

```
1. //
2. //  CiaolAppDelegate.h
3. //  Ciaol
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates localized nibs.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @class CiaolViewController;
15.
16. @interface CiaolAppDelegate : NSObject <UIApplicationDelegate> {
17. }
18.
19. @property (nonatomic, retain) IBOutlet CiaolViewController *viewController;
20. @property (nonatomic, retain) IBOutlet UIWindow *window;
21.
22. @end
```

```
1. //
2. //  CiaolAppDelegate.m
3. //  Ciaol
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates localized nibs.
10. //
11.
12. #import "CiaolAppDelegate.h"
13.
14. #import "CiaolViewController.h"
15.
16. @implementation CiaolAppDelegate
17.
18. @synthesize viewController=_viewController;
19. @synthesize window=_window;
20.
21. - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
22. {
23.     self.window.rootViewController = self.viewController;
24.     [self.window makeKeyAndVisible];
25.     return YES;
26. }
27.
28.
29. - (void)dealloc
30. {
31.     [_viewController release];
32.     [_window release];
33.     [super dealloc];
34. }
35.
36. @end
```

```
1. //
2. //  CiaoViewController.h
3. //  Ciao1
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates localized nibs.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @interface CiaoViewController : UIViewController {
15. }
16.
17. @end
```

```
1. //
2. //  CiaoViewController.m
3. //  Ciao1
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates localized nibs.
10. //
11.
12. #import "CiaoViewController.h"
13.
14. @implementation CiaoViewController
15.
16. @end
```

```
1. //
2. //  Ciao2AppDelegate.h
3. //  Ciao2
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates localized strings.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @class Ciao2ViewController;
15.
16. @interface Ciao2AppDelegate : NSObject <UIApplicationDelegate> {
17. }
18.
19. @property (nonatomic, retain) IBOutlet Ciao2ViewController *viewController;
20. @property (nonatomic, retain) IBOutlet UIWindow *window;
21.
22. @end
```

```
1. //
2. //  Ciao2AppDelegate.m
3. //  Ciao2
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates localized strings.
10. //
11.
12. #import "Ciao2AppDelegate.h"
13.
14. #import "Ciao2ViewController.h"
15.
16. @implementation Ciao2AppDelegate
17.
18. @synthesize viewController=_viewController;
19. @synthesize window=_window;
20.
21. - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
22. {
23.     self.window.rootViewController = self.viewController;
24.     [self.window makeKeyAndVisible];
25.     return YES;
26. }
27.
28. - (void)dealloc
29. {
30.     [_window release];
31.     [_viewController release];
32.     [super dealloc];
33. }
34.
35. @end
```

```
1. //
2. //  Ciao2ViewController.h
3. //  Ciao2
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates localized strings.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @interface Ciao2ViewController : UIViewController {
15. }
16.
17. @property (nonatomic, retain) IBOutlet UILabel *label;
18.
19. @end
```

```
1. //
2. //  Ciao2ViewController.m
3. //  Ciao2
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates localized strings.
10. //
11.
12. #import "Ciao2ViewController.h"
13.
14. @implementation Ciao2ViewController
15.
16. @synthesize label=_label;
17.
18. - (void)dealloc
19. {
20.     [_label release];
21.     [super dealloc];
22. }
23.
24. - (void)viewDidLoad
25. {
26.     [super viewDidLoad];
27.
28.     // localize greeting
29.     self.label.text = NSLocalizedString(@"GREETING", @"greeting for user");
30. }
31.
32. - (void)viewDidUnload
33. {
34.     self.label = nil;
35.     [super viewDidUnload];
36. }
37.
38. @end
```



```
1. //
2. // CoreDataAppDelegate.h
3. // CoreData
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates Core Data.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @interface CoreDataAppDelegate : NSObject <UIApplicationDelegate> {
15. }
16.
