

Android 编译SDL2

ndk sdl android

Android 使用cmake 编译SDL2

SDL2官网下载



Main

- About
- Bugs
- Licensing
- Credits
- Feedback

Documentation

- Wiki
- Forums
- Mailing Lists

Download

- SDL Releases
- SDL Mercurial
- Bindings

2.0.12

SDL version 2.0.12 (stable)

Source Code:

- SDL2-2.0.12.zip - GPG signed
- SDL2-2.0.12.tar.gz - GPG signed

Runtime Binaries:

Windows:

- SDL2-2.0.12-win32-x86.zip (32-bit Windows)
- SDL2-2.0.12-win32-x64.zip (64-bit Windows)

Mac OS X:

- SDL2-2.0.12.dmg

Linux:

Please contact your distribution maintainer for updates.

GPG Signature:

The source code to this release has been signed by Sam Lantinga.

You can get the public key from any keyserver with the ID 1024D/A7763BE6, or directly from Sam's home page: [slopesofmud@pubkey.asc](#)

The public key fingerprint should be:

```
pub 1024D/A7763BE6 2001-01-05 Sam Lantinga <slopesofmud@lunarg.com>
Key fingerprint = 1528 635D 8053 A57F 77D1 E08E
```

For more information about public key signatures, see <http://www.gnupg.org/>

如这里解压得到 SDL2-2.0.12:

名称	修改日期	类型	大小
acinclude	2020/3/11 9:36	文件夹	
android-project	2020/3/11 9:36	文件夹	
build-scripts	2020/3/11 9:36	文件夹	
cmake	2020/3/11 9:36	文件夹	
debian	2020/3/11 9:36	文件夹	
docs	2020/3/11 9:36	文件夹	
include	2020/3/11 9:36	文件夹	
src	2020/3/11 9:36	文件夹	
test	2020/3/11 9:36	文件夹	
VisualC	2020/3/11 9:36	文件夹	
VisualC-WinRT	2020/3/11 9:36	文件夹	
wayland-protocols	2020/3/11 9:36	文件夹	
Xcode	2020/3/11 9:36	文件夹	
Xcode-iOS	2020/3/11 9:36	文件夹	
Android.mk	2020/3/11 9:36	MK 文件	4 KB
autogen.sh	2020/3/11 9:36	SH 文件	1 KB
BUGS.txt	2020/3/11 9:36	TXT 文件	1 KB
cmake_uninstall.cmake.in	2020/3/11 9:36	IN 文件	1 KB
CMakeLists.txt	2020/3/11 9:36	TXT 文件	75 KB
configure	2020/3/11 9:36	文件	823 KB
configure.ac	2020/3/11 9:36	AC 文件	158 KB
COPYING.txt	2020/3/11 9:36	TXT 文件	1 KB
CREDITS.txt	2020/3/11 9:36	TXT 文件	2 KB
INSTALL.txt	2020/3/11 9:36	TXT 文件	2 KB
Makefile.in	2020/3/11 9:36	IN 文件	8 KB
Makefile.minimal	2020/3/11 9:36	MINIMAL 文件	1 KB
Makefile.os2	2020/3/11 9:36	OS2 文件	5 KB
Makefile.pandora	2020/3/11 9:36	PANDORA 文件	2 KB
Makefile.psp	2020/3/11 9:36	PSP 文件	4 KB
Makefile.wiz	2020/3/11 9:36	WIZ 文件	3 KB
README.txt	2020/3/11 9:36	TXT 文件	1 KB
README-SDL.txt	2020/3/11 9:36	TXT 文件	1 KB
sdl2.m4	2020/3/11 9:36	M4 文件	8 KB
sdl2.pc.in	2020/3/11 9:36	IN 文件	1 KB
SDL2.spec	2020/3/11 9:36	SPEC 文件	4 KB
SDL2.spec.in	2020/3/11 9:36	IN 文件	4 KB
SDL2Config.cmake	2020/3/11 9:36	CMAKE 文件	1 KB
sdl2-config.cmake.in	2020/3/11 9:36	IN 文件	2 KB
sdl2-config.in	2020/3/11 9:36	IN 文件	2 KB
sdl2-config-version.cmake.in	2020/3/11 9:36	IN 文件	1 KB
TODO.txt	2020/3/11 9:36	TXT 文件	1 KB

