.validate.enable": true
17 20  }
No newline at end of file
```

- 5、webstorm 自行创建外部工具调用将脚本：

可以通过新建 JavaScript 调试器调用外部工具；

或将该外部工具添加到工具栏，类似于 hotfix；





编辑工具

名称:

组:

External Tools

▼

描述:

工具设置

程序: +

📁

实参: +

↗

工作目录: +

📁

> 高级选项

?

取消

确定

