

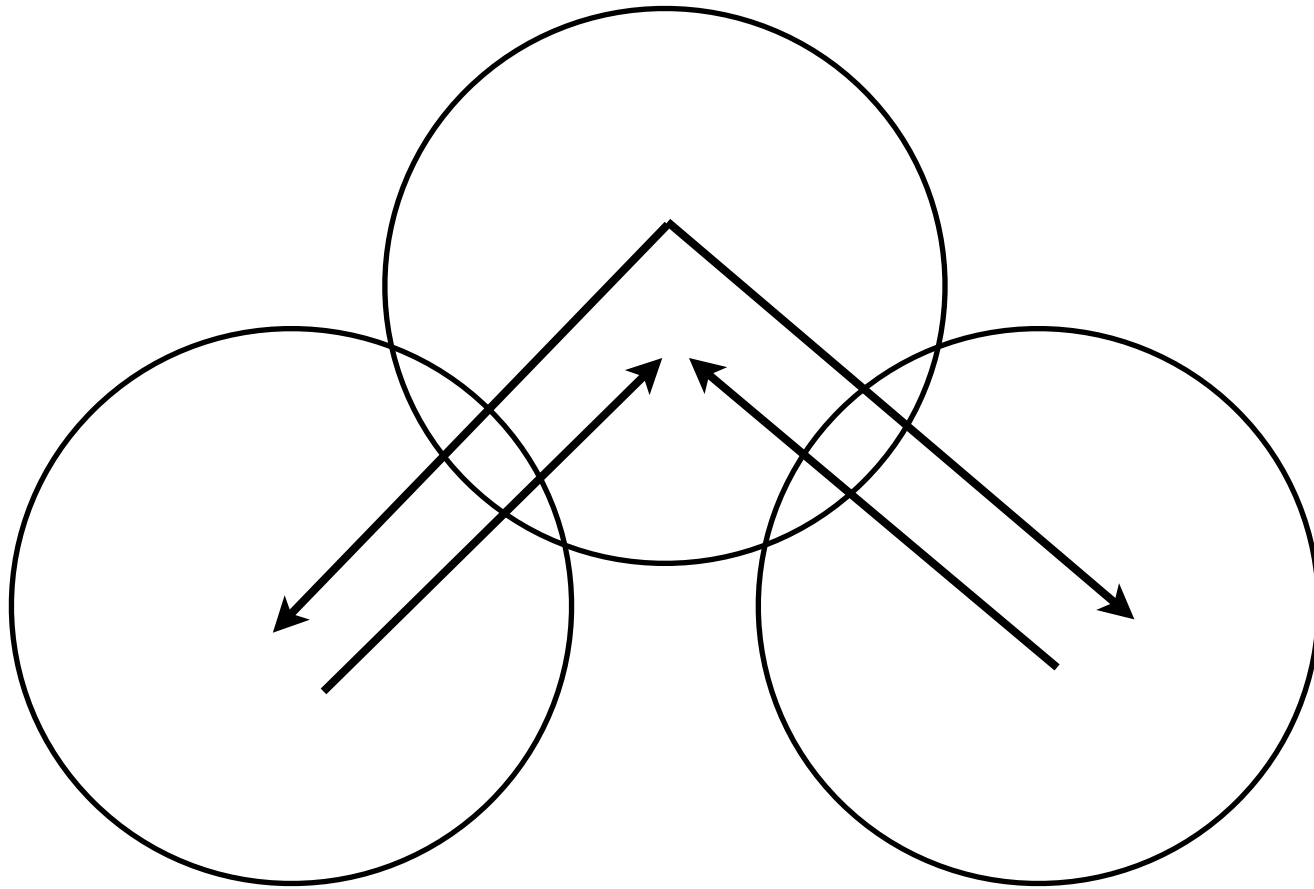
iPhone Game Development

model view controller, basic UIView

remember the homework

- mail zipfile with working project to hintze@msu.edu not later then Nov. 7th midnight!