几个重要的目录与文件说明

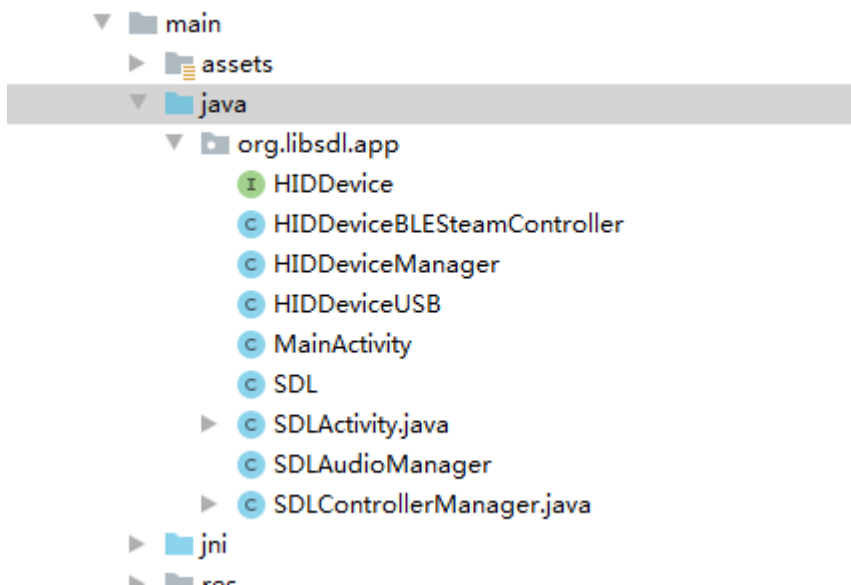
- 1 **android-project**: **Android** 项目的模板项目
- 2 **docs**: 说明文档
- 3 **include**: **SDL2**的头文件
- 4 **src**: **SDL2**的源文件
- 5 **Android.mk**: **makefile**的构建脚本, 编译动态库和静态库
- 6 **CMakeLists.txt**: **CMake**的构建脚本, 编译动态库和静态库

1、搭建项目

使用android-project 作为一个模板搭建一个项目，必须包名一致（SDLActivity 里声明了 native方法，如果包名不一致，本地方法与jni层的映射的方法对应不上，会报错，可以修改，不过先别管这么多，先跑起来再说）

1、新建一个项目，包名取org.libsdl.app

2、把模板项目android-project 下的文件复制过来，org.libsdl.app 下的所有文件，AndroidManifest.xml, app的build.gradle，复制后如下：



build.gradle

```
1
2  apply plugin: 'com.android.application'
3
4  android {
5      compileSdkVersion 30
6      buildToolsVersion "30.0.0"
7
8      defaultConfig {
9          applicationId "org.libsdl.app"
10         minSdkVersion 21
11         targetSdkVersion 30
12         versionCode 1
13         versionName "1.0"
14
15         testInstrumentationRunner "androidx.test.runner.AndroidJUnit4"
16         externalNativeBuild {
17
18             //          ndkBuild {
19             //              arguments "APP_PLATFORM=android-21"
20             //              abiFilters 'armeabi-v7a'
21             //          }
22
```

```

23
24         cmake {
25             arguments "-DANDROID_APP_PLATFORM=android-21", "-DANI
26             cppFlags ""
27             abiFilters 'armeabi-v7a'
28         }
29     }
30 }
31
32 buildTypes {
33     release {
34         minifyEnabled false
35         proguardFiles getDefaultProguardFile('proguard-android-opti
36     }
37 }
38
39 sourceSets.main {
40     jniLibs.srcDir 'libs'
41 }
42 externalNativeBuild {
43
44     // ndkBuild {
45     //     path 'src/main/jni/Android.mk'
46     // }
47
48     cmake {
49         //这里需要指定到 jni/CMakeLists.txt 文件
50         path "src/main/jni/CMakeLists.txt"
51         version "3.10.2"
52     }
53 }

