> 摘要： 本文讲的是iOS下PDF文件的浏览和涂鸦效果的简单实现 iOS， 浏览 PDF 的效果方法一：利用 webview 复制代码 代码如下: - (void)loadDocument:(NSString *)documentName inView:(UIWebView *)webView

浏览PDF的效果

方法一：利用webview

复制代码 代码如下:

```
- (void)loadDocument:(NSString *)documentName inView:(UIWebView *)webView {
  NSString *path = [[NSBundle mainBundle] pathForResource:documentName ofType:nil]
;
  NSURL *url = [NSURL fileURLWithPath:path];
  NSURLRequest *request = [NSURLRequest requestWithURL:url];
  [webView loadRequest:request];
}
```

利:

- 1. 实现简单
- 2. 还是实现简单

弊:

- 1.仅能浏览，拿不到任何回调，safari不会鸟任何人。
- 2.固定竖版拖动，想实现翻页动效果就扒瞎

方法二：利用CGContextDrawPDFPage

下面的方法可以解决webview 显示pdf的弊，相对的，要付出一些汗水作为代价了。

复制代码 代码如下:

```
CGPDFDocumentRef GetPDFDocumentRef(NSString *filename) {
    CFStringRef path;
    CFURLRef url;
    CGPDFDocumentRef document;
    size_t count;

    path = CFStringCreateWithCString (NULL, [filename UTF8String], kCFStringEncodi
ngUTF8);
    url = CFURLCreateWithFileSystemPath (NULL, path, kCFURLPOSIXPathStyle, 0);

    CFRelease (path);
    document = CGPDFDocumentCreateWithURL (url);
    CFRelease(url);
    count = CGPDFDocumentGetNumberOfPages (document);
    if (count == 0) {
        printf("[%s] needs at least one page!\n", [filename UTF8String]);
        return NULL;
    } else {
        printf("[%ld] pages loaded in this PDF!\n", count);
    }
    return document;
}

void DisplayPDFPage (CGContextRef myContext, size_t pageNumber, NSString *filename
) {
    CGPDFDocumentRef document;
    CGPDFPageRef page;

    document = GetPDFDocumentRef (filename);
    page = CGPDFDocumentGetPage (document, pageNumber);
    CGContextDrawPDFPage (myContext, page);
    CGPDFDocumentRelease (document);
}
```

这样显示出来的pdf单页是倒立的，Quartz坐标系和UIView坐标系不一样所致，调整坐标系，使pdf正立：

复制代码 代码如下:

```
CGContextRef context = UIGraphicsGetCurrentContext();
CGContextTranslateCTM(context, 80, self.frame.size.height-60);
CGContextScaleCTM(context, 1, -1);
```

配合iOS5强大的UIPageViewController实现翻页浏览

复制代码 代码如下:

```
- (PDFViewController *)viewControllerAtIndex:(NSUInteger)index {
```

```
    //Return the PDFViewController for the given index.
    if (([self.pagePDF count] == 0 )|| (index > [self.pagePDF count]) ) {
        return nil;
    }

    //Create a new view controller and pass suitable data.
    PDFViewController *dataViewController = [[PDFViewController alloc]initWithNibN
ame:@"PDFViewController" bundle:nil];
    //dataViewController.pdfview = [self.pagePDF objectAtIndex:index];
    dataViewController.pdfview = [[PDFView alloc]initWithFrame:self.view.frame atP
age:index];
    [dataViewController.view addSubview:dataViewController.pdfview];
    NSLog(@"index = %d",index);
    return dataViewController;
}

- (NSUInteger) indexOfViewController:(PDFViewController *)viewController {
    return [self.pagePDF indexOfObject:viewController.pdfview];
}

- (UIViewController *)pageViewController:(UIPageViewController *)pageViewControlle
r viewControllerBeforeViewController:(UIViewController *)viewController {
    NSUInteger index = [self indexOfViewController:(PDFViewController *)viewContro
ller];
    if ((index == 0 ) || (index == NSNotFound)){
        return nil;
    }

    index--;
    return  [self viewControllerAtIndex:index];
}

- (UIViewController *)pageViewController:(UIPageViewController *)pageViewControlle
r viewControllerAfterViewController:(UIViewController *)viewController {
    NSUInteger index = [self indexOfViewController:(PDFViewController *)viewContro
ller];
    if (index == NSNotFound)
    {
        return nil;
    }

    index++;

    if (index == [self.pagePDF count]){
        return  nil;
    }

    return [self viewControllerAtIndex:index];
```

```
}
```

涂鸦效果

主要涉及：

1. 多context，分层画画

复制代码 代码如下:

```
- (void)drawLayer:(CALayer *)layer inContext:(CGContextRef)ctx
```

2. 触摸事件touches族那些event

复制代码 代码如下:

```
- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
- (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event
...
```

3. 初始化单页view传页码

复制代码 代码如下:

```
- (id)initWithFrame:(CGRect)frame onPage:(NSInteger)page
```

4.画轨迹方法

复制代码 代码如下:

```
CG_EXTERN void CGPathMoveToPoint(CGMutablePathRef path,
  const CGAffineTransform *m, CGFloat x, CGFloat y)
CG_EXTERN void CGPathAddLineToPoint(CGMutablePathRef path,
  const CGAffineTransform *m, CGFloat x, CGFloat y)
```