

自定义有多个按钮节点的SliderView



作者 十字云 (/u/375ab64a5fb7) + 关注

2017.03.02 12:56* 字数 536 阅读 436 评论 6 喜欢 33

(/u/375ab64a5fb7)

前言

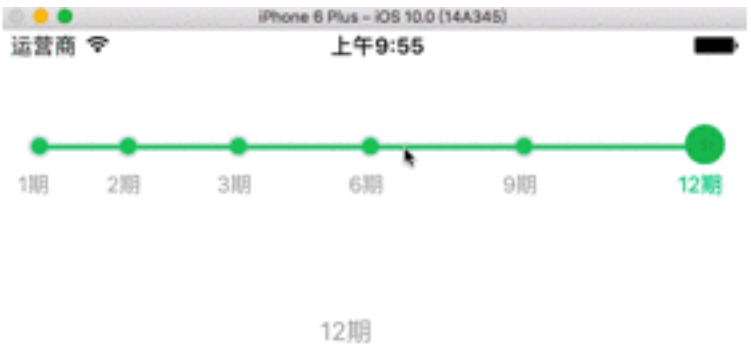
前些天看到一个设计图，关于分期付款选择期数的，有多个节点。它像是一个 sliderView,但是sliderView实现不了多个节点按钮。所以，我就想到了自定义 sliderView。DCSliderView (<https://github.com/XDChang/DCSliderView>)

设计图如下：



期数设计图.png

最终效果图如下：



DCSliderView.gif

设计思路

1. 先添加一个底层view，然后在底层view上画出背景layer,这里是六个小圆，和一个细长矩形。
2. 小圆点是可点的，所以还要创建六个btn,并添加下标题。

- 3. 在底层view的上方添加一个view，充当滑动控制器。
- 4. 在滑动控制器上添加拖拽手势，并且控制滑动时，只改变控制器的X坐标，Y轴保持不变。
- 5. 绘制绿色layer跟随滑动控制器而动。
- 6. 处理各个按钮的点击事件，让滑动控制器跟绿色layer随之改变。
- 7. 处理细节，吸附功能，点亮下标题，对滑动控制器最小和最大X轴位移的控制。
- 8. 设置代理，在各个方法里触发代理方法。

