

```
1: //
2: // HappinessViewController.h
3: // Happiness
4: //
5: // Created by Gabriel Parriaux on 11.10.12.
6: // Copyright (c) 2012 gymo. All rights reserved.
7: //
8:
9: #import <UIKit/UIKit.h>
10:
11: @interface HappinessViewController : UIViewController
12:
13: @property (nonatomic) int happiness; // 0 est triste, 100 est très heureux
14:
15:
16:
17: @end
```

```
//
// HappinessViewController.m
// Happiness
//
// Created by Gabriel Parriaux on 11.10.12.
// Copyright (c) 2012 gymo. All rights reserved.
//

#import "HappinessViewController.h"
#import "FaceView.h"

@interface HappinessViewController () <FaceViewDataSource>

@property (nonatomic, weak) IBOutlet FaceView *faceView;

@end

@implementation HappinessViewController

@synthesize happiness = _happiness;
@synthesize faceView = _faceView;

- (void)setFaceView:(FaceView *)faceView
{
    _faceView = faceView;
    [self.faceView addGestureRecognizer:[UIPinchGestureRecognizer alloc]
initWithTarget:self.faceView action:@selector(pinch:)]];
    [self.faceView addGestureRecognizer:[UIPanGestureRecognizer alloc] initWithTarget:self
action:@selector(gererHappinessGesture:)]];
    self.faceView.dataSource = self;
}

- (void)setHappiness:(int)happiness
{
    _happiness = happiness;
    [self.faceView setNeedsDisplay];
}

- (void)gererHappinessGesture:(UIPanGestureRecognizer *)gesture
{
    if ((gesture.state == UIGestureRecognizerStateChanged || gesture.state ==
UIGestureRecognizerStateEnded)) {
        CGPoint translation = [gesture translationInView:self.faceView];
        // NSLog(@"translation.y = %.2f\n translation.x = %.2f", translation.y, translation.x);
        self.happiness -= translation.y / 2;
        [gesture setTranslation:CGPointZero inView:self.faceView];
    }
}

- (float)smileForFaceView:(FaceView *)sender
{
    return (self.happiness - 50) / 50.0;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)toInterfaceOrientation
{
    return YES;
}

@end
```

```
1: //
2: // FaceView.h
3: // Happiness
4: //
5: // Created by Gabriel Parriaux on 11.10.12.
6: // Copyright (c) 2012 gymo. All rights reserved.
7: //
8:
9: #import <UIKit/UIKit.h>
10:
11: @protocol FaceViewDataSource;
12:
13: @interface FaceView : UIView
14:
15: @property (nonatomic) CGFloat scale;
16:
17: - (void)pinch:(UIPinchGestureRecognizer *)gesture;
18:
19: @property (weak, nonatomic) IBOutlet id <FaceViewDataSource> dataSource;
20:
21: @end
22:
23: @protocol FaceViewDataSource
24:
25: - (float)smileForFaceView:(FaceView *)sender;
26:
27: @end
```

```
1: //
2: // FaceView.m
3: // Happiness
4: //
5: // Created by Gabriel Parriaux on 11.10.12.
6: // Copyright (c) 2012 gymo. All rights reserved.