```

(想用cmake 编译就打开cmake，想用makefile 方式就打开ndkBuild，这里使用了cmake方式)

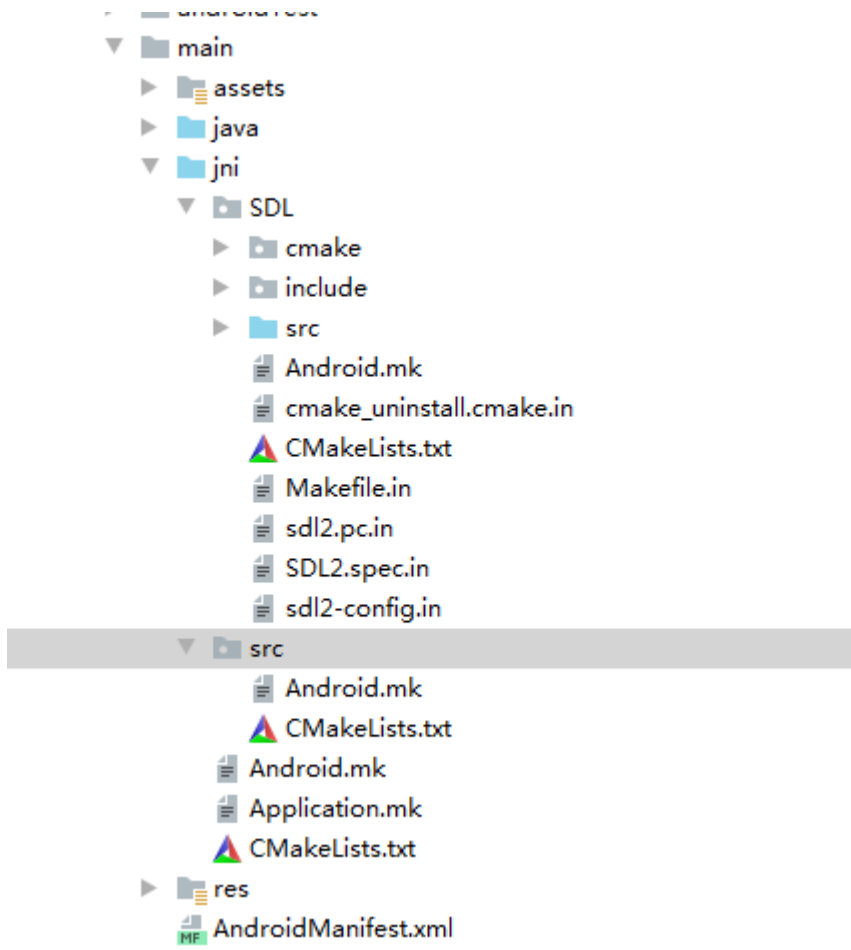
AndroidManifest.xml 文件不贴了，原样照搬

3、

1. 新建jni 文件夹，在jni 文件中新建SDL文件夹，把解压文件（SDL2-2.0.12）中的cmake、include、src文件夹直接复制到SDL文件夹中

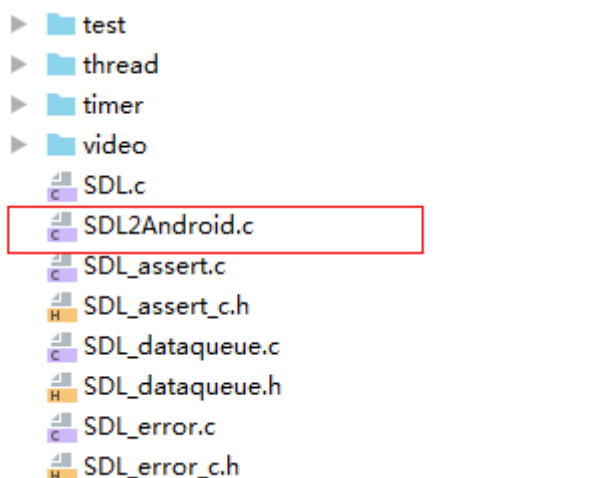
2. 把解压文件夹（SDL2-2.0.12） 中的所有 .in 结尾的文件和Android.mk、CMakeLists.txt也复制到SDL 文件中
3. 在把模板项目中SDL2-2.0.12\android-project\app\jni 下的所有文件复制到jni 文件夹下