实现相关功能

- 1. 创建底层view，在view上添加各种layer；创建btn和下标题。

```
9  #import "DCSliderView.h"
10 #define WIDTH self.frame.size.width
11 #define TITLE @"1期",@"2期",@"3期",@"6期",@"9期",@"12期";
12 @interface DCSliderView ()
13
14
15 @property (nonatomic,strong) NSMutableArray *btnArr; // 创建的btn
16 @property (nonatomic,strong) NSMutableArray *btnOriginXArr;// 每个btn的X坐标
17 @property (nonatomic,strong) NSMutableArray *btnLayerArr; // 多个圆的绿色layer
18 @property (nonatomic,strong) NSMutableArray *titleLabelArr; // 标题label数组
19 @property (nonatomic,assign) float xx; // 圆心系数X
20 @property (nonatomic,assign) float yy; // 圆心系数Y
21 @property (nonatomic,assign) float middleGap;//圆之间的中点系数
22 @end
23
24 @implementation DCSliderView
25 {
26     UIView *_holeShapeView; // 底层灰色背景view
27     UIImageView *_targetView; // 大按钮
28     UIBezierPath *_recPath; // 创建绿色layer的贝塞尔
29     CAShapeLayer *_tubeShape; // 创建绿色layer的ShapeLayer
30     CGColorRef K_CGColor; // layer的颜色
31 }
```

```
#pragma mark --- 加载所有的layer
- (void)drawWholeShape
{
    CGFloat gapX = self.frame.origin.x; //父视图距离屏幕左边的距离(实现各个圆之间的间距逐渐增大，我自己设置了几个参数，大家可以根据自己的实际情况去改变圆之间的间距。不是非要按照这个来，这里只是提供思路。)
    // 用贝塞尔函数画出细长矩形路径
    UIBezierPath *recPath = [UIBezierPath bezierPath];
    [recPath moveToPoint:CGPointMake(8, 4)];//上起点
    [recPath addLineToPoint:CGPointMake(8, 8)];//下起点
    [recPath addLineToPoint:CGPointMake(8+WIDTH-2*gapX, 8)];//下结束点
    [recPath addLineToPoint:CGPointMake(8+WIDTH-2*gapX, 4)];//上结束点
    // 用CAShapeLayer绘制细长矩形
    CAShapeLayer *tubeShape = [[CAShapeLayer alloc]init];
    tubeShape.path = recPath.CGPath;
    tubeShape.strokeColor = [UIColor colorWithRed:224/255.0 green:224/255.0 blue:224/255.0 alpha:1].CGColor;// 外边框颜色
    tubeShape.fillColor = [UIColor colorWithRed:224/255.0 green:224/255.0 blue:224/255.0 alpha:1].CGColor;// 内部填充颜色

    [_holeShapeView.layer addSublayer:tubeShape];

    NSArray *title = TITLE;
    // for 循环绘制六个灰色小圆跟绿色小圆，创建六个btn,下标题并添加进数组
    for (int i = 0; i <6; i ++) {

        //灰色小圆
        UIBezierPath *leftSemiPath1 = [UIBezierPath bezierPath];

        CGPoint pointR1 = CGPointMake(12 +(_yy+_xx*i)*i, 6);

        [leftSemiPath1 addArcWithCenter:pointR1 radius:6 startAngle:(0.0 * M_PI) endAngle:(2.0 * M_PI) clockwise:YES];

        CAShapeLayer *leftSemiShape1 = [[CAShapeLayer alloc]init];
```

```

        leftSemiShape1.path = leftSemiPath1.CGPath;

        leftSemiShape1.strokeColor = [UIColor colorWithRed:224/255.0 green:224/25
5.0 blue:224/255.0 alpha:1].CGColor;
        leftSemiShape1.fillColor = [UIColor colorWithRed:224/255.0 green:224/255.
0 blue:224/255.0 alpha:1].CGColor;

        [_holeShapeView.layer addSublayer:leftSemiShape1];

        // 绿色小圆
        UIBezierPath *leftSemiPath2 = [UIBezierPath bezierPath];

        CGPoint pointR2 = CGPointMake(12 +(_yy+_xx*i)*i, 6);

        [leftSemiPath2 addArcWithCenter:pointR2 radius:4 startAngle:(0.0 * M_PI)
endAngle:(2.0 * M_PI) clockwise:YES];

        CAShapeLayer *leftSemiShape2 = [[CAShapeLayer alloc]init];

        leftSemiShape2.path = leftSemiPath2.CGPath;

        leftSemiShape2.strokeColor = K_CGColor;
        leftSemiShape2.fillColor = K_CGColor;

        [self.btnLayerArr addObject:leftSemiShape2];

        if (i==0) {
            // 将第一个绿色小圆添加到底层view上
            [_holeShapeView.layer addSublayer:leftSemiShape2];
        }

        float x = 4 +(_yy+_xx*i)*i;
        // 创建btn
        UIButton *stepBtn = [[UIButton alloc]initWithFrame:CGRectMake(x, -2, 14,
14)];

        [_btnArr addObject:stepBtn];
        [self.btnOriginXArr addObject:@(x)];

        stepBtn.tag = i;

        [stepBtn addTarget:self action:@selector(onBtnClick:) forControlEvents:UI
ControlEventTouchUpInside];
        [self addSubview:stepBtn];

        // 创建下标题
        UILabel *qiShuLabel = [[UILabel alloc]init];

        qiShuLabel.center = CGPointMake(x-4, 20);
        qiShuLabel.text = title[i];
        qiShuLabel.textColor = [UIColor colorWithRed:153/255.0 green:153/255.0 bl
ue:153/255.0 alpha:1];
        qiShuLabel.font = [UIFont systemFontOfSize:12];

        [qiShuLabel sizeToFit];

        [self addSubview:qiShuLabel];

        [self.titleLabelArr addObject:qiShuLabel];
    }

}

```

2. 创建滑动控制器view，并添加滑动手势。

```

- (void)initTargetView
{
    _targetView = [[UIImageView alloc]initWithFrame:CGRectMake(0, -6, 22, 22)];
    _targetView.image = [UIImage imageNamed:@"target"];

    _targetView.userInteractionEnabled = YES;

    UIPanGestureRecognizer *imageViewPanGesture = [[UIPanGestureRecognizer alloc]
initWithTarget:self action:@selector(panGesture:)];

    [_targetView addGestureRecognizer:imageViewPanGesture];

    [self addSubview:_targetView];

}

