# 修改代码

## 1.1 CCFileUtils.cpp

cocos2d-x\cocos\platform\CCFileUtils.cpp

覆盖三个方法的代码:

|  |
| --- |
| #include "xxtea.h"  std::string FileUtils::getStringFromFile(const std::string& filename)  {  auto d = getDataFromFile(filename);  if (d.isNull())  {  return "";  }  ssize\_t size = 0;  char \* buf = (char \*)d.getBytes();  std::string s = buf;  s.resize(d.getSize());  return s;  }  Data FileUtils::getDataFromFile(const std::string& filename)  {  Data d;  getContents(filename, &d);  if (d.isNull())  {  return d;  }  unsigned char\* result = d.getBytes();  if (jyx\_check\_xxtea((const char\*)result))  {  xxtea\_long len = 0;  unsigned char\* temp = jyx\_xxtea\_decrypt((unsigned char\*)result, (xxtea\_long)d.getSize(), &len, filename.rfind(".lua") != std::string::npos);  d.copy(temp, len);  free(temp);  }  return d;  }  unsigned char\* FileUtils::getFileData(const std::string& filename, const char\* mode, ssize\_t \*size)  {  CCASSERT(!filename.empty() && size != nullptr && mode != nullptr, "Invalid parameters.");  (void)(mode); // mode is unused, as we do not support text mode any more...  auto d = getDataFromFile(filename);  if (d.isNull()) {  \*size = 0;  return nullptr;  }  return d.takeBuffer(size);  } |

## 1.2覆盖xxtea

Xxtea 目录拷贝到 cocos2d-x\external

## 1.3修改 AppDelegate.cpp

如:client\Classes\AppDelegate.cpp

|  |
| --- |
| #include "package\_quick\_register.h"  在程序applicationDidFinishLaunching 方法增加 :  package\_quick\_register();  修改启动搜索文件路径  string scriptFile = "src/main.lua"; // 源码目录  string uploadPath = FileUtils::getInstance()->getWritablePath() + "upload/bin/main.lua";  if (FileUtils::getInstance()->isFileExist(uploadPath))  {  scriptFile = uploadPath; // 更新目录  }  else if (FileUtils::getInstance()->isFileExist("res/bin/main.lua"))  {  scriptFile = "res/bin/main.lua"; // 安装目录  }  if (engine->executeScriptFile(scriptFile.c\_str()))  {  return false;  }  return true; |

## 1.4 CCFileUtils-win32.cpp

在第222行(非必要.只是增加大小写判断)

|  |
| --- |
| std:wstring str = StringUtf8ToWideChar(strPath);  const wchar\_t\* chr = str.c\_str();  DWORD attr = GetFileAttributesW(chr);  if(attr == INVALID\_FILE\_ATTRIBUTES || (attr & FILE\_ATTRIBUTE\_DIRECTORY))  return false; // not a file  const int len = str.length();  int index = -1;  for (int i = len - 1; i >= 0; i--)  {  char s = chr[i];  if (s == '/' || s == '\\')  {  index = i;  break;  }  }  int fileNameLen = len - index;  WCHAR\* fileName = new WCHAR[fileNameLen];  for (int i = 0; i < fileNameLen; i++)  {  fileName[i] = 0;  }  for (int i = index + 1; i < len; i++)  {  fileName[i - index - 1] = chr[i];  }  WIN32\_FIND\_DATA FindFileData;  HANDLE hFind = ::FindFirstFile(chr, &FindFileData);  if (INVALID\_HANDLE\_VALUE == hFind)  {  return true;  }  BOOL result = true;  while (TRUE)  {  if (FindFileData.dwFileAttributes & FILE\_ATTRIBUTE\_DIRECTORY)  {  }  else  {  WCHAR\* tmpName = FindFileData.cFileName;  WCHAR\* tmp1 = new WCHAR[fileNameLen];  WCHAR\* tmp2 = new WCHAR[fileNameLen];  wcscpy(tmp1, fileName);  wcscpy(tmp2, tmpName);  wcsupr(tmp1);  wcsupr(tmp2);  if (0 == wcscmp(tmp1, tmp2))  {  delete tmp1;  delete tmp2;  if (0 != wcscmp(fileName, tmpName))  {  CCLOG("JYX ERROR : Wrong file name : %s", strPath.c\_str());  char\* tmp = new char[fileNameLen];  for (int i = 0; i < fileNameLen; i++)  {  tmp[i] = tmpName[i];  }  CCLOG("System file name : %s", tmp);  delete tmp;  result = false;  break;  }  break;  }  else  {  delete tmp1;  delete tmp2;  }  }  if (!FindNextFile(hFind, &FindFileData))  break;  }  FindClose(hFind);  delete fileName;  return result; |

