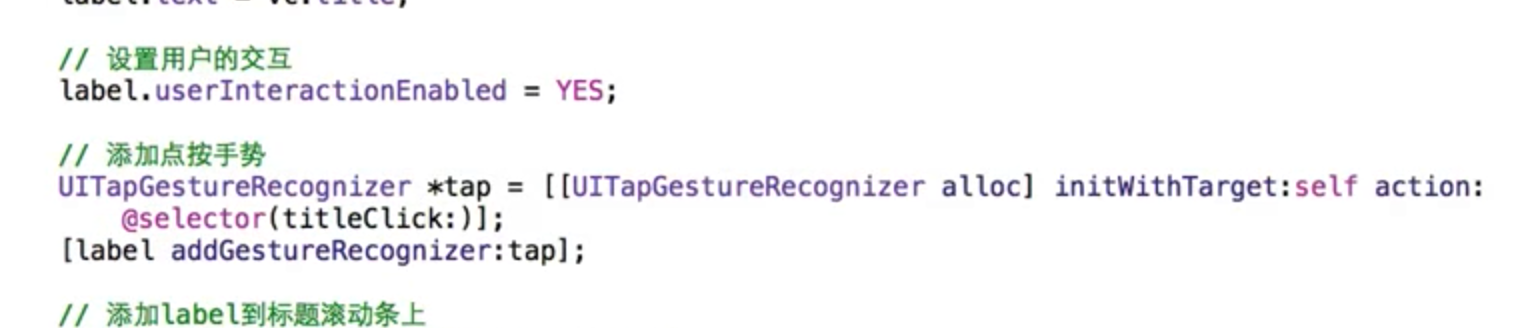
Ios学习笔记：

1:给uilable添加点击事件



2:app的可视范围由5启动页决定，（默认3.5寸）所以不设置启动页上下有黑边

3:edit seheml里面可以设置运行，编译时使用调试模式还是发布模式，可以试试设置发布模式时推送能不能生效

4:每个vc都可以实现方法去控制状态栏。如果想要用uiapplication控制，需要现在info里面声明。声明后，uiapplication控制的优先级更高，在视频1的05里面

5:程序的window可以不止一个，创建一个uiwindow，并设置hidden=NO，就会显示在屏幕上。Application.windows可以获取所有窗口，状态栏和输入法都是窗口,窗口还有windowLevl字段表示层级，层级别越高，显示在越前。层级相同，按时间，后加入的在前.但是，2020/9/15亲自测试，加多个window会报错，只能有一个

注意：新版xcode创建的项目，修改window和mian interface无效，需要删掉info里面的字段，并且删掉appdelegate里面两个方法才可以像以前一样：

**Application scene manifest**

**configurationForConnectingSceneSession**

**didDiscardSceneSessions**

6:storyboard和xib区别，s用来定义viewController，包含view和控制器。Xib只定义view。所以用uistoryboard可以获取一个完整vc，尔xib只能获取view。（并不对，两个都可以，需要查查两个的区别）

Vc的loadView方法加载view，可以重写，在里面self。View=【uiview new】可以替换掉默认加载的view

一个vc如果用initWithNibName创建，会创建指定的名字xib，如果传nil，会自动检测有没有和vc同名的xib，有就创建，没有就创建空白view。Init方法也会做这个检测

loadView调用完成后，回掉viewDidLoad方法。很熟悉的地方

7:alpha属性，如果<=0.01,表示完全透明，可以穿透，即可以穿过它触碰到它下面的控件 ,0.99完全不透明。但等于1的时候，会有点透明

8:uiview如果是从xib或storboard创建，创建完成有awakeFormNib回掉。

9:uitextfile的inputVIew属性，设置后输入时会弹出自定义的键盘。

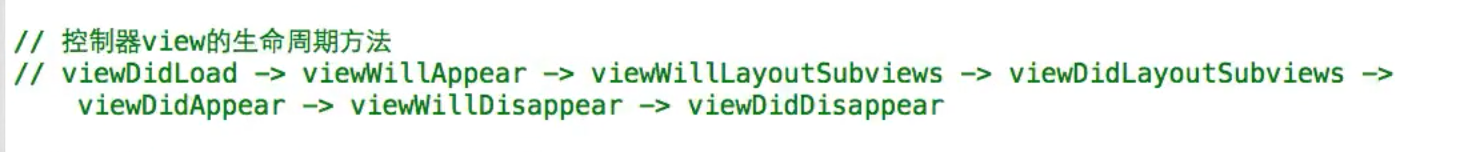
10:导航控制器：【【uinavigationcontroller alloc】 initviewrootviewcontroller：vc】创建，并添加为uiwindow的rootViewController。接下来使用就和开发中一样了。他有.viewcontrollers,和childViewControllers两个属性可以获取子控制器列表。