完成后项目结构如下：



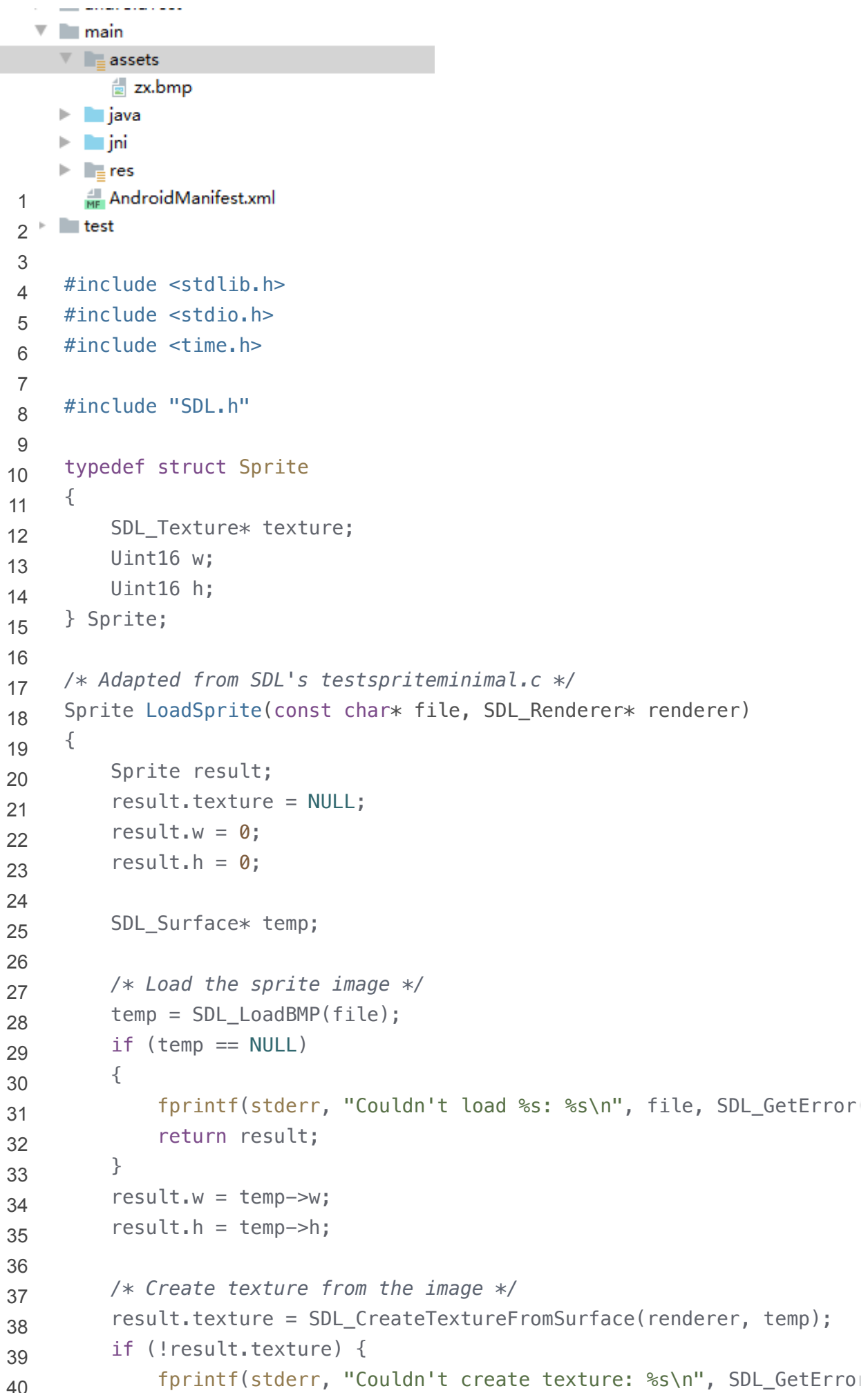
（这里也把.mk 文件放进来，如果只是想用cmake 方式也不复制过来）

- 4、新建SDL2Android.c文件放到 jni/SDL/src 中



这里SDL2Android.c 文件作用是绘制一张图片,内容如下：[SDL详细使用](#)

