

Разработване на приложения за iOS



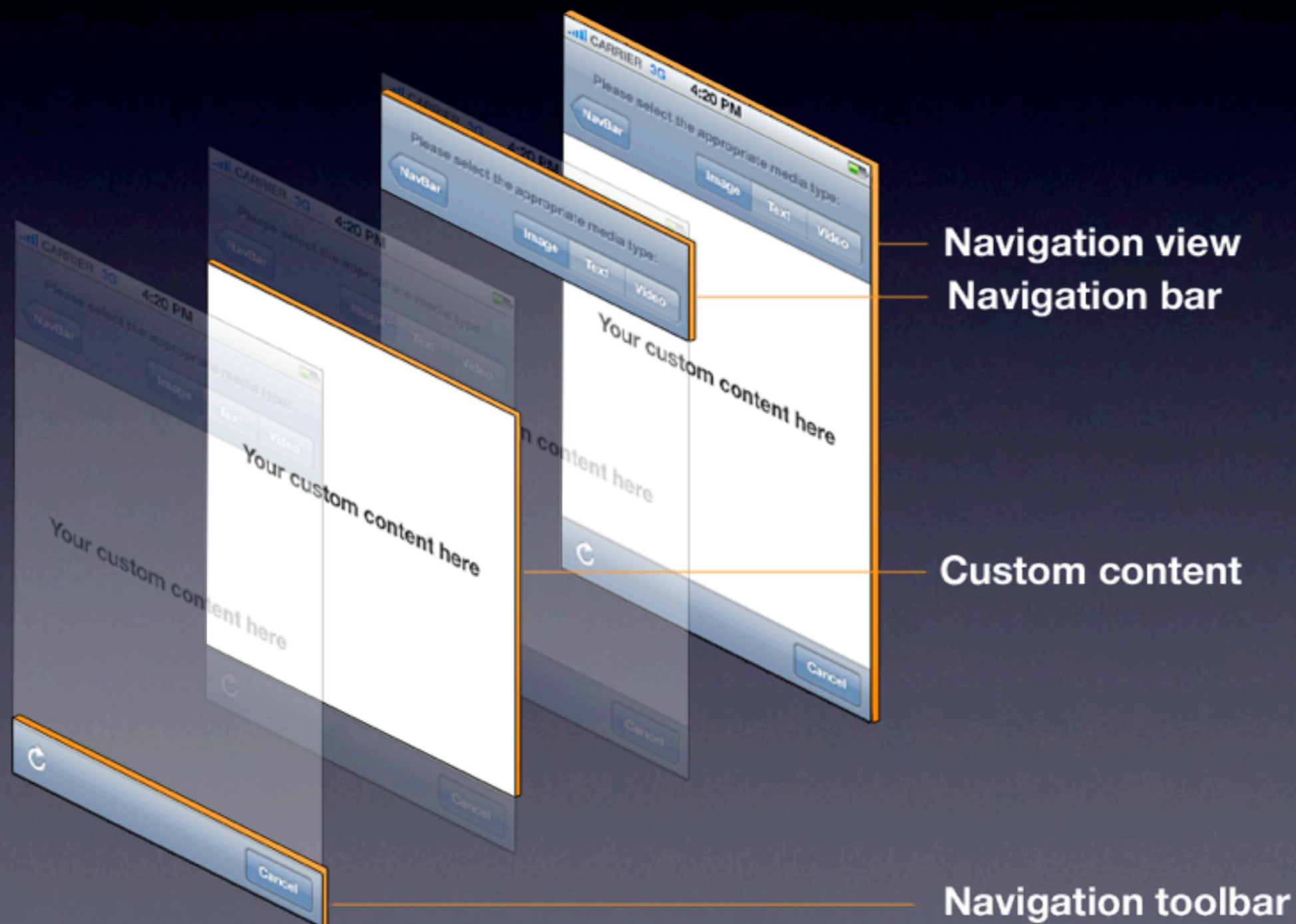
Петко Пенчев, TapBits.com

Компоненти на навигацията

- Status Bar
- Navigation bar
- Window, View(s)
- Tab Bar

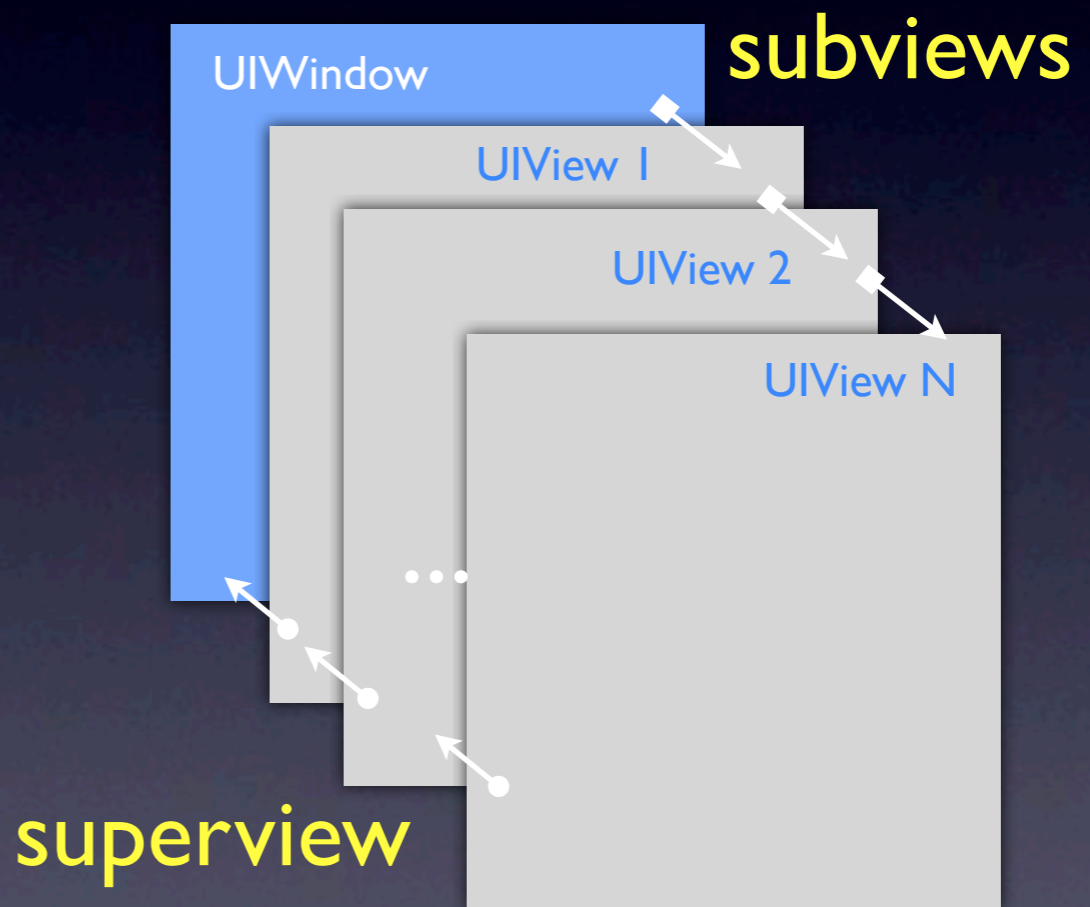


Иерархия на изгледите



Йерархия на изгледите

- UIWindow, UIView
- Дефинира
 - геометрична област
Frame, Bound ..
 - чертае
 - обслужва събития
- Йерархия
 - (UIView *)superview;
 - (NSArray *)subviews;



UIView

- Създаване

Interface Builder или програмно

- - (void)addSubview:(UIView *)aView;
 - (void)removeFromSuperview;
- - (void)insertSubview:(UIView *)aView atIndex:(int)index;
 - (void)insertSubview:(UIView *)aView belowSubview:(UIView *)otherView;
 - (void)insertSubview:(UIView *)aView aboveSubview:(UIView *)otherView;