导航控制器有两个view，顶部view（状态栏和titleview），存放子控件view的view。当一个vc加入顶部后，之前的vc的view会从导航控制器的viw里移除，即里面永远只有一个vc的view。有两个方法移除控制器，移除最顶部的和回到指定控制器。疑问：第三个控制器上指定回到第一个的话，viewControllers里面第2个还在不？

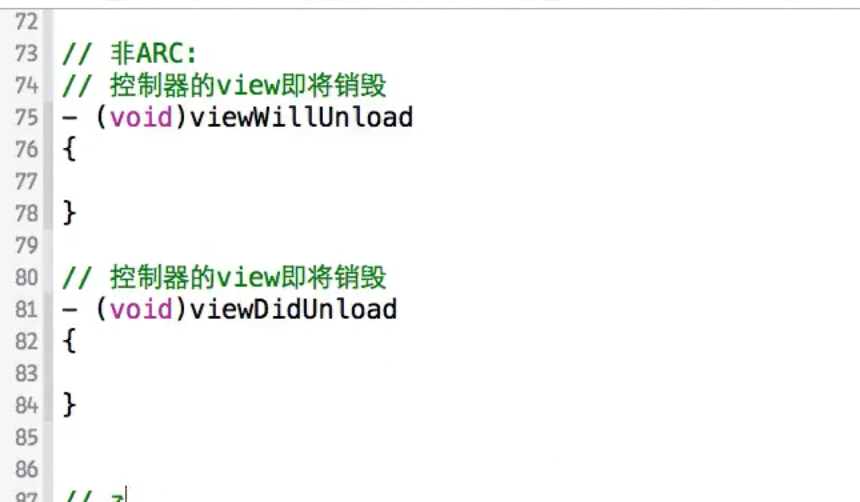
导航条的内容是有栈顶控制器的navigationItem控制，而不是导航控制器的，所以每次跳到新vc，顶部都会变化

顶部按钮的图片系统会自动渲染成蓝色，如果不需要自动渲染，需要用【image imageWithRenderingModel:UiImageRenderingModelOrigial】设置使用原始图片，或者直接在图片文件夹里设置图片的属性。开发中底部栏有时变蓝色的bug原因找到了

11:/生命周期，arc下



非arc下还有这两个方法，在内存不足时由系统回掉。里面需要进行资源释放



12:view.contentInset为额外滚动区域，uitableviewController里的uitableview默认设置了（64，0，0，0），所以向下滚了64.

Ios7以后苹果自动为所有控制器下的scrollview自动添加了64.x可能更多

用self.automaticallyAdjustsScrollViewInsets = NO;设置不需要这个区域

13:归档：*NSKeyedArchive*r，可以用来存储自定义对象，对象需要实现nscoding协议，的两个方法，就可以存和读。使用方法类似nsdic差不多

14:形变：view.transform=CGAffinetransformMakeTranslation(x,y)位移形变，xy移动的值

view.transform=CGAffinetransformMakeRotation(角度)旋转形变

view.transform=CGAffinetransformMakeScale(sx，sy)缩放形变，宽高缩放比例。

上面三种都是以原始位置形变，想要在形变基础上形变，需要调用另一个方法如：CGAffinetransformTranslate（view.transform，x，y）传入上一次形变。其他类型以此类推

15:uitablebarcontroller的使用：创建uitablebarcontroller。然后addchildvc。底部会自动分成多分。给每个vc的tabbaritem的tiitle,image,badgevalue设置值就。