文件中使用了zx.bmp图片，建一个assets文件，随便把一张图片放进去就行



```

1
2
3
4 #include <stdlib.h>
5 #include <stdio.h>
6 #include <time.h>
7
8 #include "SDL.h"
9
10 typedef struct Sprite
11 {
12     SDL_Texture* texture;
13     Uint16 w;
14     Uint16 h;
15 } Sprite;
16
17 /* Adapted from SDL's testspriteminimal.c */
18 Sprite LoadSprite(const char* file, SDL_Renderer* renderer)
19 {
20     Sprite result;
21     result.texture = NULL;
22     result.w = 0;
23     result.h = 0;
24
25     SDL_Surface* temp;
26
27     /* Load the sprite image */
28     temp = SDL_LoadBMP(file);
29     if (temp == NULL)
30     {
31         fprintf(stderr, "Couldn't load %s: %s\n", file, SDL_GetError());
32         return result;
33     }
34     result.w = temp->w;
35     result.h = temp->h;
36
37     /* Create texture from the image */
38     result.texture = SDL_CreateTextureFromSurface(renderer, temp);
39     if (!result.texture) {
40         fprintf(stderr, "Couldn't create texture: %s\n", SDL_GetError());

```

```

41         SDL_FreeSurface(temp);
42         return result;
43     }
44     SDL_FreeSurface(temp);
45
46     return result;
47 }
48
49 void draw(SDL_Window* window, SDL_Renderer* renderer, const Sprite s)
50 {
51     int w, h;
52     SDL_GetWindowSize(window, &w, &h);
53     SDL_Rect destRect = {w/2 - sprite.w/2, h/2 - sprite.h/2, sprite.w,
54     /* Blit the sprite onto the screen */
55     SDL_RenderCopy(renderer, sprite.texture, NULL, &destRect);
56 }
57
58 /**
59  * 图片绘制，一定是个main方法
60  * SDL 会自动调用这个方法，完成图片绘制与显示
61  * @param argc
62  * @param argv
63  * @return
64  */
65 int main(int argc, char *argv[])
66 {
67     SDL_Window *window;
68     SDL_Renderer *renderer;
69
70     if(SDL_CreateWindowAndRenderer(0, 0, 0, &window, &renderer) < 0)
71         exit(2);
72
73     Sprite sprite = LoadSprite("zx.bmp", renderer);
74     if(sprite.texture == NULL)
75         exit(2);
76
77     /* Main render loop */
78     Uint8 done = 0;
79     SDL_Event event;
80     while(!done)
81     {
82         /* Check for events */
83         while(SDL_PollEvent(&event))
84         {
85             if(event.type == SDL_QUIT || event.type == SDL_KEYDOWN ||
86             {
87                 //done = 1;

```

```

88         }
89     }
90
91     /* Draw a gray background */
92     SDL_SetRenderDrawColor(renderer, 0xA0, 0xA0, 0xA0, 0xFF);
93     SDL_RenderClear(renderer);
94
95     draw(window, renderer, sprite);
96
97     /* Update the screen! */
98     SDL_RenderPresent(renderer);
99
10    SDL_Delay(10);
0    }
10
1    exit(0);
10 }
2

```

5、修改 jni/src/CMakeLists.txt 文件，修改后如下，注释已经说明了

```

1  cmake_minimum_required(VERSION 3.4.1)
2
3  project(MY_APP)
4
5  #这里添加了头文件查找目录
6  include_directories(${CMAKE_SOURCE_DIR}/SDL/include)
7
8  #生成main.so 需要引用SDL2.so库 (SDL2.so是脚本文件jni/SDL/CMakeLists.txt
9  find_library(SDL2 SDL2)
10
11  add_library(main SHARED)
12
13  #添加这个main.so 的原文件，把 SDL2Android.c 放到这个so库中 (SDL_android_ma
14  target_sources(main PRIVATE ${CMAKE_SOURCE_DIR}/SDL/src/main/android,
15  ${CMAKE_SOURCE_DIR}/SDL/src/SDL2Android.c)
16
17  #把SDL2.so 加入main.so 中
18  target_link_libraries(main SDL2)
19
20

