绘制纯背景的 UISearchBar

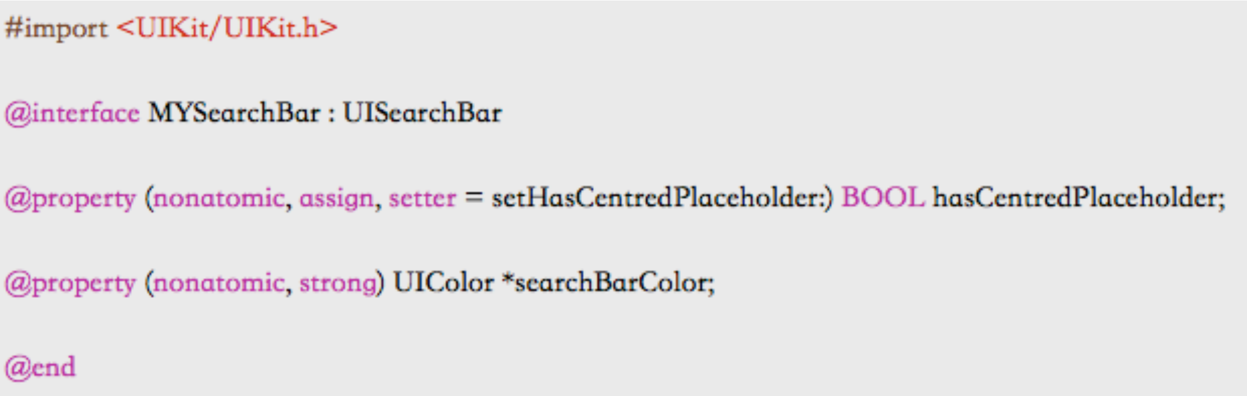
1. 简述：

UI 设计需求，纯白搜索框，提示文字默认居左显示。

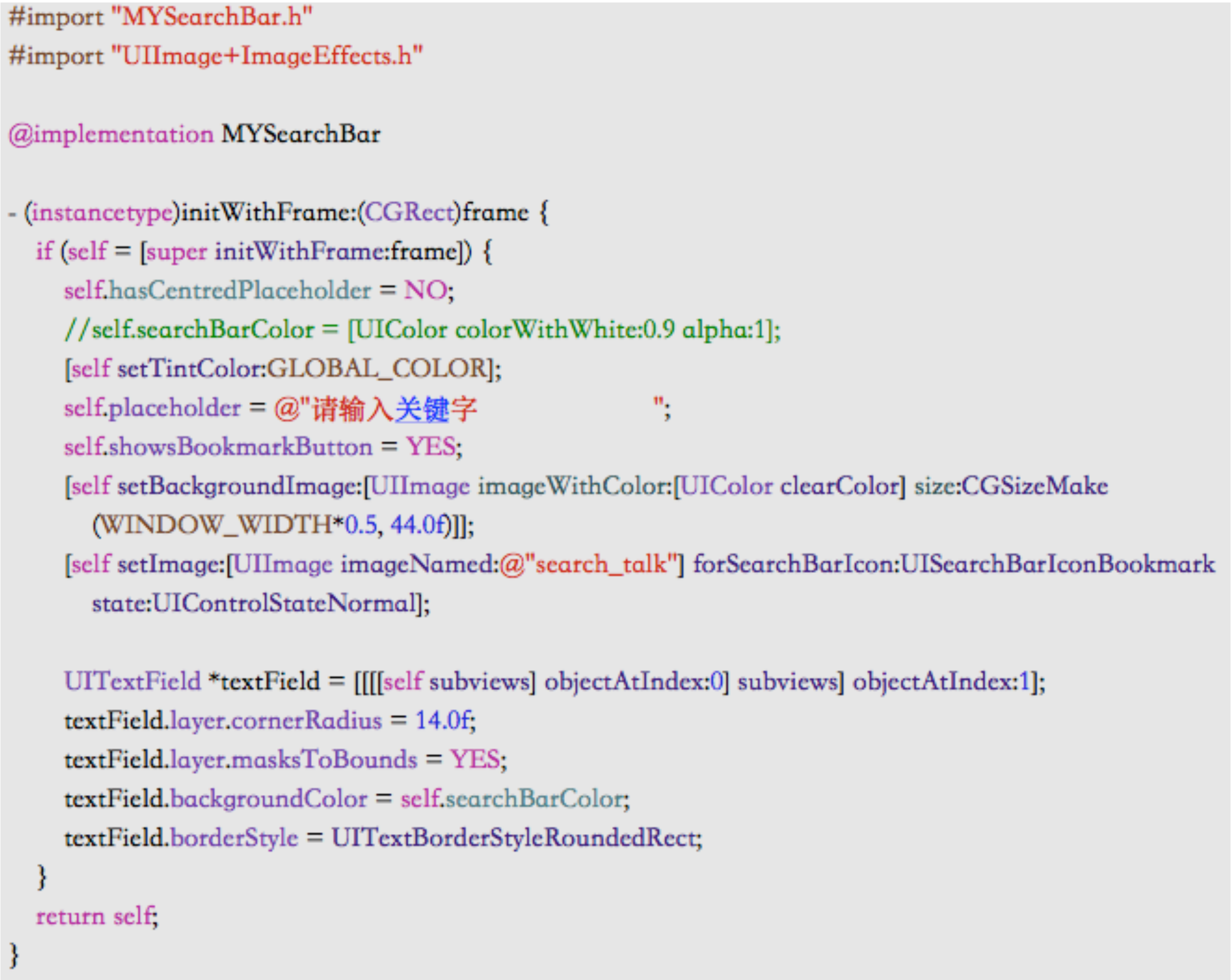
（1）search style 默认灰背景，minimal 灰文本框，无法纯色背景。

