动态切换龙骨动画也需要先设为不可见，改完数据后再设为可见，否则不会生效

Orange 账号 ：[1395945956@qq.com](mailto:1395945956@qq.com)

密码：Lp17674907410

适配挖孔屏：

api要在android9.0可用，所以compileSdkVersion要 >=28,目标sdkversion为28，最低19

导入模块：**import** android.os.Build;

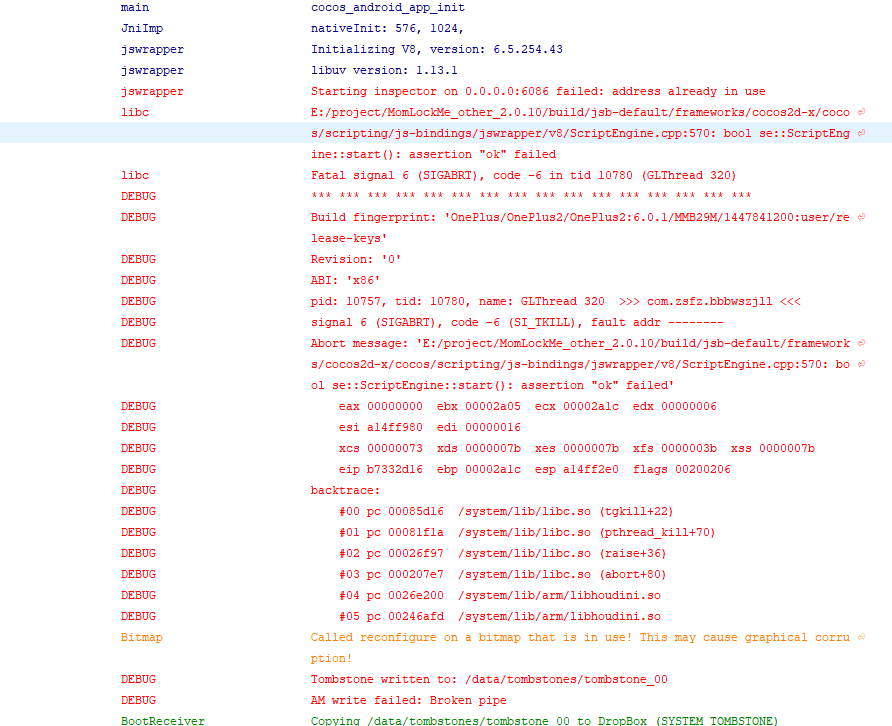
**import** android.view.WindowManager;

（在AndroidStudio中快速导入模块，选中类名，alt +enter可快速导入头文件）

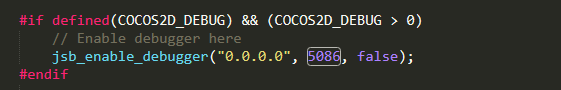
在onCreat中加入

**if** (android.os.Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***P***) {  
 getWindow().getDecorView().setSystemUiVisibility(View.***SYSTEM\_UI\_FLAG\_FULLSCREEN*** | View.***SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN***);  
 WindowManager.LayoutParams lp = getWindow().getAttributes();  
 lp.**layoutInDisplayCutoutMode** = WindowManager.LayoutParams.***LAYOUT\_IN\_DISPLAY\_CUTOUT\_MODE\_SHORT\_EDGES***;  
}

Creator 两个安卓包不能同时运行，即后台挂起一个安卓包后，再去打开另一个安卓包会闪退，报错信息：



解决方案：修改AppDelegate.cpp下的端口号为不一样的，默认统一为6086.



包体构建选release debug暂未尝试

release和debug的区别在于debug是调试包，相较于release包体略大

Js 中输出 false == 0 的值为 true。 注意这个细节

检查ndk ndk-build

sdk adb

Nodejs node -v

忽略谷歌警告：

app 下build.grade 下添加

lintOptions{

disable 'GoogleAppIndexingWarning'

}

Ts定义初值的时候一定要设置默认值为null，不然在creator中会有警告。

正常情况下：

   @property(cc.Node)