```

6、jni/CMakeLists.txt 文件内容


```
1  cmake_minimum_required(VERSION 3.4.1)
2
3  project(GAME)
4
5  #编译 SDL 目录下的cmake文件
6  add_subdirectory(SDL)
7
8  #编译 src 目录下的cmake文件
9  add_subdirectory(src)
```

7、记得把app 中build.gradle 中cmake 指定到 jni/CMakeLists.txt 文件



```
    }
    externalNativeBuild {
        // ndkBuild {
        //     path 'src/main/jni/Android.mk'
        // }
        cmake {
            //这里需要指定到 jni/CMakeLists.txt 文件
            path "src/main/jni/CMakeLists.txt"
            version "3.10.2"
        }
    }
}
```

8、编译项目，会在build/intermediates/cmake/debug/obj/armeabi-v7a 下生成想要的动态库与静态库（这里只编译了armv7a的）

▼ app

▶ .cxx

▼ build

▶ generated

▼ intermediates

▶ annotation_processor_list

▶ blame

▶ bundle_manifest

▼ cmake

▼ debug

▼ obj

▼ armeabi-v7a

libc++_static.a

libc++abi.a

libhidapi.so

libmain.so

libSDL2.so

libunwind.a

▶ compatible_screen_manifest

3

3

3

3

3

3

3

3

3

3

4

4

4

4

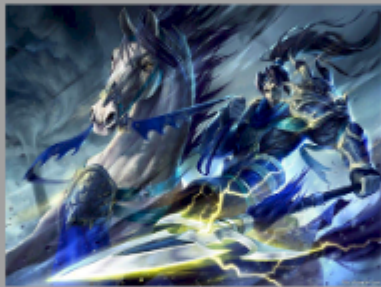
4

4

4

4

9、运行项目：



版权声明：本文为CSDN博主「qq_37382732」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/qq_37382732/article/details/108796638

更多相关推荐

交叉编译sdl2成android的.so库

sdl **android** 交叉编译

去官网下载sdl2的源码，把源码解压，我下载的是2.0.7，解压会生成SDL2-2.0.7。如果大家还有不了解的可以去我的之前的博文<http://blog.csdn.net/u013654125/article/details/79157944>的一、二小节了解。博文里有SDL2-...

sdl for android 编译,SDL2库(1)-移植Android 端 CMakeList 集成

sdl for android 编译

SDL.png系列内容导读简单的集成1.获取源代码SDL的源代码获取十分简单。访问SDL的官方网站(<http://www.libsdl.org/>)，单击左侧的“Download”进入下载页面，然后下载“SourceCode”栏目下的文件就可以了。解压完后的文件....

android 编译 sdl,使用android ndk编译SDL2示例错误r14

android 编译 **sdl**

我已经测试过在我的ubuntu16.04机器上构建SDL2源代码(2.0.5)中的示例。使用androidndk编译SDL2示例错误r14根据<https://wiki.libsdl.org/Android>，我安装了androidsdk和ndkr14并设置了我的环境变量。但是我在“简单构....

sdl2编译成ios .a库

ios **sdl**

一、编译环境sdl库编译成.a库，应该没有什么环境的要求，但是在这里还是先说一下自己的环境。
mac10.12.6Xcode9.2sdl2SDL2-2.0.7二、准备工作大家先在sdl官网下载sdl2源码其中SDL2-2.0.7.zip和SDL2-2.0.7.tar.gz里是....

SDL2像素操作

sdl **sdl2**

SDL2像素操作经过几天的研究,摸索出SDL2像素操作的方式操作surface中的像素操作texture中的像素SDL2中,surface是直接放在内存中的,没有硬件加速.而texture放在显存中,有硬件加速,原理上应该效率更高.直接贴两段代码s...

[首页](#)

[技术博客](#)

[联系我们](#)

[版权申明](#)

[隐私条款](#)

© 2023 All rights reserved by CodeAntenna.com.