Uitablebarvc分三个view，本身的view，tabbar的view，子控制器的view空间。

Tabbar里面每个对象是uitablebarButton。

16:打开新vc的另一个方式：modal，不依赖以导航控制器，可以随意弹出：

self presentViewController：xxx 弹出

self dismissViewController 收起最后modal出来的vc

modal出来的vc会被前一个vc .presentViewcontroller强引用，不会销毁

17:触摸事件：对象继承自UiResponder就能处理事件。只要实现touchBegin，touchMove，touchEnd，touchCancel四个方法，就能接收到事件进行处理。

UiApplication，UiWindow，UiViewController，UiView都是UiResponder的子类。

如果不实现方法或方法里调用super实现，就会把事件传递给上一个相应者处理。顺序subView🡪 UiView🡪 VC🡪🡪 UiWindow🡪 UiApplication

发生了事件后，系统传递顺序：UiApplication—>UiWindow🡪VC🡪UiView🡪subView。

自己做了个实验：view只实现end方法，vc实现全部，事件会被vc完全捕获。

并且view也能捕获end方法；view实现move和end方法，依然是vc完全捕获，view只捕获end，但view实现了begin方法，vc就捕获不到事件了。结论是：如果实现了begin，那该事件就独占了。如果没有实现begin，end会捕获，其他捕获不到，并且事件被上一个相应者处理

如果对象不能处理事件，也就不能再传给下级。即下级的事件也不会触发

不能处理事件的情况（Uiimageview默认userInterctionEnable=NO）：

view.userInterctionEnable=NO;//不接受用户交互

view.hidden=YES;//隐藏了

view.alpha=0.0-0.01；//透明

hitTest方法，每个对象接收到事件以后，会调用这个方法去判断事件该交给子控件处理还是自己处理。Return自己或子view逻辑：

1判断自己能否接受事件，并且点击位置是否在自己身上(hite里调用self pointInside::方法判断点是否在自己身上)

2从后到前判断子view是否能处理，能的话交给子view，子view也是以1为判断

3如果自己能处理但找不到子view接盘，就自己处理。

18:手势：因为上面的触摸事件提供的东西过于简单，难以实现复杂手势判断，所以需要UiXXXGestureRecognizer处理各种手势，点按，长按，拖动，捏合等等。代码：

UiXXXGestureRecognizer \*xxxGesture=【【UiXXXGestureRecognizer alloc】initWithTarget：self action：@selector（xxx）】

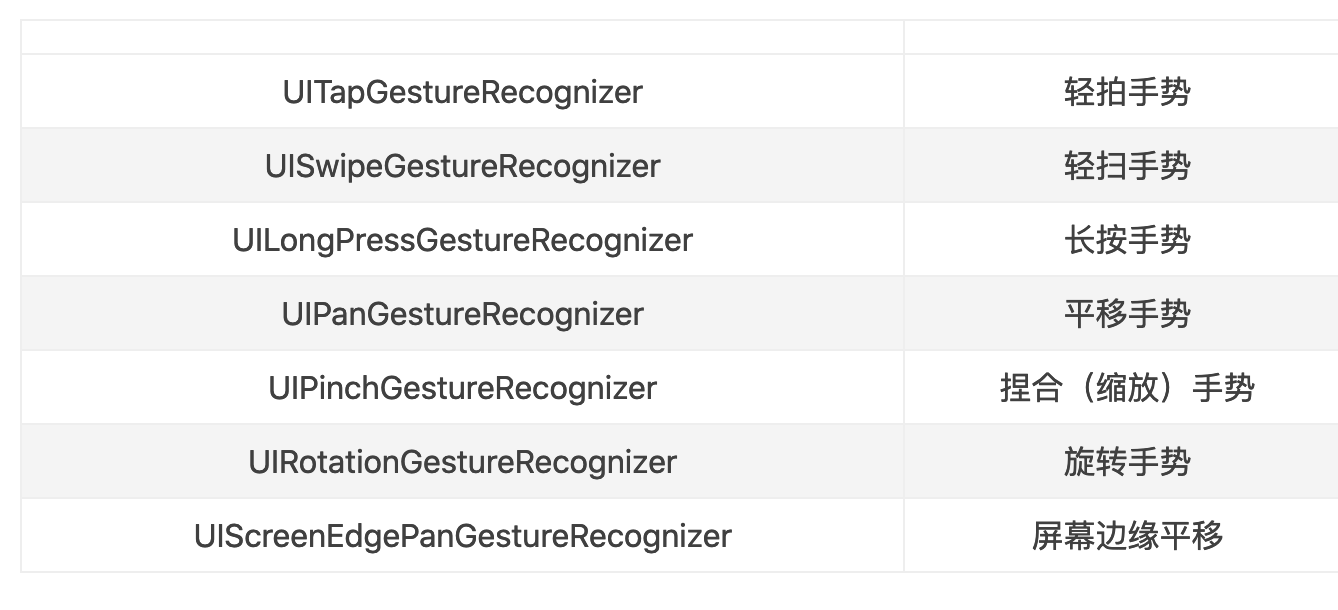
[View ddGuestureRecognizer: xxxGesture]