7: //
8:
9: #import "FaceView.h"
10:
11: @implementation FaceView
12:
13: @synthesize scale = _scale;
14:
15: @synthesize dataSource = _dataSource;
16:
17: #define DEFAULT_SCALE 0.90
18:
19: - (CGFloat)scale
20: {
21:     if (!_scale) {
22:         return DEFAULT_SCALE;
23:     } else return _scale;
24: }
25:
26: - (void)setScale:(CGFloat)scale
27: {
28:     if (_scale != scale) {
29:         _scale = scale;
30:         [self setNeedsDisplay];
31:     }
32: }
33:
34: - (void)pinch:(UIPinchGestureRecognizer *)gesture
35: {
36:     if (gesture.state == UIGestureRecognizerStateChanged || gesture.state ==
    UIGestureRecognizerStateEnded) {
37:         self.scale *= gesture.scale;
38:         gesture.scale = 1;
39:     }
40: }
41:
42: - (void)drawCircleAtPoint:(CGPoint)p
43:     withRadius:(CGFloat)radius
44:     inContext:(CGContextRef)context
45: {
46:     UIGraphicsPushContext(context);
47:
48:     CGContextBeginPath(context);
49:     CGContextAddArc(context, p.x, p.y, radius, 0, 2*M_PI, YES);
50:     CGContextStrokePath(context);
51:
52:     UIGraphicsPopContext();
53: }
54:
55:
56: - (void)drawRect:(CGRect)rect
57: {
58:     CGContextRef context = UIGraphicsGetCurrentContext();
59:
60:     // dessiner la figure (un cercle)
61:     CGPoint midPoint;
62:     midPoint.x = self.bounds.origin.x + self.bounds.size.width / 2;
63:     midPoint.y = self.bounds.origin.y + self.bounds.size.height / 2;
64:
65:     CGFloat size = self.bounds.size.width / 2;
66:
67:     if (self.bounds.size.height < self.bounds.size.width) size = self.bounds.size.height / 2;
68:
69:     size *= self.scale;
70:
71:     CGContextSetLineWidth(context, 5.0);
72:     [[UIColor greenColor] setStroke];
73:
```

```
74:     [self drawCircleAtPoint:midPoint withRadius:size inContext:context];
75:
76:     // dessiner deux yeux (deux cercles)
77:
78:     #define EYE_H 0.35
79:     #define EYE_V 0.35
80:     #define EYE_RADIUS 0.10
81:
82:     CGPoint eyePoint;
83:     eyePoint.x = midPoint.x - size * EYE_H;
84:     eyePoint.y = midPoint.y - size * EYE_V;
85:
86:     [self drawCircleAtPoint:eyePoint withRadius:size * EYE_RADIUS inContext:context];
87:
88:     eyePoint.x += size * EYE_H * 2;
89:
90:     [self drawCircleAtPoint:eyePoint withRadius:size * EYE_RADIUS inContext:context];
91:
92:
93:     // pas de nez
94:
95:     // dessiner la bouche (courbe de Bézier)
96:     #define MOUTH_H 0.45
97:     #define MOUTH_V 0.40
98:     #define MOUTH_SMILE 0.25
99:
100:    CGPoint mouthStart;
101:    mouthStart.x = midPoint.x - MOUTH_H * size;
102:    mouthStart.y = midPoint.y + MOUTH_V * size;
103:
104:    CGPoint mouthEnd = mouthStart;
105:    mouthEnd.x += MOUTH_H * size * 2;
106:
107:    CGPoint mouthCP1 = mouthStart;
108:    mouthCP1.x += MOUTH_H * size * 2/3;
109:    CGPoint mouthCP2 = mouthEnd;
110:    mouthCP2.x -= MOUTH_H * size * 2/3;
111:
112:    // on veut que smile soit fixé entre -1.0 et 1.0
113:    float smile = [self.dataSource smileForFaceView:self];
114:    if (smile > 1) smile = 1;
115:    if (smile < -1) smile = -1;
116:
117:    CGFloat smileOffset = MOUTH_SMILE * size * smile;
118:
119:    mouthCP1.y += smileOffset;
120:    mouthCP2.y += smileOffset;
121:
122:    CGContextBeginPath(context);
123:    CGContextMoveToPoint(context, mouthStart.x, mouthStart.y);
124:    CGContextAddCurveToPoint(context, mouthCP1.x, mouthCP1.y, mouthCP2.x, mouthCP2.y,
mouthEnd.x, mouthEnd.y);
125:    CGContextStrokePath(context);
126:
127: }
128:
129:
130:
131:
132: @end
```