

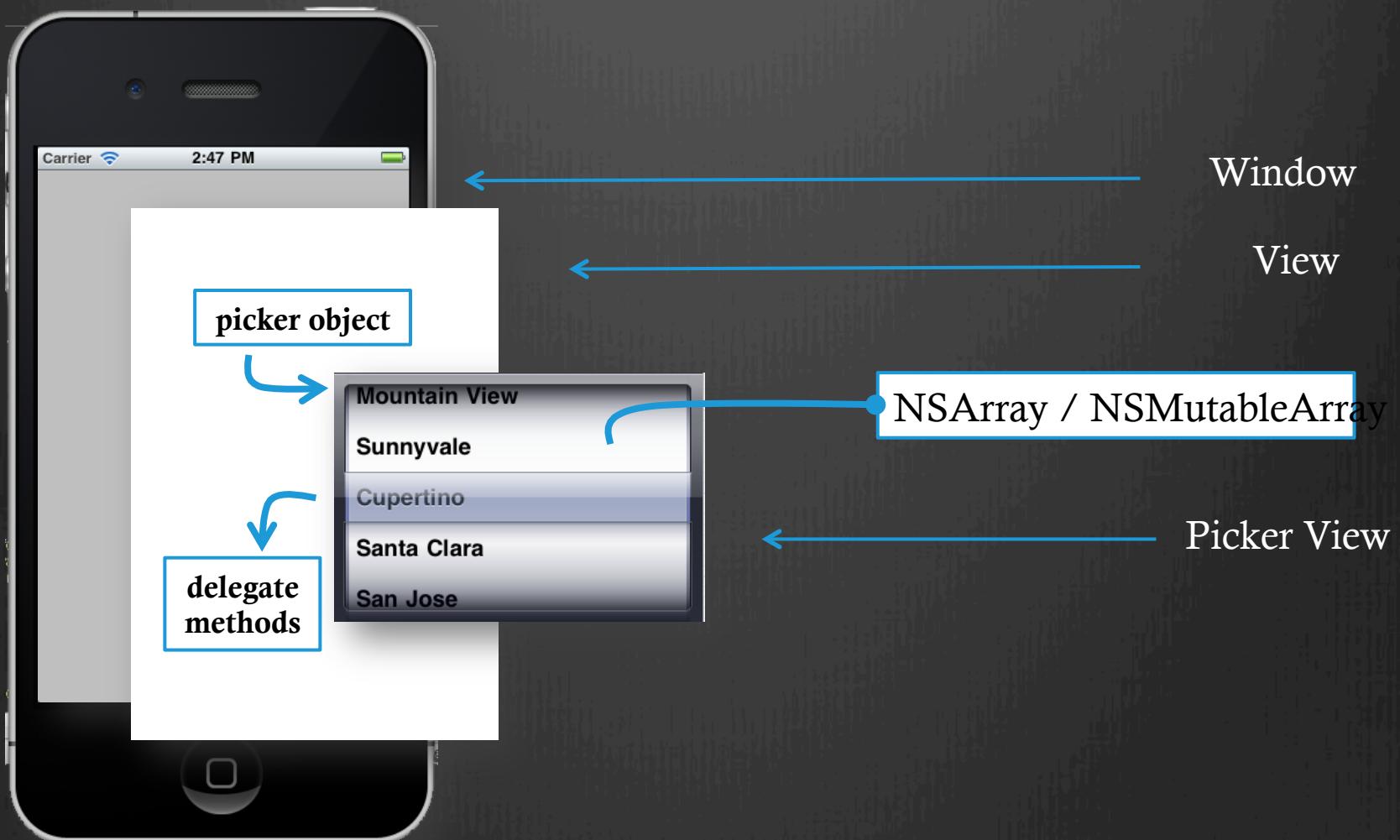
Chapter 9

iOS Delegate & Data Source

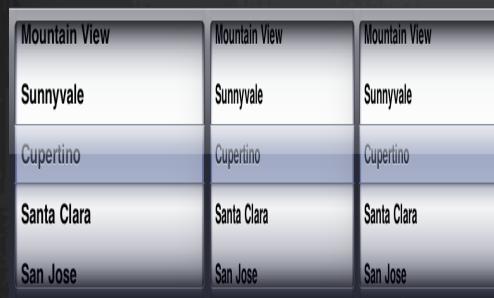
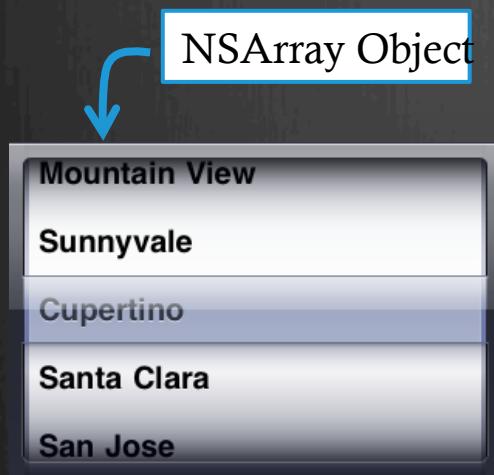
Data Source

- ◎ Control หลายๆ ตัว ใน iOS นั้นจะมีการ load ข้อมูลที่มีลักษณะเป็น array, dictionary หรือ collection ขึ้นมาแสดงผล เช่น Picker View หรือ Table View
- ◎ แหล่งของข้อมูลนั้นอาจจะเป็น file ที่ bundle ใน application หรือเป็นข้อมูลที่ load ขึ้นมาจาก database หรือ download มาจาก web services แต่เมื่อ control บน iOS จะนำมาใช้งาน ชุดข้อมูลเหล่านั้นมักจะอยู่ในรูปของ array หรือ dictionary
- ◎ Data Source ใน iOS จะเป็นเทคนิคในการเขียนโปรแกรมเพื่อให้ control load ข้อมูลขึ้นมาแสดงผล แต่การ load ข้อมูลนั้นจะมีขั้นตอนที่มากกว่า development platform อื่น
- ◎ หลักการ data source บน iOS นั้นจะใช้เทคนิค delegate แบบเดียวกับที่เรียนมาในบทที่แล้ว แต่จะแบ่ง method เป็นหลายๆ method ต่อการ load ข้อมูล 1 ชุด เช่น method หนึ่งจะใช้เพื่อบอก control ว่ามีสมาชิกใน array กี่ตัว และอีก method หนึ่ง เป็นการ load ข้อมูล cell ที่ control ต้องการเป็นต้น