17. @property (nonatomic, retain, readonly) NSManagedObjectContext *managedObjectContext;
18. @property (nonatomic, retain, readonly) NSManagedObjectContext *managedObjectContext;
19. @property (nonatomic, retain, readonly) NSPersistentStoreCoordinator *persistentStoreCoordinator;
20. @property (nonatomic, retain) IBOutlet UIWindow *window;
21.
22. - (NSURL *)applicationDocumentsDirectory;
23. - (void)saveContext;
24.
25. @end
```

```
1. //
2. // CoreDataAppDelegate.m
3. // CoreData
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates Core Data.
10. //
11.
12. #import "CoreDataAppDelegate.h"
13.
14. @implementation CoreDataAppDelegate
15.
16. @synthesize managedObjectContext=__managedObjectContext;
17. @synthesize managedObjectModel=__managedObjectModel;
18. @synthesize persistentStoreCoordinator=__persistentStoreCoordinator;
19. @synthesize window=_window;
20.
21. - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
22. {
23.     NSManagedObjectContext *context = [self managedObjectContext];
24.
25.     // Alice
26.     NSManagedObject *alice = [NSEntityDescription
27.                               insertNewObjectForEntityForName:@"Student"
28.                               inManagedObjectContext:context];
29.     [alice setValue:@"Alice" forKey:@"name"];
30.     [alice setValue:@"Computer Science" forKey:@"concentration"];
31.
32.     // Bob
33.     NSManagedObject *bob = [NSEntityDescription
34.                              insertNewObjectForEntityForName:@"Student"
35.                              inManagedObjectContext:context];
36.     [bob setValue:@"Bob" forKey:@"name"];
37.     [bob setValue:@"Computer Science" forKey:@"concentration"];
38.
39.     // Computer Science E-76
40.     NSManagedObject *course = [NSEntityDescription
41.                                 insertNewObjectForEntityForName:@"Course"
42.                                 inManagedObjectContext:context];
43.     [course setValue:@"Computer Science E-76" forKey:@"title"];
44.     [course setValue:[NSSet setWithObjects:alice,bob,nil] forKey:@"students"];
45.     NSError *error;
46.     if (![context save:&error]) {
47.         NSLog(@"%@", [error localizedDescription]);
48.     }
```

```
49.
50. // iterate over courses
51. NSFetchRequest *fetchRequest = [[NSFetchRequest alloc] init];
52. NSEntityDescription *entity = [NSEntityDescription
53.     entityForName:@"Course" inManagedObjectContext:context];
54. [fetchRequest setEntity:entity];
55. NSArray *fetchedObjects = [context executeFetchRequest:fetchRequest error:&error];
56. for (NSManagedObject *info in fetchedObjects) {
57.
58.     // output course's title
59.     NSLog(@"%@", [info valueForKey:@"title"]);
60.
61.     // iterate over students
62.     NSSet *students = [info valueForKey:@"students"];
63.     for (NSManagedObject *student in students) {
64.
65.         // output student's name and concentration
66.         NSLog(@"- %@ (%@)", [student valueForKey:@"name"], [student valueForKey:@"concentration"]);
67.     }
68. }
69. [fetchRequest release];
70.
71. [self.window makeKeyAndVisible];
72. return YES;
73. }
74.
75. - (void)applicationWillTerminate:(UIApplication *)application
76. {
77.     [self saveContext];
78. }
79.
80. - (void)dealloc
81. {
82.     [__managedObjectContext release];
83.     [__managedObjectModel release];
84.     [__persistentStoreCoordinator release];
85.     [_window release];
86.     [super dealloc];
87. }
88.
89. - (void)saveContext
90. {
91.     NSError *error = nil;
92.     NSManagedObjectContext *managedObjectContext = self.managedObjectContext;
93.     if (managedObjectContext != nil)
94.     {
95.         if ([managedObjectContext hasChanges] && ![managedObjectContext save:&error])
96.         {
```

```
97.         NSLog(@"Unresolved error %@, %@", error, [error userInfo]);
98.         abort();
99.     }
100. }
101. }
102.
103. #pragma mark - Core Data stack
104.
105. /**
106.  Returns the managed object context for the application.
107.  If the context doesn't already exist, it is created and bound to the persistent store coordinator for the application.