## 1.5 修改动画时间根据设置时间

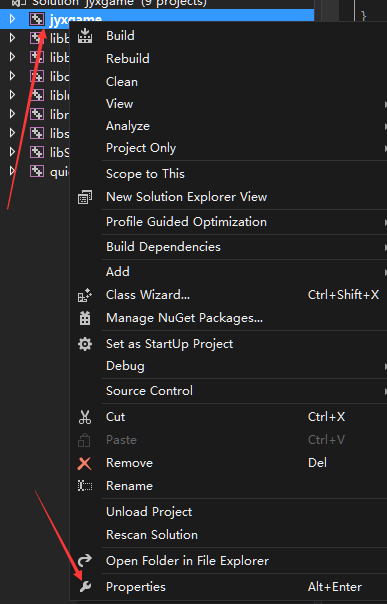
|  |
| --- |
| cocos2d-x\cocos\editor-support\cocostudio\ActionTimeline\CCActionTimeline.cpp  #include "cocos/base/CCDirector.h"  构造方法增加方法:  \_frameInternal = Director::getInstance()->getAnimationInterval(); |

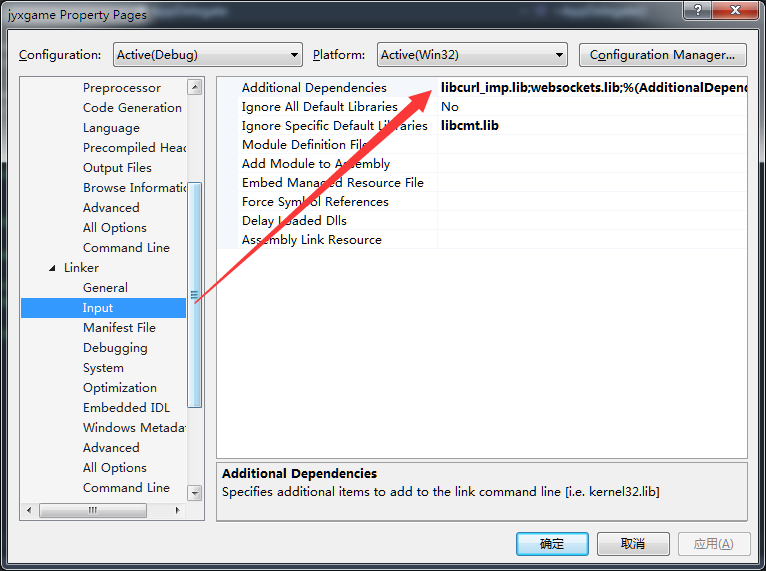
## 1.6 覆盖zlib/include

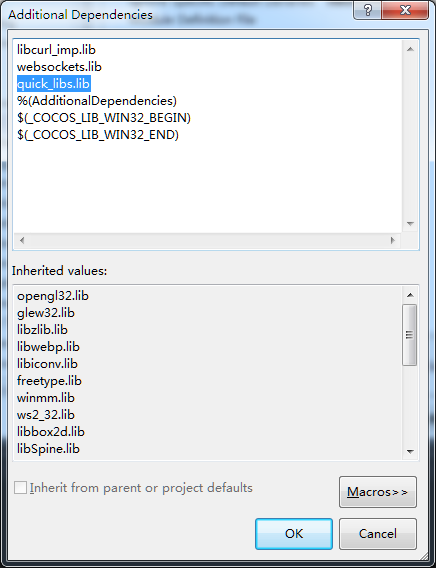
# 3.WIN32:

## 3.1.将项目 quick\_lib 增加到工程

在原项目使用link 连接到quick\_libs.lib







## 3.2 为编译项目增加头文件搜索路径:

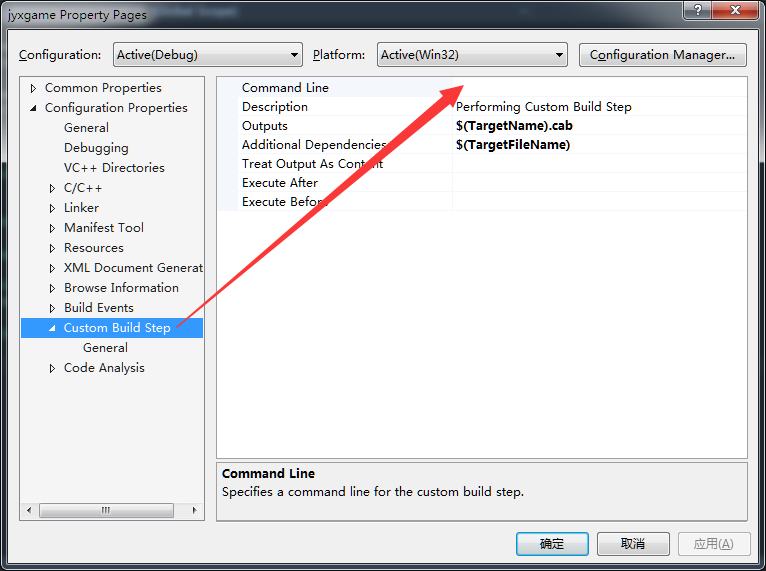
$(EngineRoot)..\quick\_libs\

$(EngineRoot)..\quick\_libs\src\extra

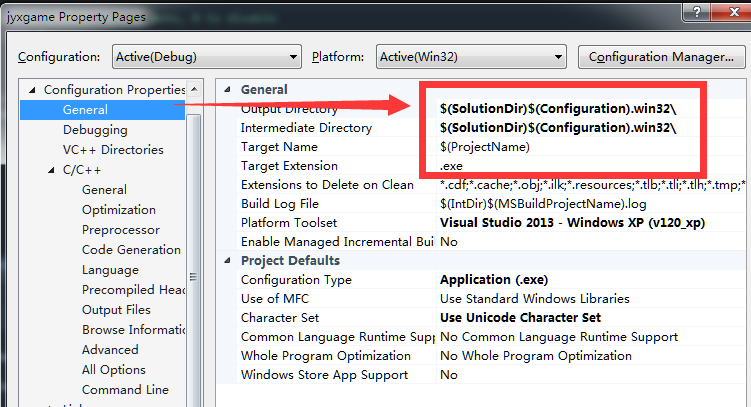
$(EngineRoot)..\quick\_libs\src\extensions

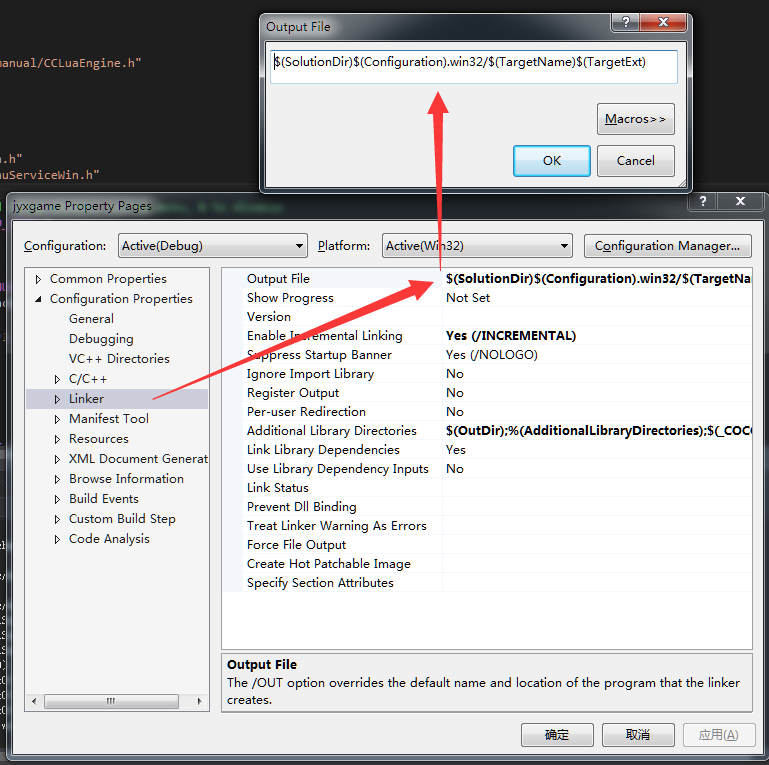
$(EngineRoot)cocos\scripting\lua-bindings\manual