Picker View



Data Source & Delegate Example



UIPickerViewDataSource

```
- (NSInteger)numberOfComponentsInPickerView:  
    (UIPickerView *)pickerView  
  
- (NSInteger)pickerView:(UIPickerView *)pickerView  
    numberOfRowsInComponent:(NSInteger)rowNum
```

UIPickerViewDelegate

```
- (NSString *)pickerView:(UIPickerView *)pickerView  
    titleForRow:(NSInteger)row  
    forComponent:(NSInteger)component  
  
- (CGFloat)pickerView:(UIPickerView *)pickerView  
    widthForComponent:(NSInteger)component  
  
- (void)pickerView:(UIPickerView *)pickerView  
    didSelectRow:(NSInteger)row  
    inComponent:(NSInteger)component
```

Lab : Custom Picker View (1/8)

◎ วัตถุประสงค์

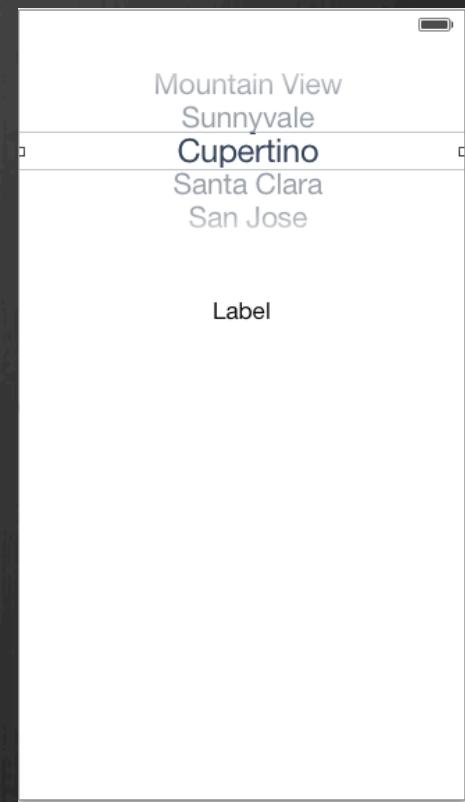
- ◎ เพื่อให้เข้าใจหลักการของ delegate และ data source ของ control บน iOS โดยศึกษาจากวิธีใช้ Picker View
- ◎ เข้าใจ NSArray ใน iOS และสามารถนำไปใช้งานได้

