

```
1. //
2. // Account.h
3. // ATM
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Simulates an ATM (that only accepts deposits).
10. //
11.
12. #import <Foundation/Foundation.h>
13.
14. @interface Account : NSObject {
15. }
16.
17. @property (assign, nonatomic) unsigned long long balance;
18.
19. @end
```

```
1. //
2. // Account.m
3. // ATM
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Simulates an ATM (that only accepts deposits).
10. //
11.
12. #import "Account.h"
13.
14. @implementation Account
15.
16. @synthesize balance=_balance;
17.
18. - (id)init
19. {
20.     if ((self = [super init])) {
21.         self.balance = 0;
22.     }
23.     return self;
24. }
25.
26. @end
```

```
1. //
2. // Controller.h
3. // ATM
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Simulates an ATM (that only accepts deposits).
10. //
11.
12. #import <Foundation/Foundation.h>
13. #import "Account.h"
14.
15. @interface Controller : NSObject {
16. }
17.
18. @property (nonatomic, retain) Account *account;
19. @property (assign, nonatomic) unsigned long long amount;
20. @property (nonatomic, retain) IBOutlet UILabel *balanceLabel;
21. @property (nonatomic, retain) IBOutlet UILabel *depositLabel;
22.
23. - (IBAction)clear:(id)sender;
24. - (IBAction)deposit:(id)sender;
25. - (IBAction)digit:(id)sender;
26. - (void)show;
27.
28. @end
```

```
1. //
2. // Controller.m
3. // ATM
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Simulates an ATM (that only accepts deposits).
10. //
11.
12. #import "Controller.h"
13.
14. @implementation Controller
15.
16. @synthesize account=_account;
17. @synthesize amount=_amount;
18. @synthesize balanceLabel=_balanceLabel;
19. @synthesize depositLabel=_depositLabel;
20.
21. - (void)awakeFromNib
22. {
23.     // create account
24.     Account *account = [[Account alloc] init];
25.     self.account = account;
26.     [account release];
27.
28.     // show UI
29.     [self show];
30. }
31.
32. - (IBAction)clear:(id)sender
33. {
34.     // clear input
35.     self.amount = 0;
36.     [self show];
37. }
38.
39. - (IBAction)deposit:(id)sender
40. {
41.     // deposit amount
42.     self.account.balance += self.amount;
43.
44.     // clear input
45.     self.amount = 0;
46.     [self show];
47. }
48.
```

```
49. - (IBAction)digit:(id)sender
50. {
51.     // append digit to amount
52.     UIButton *b = (UIButton *)sender;
53.     self.amount = self.amount * 10 + b.tag;
54.     [self show];
55. }
56.
57. - (void)show
58. {
59.     // show balance
60.     self.balanceLabel.text = [NSString stringWithFormat:@"%llu", self.account.balance];
61.
62.     // show input
63.     self.depositLabel.text = [NSString stringWithFormat:@"%llu", self.amount];
64. }
65.
66. - (void)dealloc
67. {
68.     [_account release];
69.     [_balanceLabel release];
70.     [_depositLabel release];
71.     [super dealloc];
72. }
73.
74. @end
```

```
1. //
2. // GesturesViewController.h
3. // Gestures
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates gestures.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @interface GesturesViewController : UIViewController <UIAlertViewDelegate> {
15. }
16.
17. @property (assign, nonatomic) BOOL alertInProgress;
18. @property (nonatomic, retain) IBOutlet UIImageView *imageView;
19. @property (assign, nonatomic) int index;
20. @property (nonatomic, retain) NSArray *tommies;
21.
22. - (void)alertView:(UIAlertView *)alertView didDismissWithButtonIndex:(NSInteger)buttonIndex;
23. - (void)handleLongPress:(UILongPressGestureRecognizer *)sender;
24. - (void)handleSwipe:(UISwipeGestureRecognizer *)sender;
25.
26.
27. @end
```

```
1. //
2. // GesturesViewController.m
3. // Gestures
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates gestures.
10. //
11.
12. #import "GesturesViewController.h"
13.
14. @implementation GesturesViewController
15.
16. @synthesize alertInProgress=_alertInProgress;
17. @synthesize imageView=_imageView;
18. @synthesize index=_index;
19. @synthesize tommies=_tommies;
20.
21. - (void)alertView:(UIAlertView *)alertView didDismissWithButtonIndex:(NSInteger)buttonIndex
22. {
23.     self.alertInProgress = NO;
24. }
25.