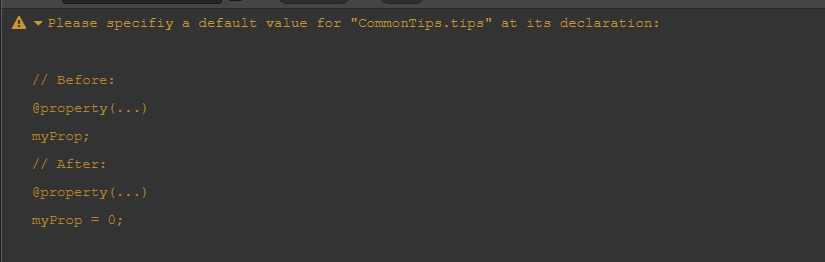
    tips:cc.Node = null;

如果没有设置初值

   @property(cc.Node)

    tips:cc.Node = null;

就会报警告



splice() 方法向/从数组中添加/删除项目，然后返回被删除的项目。

**注释：**该方法会改变原始数组。

|  |  |
| --- | --- |
| **参数** | **描述** |
| index | 必需。整数，规定添加/删除项目的位置，使用负数可从数组结尾处规定位置。 |
| howmany | 必需。要删除的项目数量。如果设置为 0，则不会删除项目。 |
| item1, ..., itemX | 可选。向数组添加的新项目。 |

删除数组中首个元素

  self.\_getFoodList.splice(0,1);

**返回的是一个数组，数组中包含删除的元素**

slice() 方法可从已有的数组中返回选定的元素。

|  |  |
| --- | --- |
| **参数** | **描述** |
| start | 必需。规定从何处开始选取。如果是负数，那么它规定从数组尾部开始算起的位置。也就是说，-1 指最后一个元素，-2 指倒数第二个元素，以此类推。 |
| end | 可选。规定从何处结束选取。该参数是数组片断结束处的数组下标。如果没有指定该参数，那么切分的数组包含从 start 到数组结束的所有元素。如果这个参数是负数，那么它规定的是从数组尾部开始算起的元素。 |

self.\_getFoodList[0].slice(0,5) //返回数组中前5个值

self.\_getFoodList[0].slice(-6) //返回数组中最后六个值

例如： “asdfghjkl”.slice(0,2) = as;

“asdfghjkl”.slice(-3) = jkl;

## **提示和注释**

**注释：**请注意，splice() 方法与 slice() 方法的作用是不同的，splice() 方法会直接对数组进行修改。

图片渲染模式选择平铺，当图片尺寸过大（超过10W)会严重降低frame time值

cc.view.enableAntiAlias(true);  
开启抗锯齿

Android 设备的CPU类型(通常称为”ABIs”)

armeabi-v7a: 第7代及以上的 ARM 处理器。2011年15月以后的生产的大部分Android设备都使用它.  
arm64-v8a: 第8代、64位ARM处理器，很少设备，三星 Galaxy S6是其中之一。  
armeabi: 第5代、第6代的ARM处理器，早期的手机用的比较多。  
x86: 平板、模拟器用得比较多。  
x86\_64: 64位的平板。

1.armeabi：默认选项，将创建以基于

ARM\* v5TE 的设备为目标的库。 具有这种目标的浮点运算使用软件浮点运算。 使用此 ABI （二进制接口）

创建的二进制代码将可以在所有 ARM\*  
设备上运行。所以armeabi通用性很强。但是速度慢

2.armeabi-v7a：创建支持基于

ARM\* v7 的设备的库，并将使用硬件 FPU 指令。armeabi-v7a是针对有浮点运算或高级扩展功能的arm v7 cpu。

3.x86:支持基于硬件的浮点运算的

IA-32 指令集。x86是可以兼容armeabi平台运行的，无论是armeabi-v7a还是armeabi，同时带来的也是性能上的损耗，

另外需要指出的是，打包出的x86的so，总会比armeabi平台的体积更小。

总结

如果项目只包含了 armeabi，那么在所有Android设备都可以运行；  
如果项目只包含了 armeabi-v7a，除armeabi架构的设备外都可以运行；

如果项目只包含了 x86，那么armeabi架构和armeabi-v7a的Android设备是无法运行的； 如果同时包含了 armeabi， armeabi-v7a和x86，所有设备都可以运行，程序在运行的时候去加载不同平台对应的so，这是较为完美的一种解决方案，同时也会导致包变大。

最后，如果我们只想支持armeabi-v7a，那么需要在gradle中配置

defaultConfig {

ndk {

abiFilters "armeabi-v7a"