（2）placeholder 默认居中，点击后 placeholder 居左，并出现闪烁光标。

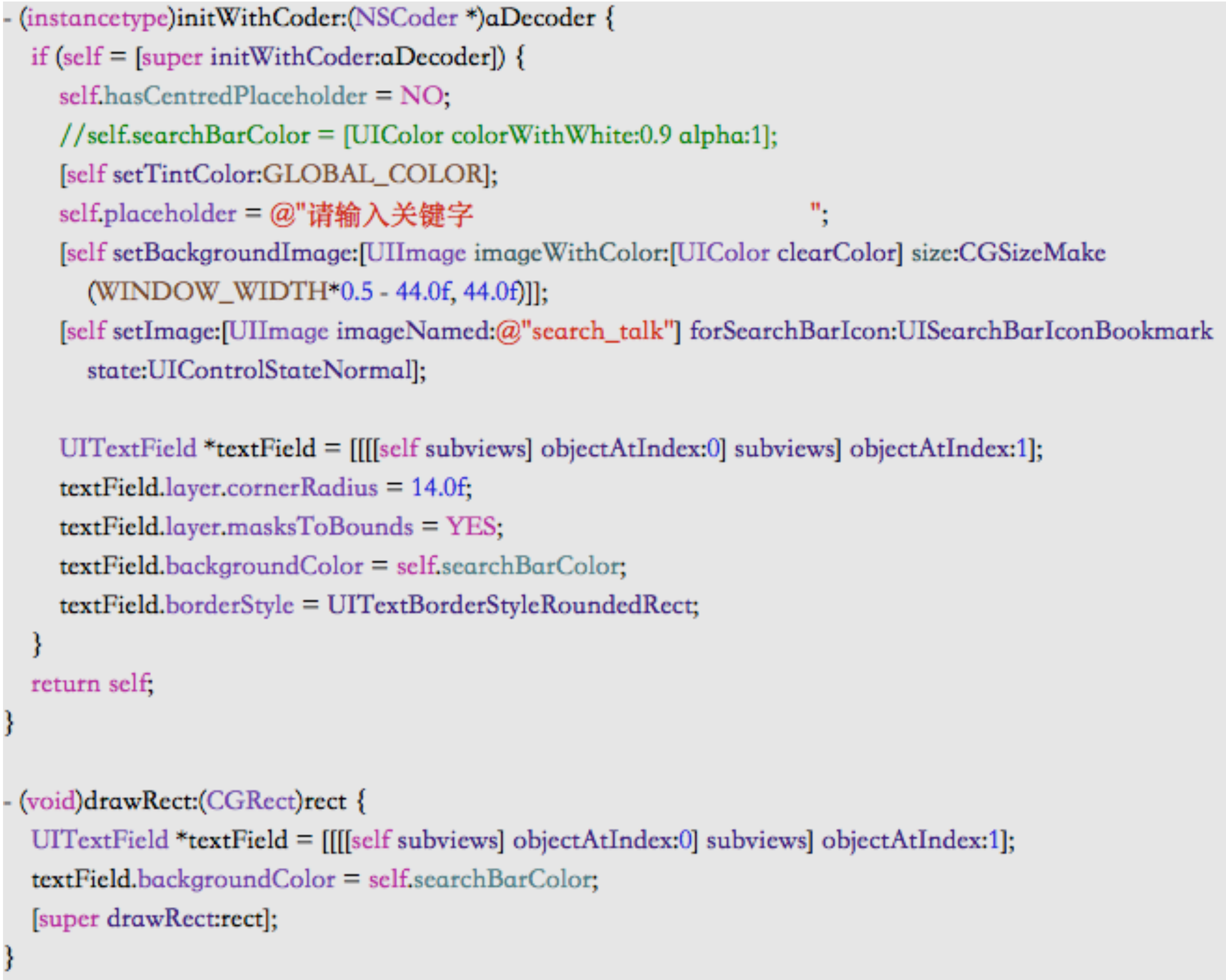
2. 实现：



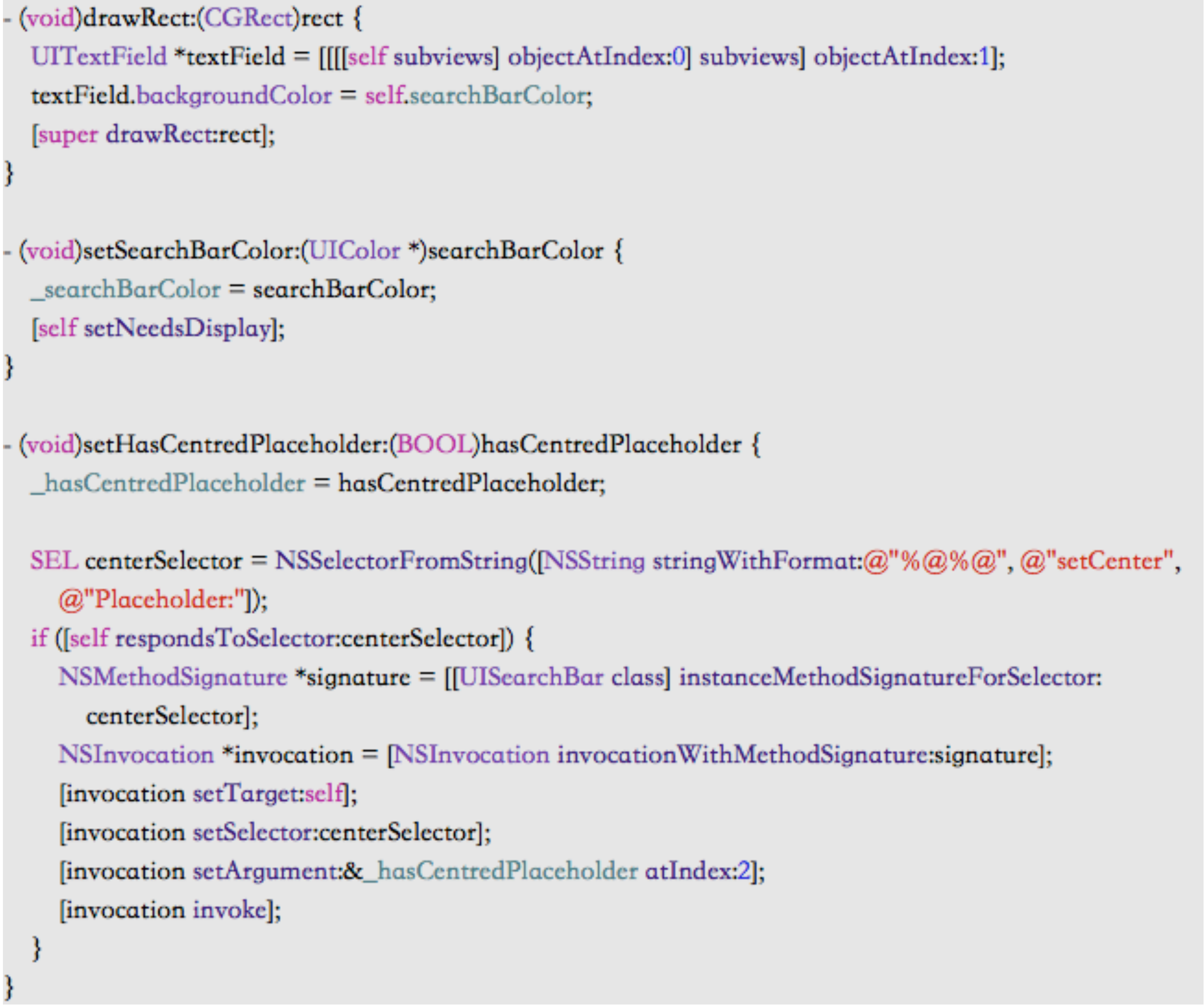
```
#import <UIKit/UIKit.h>

@interface MYSearchBar : UISearchBar

@property (nonatomic, assign, setter = setHasCentredPlaceholder:) BOOL hasCentredPlaceholder;

@property (nonatomic, strong) UIColor *searchBarColor;

@end
```



```
#import "MYSearchBar.h"
#import "UIImage+ImageEffects.h"

@implementation MYSearchBar

- (instancetype)initWithFrame:(CGRect)frame {
    if (self = [super initWithFrame:frame]) {
        self.hasCentredPlaceholder = NO;
        //self.searchBarColor = [UIColor colorWithWhite:0.9 alpha:1];
        [self setTintColor:GLOBAL_COLOR];
        self.placeholder = @"请输入关键字                    ";
        self.showsBookmarkButton = YES;
        [self setBackgroundImage:[UIImage imageWithColor:[UIColor clearColor] size:CGSizeMake
            (WINDOW_WIDTH*0.5, 44.0f)]];
        [self setImage:[UIImage imageNamed:@"search_talk"] forSearchBarIcon:UISearchBarIconBookmark
            state:UIControlStateNormal];

        UITextField *textField = [[[[self subviews] objectAtIndex:0] subviews] objectAtIndex:1];
        textField.layer.cornerRadius = 14.0f;
        textField.layer.masksToBounds = YES;
        textField.backgroundColor = self.searchBarColor;