```

//在移动过程中, UIGestureRecognizerStateChanged 这个状态会调用很多次, 在这里面处理绿色细长矩形的绘制, 添加或删除绿色小圆layer。

//在移动结束时, UIGestureRecognizerStateChanged 这个状态只调用一次, 在这里处理最终的绿色细长矩形, 绿色小圆, 下标题的点亮, 吸附功能。

```
- (void)panGesture:(UIPanGestureRecognizer *)gesture
{
    CGFloat y;

    switch (gesture.state) {

        case UIGestureRecognizerStateBegan:
        {

            CGRect rect = gesture.view.frame;
            y = rect.origin.y ;
        }

        break;
        case UIGestureRecognizerStateChanged:
        {
            // 获得添加手势的对象
            // 获得滑动的距离 包含 x y 移动的数值
            CGPoint point =[gesture translationInView:gesture.view];

            CGRect targetRect = _targetView.frame;

            CGFloat targetX = targetRect.origin.x;

            // 绿色的细长矩形
            [_recPath removeAllPoints];// 这个方法会调用很多次, 每次调用都会绘制一条路径
            , 为了实现绿色路径跟随滑动控制器而动的效果, 所有每次绘制之前都移除掉所有的点, 其它地方有这样的处理
            都是一个道理。
            [_recPath moveToPoint:CGPointMake(8, 5.8)];
            [_recPath addLineToPoint:CGPointMake(8, 7)];

            if (targetX>8) { // 避免超出最小范围
                [_recPath addLineToPoint:CGPointMake(targetX, 7)];
                [_recPath addLineToPoint:CGPointMake(targetX, 5.8)];
            }

            [_recPath closePath];

            _tubeShape.path = _recPath.CGPath;
            [_tubeShape setNeedsDisplay];
            [self.layer addSublayer:_tubeShape];

            NSArray *titleArr = TITLE;

            for (int i = 0; i <6; i ++) {

                if (i!=5) {
                    // 滑动过程中添加和删除绿色圆layer
                    if (targetX >= [self.btnOriginXArr[i]integerValue] && targetX
< [_btnOriginXArr[i+1]integerValue]) {
                        // 删除上一个绿色小圆layer
                        CAShapeLayer *layer = self.btnLayerArr[i+1];
                        if (layer) {
                            [layer removeFromSuperlayer];
                        }
                        // 添加新的绿色小圆layer
                        [_holeShapeView.layer addSublayer:self.btnLayerArr[i]];
                        [_shapeViewDelegate onShapeViewDelegateEventWithString:ti
tleArr[i]];// 调用代理方法, 回调期数
                    }

                }

            }

        }

        //CGRectOffset是以试图的原点为起始 移动 dx x移动距离 dy y移动距离

        gesture.view.frame =CGRectOffset(gesture.view.frame, point.x, y );//
        改变滑动控制器的frame, 只改变X, Y坐标保持不变。

        //清空移动距离
        [gesture setTranslation:CGPointZero inView:gesture.view];

    }

    break;
    case UIGestureRecognizerStateEnded:
    {

        CGRect targetRect = _targetView.frame;
```

```

CGFloat targetX = targetRect.origin.x;

float btnX = [self.btnOriginXArr.lastObject integerValue];
// targetView在第一个圆
if (targetX<0) {

    targetRect.origin.x = 0;

    _targetView.frame = targetRect;

    [_shapeViewDelegate onShapeViewDelegateEventWithString:@"1期"];
    // 改变下标题颜色
    for (UILabel *label in self.titleLabelArr) {

        label.textColor = [UIColor colorWithRed:153/255.0 green:153/2
55.0 blue:153/255.0 alpha:1];
    }
    UILabel *firstLabel = self.titleLabelArr.firstObject;

    firstLabel.textColor = [UIColor colorWithCGColor:K_CGColor];
    break;
}
// targetView在最后一个圆
if (targetX >btnX) {

    targetRect.origin.x = btnX;

    _targetView.frame = targetRect;
    [_shapeViewDelegate onShapeViewDelegateEventWithString:@"12期"];
    // 改变下标题颜色
    for (UILabel *label in self.titleLabelArr) {

        label.textColor = [UIColor colorWithRed:153/255.0 green:153/2
55.0 blue:153/255.0 alpha:1];
    }
    UILabel *firstLabel = self.titleLabelArr.lastObject;

    firstLabel.textColor = [UIColor colorWithCGColor:K_CGColor];
    break;
}

NSArray *titleArr = TITLE;
// targetView 在中间各个圆
for (int i = 0; i <6; i ++) {

    if (i!=5) {

        if (targetX >= [self.btnOriginXArr[i]integerValue] && targetX
< [_btnOriginXArr[i]integerValue]+15.0 +_middleGap*i) {

            NSLog(@"%ld", (long)[_btnOriginXArr[i]integerValue]);

            targetRect.origin.x = [_btnOriginXArr[i]integerValue];
            _targetView.frame = targetRect;

            [_shapeViewDelegate onShapeViewDelegateEventWithString:ti
tleArr[i]];

            for (UILabel *label in self.titleLabelArr) {

                label.textColor = [UIColor colorWithRed:153/255.0 gre
en:153/255.0 blue:153/255.0 alpha:1];
            }
            UILabel *firstLabel = self.titleLabelArr[i];

            firstLabel.textColor = [UIColor colorWithCGColor:K_CGColo
r];

        }
        else if(targetX >=[_btnOriginXArr[i]integerValue]+10.0 + _mid
dleGap*i)
        {
            targetRect.origin.x = [_btnOriginXArr[i+1]integerValue];
            _targetView.frame = targetRect;
            [_shapeViewDelegate onShapeViewDelegateEventWithString:ti
tleArr[i+1]];

            // 改变下标题颜色
            for (UILabel *label in self.titleLabelArr) {

                label.textColor = [UIColor colorWithRed:153/255.0 gre
en:153/255.0 blue:153/255.0 alpha:1];
            }
            UILabel *firstLabel = self.titleLabelArr[i+1];