◎ ขั้นตอน

- ◎ ออกแบบ UI
- ◎ Implement Data Source
- ◎ Implement Delegate

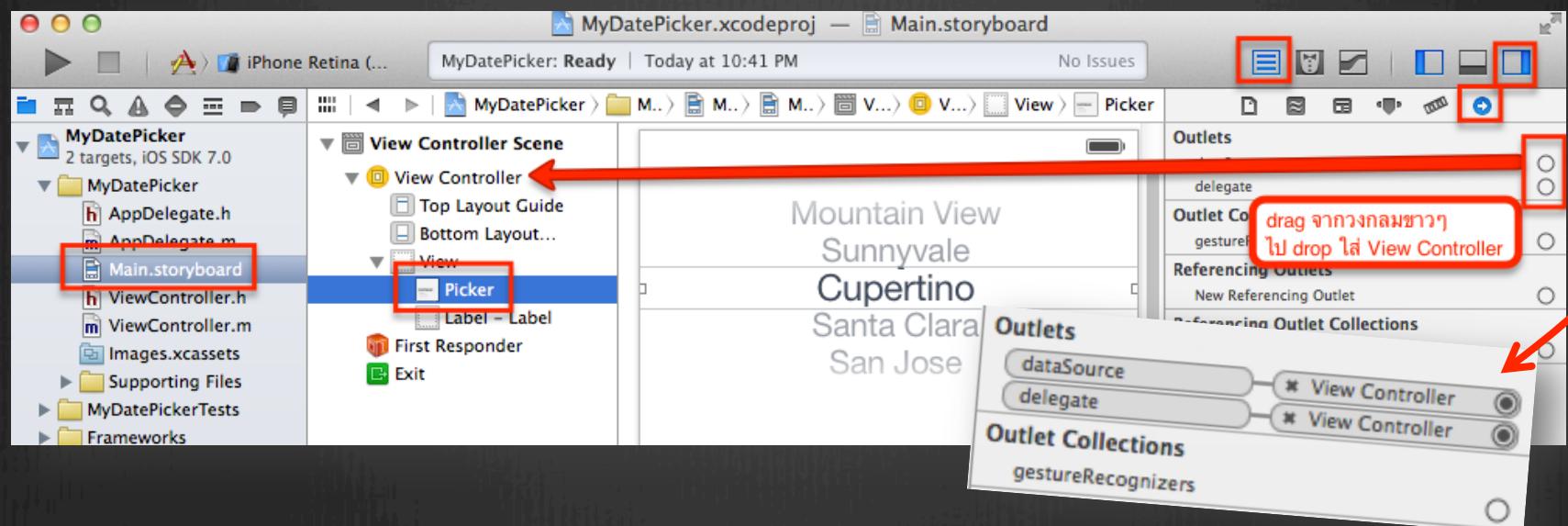
Task : Create Project (2/8)

1. จาก Xcode สร้าง project ใหม่โดยเลือก iOS > Application > Single View Application
2. ตั้งชื่อ project ว่า “MyDatePicker” และเลือก Devices เป็น iPhone
3. Click “Next” เลือก folder ที่จะ save project และ click ปุ่ม “Create”
4. เปิด Main.storyboard และวาง control ตามรูป
 - Picker View
 - Label



Task : Binding Data Source & Delegate (3/8)