108.  */
109. - (NSManagedObjectContext *)managedObjectContext
110. {
111.     if (__managedObjectContext != nil)
112.     {
113.         return __managedObjectContext;
114.     }
115.
116.     NSPersistentStoreCoordinator *coordinator = [self persistentStoreCoordinator];
117.     if (coordinator != nil)
118.     {
119.         __managedObjectContext = [[NSManagedObjectContext alloc] init];
120.         [__managedObjectContext setPersistentStoreCoordinator:coordinator];
121.     }
122.     return __managedObjectContext;
123. }
124.
125. /**
126.  Returns the managed object model for the application.
127.  If the model doesn't already exist, it is created from the application's model.
128.  */
129. - (NSManagedObjectModel *)managedObjectModel
130. {
131.     if (__managedObjectModel != nil)
132.     {
133.         return __managedObjectModel;
134.     }
135.     NSURL *modelURL = [[NSBundle mainBundle] URLForResource:@"CoreData" withExtension:@"momd"];
136.     __managedObjectModel = [[NSManagedObjectModel alloc] initWithContentsOfURL:modelURL];
137.     return __managedObjectModel;
138. }
139.
140. /**
141.  Returns the persistent store coordinator for the application.
142.  If the coordinator doesn't already exist, it is created and the application's store added to it.
143.  */
144. - (NSPersistentStoreCoordinator *)persistentStoreCoordinator
```

```
145. {
146.     if (__persistentStoreCoordinator != nil)
147.     {
148.         return __persistentStoreCoordinator;
149.     }
150.
151.     NSURL *storeURL = [[self applicationDocumentsDirectory] URLByAppendingPathComponent:@"CoreData.sqlite"];
152.
153.     NSError *error = nil;
154.     __persistentStoreCoordinator = [[NSPersistentStoreCoordinator alloc] initWithManagedObjectModel:[self managedObjectModel]];
155.     if (![__persistentStoreCoordinator addPersistentStoreWithType:NSSQLiteStoreType configuration:nil URL:storeURL options:nil error:&error])
156.     {
157.         /*
158.          * Replace this implementation with code to handle the error appropriately.
159.          *
160.          * abort() causes the application to generate a crash log and terminate. You should not use this function in a shipping application,
161.          * although it may be useful during development. If it is not possible to recover from the error, display an alert panel that instructs the user
162.          * to quit the application by pressing the Home button.
163.          *
164.          * Typical reasons for an error here include:
165.          * The persistent store is not accessible;
166.          * The schema for the persistent store is incompatible with current managed object model.
167.          * Check the error message to determine what the actual problem was.
168.          *
169.          * If the persistent store is not accessible, there is typically something wrong with the file path. Often, a file URL is pointing into
170.          * the application's resources directory instead of a writeable directory.
171.          *
172.          * If you encounter schema incompatibility errors during development, you can reduce their frequency by:
173.          * Simply deleting the existing store:
174.          * [[NSFileManager defaultManager] removeItemAtURL:storeURL error:nil]
175.          *
176.          * Performing automatic lightweight migration by passing the following dictionary as the options parameter:
177.          * [NSDictionary dictionaryWithObjectsAndKeys:[NSNumber numberWithInt:YES], NSMigratePersistentStoresAutomaticallyOption, [NSNumber
178.          * numberWithInt:YES], NSInferMappingModelAutomaticallyOption, nil];
179.          *
180.          * Lightweight migration will only work for a limited set of schema changes; consult "Core Data Model Versioning and Data Migration
181.          * Programming Guide" for details.
182.          *
183.          */
184.         NSLog(@"Unresolved error %@, %@", error, [error userInfo]);
185.         abort();
186.     }
187.     return __persistentStoreCoordinator;
188. }
```

#pragma mark - Application's Documents directory

```
188.  
189. /**  
190.  Returns the URL to the application's Documents directory.  
191.  */  
192. - (NSURL *)applicationDocumentsDirectory  
193. {  
194.     return [[[NSFileManager defaultManager] URLsForDirectory:NSDocumentDirectory inDomains:NSUserDomainMask] lastObject];  
195. }  
196.  