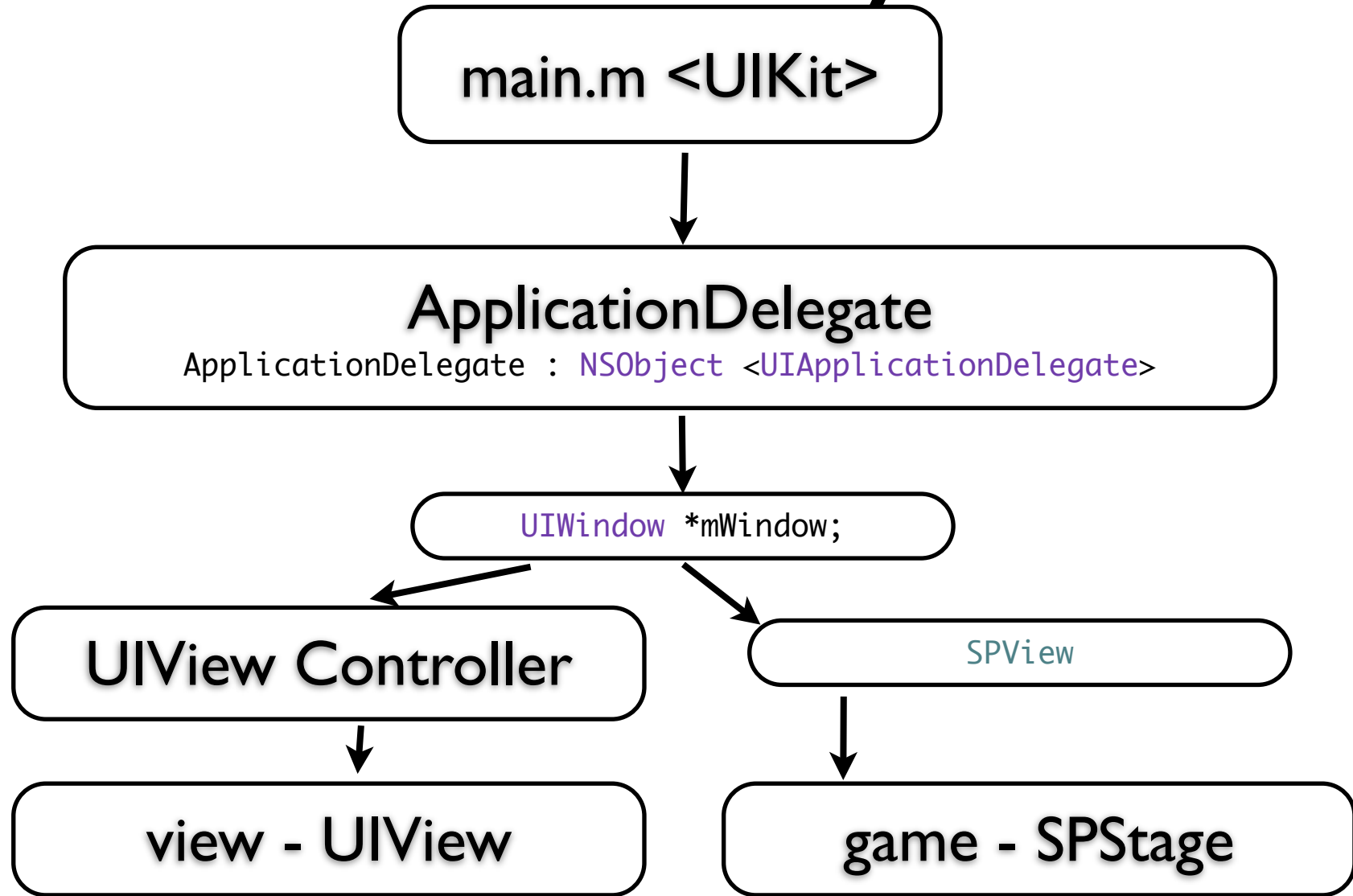
controller

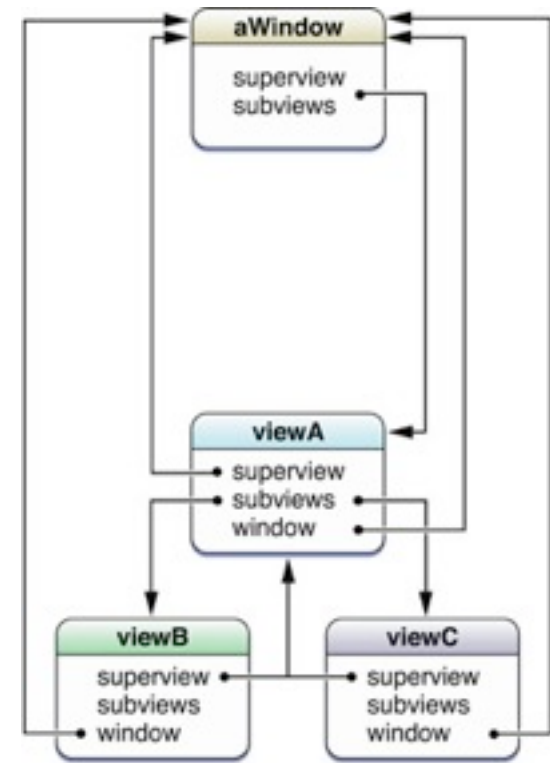
