

```
1: //
2: //  RectangleView.h
3: //  Rectangle
4: //
5: //  Created by Gabriel Parriaux on 08.03.13.
6: //  Copyright (c) 2013 pxgab. All rights reserved.
7: //
8:
9: #import <UIKit/UIKit.h>
10:
11: @protocol RectangleViewDataSource;
12:
13: @interface RectangleView : UIView
14:
15: @property (nonatomic, weak) IBOutlet id <RectangleViewDataSource> dataSource;
16:
17: @end
18:
19: @protocol RectangleViewDataSource
20:
21: - (CGFloat)largeurPourRectangleView:(RectangleView *)sender;
22: - (CGFloat)hauteurPourRectangleView:(RectangleView *)sender;
23:
24: @end
```

```
1: //
2: // RectangleView.m
3: // Rectangle
4: //
5: // Created by Gabriel Parriaux on 08.03.13.
6: // Copyright (c) 2013 pxgab. All rights reserved.
7: //
8:
9: #import "RectangleView.h"
10:
11: @interface RectangleView ()
12:
13: @property (nonatomic) CGFloat largeurRectangle;
14: @property (nonatomic) CGFloat hauteurRectangle;
15:
16: @end
17:
18: @implementation RectangleView
19:
20: - (void)drawRect:(CGRect)rect
21: {
22:     self.hauteurRectangle = [self.dataSource hauteurPourRectangleView:self];
23:     if (self.hauteurRectangle > (self.bounds.size.height / 2 * .9)) {
24:         self.hauteurRectangle = (self.bounds.size.height / 2 * .9);
25:     }
26:     if (self.hauteurRectangle < (self.bounds.size.height / 2 * .9 * -1)) {
27:         self.hauteurRectangle = (self.bounds.size.height / 2 * .9 * -1);
28:     }
29:     self.largeurRectangle = [self.dataSource largeurPourRectangleView:self];
30:     if (self.largeurRectangle > (self.bounds.size.width / 2 * .9)) {
31:         self.largeurRectangle = (self.bounds.size.width / 2 * .9);
32:     }
33:     if (self.largeurRectangle < (self.bounds.size.width / 2 * .9 * -1)) {
34:         self.largeurRectangle = (self.bounds.size.width / 2 * .9 * -1);
35:     }
36:
37:     CGPoint pointOrigine;
38:     pointOrigine.x = self.bounds.origin.x + (self.bounds.size.width / 2);
39:     pointOrigine.y = self.bounds.origin.y + (self.bounds.size.height / 2);
40:
41:     CGRect rectangle = CGRectMake(pointOrigine.x, pointOrigine.y, self.largeurRectangle,
self.hauteurRectangle);
42:     CGContextRef context = UIGraphicsGetCurrentContext();
43:
44:     [[UIColor lightGrayColor] setFill];
45:
46:     CGContextBeginPath(context);
47:     CGContextAddRect(context, rectangle);
48:     CGContextFillRect(context, rectangle);
49:
50: }
51:
52:
53: @end
```

```
1: //
2: // RectangleViewController.h
3: // Rectangle
4: //
5: // Created by Gabriel Parriaux on 08.03.13.
6: // Copyright (c) 2013 pxgab. All rights reserved.
7: //
8:
9: #import <UIKit/UIKit.h>
10: #import "RectangleView.h"
11:
12: @interface RectangleViewController : UIViewController
13:
14:     // modèle: deux valeurs entières
15: @property (nonatomic) int largeur;
16: @property (nonatomic) int hauteur;
17:
18: @end
```

```
1: //
2: // RectangleViewController.m
3: // Rectangle
4: //
5: // Created by Gabriel Parriaux on 08.03.13.
6: // Copyright (c) 2013 pxgab. All rights reserved.
7: //
8:
9: #import "RectangleViewController.h"
10:
11: @interface RectangleViewController () <RectangleViewDataSource>
12:
13: @property (weak, nonatomic) IBOutlet RectangleView *rectangleView;
14:
15: @end
16:
17: @implementation RectangleViewController
18:
19: - (void)setRectangleView:(RectangleView *)rectangleView
20: {
21:     _rectangleView = rectangleView;
22:     self.rectangleView.dataSource = self;
23:     [self.rectangleView addGestureRecognizer:[[UIPanGestureRecognizer alloc]
initWithTarget:self action:@selector(gererRectangleViewGesture:)]];
24: }
25:
26: #pragma mark -
27: #pragma mark setters du modèle
28:
29: - (void)setLargeur:(int)largeur
30: {
31:     _largeur = largeur;
32:     NSLog(@"largeur = %d", self.largeur);
33:     [self.rectangleView setNeedsDisplay];
34: }
35:
36: - (void)setHauteur:(int)hauteur
37: {
38:     _hauteur = hauteur;
39:     NSLog(@"hauteur = %d", self.hauteur);
40:     [self.rectangleView setNeedsDisplay];
41: }
42:
43: #pragma mark -
44: #pragma mark delegate
45:
46: - (CGFloat)hauteurPourRectangleView:(RectangleView *)sender
47: {
48:     return (CGFloat)self.hauteur;
49: }
50:
51: - (CGFloat)largeurPourRectangleView:(RectangleView *)sender
52: {
53:     return (CGFloat)self.largeur;
54: }
55:
56: #pragma mark -
57: #pragma mark target pour recognizer
58:
59: - (void)gererRectangleViewGesture:(UIPanGestureRecognizer *)gesture
60: {
61:     if (gesture.state == UIGestureRecognizerStateChanged || gesture.state ==
UIGestureRecognizerStateEnded) {
62:         CGPoint translation = [gesture translationInView:self.rectangleView];
63:         self.largeur += translation.x * 1.5;
64:         self.hauteur += translation.y * 1.5;
65:         [gesture setTranslation:CGPointZero inView:self.rectangleView];
66:     }
67: }
68:
69: @end
```