}

}

因为默认情况下，打包后会自动生成armeabi 到 x86的所有文件夹。这就有可能导致一些x86的设备因为在x86文件夹下找不到so文件而崩溃。

* default，使用默认的 cocos2d-x 源码版引擎构建项目
* link，与 default 模板不同的是，link 模板不会拷贝 cocos2d-x 源码到构建目录下，而是使用共享的 cocos2d-x 源码。这样可以有效减少构建目录占用空间，以及对 cocos2d-x 源码的修改可以得到共享

物理碰撞体一定要记得开enabledContactListener

才能进入碰撞回调。

Projec.json下有版本信息

Ts和js 混用，最终都会编译成es5 的js代码。

instanceof 就是判断一个实例是否属于某种类型

cc.assertID(scene instanceof cc.Scene, 1216);

ES6标准新增了一种新的函数：Arrow Function（箭头函数）。

箭头函数看上去是匿名函数的一种简写，但实际上，箭头函数和匿名函数有个明显的区别：箭头函数内部的this是词法作用域，由上下文确定。

回顾前面的例子，由于[JavaScript](http://lib.csdn.net/base/18" \o "JavaScript知识库" \t "https://www.cnblogs.com/hailun/p/_blank)函数对this绑定的错误处理，下面的例子无法得到预期结果：

var obj = {

birth: 1990,

getAge: function () {

var b = this.birth; // 1990

var fn = function () {

return new Date().getFullYear() - this.birth; // this指向window或undefined };

return fn();

}

};

现在，箭头函数完全修复了this的指向，this总是指向词法作用域，也就是外层调用者obj：

var obj = {

birth: 1990,

getAge: function () {

var b = this.birth; // 1990

var fn = () => new Date().getFullYear() - this.birth; // this指向obj对象

return fn();

}

};

obj.getAge(); // 25

如果使用箭头函数，以前的那种hack写法：

var that = this;

就不再需要了。

由于this在箭头函数中已经按照词法作用域绑定了，所以，用call()或者apply()调用箭头函数时，无法对this进行绑定，即传入的第一个参数被忽略：

var obj = {

birth: 1990,

getAge: function (year) {

var b = this.birth; // 1990

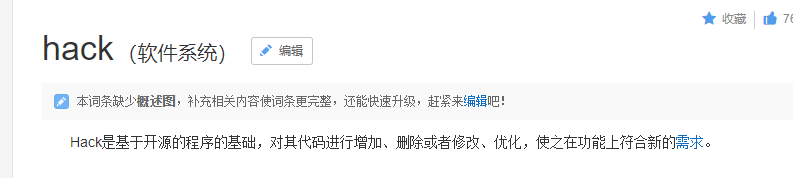
var fn = (y) => y - this.birth; // this.birth仍是1990

return fn.call({birth:2000}, year);

}

};

obj.getAge(2015); // 25



AndroidStudio的assets目录：

在src/main下面，如果没有就右键新建文件夹main，再新建文件夹assets在main文件夹下。

接入sdk一定要先看xml文件，先看要接入平台demo的xml文件，文档里的权限有可能写的不全面。

Vivo的sdk在targetSdkVersion = 28 的时候没有广告展示，不知道为什么，改成26就可以正常展示了。

图片Opacity和点击一起用会有问题。

Picasso，可译为“毕加索”，是Android中一个图片加载开源库。

使用的时候要导入依赖项，在build.grade下的dependencies添加

implementation **'com.squareup.picasso:picasso:2.5.2'**

**头文件导入import** com.squareup.picasso.Picasso;

（头文件导入的时候，如果有红色的东西出来，快速导入没有用，就要考虑是不是要引入依赖库了）。

Java的构造函数和C++一样，类名+括号

**implements**要继承所有的方法，不然会报错

AndroidStudio报错

Error: Program type already present: okio.AsyncTimeout$Watchdog

1. 手动删除libs下依赖的okio.jar包。
2. 在build.grade的defaultConfig加上

configurations {  
 all\*.exclude **group**: **'com.google.code.gson'** all\*.exclude **group**: **'com.squareup.okio'**}

原因：1、通过在AS双击shift键盘，然后在输入框里面输入AsyncTimeout，发现是因为有两个okio.jar包

2、可以现在project删除bmob的okio

3、要想彻底解决，（在app下的build文件下的android里面添加如下语句）

configurations {  
 all\*.exclude **group**: **'com.google.code.gson'** all\*.exclude **group**: **'com.squareup.okio'**}