        textField.borderStyle = UITextBorderStyleRoundedRect;
    }
    return self;
}
```



```
- (instancetype)initWithCoder:(NSCoder *)aDecoder {
  if (self = [super initWithCoder:aDecoder]) {
    self.hasCentredPlaceholder = NO;
    //self.searchBarColor = [UIColor colorWithWhite:0.9 alpha:1];
    [self setTintColor:GLOBAL_COLOR];
    self.placeholder = @"请输入关键字                              ";
    [self setBackgroundImage:[UIImage imageWithColor:[UIColor clearColor] size:CGSizeMake
      (WINDOW_WIDTH*0.5 - 44.0f, 44.0f)]];
    [self setImage:[UIImage imageNamed:@"search_talk"] forSearchBarIcon:UISearchBarIconBookmark
      state:UIControlStateNormal];

    UITextField *textField = [[[[self subviews] objectAtIndex:0] subviews] objectAtIndex:1];
    textField.layer.cornerRadius = 14.0f;
    textField.layer.masksToBounds = YES;
    textField.backgroundColor = self.searchBarColor;
    textField.borderStyle = UITextBorderStyleRoundedRect;
  }
  return self;
}

- (void)drawRect:(CGRect)rect {
  UITextField *textField = [[[[self subviews] objectAtIndex:0] subviews] objectAtIndex:1];
  textField.backgroundColor = self.searchBarColor;
  [super drawRect:rect];
}
```



```
- (void)drawRect:(CGRect)rect {
  UITextField *textField = [[[[self subviews] objectAtIndex:0] subviews] objectAtIndex:1];
  textField.backgroundColor = self.searchBarColor;