UIView

- UIView 1, UIView 2

- *Frame:*

абсолютни координати

`UIView *v1;`

`v1.frame = CGRect(0,0,H,W);`

`UIView *v2;`

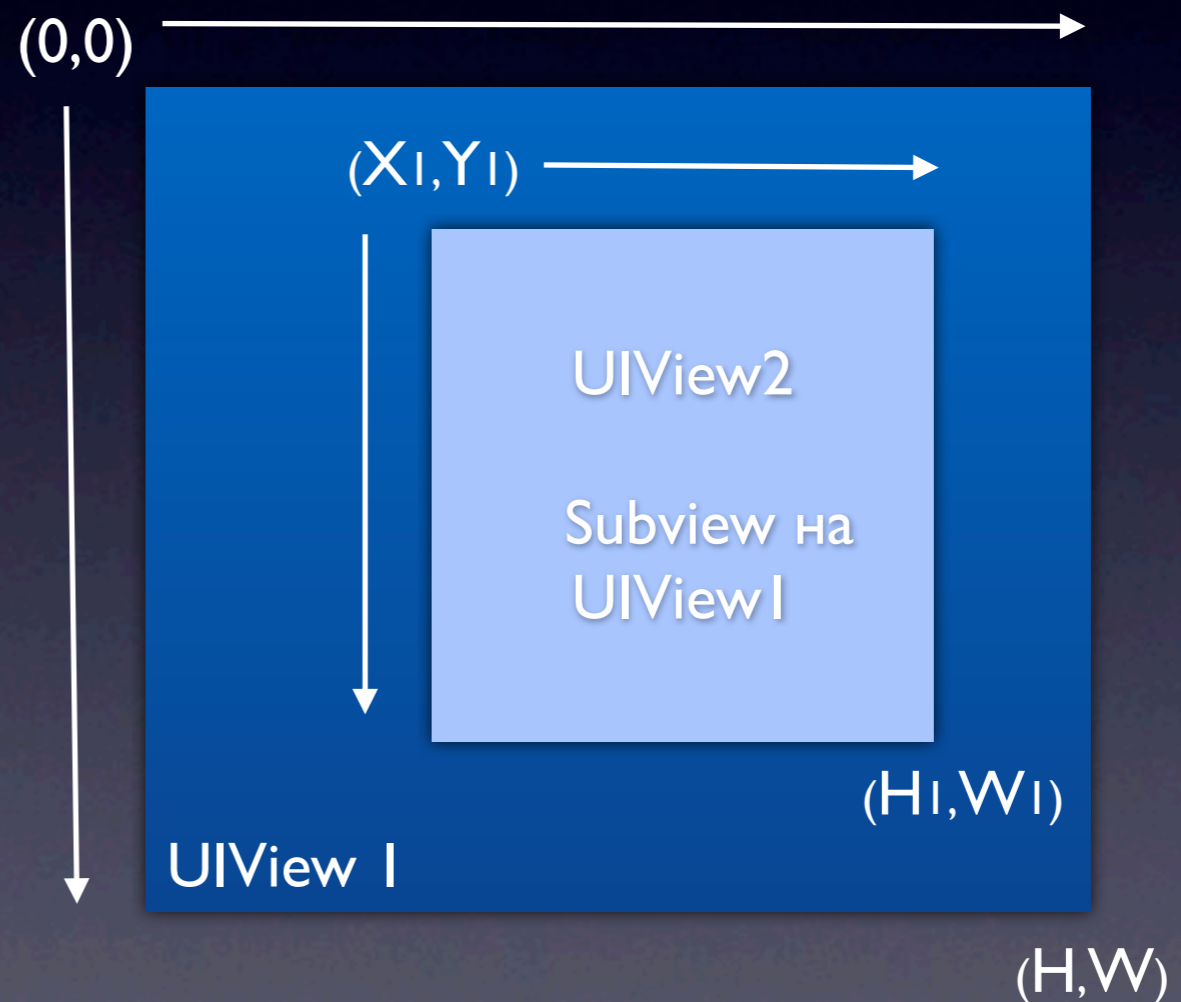
`v2.frame = CGRect(0,0,H1,W1);`

- *Bounds:*

спрямо superview

`v1.bounds = CGRect(0,0,H,W);`

`v2.bounds = CGRect(X1,Y1,H,W);`



UIView

- Други свойства:

Tag, Orientation, Background, Hidden, Alpha, Opaque, contentScaleFactor

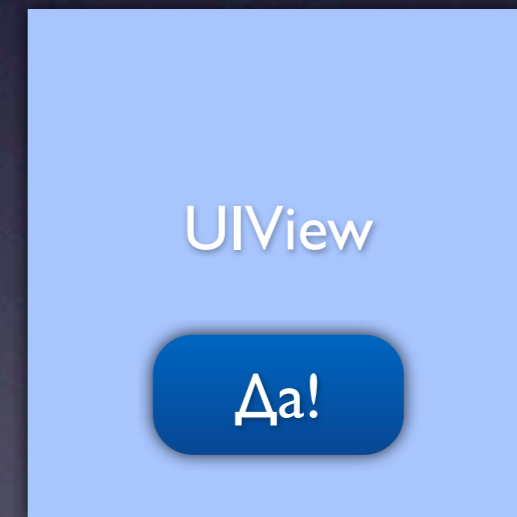
- Създаване на UIView програмно

```
CGRect buttonRect = CGRectMake(20, 20, 120, 37);
```

```
UIButton *button = [[UIButton alloc] initWithFrame:buttonRect];  
button.titleLabel.text = @"Да!";
```

```
[window addSubview:button];  
// view йерархията се грижи да retain UIView обектите
```

```
[button release];
```



view.tag и прелистване

```
CGRect viewRect = CGRectMake(10, 10, 100, 100);
```

```
UIView* myView = [[UIView alloc] initWithFrame:viewRect];
```

```
...
```

```
for (UIImageView *anImage in [self.view subviews]) {  
    if (anImage.tag == 1) {  
        // do something  
    }  
}
```

Координати

- **CGFloat**

реално е float, използва се винаги при графика вместо float

- **CGPoint**

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

```
CGPoint p = CGPointMake(34.5, 22.0);  
p.x+=20;
```

Координати

- **CGSize**

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

```
CGSize s = CGSizeMake(100.0, 200.0);  
s.height += 50;
```

- **CGRect**

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

```
CGRect aRect = CGRectMake(45.0, 75.5, 300, 500);  
aRect.size.height += 45;  
aRect.origin.x += 30;
```

Създаване на UIView

- Наследяваме UIView
- Пренаписваме
 - `(void)drawRect:(CGRect)aRect;`
- Важно - директно не извикваме никога `drawRect`
- Важно за обновяване (пречертване) използваме
 - `(void)setNeedsDisplay;`
 - `(void)setNeedsDisplayInRect:(CGRect)aRect;`

“Рисуване” в UIView

- Core Graphics Framework
- “взимане” на context-а (iOS се грижи да създаде context при всяко извикване на `drawRect:`)
- Изчертаване на линии/фигури чрез пътеки от точки
- устанавяване на текстуре, цветове и шрифтове и т.н.

Изчертаване

- Взимане на context
`CGContextRef context = UIGraphicsGetCurrentContext();`
- Започване на пътека за изчертаване
`CGContextBeginPath(context);`
- Разширяване на пътека за изчертаване
`CGContextMoveToPoint(context, 10, 10);`
`CGContextAddLineToPoint(context, 50, 50);`
`CGContextAddLineToPoint(context, 100, 10);`
- Започване на пътека за изчертаване (препоръчително)
`CGContextClosePath(context);`
- `CGPath` вместо `CGContext`

UIColor

- UIColor class

```
UIColor *red = [UIColor redColor];
```

или

```
UIColor *custom = [[UIColor alloc] initWithRed:(CGFloat)red  
                blue:(CGFloat)blue  
                green:(CGFloat)green  
                alpha:(CGFloat)alpha; // 0.0 to 1.0 (opaque)
```

```
[redsetFill];  
[custom set];
```

- UIColor class и RGB

```
#define UIColorFromRGB(rgbValue) [UIColor \  
    colorWithRed:((float)((rgbValue & 0xFF0000) >> 16))/255.0 \  
    green:((float)((rgbValue & 0xFF00) >> 8))/255.0 \  
    blue:((float)(rgbValue & 0xFF))/255.0 alpha:1.0]
```

```
tintColor = UIColorFromRGB(0x85c900);
```

- CGContextSetLineWidth(context, 1.0); //linewidthinpoints(notpixels)
CGContextSetFillPattern(context, (CGPatternRef)pattern, (CGFloat[])components));