5. เปิดไฟล์ Main.storyboard และเปิด Connections Inspector ใน Inspector Pane และ click เลือก Picker View บน View จะเห็น “dataSource” และ “delegate” ในกลุ่ม Outlet ของ Picker View ให้ทำการ drag จาก connector dataSource และ delegate ไป drop ยังรายการ View Controller ใน Document Outline



Task : Add Attributes (4/8)

6. เปิดไฟล์ ViewController.h เปลี่ยน editor mode เป็น Assistant Editor และผูก outlet ของ Label กับ class View Controller
7. เพิ่ม code ในไฟล์ ViewController.h ดังนี้

```
#import <UIKit/UIKit.h>

@interface ViewController : UIViewController
    <UIPickerViewDelegate, UIPickerViewDataSource>
{
    NSArray * monthArray;
    NSArray * yearArray;

    NSString * selectedYear;
    NSString * selectedMonth;
}

@property (weak, nonatomic) IBOutlet UILabel *txtResult;

- (void)setResultText;

@end
```

Task : Initial Array (5/8)

8. เปิดไฟล์ ViewController.m เพิ่ม code สำหรับ initial array ที่เตรียมไว้

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    yearArray = [[NSArray alloc] initWithObjects:
                 @"2553", @"2554", @"2555", @"2556",
                 @"2557", @"2558", nil];

    monthArray = [[NSArray alloc] initWithObjects:
                  @"มกราคม", @"กุมภาพันธ์", @"มีนาคม", @"เมษายน",
                  @"พฤษภาคม", @"มิถุนายน", @"กรกฎาคม", @"สิงหาคม",
                  @"กันยายน", @"ตุลาคม", @"พฤศจิกายน", @"ธันวาคม", nil];
    selectedYear = [yearArray objectAtIndex:0];
    selectedMonth = [monthArray objectAtIndex:0];
    [self setResultText];
}

- (void)setResultText
{
    self.txtResult.text = [NSString stringWithFormat:
                           @"คุณเลือก %@ %@", selectedMonth, selectedYear];
}
```

Task : Picker View Data Source (6/8)

9. เพิ่ม Data Source Method ของ Picker View เพื่อแสดงข้อมูลใน array

```
#pragma mark – Picker View Data Source

- (NSInteger)numberOfComponentsInPickerView:(UIPickerView *)pickerView
{
    return 2;
}

- (NSInteger)pickerView:(UIPickerView *)pickerView
    numberOfRowsInComponent:(NSInteger)component
{
    switch (component) {
        case 0:
            return [yearArray count];
            break;
        case 1:
            return [monthArray count];
            break;
        default:
            return 0;
            break;
    }
}
```

Task : Picker View Delegate (7/8)

10. เพิ่ม Data Source Method ของ Picker View เพื่อแสดงข้อมูลใน array

```
#pragma mark - implement UIPickerViewDelegate

- (NSString *)pickerView:(UIPickerView *)pickerView titleForRow:(NSInteger)row
    forComponent:(NSInteger)component
{
    switch (component) {
        case 0:
            return [yearArray objectAtIndex:row]; break;
        case 1:
            return [monthArray objectAtIndex:row]; break;
        default:
            return nil; break;
    }
}

- (CGFloat)pickerView:(UIPickerView *)pickerView widthForComponent:(NSInteger)component
{
    switch (component) {
        case 0:
            return 128.0f; break;
        case 1:
            return 192.0f; break;
        default:
            return 0; break;
    }
}
```

Task : Picker View Data Source (8/8)

11. เพิ่ม delegate method didSelectRow เพื่ออ่านค่าของ Picker ที่ถูกเลือก

```
- (void)pickerView:(UIPickerView *)pickerView didSelectRow:(NSInteger)row
           inComponent:(NSInteger)component
{
    switch (component) {
        case 0:
            selectedYear = [yearArray objectAtIndex:row];
            break;
        case 1:
            selectedMonth = [monthArray objectAtIndex:row];
            break;
    }

    self.txtResult.text = [NSString stringWithFormat:
                           @"คุณเลือก %@ %@", selectedMonth, selectedYear];
}
```

12. Run โปรแกรมเพื่อดูผลลัพธ์