可 add多个Gesture；

触发了手势就会调用xxx方法

gesture.delegate，还可以实现手势的代理，去更精细的处理手势。比如是否响应这个手势，是否通知支持两种手势（默认支持一种手势）

每个手势都有对应的属性可以设置，比如轻拍设置需要几个手指等等





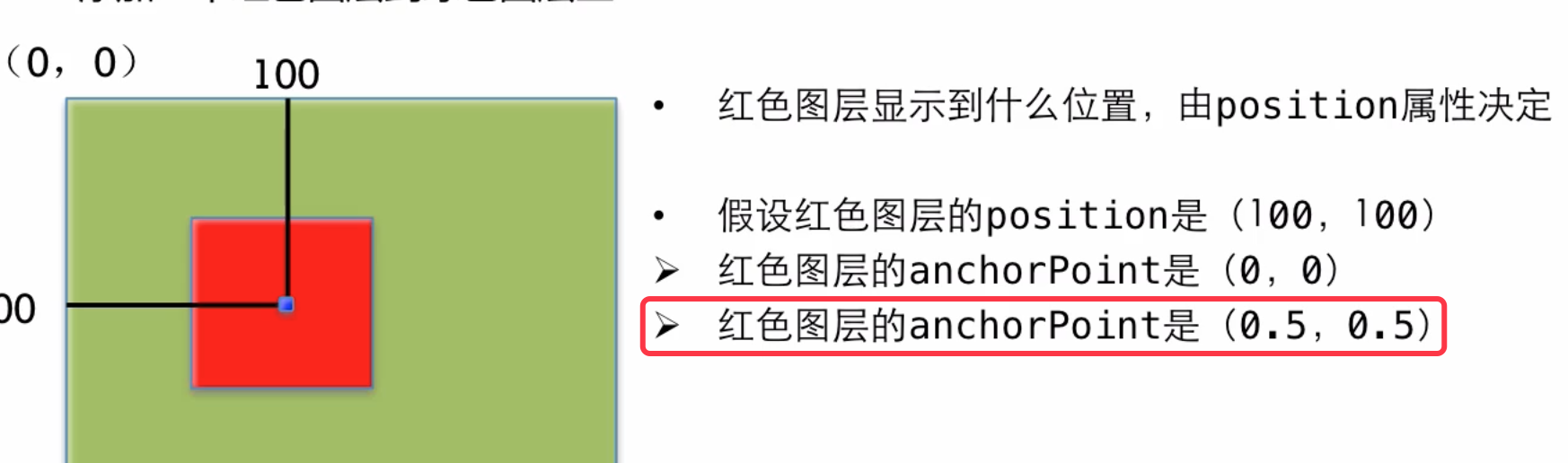
19:layer：是uiview上面的一个图层，决定uiview显示的内容。Uiview能显示，正是因为它的存在。Uiview子控件的layer，也是放在uiview的layer上面的。

可以设置圆角，边框，阴影等，平时用的少的还有transform，只是它对应需要cgaffinxxx3D的形变对象，用法和2d的类似。

也可以new一个calayer，放到view。Layer上面去，newLayer可以设置bounds，posioton等，类似view，设置content（例如cgimage）。做任何节目展示。但是无法接受触摸事件。这也是他和uiview的区别。

他上面的color，image等都需要使用cgcolor，cgimage等，因为cgxx是定义在CoreGraphics里面的，CALayer是定义在QuartzCore里面的。这两个mac，ios通用。而UIKit是ios专用，里面的uiimage等如果放到mac上面就会报错。所以CALayer里面的对象都用CGXXX。

