autoLayout自动布局

1. 简介：
2. autolayout是一种布局技术，专门用来布局UI界面的，用来取代frame布局在遇到多屏幕尺寸多种多样的不足。从ios6开始引入，由于xcode4不给力，当时没有得到很大推广。在ios7(xcode5)开始，autolayout的开发效率得到了很大的提升。苹果官方也推荐使用autolayout来解决屏幕适配的问题。
3. autolayout的实现很多种：苹果api，之后的vfl，storyboard，第三方masonry等，下面就分别举例介绍各种使用方法。
4. 苹果API NSLayoutConstraint 注意参照关系，有时constant为负有时为正

例：设置redView距离父视图左边距20，底边距20，宽50，高50

UIView \*redView = [[UIView alloc] init];

[redView setBackgroundColor:[UIColor redColor]];

[redView setTranslatesAutoresizingMaskIntoConstraints:NO];//不以autoresizingMask进行计算(ios6以前只有3.5寸屏幕)

//autoresizing是一个相对父控件的布局解决方法，只有上下左右宽高可以调用

//由于autolayout和autoresizing是相对冲突的，所以此处禁用autoresizing

[self.view addSubView:redView];

NSLayoutConstraint \*redLeft = [NSLayoutConstraint constraintWithItem:redView attribute:NSLayoutAttributeLeft relateBy:NSLayoutRelationEqual toItem:self.view attribute NSLayoutAttributeLeft multiplier:1.0f constant:20.0];

NSLayoutConstraint \*redBottom = [NSLayoutConstraint constraintWithItem:redView attribute:NSLayoutAttributeBottom relateBy:NSLayoutRelationEqual toItem:self.view attribute NSLayoutAttributeBottom multiplier:1.0f constant:20.0];

NSLayoutConstraint \*redWidth = [NSLayoutConstraint constraintWithItem:redView attribute:NSLayoutAttributeWidth relateBy:NSLayoutRelationEqual toItem:nil attribute NSLayoutAttributeNotAnAttribute multiplier:1.0f constant:50.0f];

NSLayoutConstraint \*redHeight = [NSLayoutConstraint constraintWithItem:redView attribute:NSLayoutAttributeHeight relateBy:NSLayoutRelationEqual toItem:nil attribute NSLayoutAttributeNotAnAttribute multiplier:1.0f constant:50.0f];

[self.view addConstraint:redLeft];//有参照关系，约束加到父视图

[self.view addConstraint:redBottom];

[redView addConstraint:redWidth];//无参照关系，约束加到自身

[redView addConstraint:redHeight];

1. VFL语法
2. 简述：vfl的出现是为了减轻原api带来的烦琐。Vfl思想与其它实现方法不同，它更宏观化，它将约束分成了两块：水平方向H:和竖直方向V:，也就是说，建立约束时，得把水平与垂直方向的约束用字符串一并表达出。
3. 例子：在2例子基础上水平放置3个view，互相间隔20

//redView,blueView,yellowView

//view添加到self.view

//开始建立约束

NSString \*Hvfl = @”H:|-margin-[redView(50)-margin-[blueView(==redView)]-margin-[yellowView(==redView)]]”;

//设置替换的数字(用字典表示)

NSDictionary \*metrics = @{@”margin”:@20};

//把药添加约束的view转成字典

NSDictionary \*views = NSDictionaryOfVariableBindings(redView,blueView,yellowView);

//设置对齐方式，顶部和底部均与redView对齐

NSLayoutFormatOptions ops = NSLayoutFormateAllTop|NSLayoutFormatAllBottom;

//创建水平约束

NSArray \*Hconstraints = [NSLayoutConstraint constraintsWithVisualFormat:Hvfl options:ops metrics:metrics views:views];

//创建竖直约束

NSString \*Vvfl = @”V:[redView(50)]-margin-|”;

NSArray \*Vconstraints = [NSLayoutConstraint constraintsWithVisualFormat:Vvfl options:ops metrics:metrics views:views];

//父视图添加约束

[self.view addConstraint:Hconstraints];

[self.view addConstraint:Vconstraints];

1. 总结：
2. VFL实现的autolayout是有局限性的，如果想设置约束 优先级或是传入倍数关系必须调用苹果原API [NSLayoutConstraint constraintWithItem:...]
3. VFL仅供了解，最好用的莫过于第三方框架masonry了
4. Masonry框架
5. 简介：这是一款公认的非常简洁优美的autolayout框架，这框架约束添加思维和原API添加约束是相同的，只是Masonry语法更简洁，代码更美观。

注：autolayout目的是为了确定控件位置及尺寸，除此外不要添加多余约束，否则容易出现约束冲突，除非你想使用约束优先级

1. 例子讲解：
2. 将屏幕平分四块，redView blueView yellowView greenView

//添加四个view至父视图

//添加红色view约束

[redView mas\_makeConstraints:^(MASConstraintMaker \*make){

make.left.equalTo(self.view.mas\_left).offset(0);//使左边等于self.view的左边，间距为0

make.left.equalTo(self.view.mas\_top).offset(0);

make.width.equalTo(self.view.mas\_width).multipliedBy(0.5);//宽是self.view 的0.5倍

make.height.equalTo(self.view.mas\_height).multipliedBy(0.5);  
}];//至此红色view约束添加完成，其它view以其确定自己位置和尺寸如：

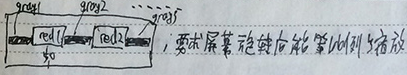
[blueView mas\_makConstraints:^(MASConstraintMaker \*make){

make.width.and.height.equalTo(redView);//使宽高等与redView的宽高

make.top.equalTo(redView.mas\_top);//与redView顶部对齐

make.leading.equalTo(redView.mas\_right);//与redView间距为0  
}];

1. 要求屏幕旋转后能等比例缩放，效果如下：



[gray1 mas\_makeConstraints:^(MASConstraintMaker \*maker){

make.height.mas\_equalTo(20);

make.leading.equalTo(self.view.mas\_leading).offset(0);

make.bottom.equalTo(red1.mas\_bottom);  
}];

[red1 mas\_makeConstraints:^(MASConstraintMaker \*maker){

make.width.mas\_equalTo(100);

make.height.mas\_equalTo(50);

make.left.equalTo(gray1.mas\_right);

make.bottom.equalTo(self.view.mas\_bottom).offset(-50);  
}];

[gray2 mas\_makeConstraints:^(MASConstraintMaker \*make){

make.height.and.width.equalTo(gray1);

make.leading.equalTo(red1.mas\_right);

make.bottom.equalTo(red1.mas\_bottom);

}];

[red2 mas\_makeConstraints:^(MASConstraintMaker \*make){

make.height.and.width.equalTo(red1);

make.leading.equalTo(gray2.mas\_right);

make.bottom.equalTo(red1.mas\_bottom);  
 }];

[gray3 mas\_makeConstraints:^(MASConstraintMaker \*make){

make.height.and.width.equalTo(gray1);

make.leading.equalTo(red2.mas\_right);

make.trailing.equalTo(self.view.mas\_right);

make.bottom.equalTo(red1.mas\_bottom);  
}]

1. storyboard实现autolayout
2. 右键拖拽控件建立约束
3. 右下角添加对齐和约束(可选参照是view还是layoutGuide)
4. 例子：略