# 安装

## win32 android

### 说明

【18:41 2014-9-29】单独导入一个android项目编译出错

//================================================

这是因为缺少Cocos2d-x Android JNI桥接库（bridge library）。解决方法：导入另一个Android项目，项目位置为“C:\cocos2d-x-2.2.0\cocos2dx\platform\android”。导入项目之后，错误即会消失

【12:58 2014-9-24】移植卡牌塔防说明

//================================================

Q,在mygame项目中MyGame.java文件中的package com.MyCompany.AwesomeGame;是什么时候指定的?

A,是由create\_project.py指定的参数

Q,MyGame.java下的System.loadLibrary("cocos2dcpp");名字是怎么指定的？

A,表示libcocos2dcpp.so java动态库，这个与jni\Android.mk下的

LOCAL\_MODULE\_FILENAME := libcocos2dcpp 一致

\*\*\*\*\*改了代码先在cygwin下调用./build\_native.sh 编译，完后在eclipse下执行

(暂时没搞懂为什么eclipse不调用build\_native.sh编译，之前都可以的。。)

【15:49 2014-9-22】安卓环境

//================================================

参考链接：

http://cn.cocos2d-x.org/article/index?type=cocos2d-x&url=/doc/cocos-docs-master/manual/framework/native/v2/getting-started/setting-up-development-environments/windows-7-environment-setup/how-to-set-up-the-android-cocos2d-x-development-environment-on-windows7/zh.md

http://cn.cocos2d-x.org/article/index?type=cocos2d-x&url=/doc/cocos-docs-master/manual/framework/native/v3/getting-started/setting-up-development-environments-on-windows7-with-eclipse/zh.md

//需要软件

python jdk adt(eclipse 、androidsdk) NDK、cygwin、cocos2dx

D:\Python27 (2.7.6)

D:\Program Files\Java\jdk1.7.0\_40

E:\AppDev\adt-bundle-windows-x86-20140702\eclipse

E:\AppDev\adt-bundle-windows-x86-20140702\sdk

E:\AppDev\android-ndk32-r10b-windows-x86

D:\cygwin (4.1.10)

E:\AppDev\cocos2d-x-2.2.4

说明：

ADT //eclipse插件

NDK //eclipse C++编译

\*\*\*\*\*日，用 "create\_project.py -project MyGame -package com.MyCompany.AwesomeGame -language cpp"

创建项目老出错，把python从3.3换回到2.7版本就OK了(2.7.6)

\*\*\*\*\*编译时找不到bash...

cygwin没加入到系统环境变量里,加入环境后重启电脑后才没报错...

环境变量说明：

ANDROID\_SDK: E:\AppDev\adt-bundle-windows-x86-20140702\sdk\platforms;E:\AppDev\adt-bundle-windows-x86-20140702\sdk\platform-tools;E:\AppDev\adt-bundle-windows-x86-20140702\sdk\tools

classpath: %JAVA\_HOME%\lib;%JAVA\_HOME%\lib\tools.jar;

CYGWIN: D:\cygwin

java\_home: D:\Program Files\Java\jdk1.7.0\_40

NDK\_ROOT: E:\AppDev\android-ndk32-r10b-windows-x86

Path: D:\cygwin\bin;D:\cygwin;%NDK\_ROOT%;%ANDROID\_SDK%;D:\Python27;%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;D:\Microsoft DirectX SDK (March 2008)\Utilities\Bin\x86;C:\Program Files\NVIDIA Corporation\PhysX\Common;%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;C:\Program Files\Kaspersky Lab\Kaspersky Anti-Virus 6.0 for Windows Workstations MP4\;C:\Program Files\Microsoft SQL Server\90\Tools\binn\;D:\Program Files\Perforce;C:\Program Files\Common Files\Ulead Systems\MPEG;d:\Program Files\IDM Computer Solutions\UltraEdit-32;C:\Program Files\CMake 2.8\bin;c:\Program Files\Microsoft SQL Server\100\Tools\Binn\;c:\Program Files\Microsoft SQL Server\100\DTS\Binn\;D:\Program Files\TortoiseHg\

# 编码

## 消息相关

### 初步认识

【19:18 2014-9-30】压缩 解压

//================================================

ztServer\zlib-1.2.3.win32\include\zconf.h

【19:18 2014-9-30】加密 解密

//================================================

ztServer\libsrvcore\miniCrypt.c

【14:57 2014-9-30】Server

//================================================

int WorldSocket::handle\_output\_queue (GuardType& g)

//循环从命令队列里取数据包组合起来，直到取完或发送区满

while(PacketBufferQueue::BuffPkt\* pb = m\_CmdSendQueue.PeekReadPacket())

。。。

peer().send...

WorldSocket::SendPacket (const WorldPacket& pct)

QueueCmdPacket(const WorldPacket & pkt)

m\_CmdSendQueue.WritePacket(pkt);

【14:04 2014-9-30】LZCS消息发送

//================================================

SocketBuffer \*sendBuffer = SocketBuffer::defaultSocketBuffer();

sendBuffer->writeShort(LoginDataCenter::USER\_SOCKET\_APPLICATION\_ACCOUNT);

LoginDataCenter::sharedLoginDataCenter()->send(sendBuffer,this);

UserSocketManager::sharedUserSocketManager()->send(sendBuffer);

\_userSendQueue.Push(data);

【18:41 2014-9-29】tsdemo消息结构

17+要传送的buff长度

//================================================

char HEAD0;

char HEAD1;

char HEAD2;

char HEAD3;

char ProtoVersion;

byte serverVersion[4];

byte length[4];

byte commandId[4];

/\*\*

\* 消息的数据

\*/

char\* data;

**发送：**

Json::FastWriter writer;

Json::Value person;

person["username"]=userName;

person["password"]=password;

std::string json\_file=writer.write(person);//192.168.1.210 114.252.70.61 183.60.243.195

CCLog("%s",json\_file.c\_str());

SocketManager::getInstance()->sendMessage(json\_file.c\_str(), 101);

### 标准消息

2014-10-9

//================================================

PackHead(4字节) + 内容(PackLength个字节)

Struct{

unsigned long packHead;

stNullUserCmd cmd;

}

注：

packHead除了包含包的长度信息，还包含是否压缩加密的信息

packHead && 0x00ffffff 表示包的长度（最长0x00ffffff， 长度是实际内容的长度，不包含这个packHead自身）

packHead && 0x80000000 表示加密

packHead && 0x40000000 表示压缩

加密：根据不同登入状态以RC5或者DES进行加密。 从包头开始(从packHead),每8个字节一组进行加密

加密前整个包的长度(含包头)调整成8的整数倍

发送消息在主线程中

接受消息有独立线程

# 调试

# 发布

## 11

# 更新