Layer的position（x，y）和anchorPoint（锚点）的关系：锚点的xy取值范围0-1，默认0.5，即中间位置。设置position实际上是把锚点放在position上。例如锚点为x和y=0.5，position：x=100，y=100.世纪是把layer的中间点放在父layer的100，100上



自定义calayer的方法，一个是继承calayer，重写drawinrect方法。一个是不继承，设置delegate，delegate里面实现-drawlayer：incontent方法。

20:创建定时器，加到runloop里面去。这个定时器每当屏幕刷新就会调用，屏幕每秒刷新60次。可以用来做动画之类的

CADisplayLink \*link=[CADisplayLink displayLinkWithTarget:self selector:@selector(xxx)]

[link addToRunloop:[NSRunloop mainRunloop] forMode:NSDefaultRunloopMode];

21:绘画：

1:基本图形的绘制，重绘

2:uikey的快捷绘制带裁剪：文字绘制，图片绘制，图片绘制的平铺方法

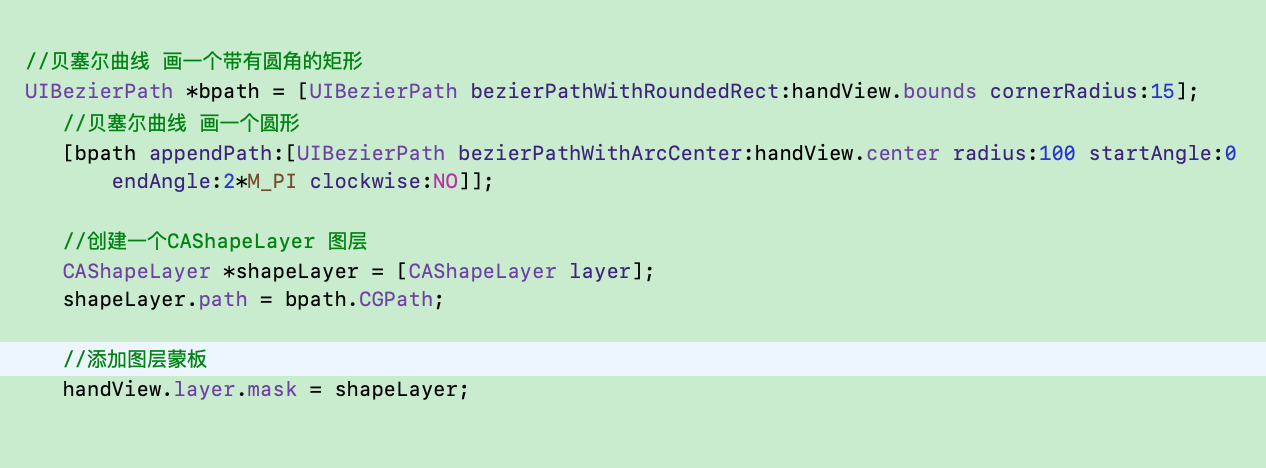
3:上下文状态栈，上下文矩阵。

4:开启关闭Image的上下文，生成图片

5: Path设置为裁剪区域：【path addCilp】

6:判断一个点是否在rect里面的方法：CGRectContainsPoint(rect,point)

7:镂空方法：核心是appendpath后，如果两个path有重叠的部分，这部分不会绘制，寄呈现透明效果



22:隐式动画：非根layer的大小，位置，颜色改变，都默认有动画。需要关闭的话，使用事务的操作关闭，也可以修改事物的事件让动画时长修改。

23:核心动画：

1:uiview的动画会修改frame，但不修改center，核心动画作用于layer层，动画不修改frame，也不修改center 。

2:动画类型：

CABasicAnimation：keyPath设置动画对象，position，bounds等，fromValue设置其实值，toValue设置结束值。

CAKeyframeAnimation：关键帧动画，和上面的区别是values值，可以设置多个值做一系列动画，而上面的只有一个开始结束值。还有path值，可以传路一个路径，直接绕path做动画keyTimes设置哥哥value对应的时间。