```

```
firstLabel.textColor = [UIColor colorWithCGColor:K_CGColor];
    }
    }
}
// 先移除贝塞尔所有的点,然后重新绘制贝塞尔路径
[_recPath removeAllPoints];
[_recPath moveToPoint:CGPointMake(8, 5.8)];
[_recPath addLineToPoint:CGPointMake(8, 7)];

[_recPath addLineToPoint:CGPointMake(_targetView.frame.origin.x, 7)];
[_recPath addLineToPoint:CGPointMake(_targetView.frame.origin.x, 5.8)
];

[_recPath closePath];
_tubeShape.path = _recPath.CGPath;
_tubeShape setNeedsDisplay];
[self.layer addSublayer:_tubeShape];
}
break;
default:
break;
}
}
```

3 . 处理按钮的点击事件。

```
- (void)onBtnClick:(UIButton *)btn
{
    NSArray *titleArr = TITLE;

    [_shapeViewDelegate onShapeViewDelegateEventWithString:titleArr[btn.tag]];//
回调代理

// 滑动控制器frame动画
[UIView animateWithDuration:0.3 animations:^(

    NSInteger x    = [_btnOriginXArr[btn.tag] integerValue];
    CGRect rect    = _targetView.frame;
    rect.origin.x = x;
    _targetView.frame = rect;

    } completion:^(BOOL finished) {
        // 改变下标题颜色
        for (UILabel *label in self.titleLabelArr) {

            label.textColor = [UIColor colorWithRed:153/255.0 green:153/255.0 blue:153/255.0 alpha:1];
        }
        UILabel *firstLabel = self.titleLabelArr[btn.tag];

        firstLabel.textColor = [UIColor colorWithCGColor:K_CGColor];
    }];

// layer的动画没处理好，这里通过延迟处理，实现相关功能，下次layer动画处理好了再补充上来。
dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(0.15 * NSEC_PER_SEC)), dispatch_get_main_queue(), ^{

//绿色小圆layer的添加和删除
    for (CAShapeLayer *layer in self.btnLayerArr) {
        [layer removeFromSuperlayer];
    }
    for (int i = 0; i < btn.tag+1; i ++) {

        [_holeShapeView.layer addSublayer:self.btnLayerArr[i]];

    }
    // 先移除贝塞尔所有的点,然后重新绘制贝塞尔路径
    [_recPath removeAllPoints];
    [_recPath moveToPoint:CGPointMake(8, 5.8)];
    [_recPath addLineToPoint:CGPointMake(8, 7)];

    if (_targetView.frame.origin.x > 8) { // 控制最小距离

        [_recPath addLineToPoint:CGPointMake(_targetView.frame.origin.x, 7)];
        [_recPath addLineToPoint:CGPointMake(_targetView.frame.origin.x, 5.8)
    ];

    }

    [_recPath closePath];

    _tubeShape.path = _recPath.CGPath;
    [_tubeShape setNeedsDisplay];
    [self.layer addSublayer:_tubeShape];

    });

}
```