26. - (void)handleLongPress:(UILongPressGestureRecognizer *)sender
27. {
28.     if (!self.alertInProgress) {
29.         self.alertInProgress = YES;
30.         UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Hey!"
31.                                                                    message:@"Stop that."
32.                                                                    delegate:self
33.                                                                    cancelButtonTitle:@"Fine"
34.                                                                    otherButtonTitles:nil];
35.         [alertView show];
36.         [alertView release];
37.     }
38. }
39.
40. - (void)handleSwipe:(UISwipeGestureRecognizer *)sender
41. {
42.     // handle swipe
43.     UISwipeGestureRecognizerDirection direction = [(UISwipeGestureRecognizer *)sender direction];
44.     switch (direction) {
45.
46.         // ignore up, down
47.         case UISwipeGestureRecognizerDirectionUp:
48.         case UISwipeGestureRecognizerDirectionDown:
```

```
49.         break;
50.
51.         // left
52.     case UISwipeGestureRecognizerDirectionLeft:
53.         self.index = (self.index + 1) % [self.tommies count];
54.         break;
55.
56.         // right
57.     case UISwipeGestureRecognizerDirectionRight:
58.         self.index = (self.index + [self.tommies count] - 1) % [self.tommies count];
59.         break;
60.     }
61.
62.     // update Tommy
63.     self.imageView.image = [UIImage imageNamed:[self.tommies objectAtIndex:self.index]];
64. }
65.
66. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
67. {
68.     return (interfaceOrientation == UIInterfaceOrientationPortrait);
69. }
70.
71. - (void)viewDidLoad
72. {
73.     [super viewDidLoad];
74.
75.     // keep track of alerts
76.     self.alertInProgress = NO;
77.
78.     // prepare Tommies
79.     self.tommies = [[NSArray alloc] initWithObjects:@"tommy1.jpg", @"tommy2.jpg", @"tommy3.jpg", nil];
80.     self.index = 0;
81.
82.     // listen for long press
83.     UILongPressGestureRecognizer *longpressGesture = [[UILongPressGestureRecognizer alloc] initWithTarget:self action:@selector(
handleLongPress:)];
84.     [self.imageView addGestureRecognizer:longpressGesture];
85.     [longpressGesture release];
86.
87.     // listen for right swipe
88.     UISwipeGestureRecognizer *swipeGesture = [[UISwipeGestureRecognizer alloc] initWithTarget:self action:@selector(handleSwipe:)];
89.     swipeGesture.direction = UISwipeGestureRecognizerDirectionRight;
90.     [self.imageView addGestureRecognizer:swipeGesture];
91.     [swipeGesture release];
92.
93.     // listen for left swipe
94.     swipeGesture = [[UISwipeGestureRecognizer alloc] initWithTarget:self action:@selector(handleSwipe:)];
95.     swipeGesture.direction = UISwipeGestureRecognizerDirectionLeft;
```

```
96.     [self.imageView addGestureRecognizer:swipeGesture];
97.     [swipeGesture release];
98. }
99.
100. - (void)dealloc
101. {
102.     [_imageView release];
103.     [_tommies release];
104.     [super dealloc];
105. }
106.
107. @end
```

```
1. //
2. // RootViewController.h
3. // Plist
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates property lists.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @interface RootViewController : UITableViewController {
15. }
16.
17. @property (nonatomic, retain) NSArray *words;
18.
19. @end
```

```
1. //
2. // RootViewController.m
3. // Plist
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates property lists.
10. //
11.
12. #import "RootViewController.h"
13.
14. @implementation RootViewController
15.
16. @synthesize words=_words;
17.
18. - (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
19. {
20.     return 1;
21. }
22.
23. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
24. {
25.     return (interfaceOrientation == UIInterfaceOrientationPortrait);
26. }
27.
28. - (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
29. {
30.     // allocate cell, reusing if possible
31.     static NSString *CellIdentifier = @"Cell";
32.     UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];
33.     if (cell == nil) {
34.         cell = [[[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier] autorelease];
35.     }
36.
37.     // configure cell
38.     cell.selectionStyle = UITableViewCellSelectionStyleNone;
39.     cell.textLabel.text = [self.words objectAtIndex:indexPath.row];
40.
41.     return cell;
42. }
43.
44. - (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
45. {
46.     return [self.words count];
47. }
48.
