

windows 版本编译

库编译

具体原理参考cmd\window_engine 下 window版制作说明.md

1. 需要安装go 32 环境，下载go language 的386 压缩包 解压到D盘 最好使用go1.11.1的32windows 版本
2. 安装TDM-GCC-32 到D盘
3. git上获取库代码 使用 git@coding.xbonline.net:blockchain/go-ethereum.git上的分支代码，当前为develop分支
4. 在cmd\window_engine 目录下 参考example_local.bat 脚本 将上面 go32 和 TDM-GCC-32 目录修改到脚本
5. 执行example_local.bat 在当前目录生成windows_engine.lib windows_engine.h 和 windows_engine.dll

项目编译

安装cocos creator 2.2.2 vs2017

1. 去git上正确代码分支，目前是RMT
2. cocos creator 2.2.2 打开代码 构建windows 在build 下生成win32 工程项目
3. 将windows_engine.lib windows_engine.h 拷贝到 build\jsb-link\frameworks\runtime-src\proj.win32 目录
比较 build_kaleido\jsb-link\frameworks\runtime-src\Classes 目录下 windows_engine.h 与刚才拷贝到\build\jsb-link\frameworks\runtime-src\proj.win32下的 windows_engine.h
将

```
typedef __SIZE_TYPE__ GoUintptr;
typedef float GoFloat32;
typedef double GoFloat64;
typedef float _Complex GoComplex64;
typedef double _Complex GoComplex128;
```

改为

```
// typedef __SIZE_TYPE__ GoUintptr;
// typedef float GoFloat32;
// typedef double GoFloat64;
// typedef float _Complex GoComplex64;
// typedef double _Complex GoComplex128;
```

4. 对比build_kaleido\jsb-link\frameworks\runtime-

src\proj.win32\CocosCreator\resource\cocos2d-x\cocos\platform\CCApplication.h 和cocos 的安装目录 如 D:\CocosCreator_2.2.2\resources\cocos2d-x\cocos\platform\CCApplication.h, 修改重点是:

```
inline std::shared_ptr<Scheduler> getScheduler() const { return _scheduler; }
```

改为

```
inline Scheduler* getScheduler() const { return Application::_scheduler.get(); }
```

```
static std::shared_ptr<Scheduler> _scheduler;
```

改为

```
public :
```

```
static std::shared_ptr<Scheduler> _scheduler;
```

5. 对比\build_kaleido\jsb-link\frameworks\runtime-src\proj.win32 和 build\jsb-

link\frameworks\runtime-src\proj.win32 目录, 将差异修改一致, 注意工具的路径信息 不需要修改

主要有 BlockPoker.sln BlockPoker.vcxproj main.cpp BlockPoker.vcxproj.filters
BlockPoker.vcxproj.user

6. 对比 build_kaleido\jsb-link\frameworks\runtime-src\Classes 和

build\jsb-link\frameworks\runtime-src\Classes 目录中AppDelegate.cpp 和 jsb_module_register.cpp 修改使其一致

7. 重新构建win32 编译, 如出现错误

```
error MSB8020: 无法找到 v140_xp 的生成工具(平台工具集 =“v140_xp”)
```

用vs2017 打开 cocos 生成的 工程文件 build\jsb-link\frameworks\runtime-src\proj.win32\BlockPoker.sln

vs2017 需要安装xp支持的包 在tools -> 获取工具和能力 -> c++桌面开发 -> Select Windows XP support for C++ from the Summary section 安装即可

8. vs2017 打开BlockPoker.sln 后 项目 -> 重定解决方案 项目 属性 -> 平台工具集中选择 v141_xp 编译

出现错误在 `_scheduler = Application::getInstance()->getScheduler();` 这个地方的全部改成

```
_scheduler = Application::_scheduler;
```

主要涉及 CCDownloader-curl.cpp HttpClient.cpp 和AudioEngine-win32.cpp

注: 这样修改后在android编译也可能出现同样错误, 也需要做一样的修改: 主要是 cocos目录下 D:\CocosCreator_2.2.2\resources\cocos2d-x\cocos\network文件 HttpClient-android.cpp

9. 在vs2017 中编译生成exe, debug 和release 中都选择v141_xp指令集, 在link 依赖中添加 windows_gengine.lib 库, 保存项目配置, 这个时候在vs2017中可以生成BlockPoker.exe

10. 用cocos creator 打开项目, 重新构建, 生成项目在 build\jsb-link\publish\win32

11. 可以前面生成的engine动态库 windows_gengine.dll 拷贝到构建生成的目录build\jsb-link\frameworks\runtime-src\proj.win32\Release.win32 这样在编译生成时会copy到 build\jsb-

link\publish\win32, 也可以直接吧动态依赖库windows_gengine.dll copy到 build\jsb-link\publish\win32

12. 为了防止在没有装vc的机器上运行, 将build_kaleido\jsb-link\frameworks\runtime-src\proj.win32\Release.win32下面的动态库 拷贝到 构建生成的 build\jsb-link\frameworks\runtime-src\proj.win32\Release.win32目录, 这样在cocos 编译都 这些库会直接拷贝入publish\win32目录下
13. 最后将 build_kaleido\jsb-link\frameworks\runtime-src\proj.android-studio\app\res\raw 下 portal.lua copy到 build\jsb-link\publish\win32 , 这样 BlockPoker.exe 就可以直接在本目录运行
14. 打包只需要将build\jsb-link\publish\win32 压缩成rar, 转换成exe 就可以了