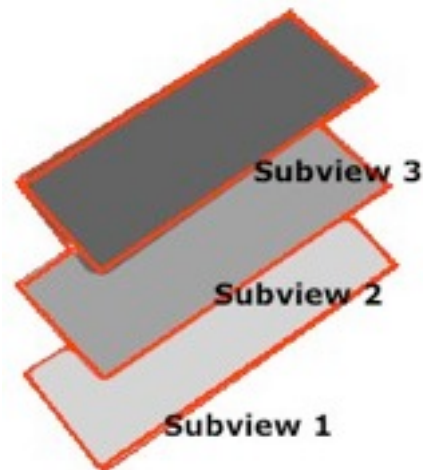
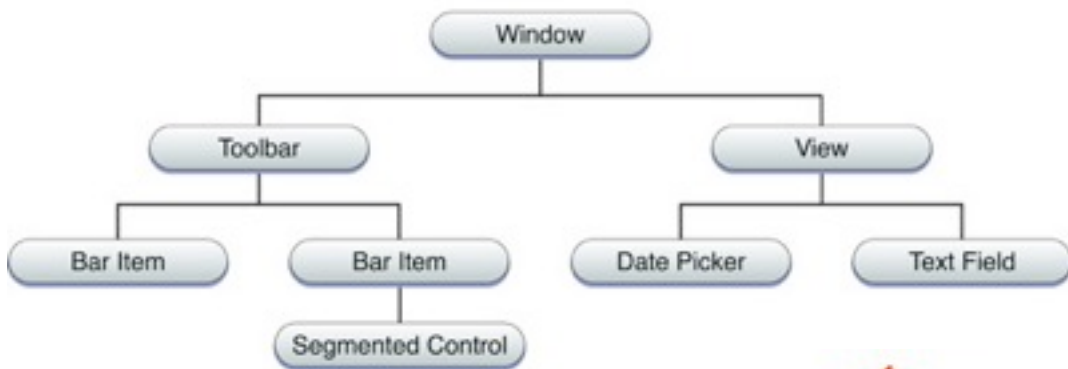