```

```
49. - (void)viewDidLoad
50. {
51.     [super viewDidLoad];
52.
53.     // load words
54.     NSString *path = [[NSBundle mainBundle] pathForResource:@"words" ofType:@"plist"];
55.     NSArray *words = [[NSArray alloc] initWithContentsOfFile:path];
56.     self.words = words;
57.     [words release];
58. }
59.
60. @end
```

```
1. //
2. // RootViewController.h
3. // Sqlite
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates SQLite.
10. //
11.
12. #import <UIKit/UIKit.h>
13. #import "sqlite3.h"
14.
15. @interface RootViewController : UITableViewController {
16. }
17.
18. @property (nonatomic, retain) NSArray *words;
19.
20. @end
```

```
1. //
2. // RootViewController.m
3. // Sqlite
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates SQLite.
10. //
11.
12. #import "RootViewController.h"
13.
14. @implementation RootViewController
15.
16. @synthesize words=_words;
17.
18. - (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
19. {
20.     return 1;
21. }
22.
23. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
24. {
25.     return (interfaceOrientation == UIInterfaceOrientationPortrait);
26. }
27.
28. - (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
29. {
30.     // allocate cell, reusing if possible
31.     static NSString *CellIdentifier = @"Cell";
32.     UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];
33.     if (cell == nil) {
34.         cell = [[[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier] autorelease];
35.     }
36.
37.     // configure cell
38.     cell.selectionStyle = UITableViewCellSelectionStyleNone;
39.     cell.textLabel.text = [self.words objectAtIndex:indexPath.row];
40.
41.     return cell;
42. }
43.
44. - (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
45. {
46.     return [self.words count];
47. }
48.
```

```
49. - (void)viewDidLoad
50. {
51.     [super viewDidLoad];
52.
53.     // prepare for words
54.     NSMutableArray *words = [[NSMutableArray alloc] init];
55.
56.     // connect to SQLite database
57.     sqlite3 *db;
58.     NSString *path = [[NSBundle mainBundle] pathForResource:@"words" ofType:@"sqlite"];
59.     sqlite3_open([path UTF8String], &db);
60.
61.     // select four-letter words words
62.     NSString *sql = @"SELECT word FROM words WHERE LENGTH(word) = 4";
63.     sqlite3_stmt *statement;
64.     sqlite3_prepare_v2(db, [sql UTF8String], -1, &statement, nil);
65.
66.     // iterate over results
67.     while (sqlite3_step(statement) == SQLITE_ROW) {
68.         char *c = (char *) sqlite3_column_text(statement, 0);
69.         NSString *s = [[NSString alloc] initWithUTF8String:c];
70.         [words addObject:s];
71.         [s release];
72.     }
73.
74.     // close database
75.     sqlite3_close(db);
76.
77.     // retain words
78.     self.words = words;
79.     [words release];
80. }
81.
82. @end
```

```
1. //
2. // FirstViewController.h
3. // TabBar
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Tab Bar Application template.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14.
15. @interface FirstViewController : UIViewController {
16.
17. }
18.
19. @end
```

```
1. //
2. // FirstViewController.m
3. // TabBar
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Tab Bar Application template.
10. //
11.
12. #import "FirstViewController.h"
13.
14.
15. @implementation FirstViewController
16.
17. /*
18. // Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
19. - (void)viewDidLoad
20. {
21.     [super viewDidLoad];
22. }
23. */
24.
25. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
26. {
27.     // Return YES for supported orientations
28.     return (interfaceOrientation == UIInterfaceOrientationPortrait);
29. }
30.
31.
32. - (void)didReceiveMemoryWarning
33. {
34.     // Releases the view if it doesn't have a superview.
35.     [super didReceiveMemoryWarning];
36.
37.     // Release any cached data, images, etc. that aren't in use.
38. }
39.
40.
41. - (void)viewDidUnload
42. {
43.     [super viewDidUnload];
44.
45.     // Release any retained subviews of the main view.
46.     // e.g. self.myOutlet = nil;
47. }
48.
```

```
49.  
50. - (void)dealloc  
51. {  
52.     [super dealloc];  
53. }  
54.  
55. @end
```

```
1. //
2. // SecondViewController.h
3. // TabBar
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Tab Bar Application template.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14.
15. @interface SecondViewController : UIViewController {
16.
17. }
18.
19. @end
```

```
1. //
2. // SecondViewController.m
3. // TabBar
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Tab Bar Application template.