## 3.3 删除编译后的CMD命令



## 3.4 修改输出路径和运行路径





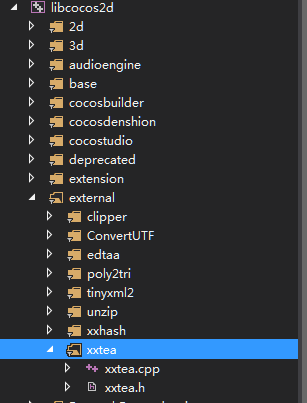
和上图一样: $(SolutionDir)$(Configuration).win32/$(TargetName)$(TargetExt)

## 3.5 为libcocos2d项目增加头文件搜索路径和增加xxtea引用:

$(EngineRoot)external\xxtea

## 3.6 为libluacocos2d项目增加头文件搜索

$(EngineRoot)external\win32-specific\zlib\include



# 4.android:

## 4.1增加安卓引用

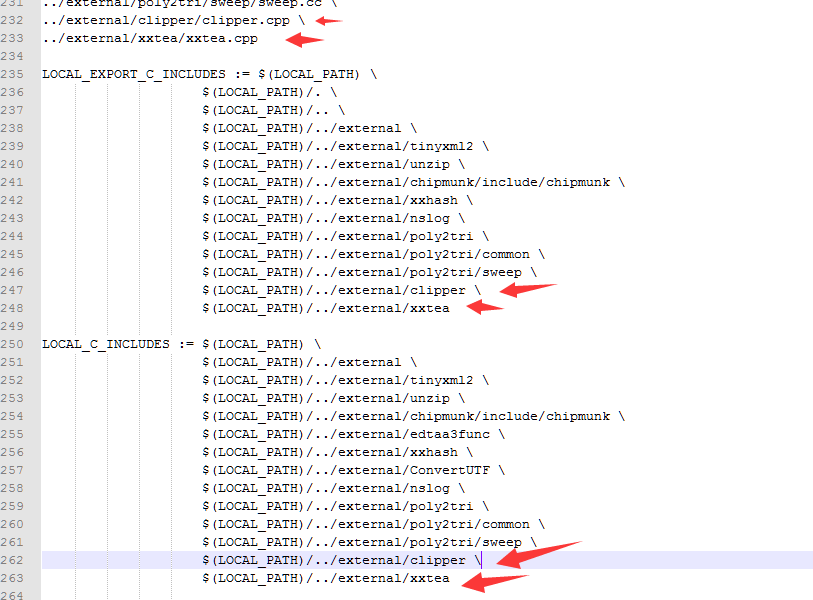
cocos2d-x\cocos\Android.mk

最下面增加(235行左右)引用

../external/xxtea/xxtea.cpp

LOCAL\_EXPORT\_C\_INCLUDES 和 LOCAL\_C\_INCLUDES 中增加 :

$(LOCAL\_PATH)/../external/xxtea



## 4.2 编译项目中的Android.mk 增加

如: client\android\debug\jni\Android.mk

# QUICK\_LIB\_DEPENDENCY

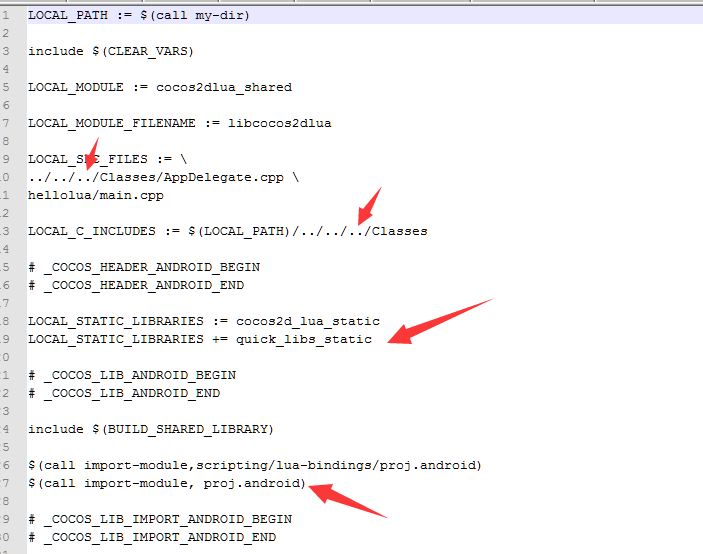
LOCAL\_STATIC\_LIBRARIES += quick\_libs\_static