197. @end
```

```
1. //
2. // PaddleAppDelegate.h
3. // Paddle
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates Core Graphics with a rectangle.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @class PaddleViewController;
15.
16. @interface PaddleAppDelegate : NSObject <UIApplicationDelegate> {
17. }
18.
19. @property (nonatomic, retain) IBOutlet PaddleViewController *viewController;
20. @property (nonatomic, retain) IBOutlet UIWindow *window;
21.
22. @end
```

```
1. //
2. // PaddleAppDelegate.m
3. // Paddle
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates Core Graphics with a rectangle.
10. //
11.
12. #import "PaddleAppDelegate.h"
13. #import "PaddleViewController.h"
14.
15. @implementation PaddleAppDelegate
16.
17. @synthesize viewController=_viewController;
18. @synthesize window=_window;
19.
20. - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
21. {
22.     self.window.rootViewController = self.viewController;
23.     [self.window makeKeyAndVisible];
24.     return YES;
25. }
26.
27. - (void)dealloc
28. {
29.     [_window release];
30.     [_viewController release];
31.     [super dealloc];
32. }
33.
34. @end
```



```
1. //
2. // PaddleView.h
3. // Paddle
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates Core Graphics with a rectangle.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. @interface PaddleView : UIView {
15. }
16.
17. @end
```

```
1. //
2. // PaddleView.m
3. // Paddle
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates Core Graphics with a rectangle.
10. //
11.
12. #import "PaddleView.h"
13.
14. @implementation PaddleView
15.
16. - (void)drawRect:(CGRect)rect
17. {
18.     CGRect square = CGRectMake(0.0f, 0.0f, 10.0f, 60.0f);
19.     [[UIColor whiteColor] set];
20.     UIRectFill(square);
21. }
22.
23. @end
```

```
1. //
2. // PaddleViewController.h
3. // Paddle
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates Core Graphics with a rectangle.
10. //
11.
12. #import <UIKit/UIKit.h>
13. #import "PaddleView.h"
14.
15. @interface PaddleViewController : UIViewController {
16. }
17.
18. @property (nonatomic, retain) PaddleView *paddleView;
19.
20. @end
```

```
1. //
2. // PaddleViewController.m
3. // Paddle
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates Core Graphics with a rectangle.
10. //
11.
12. #import "PaddleView.h"
13. #import "PaddleViewController.h"
14.
15. @implementation PaddleViewController
16.
17. @synthesize paddleView=_paddleView;
18.
19. - (void)dealloc
20. {
21.     [_paddleView release];
22.     [super dealloc];
23. }
24.
25. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
26. {
27.     return (interfaceOrientation == UIInterfaceOrientationLandscapeRight);
28. }
29.
30. - (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event
31. {
32.     // follow user's finger vertically with paddle
33.     UITouch *touch = [[event allTouches] anyObject];
34.     CGPoint location = [touch locationInView:self.view];
35.     self.paddleView.center = CGPointMake(self.paddleView.center.x, location.y);
36. }
37.
38. - (void)viewDidLoad
39. {
40.     [super viewDidLoad];
41.
42.     // add paddle
43.     self.paddleView = [[PaddleView alloc] initWithFrame:CGRectMake(10.0f, 10.0f, 10.0f, 60.0f)];
44.     [self.view addSubview:self.paddleView];
45.     [self.paddleView release];
46. }
47.
48. - (void)viewDidUnload
```

```
49. {  
50.     self.paddleView = nil;  
51. }  
52.  
53. @end
```

```
1. //
2. // PongAppDelegate.h
3. // Pong
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates Core Graphics.
10. // Inspired by iTennis by Brandon Trebitowski.
11. //
12.
13. #import <UIKit/UIKit.h>
14.
15. @class PongViewController;
16.
17. @interface PongAppDelegate : NSObject <UIApplicationDelegate> {
18. }
19.