10. //
11.
12. #import "SecondViewController.h"
13.
14.
15. @implementation SecondViewController
16.
17. /*
18. // Implement viewDidLoad to do additional setup after loading the view, typically from a nib.
19. - (void)viewDidLoad
20. {
21.     [super viewDidLoad];
22. }
23. */
24.
25. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
26. {
27.     // Return YES for supported orientations
28.     return (interfaceOrientation == UIInterfaceOrientationPortrait);
29. }
30.
31.
32. - (void)didReceiveMemoryWarning
33. {
34.     // Releases the view if it doesn't have a superview.
35.     [super didReceiveMemoryWarning];
36.
37.     // Release any cached data, images, etc. that aren't in use.
38. }
39.
40.
41. - (void)viewDidUnload
42. {
43.     [super viewDidUnload];
44.
45.     // Release any retained subviews of the main view.
46.     // e.g. self.myOutlet = nil;
47. }
48.
```

```
49.  
50. - (void)dealloc  
51. {  
52.     [super dealloc];  
53. }  
54.  
55. @end
```

```
1. //
2. // TransformationsViewController.h
3. // Transformations
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates affine transformations of Tommy.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @interface TransformationsViewController : UIViewController {
15.     CGPoint _translation;
16. }
17.
18. @property (nonatomic, retain) IBOutlet UIImageView *imageView;
19. @property (assign, nonatomic) float scale;
20.
21. - (void)handlePan:(UIPanGestureRecognizer *)sender;
22. - (void)handlePinch:(UIPinchGestureRecognizer *)sender;
23.
24. @end
```

```
1. //
2. // TransformationsViewController.m
3. // Transformations
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates affine transformations of Tommy.
10. //
11.
12. #import "TransformationsViewController.h"
13.
14. @implementation TransformationsViewController
15.
16. @synthesize imageView=_imageView;
17. @synthesize scale=_scale;
18.
19. - (void)handlePan:(UIPanGestureRecognizer *)sender
20. {
21.     // translate Tommy
22.     CGPoint translation = [sender translationInView:self.imageView];
23.     CGAffineTransform scale = CGAffineTransformMakeScale(self.scale, self.scale);
24.     CGAffineTransform translate = CGAffineTransformMakeTranslation(_translation.x + translation.x * self.scale, _translation.y + translation.y
 * self.scale);
25.     sender.view.transform = CGAffineTransformConcat(scale, translate);
26.
27.     // remember translation once done panning
28.     if (sender.state == UIGestureRecognizerStateEnded) {
29.         _translation.x += translation.x * self.scale;
30.         _translation.y += translation.y * self.scale;
31.     }
32. }
33.
34. - (void)handlePinch:(UIPinchGestureRecognizer *)sender
35. {
36.     // scale Tommy
37.     CGFloat factor = [sender scale];
38.     CGAffineTransform scale = CGAffineTransformMakeScale(self.scale * factor, self.scale * factor);
39.     CGAffineTransform translate = CGAffineTransformMakeTranslation(_translation.x, _translation.y);
40.     sender.view.transform = CGAffineTransformConcat(scale, translate);
41.
42.     // remember scale once done pinching
43.     if (sender.state == UIGestureRecognizerStateEnded) {
44.         self.scale *= factor;
45.     }
46. }
47.
```

```
48. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
49. {
50.     return (interfaceOrientation == UIInterfaceOrientationPortrait);
51. }
52.
53. - (void)viewDidLoad
54. {
55.     [super viewDidLoad];
56.
57.     // initial scale
58.     self.scale = 1.0;
59.
60.     // initial translation
61.     _translation.x = 0.0;
62.     _translation.y = 0.0;
63.
64.     // listen for pan
65.     UIPanGestureRecognizer *panGesture = [[UIPanGestureRecognizer alloc] initWithTarget:self action:@selector(handlePan:)];
66.     [self.imageView addGestureRecognizer:panGesture];
67.     [panGesture release];
68.
69.     // listen for pinch
70.     UIPinchGestureRecognizer *pinchGesture = [[UIPinchGestureRecognizer alloc] initWithTarget:self action:@selector(handlePinch:)];
71.     [self.imageView addGestureRecognizer:pinchGesture];
72.     [pinchGesture release];
73. }
74.
75. - (void)dealloc
76. {
77.     [_imageView release];
78.     [super dealloc];
79. }
80.
81. @end
```