model

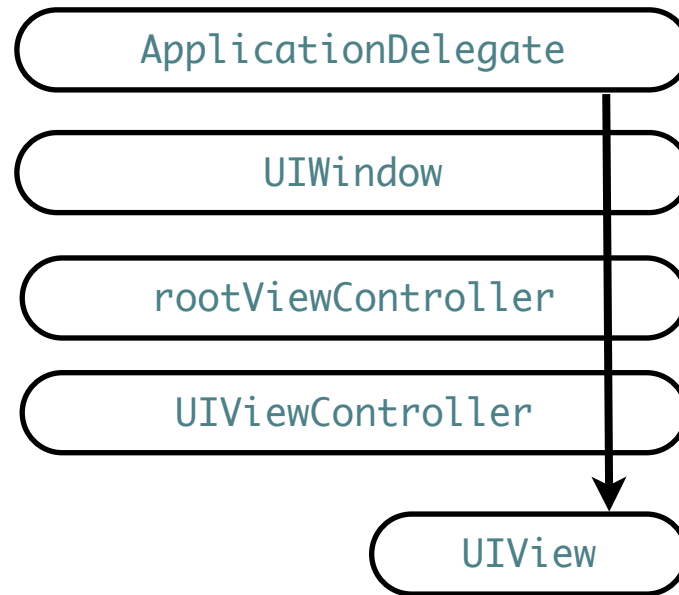
view

hierarchy





iPhone OS

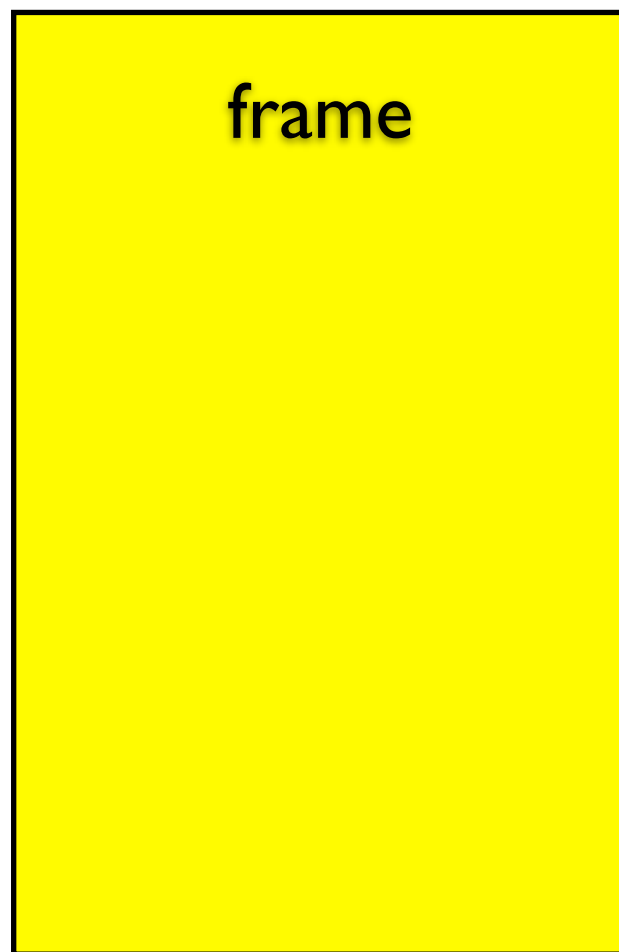


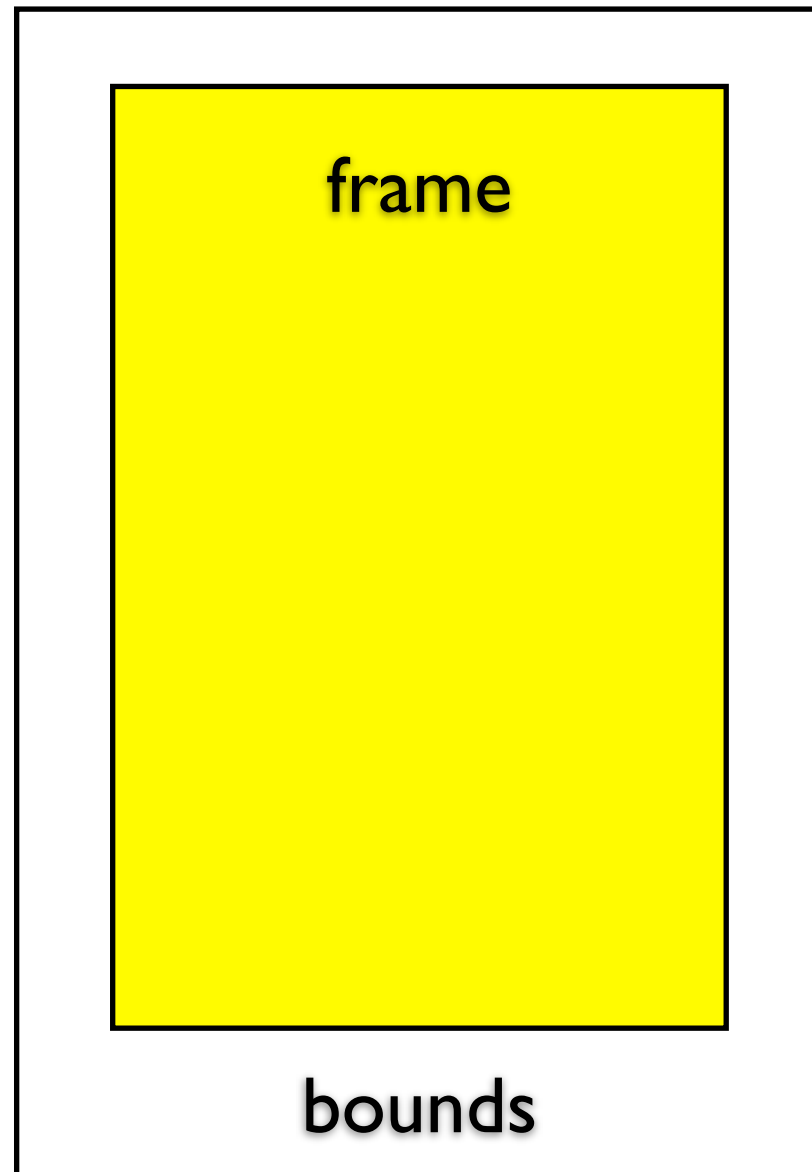
UIView

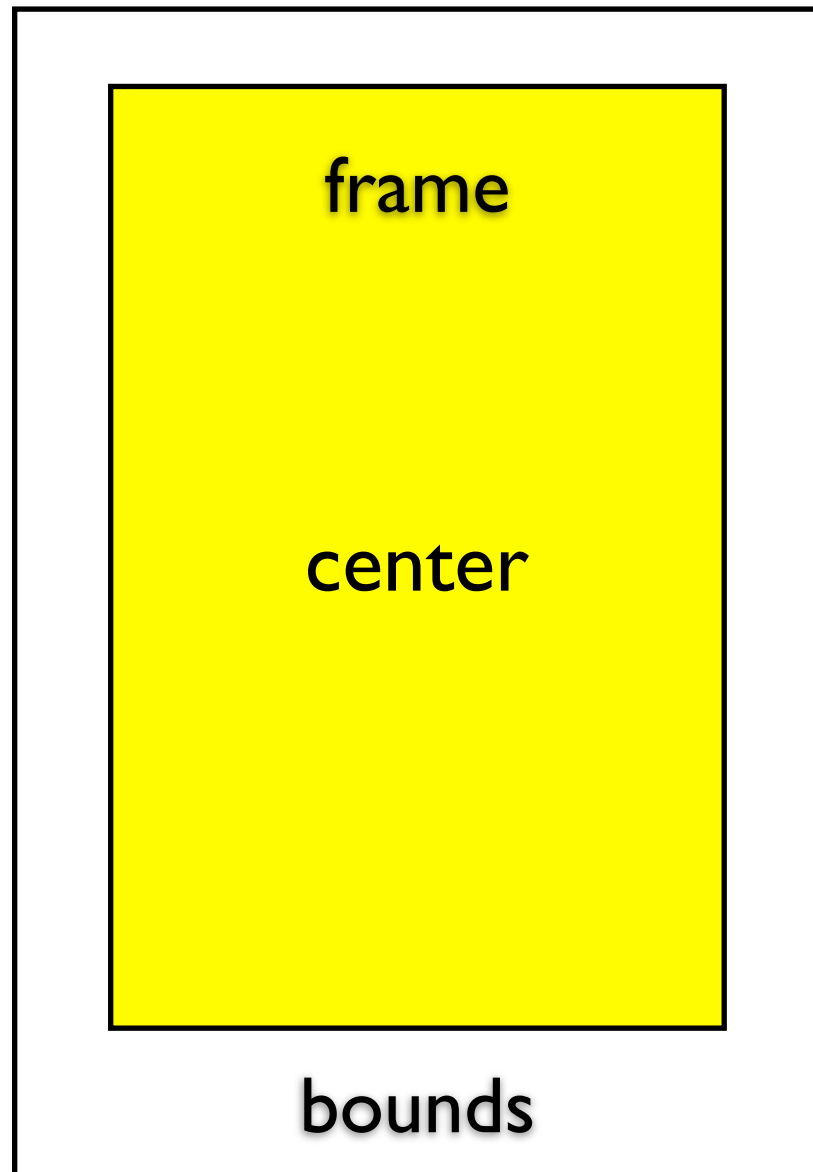
- central display element
- added to the display cue
- always has a UIViewController (which can be hidden)
- draw function defines its content
- [self setNeedsDisplay] [self setNeedsLayout]
refresh the view