20. @property (nonatomic, retain) IBOutlet PongViewController *viewController;
21. @property (nonatomic, retain) IBOutlet UIWindow *window;
22.
23. @end
```

```
1. //
2. // PongAppDelegate.m
3. // Pong
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates Core Graphics.
10. // Inspired by iTennis by Brandon Trebitowski.
11. //
12.
13. #import "PongAppDelegate.h"
14.
15. #import "PongViewController.h"
16.
17. @implementation PongAppDelegate
18.
19. @synthesize viewController=_viewController;
20. @synthesize window=_window;
21.
22. - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
23. {
24.     self.window.rootViewController = self.viewController;
25.     [self.window makeKeyAndVisible];
26.     return YES;
27. }
28.
29. - (void)dealloc
30. {
31.     [_window release];
32.     [_viewController release];
33.     [super dealloc];
34. }
35.
36. @end
```

```
1. //
2. // PongView.h
3. // Pong
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates Core Graphics.
10. // Inspired by iTennis by Brandon Trebitowski.
11. //
12.
13. #import <UIKit/UIKit.h>
14.
15. @interface PongView : UIView {
16. }
17.
18. @end
```



```
1. //
2. // PongView.m
3. // Pong
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates Core Graphics.
10. // Inspired by iTennis by Brandon Trebitowski.
11. //
12.
13. #import "PongView.h"
14.
15. @implementation PongView
16.
17. - (void)drawRect:(CGRect)rect
18. {
19.     // get context
20.     CGContextRef context = UIGraphicsGetCurrentContext();
21.
22.     // draw dashed midfield line
23.     CGFloat dashes[] = {1,1};
24.     CGContextSetLineDash(context, 0.0, dashes, 2);
25.     CGContextSetStrokeColorWithColor(context, [UIColor whiteColor].CGColor);
26.     CGContextSetLineWidth(context, 5.0f);
27.     CGContextMoveToPoint(context, 240.0f, 0.0f);
28.     CGContextAddLineToPoint(context, 240.0f, 320.0f);
29.     CGContextStrokePath(context);
30. }
31.
32. - (void)dealloc
33. {
34.     [super dealloc];
35. }
36.
37. @end
```

```
1. //
2. // PongViewController.h
3. // Pong
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates Core Graphics.
10. // Inspired by iTennis by Brandon Trebitowski.
11. //
12.
13. #import <UIKit/UIKit.h>
14.
15. @interface PongViewController : UIViewController {
16.     CGPoint _velocity;
17. }
18.
19. @property (nonatomic, retain) IBOutlet UIImageView *ball;
20. @property (nonatomic, retain) IBOutlet UILabel *labelLeft;
21. @property (nonatomic, retain) IBOutlet UILabel *labelRight;
22. @property (nonatomic, retain) IBOutlet UIImageView *paddleLeft;
23. @property (nonatomic, retain) IBOutlet UIImageView *paddleRight;
24. @property (assign, nonatomic) BOOL paused;
25. @property (assign, nonatomic) NSInteger scoreLeft;
26. @property (assign, nonatomic) NSInteger scoreRight;
27.
28. @end
```

```
1. //
2. // PongViewController.m
3. // Pong
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates Core Graphics via Pong.
10. // Inspired by iTennis by Brandon Trebitowski.
11. //
12.
13. #import "PongViewController.h"
14.
15. #define VELOCITY 10.0f
16.
17. @implementation PongViewController
18.
19. @synthesize ball=_ball;
20. @synthesize labelLeft=_labelLeft;
21. @synthesize labelRight=_labelRight;
22. @synthesize paddleLeft=_paddleLeft;
23. @synthesize paddleRight=_paddleRight;
24. @synthesize paused=_paused;
25. @synthesize scoreLeft=_scoreLeft;
26. @synthesize scoreRight=_scoreRight;
27.
28. - (void)awakeFromNib
29. {
30.     // initialize scores
31.     self.scoreLeft = 0;
32.     self.scoreRight = 0;
33.