和:

# QUICK\_LIB\_IMPORT

$(call import-module, proj.android)

注意目录结构变化

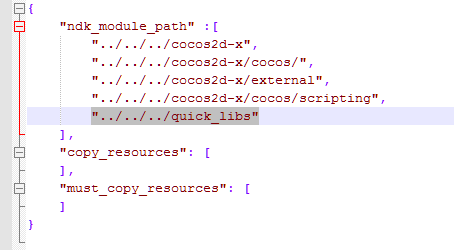


## 4.3 设置配置文件

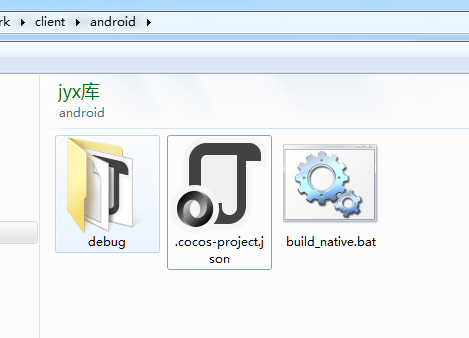
如: client\android\debug\build-cfg.json

"ndk\_module\_path" 增加 "../../../quick\_libs" ,注意目录结构原来是../../现在变成../../../三层

同时删除拷贝资源代码(如果你使用这种拷贝则不需要删除)



## 4.4 将.cocos-project.json拷贝到client/andoird目录



并且增加编译项目:

Debug就是我们的安卓目录

|  |
| --- |
| ,  "android\_cfg":{  "project\_path":"debug"  } |
|  |

同时创建批处理,并运行批处理编译

(注意:要将下载的cocos3.X版本下的tool/cocos2d-console拷贝到cocos2d-x目录

|  |
| --- |
| set DIR=%~dp0  set cocos=%DIR%..\..\cocos2d-x\tools\cocos2d-console\bin\cocos.bat  call %cocos% compile -p android --no-apk -m release -j 4  copy %DIR%debug\libs\armeabi\libcocos2dlua.so %DIR%..\android\_depends\liblyt\libs\armeabi\libcocos2dlua.so  copy %DIR%debug\obj\local\armeabi\libcocos2dlua.so %DIR%debug\libcocos2dlua\_obj.so  pause |

## 4.5必须使用10C+的NDC版本来编译

# 5.IOS

## 5.1.增加lquick\_libs 项目引用并编译包含 libquick\_libs IOS.a,增加头文件搜索路径:

"$(SRCROOT)/../../quick\_libs",

"$(SRCROOT)/../../quick\_libs/src",

"$(SRCROOT)/../../quick\_libs/proj.ios\_mac/third\_libs",

## 5.3修改cocos2dxlib 头文件搜索路径:

$(SRCROOT)/../external/xxtea



## 5.3使用新的ICON方式

## 5.4修改EditBox

|  |
| --- |
| 在 CCUIEditBoxIOS.mm 文件修改：  - (BOOL)textFieldShouldReturn:(UITextField \*)sender  {  if (sender == self.textInput) {  [sender resignFirstResponder];  getEditBoxImplIOS()->editBoxEditingDidEnd2();  }  return NO;  }  在UIEditBoxImpl-common.h 文件增加新方法  void editBoxEditingDidEnd2();  UIEditBoxImpl-common.h  void editBoxEditingDidEnd2(const std::string& text);  UIEditBoxImpl-common.cpp 增加新方法定义  void EditBoxImplCommon::editBoxEditingDidEnd2()  {  #if CC\_ENABLE\_SCRIPT\_BINDING  if (NULL != \_editBox && 0 != \_editBox->getScriptEditBoxHandler())  {  cocos2d::CommonScriptData data(\_editBox->getScriptEditBoxHandler(), "send", \_editBox);  cocos2d::ScriptEvent event(cocos2d::kCommonEvent, (void \*)&data);  cocos2d::ScriptEngineManager::getInstance()->getScriptEngine()->sendEvent(&event);  }  #endif  } |