具体使用方法

下载好我的demo (<https://github.com/XDChang/DCSliderView>),在工程中导入 DCSliderView 类，设置代理ShapeViewDelegate，具体代码如下：

```
// 1.
    DCSliderView *shapeView = [[DCSliderView alloc] initWithFrame:CGRectMake(10, 6
0, self.view.frame.size.width -20, 30) WithLayerColor:[UIColor colorWithRed:0/255
.0 green:210/255.0 blue:87/255.0 alpha:1]];
    // DCSliderView 的左右间距10 , 宽度self.view.frame.size.width -20, 最好不要变。
// 2.
    shapeView.shapeViewDelegate = self;

//3.
    [self.view addSubview:shapeView];

    _qiShuLabel = [[UILabel alloc] init];

    _qiShuLabel.center = CGPointMake(self.view.frame.size.width/2-30, 160);

    _qiShuLabel.textColor = [UIColor colorWithRed:153/255.0 green:153/255.0 blue:
153/255.0 alpha:1];

    _qiShuLabel.font = [UIFont systemFontOfSize:14];

    _qiShuLabel.text = @"1期" ;
    [_qiShuLabel sizeToFit];

    [self.view addSubview:_qiShuLabel];

// 4.代理方法
- (void)onShapeViewDelegateEventWithString:(NSString *)str
{
    _qiShuLabel.text = str ;
    [_qiShuLabel sizeToFit];
}
```

需要注意的一点是，由于各个小圆之间的间距是逐渐增大的，所以我根据屏幕的宽度设置了几个不同的系数去适配，如果你没有使用我代码中的宽度，适配就会出现问

题。其实本文只是一个引子，主讲设计思路，你可以按照自己的实际情况去具体设计。当然，如果你不想动手修改的话，那就得按照我设计的来。

转载请注明出处



十字云 (/u/375ab64a5fb7)

写了 9081 字, 被 20 人关注, 获得了 78 个喜欢

(/u/375ab64a5fb7)

+ 关注

衍道九天外,情溢武昌湖。(另：造自己的轮子，让别人搬你的轮子)

万水千山总是情,来波赞赏行不行~😂

赞赏支持



(/u/54b5900965ea)

🤍 喜欢

|

33

💬

🐼


🖼️

更多分享

(http://cwb.assets.jianshu.io/notes/images/9693319/




写下你的评论...

 wolf0803 (/u/ad49fe5440f9)
2楼 · 2017.03.02 13:39

/u/ad49fe5440f9)
不错不错哦!很有用哦👍

👍 赞


💬 回复

 By9 (/u/d87b76356fe4)
3楼 · 2017.03.02 15:18

/u/d87b76356fe4)
写的可以，给个赞👍

👍 赞


💬 回复

 靖锅锅_ (/u/8d8b564e1718)
4楼 · 2017.03.02 20:52

/u/8d8b564e1718)
好好好

👍 赞


💬 回复

 helloDolin (/u/6c0a7aeb2e84)
5楼 · 2017.03.05 12:55

/u/6c0a7aeb2e84)
mark

👍 赞

💬 回复

 黑白羽森林 (/u/5a15700eea7c)
6楼 · 2017.03.05 18:29

/u/5a15700eea7c)
码👍

👍 赞

💬 回复

 Usan (/u/d832fdf6d280)
7楼 · 2017.03.07 11:28

/u/d832fdf6d280)
mark

👍 赞

💬 回复

被以下专题收入，发现更多相似内容

+

我的专题

首页投稿

iOS开发

今日看点

程序员

iOS Dev...

动画

iOS自定义View

OC基础

iOS开发攻城...

iOS 开发

iOS开发资料收集区

iOS 开发中...

ios 进阶

UI进价

iOS第三方资料收集