34.     // initialize ball's velocity
35.     _velocity = CGPointMake(VELOCITY, VELOCITY);
36.
37.     // schedule gameplay
38.     [NSTimer scheduledTimerWithTimeInterval:0.05 target:self selector:@selector(play) userInfo:nil repeats:YES];
39. }
40.
41. - (void)dealloc
42. {
43.     [_ball release];
44.     [_labelLeft release];
45.     [_labelRight release];
46.     [_paddleLeft release];
47.     [_paddleRight release];
48.     [super dealloc];
```

```
49. }
50.
51. - (void)kickoff
52. {
53.     // update scores
54.     self.labelLeft.text = [NSString stringWithFormat:@"%u", self.scoreLeft];
55.     self.labelRight.text = [NSString stringWithFormat:@"%u", self.scoreRight];
56.
57.     // center ball
58.     self.ball.center = CGPointMake(240.0f, 160.0f);
59.
60.     // pause
61.     self.paused = YES;
62. }
63.
64. - (void)play
65. {
66.     // check whether paused
67.     if (self.paused)
68.         return;
69.
70.     // move ball
71.     self.ball.center = CGPointMake(self.ball.center.x + _velocity.x, self.ball.center.y + _velocity.y);
72.
73.     // detect goals
74.     if (self.ball.center.x < 5) {
75.         self.scoreRight++;
76.         [self kickoff];
77.     }
78.     else if (self.view.bounds.size.width - 5 < self.ball.center.x) {
79.         self.scoreLeft++;
80.         [self kickoff];
81.     }
82.
83.     // bounce off of top and bottom walls
84.     if (self.ball.center.y < 5 || self.view.bounds.size.height - 5 < self.ball.center.y) {
85.         _velocity.y = -_velocity.y;
86.     }
87.
88.     // bounce off of left paddle
89.     if (CGRectIntersectsRect(self.ball.frame, self.paddleLeft.frame)) {
90.         if (self.paddleLeft.center.x < self.ball.center.x) {
91.             _velocity.x = -_velocity.x;
92.         }
93.     }
94.
95.     // bounce off of right paddle
96.     if (CGRectIntersectsRect(self.ball.frame, self.paddleRight.frame)) {
```

```
97.         if (self.ball.center.x < self.paddleRight.center.x) {
98.             _velocity.x = -_velocity.x;
99.         }
100.     }
101.
102.     // move opponent as ball approaches
103.     if (_velocity.x < 0) {
104.
105.         // move down
106.         if (self.ball.center.y < self.paddleLeft.center.y) {
107.             self.paddleLeft.center = CGPointMake(self.paddleLeft.center.x, self.paddleLeft.center.y - 10.0f);
108.         }
109.
110.         // move up
111.         else if (self.ball.center.y > self.paddleLeft.center.y) {
112.             self.paddleLeft.center = CGPointMake(self.paddleLeft.center.x, self.paddleLeft.center.y + 10.0f);
113.         }
114.     }
115. }
116.
117. - (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
118. {
119.     // unpause if paused
120.     if (self.paused) {
121.         self.paused = NO;
122.     }
123. }
124.
125. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
126. {
127.     return (interfaceOrientation == UIInterfaceOrientationLandscapeRight);
128. }
129.
130. - (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event
131. {
132.     // follow user's finger vertically with left paddle
133.     UITouch *touch = [[event allTouches] anyObject];
134.     CGPoint location = [touch locationInView:self.view];
135.     self.paddleRight.center = CGPointMake(self.paddleRight.center.x, location.y);
136. }
137.
138. - (void)viewDidLoad
139. {
140.     [super viewDidLoad];
141.
142.     // play ball!
143.     [self kickoff];
144. }
```

```
145.  
146. - (void)viewDidUnload  
147. {  
148.     self.ball = nil;  
149.     self.labelLeft = nil;  
150.     self.labelRight = nil;  
151.     self.paddleLeft = nil;  
152.     self.paddleRight = nil;  
153.  
154.     [super viewDidUnload];  
155. }  
156.  
157. @end
```