  [super drawRect:rect];
}

- (void)setSearchBarColor:(UIColor *)searchBarColor {
  _searchBarColor = searchBarColor;
  [self setNeedsDisplay];
}

- (void)setHasCentredPlaceholder:(BOOL)hasCentredPlaceholder {
  _hasCentredPlaceholder = hasCentredPlaceholder;

  SEL centerSelector = NSSelectorFromString([NSString stringWithFormat:@"%@%@", @"setCenter",
    @"Placeholder:"]);
  if ([self respondsToSelector:centerSelector]) {
    NSMethodSignature *signature = [[UISearchBar class] instanceMethodSignatureForSelector:
      centerSelector];
    NSInvocation *invocation = [NSInvocation invocationWithMethodSignature:signature];
    [invocation setTarget:self];
    [invocation setSelector:centerSelector];
    [invocation setArgument:&_hasCentredPlaceholder atIndex:2];
    [invocation invoke];
  }
}
```

3. 拓展：

ios 直接调用某个对象方法的方式：（1）performSelector:withObject （2）NSInvocation

NSInvocation:用来包装方法和对应的对象，可存储方法名称，对应的对象，对应的参数

NSMethodSignature:方法签名，创建时必须传递一个签名对象，用于获取参数个数和方法的返回值。

创建签名对象：谁的方法，就由谁来创建

```
NSMethodSignature *signature = [[UISearchBar class] instanceMethodSignatureForSelector:@selector()];
```

创建 NSInvocation 对象

```
NSInvocation *invocat= [NSInvocation invocationWithMethodSignature:signature];

invocat.target = self;//0

invocat.selector = @selector();//1

[invocat setArgument:&_hasCenterdPlaceHolder atIndex:2];

[invocat invocate];
```