CATransition：转场动画

CAAnimationGroup：动画组，可以把多个CABasicAnimation，CAKeyframeAnimation加到进动画组，一起执行

24:一张图片里面裁剪一部分的方法UiImageCreateWithImageInRect（image，rect），注意的是，rect里面的宽高需要乘屏幕比例才是真实宽高。即@2x，@3x，可以用【UIScreen mainScreen】。Scale获取；计算已经旋转角度的办法atan2（x，x）方法

25:layer显示部分方法，渐变方法，uiview动画弹簧效果，旋转中立体效果m34，复制层的使用，自定义view重写layerClass方法就可以修改view的layer类型。形状图层CAShapeLayer（各一个path，生成图形layer）

26:计算两个点之间的距离方法

27:导航条：设置全局导航条的样式方法和单个vc导航条样式的方法：0921里

28:类方法：load第一次加载这个类的时候调用，子类再用也不再触发。Initialize第一次调用类或子类时调用（可能不止一次，子类掉时也会触发）

29:滑动返回功能系统自带有，但如果重写了返回键，就失效了，恢复方法见视频0922

30:+allocWithZone方法为类alloc时调用的方法，一半设置单例的时候重写

31:条件编译：符合条件的代码才会编译，用于arc和mrc兼容。测试代码之类

#if xxx

代码

#else

代码

#endif

32:MPMoviePlayerViewController可以直接用系统播放器播放视频，传入url即可，同样还有系统相册选择的vc，忘了名字。

33:NSInvocation可以用来访问一个对象的方法并且传多个参数，你不performSelector方法只能传一个参数的不足



34:异常捕获：NSSetUncaughtExceptionHandler(&saveException);

AppDelegate入口写这句话，再定义一个C语言的方法，就能捕获异常了，异常存起来。Exception.callStackSymbols的内容是一个arry，就是平时异常窗口输出那堆信息

35:地图相关有方法监听区域，进入区域，离开区域。。看视频地图相关吧

36:KVC是Key Value coding，实际上是所有类都实现了NSKeyValueCoding协议，提供valueForKey，setValue：forKey等方法，还可以用keyPath做二级熟悉的操作。

比如下面修改对象下address对象的street属性：

[myAccount setValue:@"中关村大街" forKeyPath:@"address.street"];

下面把数组里面对象的name属性取出，返回到一个数组

NSArray \*names = [array valueForKeyPath:@"name"];

37:KVO是key-value observing，用来监听对象值改变的。

addObserver，removeObserver来添加和取消监听，实现observeValueForKeyPath

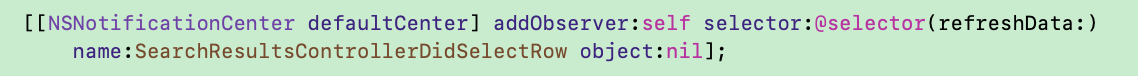
方法来处理改变事件

可以用下面的代码重写通知机制，首先禁用系统的通知，然后set方法调两个方法：

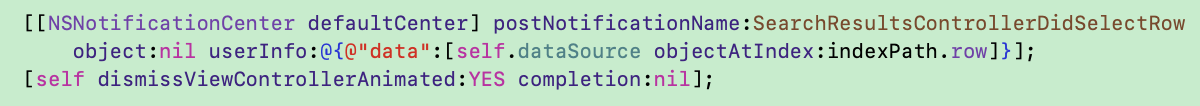
KVO底层实现原理其实就是用了runtime，被监听对象生产一个新的子类，重写了set方法，并且把isa指针指向了新动态生成的类

38:NSNotificationCenter使用，也是用来发通知和接受通知的。

1:在需要接受的地方调用下面代码定义接受方法和key



2:需要发送通知的地方用下面代码，key需要和上面一样



3:接收者需要关闭时用下面方法



39:项目中加密手段：首先参数组装成json，用XXTEA加密（使用约定的密码1），形成A，再取出当前时间戳，时间戳+A+约定密码2组成待签名字段B，把B进行md5加密形成签名字段C。把时间戳，A，C都传给服务器。

服务器首收到请求后先判断时间戳是否过期，不过期的话把时间戳+A+约定密码2进行md5加密，如果和C一致就通过，然后使用约定密码1解密A，获得参数，再做业务操作。

40: UIImagePickerController可以用来调用系统自带的选取图片，拍照，录像等功能，详细学习：<https://blog.csdn.net/lc_1581835288/article/details/50471482>