http://developer.apple.com/library/ios/#documentation/uikit/reference/uiview_class/uiview/uiview.html









CGGeometry

- CGFloat
- CGPoint
- CGSize
- CGRect

CGGeometry

- CGFloat
- CGPoint
- CGSize
- CGRect

```
struct CGPoint {
    CGFloat x;
    CGFloat y;
};
typedef struct CGPoint CGPoint;
```

CGGeometry

- CGFloat
- CGPoint
- CGSize
- CGRect

```
struct CGPoint {
    CGFloat x;
    CGFloat y;
};
typedef struct CGPoint CGPoint;

struct CGSize {
    CGFloat width;
    CGFloat height;
};
typedef struct CGSize CGSize;
```

CGGeometry

- CGFloat
- CGPoint
- CGSize
- CGRect

```
struct CGPoint {
    CGFloat x;
    CGFloat y;
};
typedef struct CGPoint CGPoint;

struct CGSize {
    CGFloat width;
    CGFloat height;
};
typedef struct CGSize CGSize;

struct CGRect {
    CGPoint origin;
    CGSize size;
};
typedef struct CGRect CGRect;
```

creating a UIView

```
[[UIView alloc] initWithFrame:[UIScreen mainScreen].bounds ];
```

Within your AppDelegate:

```
self.window.rootViewController = [[UIViewController alloc] init];  
self.viewController.view=[[UIView alloc] initWithFrame:  
    [UIScreen mainScreen].bounds  
];
```


UIImageView

- derived from UIView
- loads a UIImage
- displays the image

http://developer.apple.com/library/ios/#documentation/uikit/reference/UIImageView_Class/Reference/Reference.html

http://developer.apple.com/library/ios/#documentation/uikit/reference/UIImage_Class/Reference/Reference.html###apple_ref/doc/c_ref/UIImage

UIScrollView

- derived from UIView
- allows scrolling within its bounds
- scrolls subviews accordingly
- `contentSize` is initialized with its frame, requires to `setContentSize`

http://developer.apple.com/library/ios/#documentation/UIKit/Reference/UIScrollView_Class/Reference/UIScrollView.html#//apple_ref/occ/cl/UIScrollView

essential functions

```
- (id)initWithFrame:(CGRect)frame{
    self = [super initWithFrame:frame];
    if (self) { /* Initialization code */ }
    return self;
}

[self setNeedsDisplay];

- (void)drawRect:(CGRect)rect{
}

-(void) touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event{
}

-(void) touchesEnded:(NSSet*)touches withEvent:(UIEvent*)event{
}

-(void) touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event{
}
```

managing views

```
-(void) touchesEnded:(NSSet*)touches withEvent:(UIEvent*)event{
    UITouch *T=[touches anyObject];
    if(T!=NULL){
        if([T locationInView:self].y<240)
        {
            mainViewClass *newView;
            CGRect bounds=[super bounds];
            newView=[[mainViewClass alloc]
                    initWithFrame:CGRectMake(
                        bounds.origin.x+10 ,
                        bounds.origin.y+10,
                        bounds.size.width-20,
                        bounds.size.height-20)
                    ];
            [self addSubview:newView];
        }
        else{
            [self removeFromSuperview];
        }
    }
}
```