

ตัวอย่างที่ 1 เกมทายภาพ

สร้างโปรเจค C4_WordGame

บน iPhone วาง label 1 ตัว และ button 2 ตัว ปรับ alignment ของ label ให้อยู่ตรงกลาง สร้างลิงค์เชื่อมตัวแปร

จากนั้นลบข้อความ Label และ Button ออกจาก label และ button ทั้ง 3 ตัว

เชื่อม button เป็นทั้ง 2 แบบคือ IBOutlet และ IBAction เพราะเราต้องการทั้งอ้างอิงถึง และต้องการให้ทำงานขึ้นบน button ด้วย

import ไฟล์รูปภาพสัตว์ทั้งหมดในโฟลเดอร์ c4_animal เข้ามาในโปรเจค

สร้างไฟล์ plist ชื่อ animal.plist เพื่อไว้เก็บข้อมูลชื่อสัตว์ ใส่ข้อมูลชื่อสัตว์ให้ตรงกับชื่อไฟล์ที่มี โดยครั้งนี้ Root เป็น Array และชื่อสัตว์ทุกตัวเป็น String

ไฟล์ ViewController.m เพิ่ม code

สร้างตัวแปร เพื่อเก็บค่า

NSArray* data	เพื่อเก็บข้อมูลชื่อสัตว์จาก plist
int current	เพื่อเก็บ index ปัจจุบันที่ทำงานของ data
int score	เพื่อเก็บคะแนนจากการทายคำศัพท์

ใน viewDidLoad อ่านข้อมูลจากไฟล์ plist เก็บไว้ใน data

กำหนดค่าเริ่มต้นให้ current = -2 เพราะเราต้องการให้เมธอด setImageOnButton เพิ่มค่า current ก่อนทำงาน

เราต้องการให้มีการจัดเรียงค่าใน data ใหม่ คือสลับค่าต่างๆทุกครั้งที่รันโปรแกรม

สร้างเมธอด shuffleArray เพื่อทำการสลับข้อมูลใน data

เพิ่ม code ในเมธอด viewDidLoad

เมธอด setImageOnButton มีการทำงาน 2 อย่างคือ

นำรูปภาพจาก data ไปวางบน button ทั้งสองตัว เช่น

ถ้าตอนนี้ current = 0 จะนำภาพจาก

data[0] วางบน _imageA (button ตัวแรก)

data[1] วางบน _imageB (button ตัวที่สอง)

และแสดงข้อความบน `_display` โดยต้องการให้การเล่นแต่ละครั้งสุ่มข้อความจากภาพที่แสดงอยู่ จึงใช้ `arc4random()%2` เพื่อเช็คว่าจะแสดงภาพใด

เพิ่ม code ในเมธอด `buttonA` และเมธอด `buttonB` เพื่อเช็คว่าคำตอบถูกต้องหรือไม่

โดยการเปรียบเทียบข้อความบน `_display` กับ `data`

ในเมธอด `buttonA` ให้เปรียบเทียบกับ `data[current]`

ในเมธอด `buttonB` ให้เปรียบเทียบกับ `data[current+1]`

เช่นสมมุติตอนนี้ `current = 4`

`_imageA` จะแสดงรูป Leopard (`data[4]`)

`_imageB` จะแสดงรูป Owl (`data[5]`)

บน `_display` อาจจะแสดงคำว่า Leopard หรือ Owl ก็ได้ ขึ้นอยู่กับการสุ่มมาแสดง สมมุติสุ่มคำว่า Leopard มาแสดง

ถ้ากด `_imageA` เมธอด `buttonA` จะทำงาน เปรียบเทียบ string แล้วเหมือนกัน ทำให้ `score ++`

ถ้ากด `_imageB` เมธอด `buttonB` จะทำงาน เปรียบเทียบ string แล้วไม่เหมือนกัน ทำให้ `score` ไม่เปลี่ยน

หลังจากเช็ค `score` แล้วเรียกเมธอด `setImageOnButton` เพื่อเปลี่ยนแปลงภาพ

ทดสอบรันโปรแกรม ต้องสามารถกดบน `button` เพื่อตอบคำถามได้

เพิ่ม code ในเมธอด `setImageOnButton` เพื่อเช็คว่าถ้าหมดรูปภาพแล้วให้แสดง alert บอกคะแนน

ทดสอบรันโปรแกรม

Storyboard

ตัวอย่างที่ 2 การใช้ storyboard สร้างหลาย view

การใช้ storyboard ในการทำงาน จะใช้เมื่อโปรแกรมมีหลายวิวในการทำงาน

แนวคิดการใช้ storyboard คือ

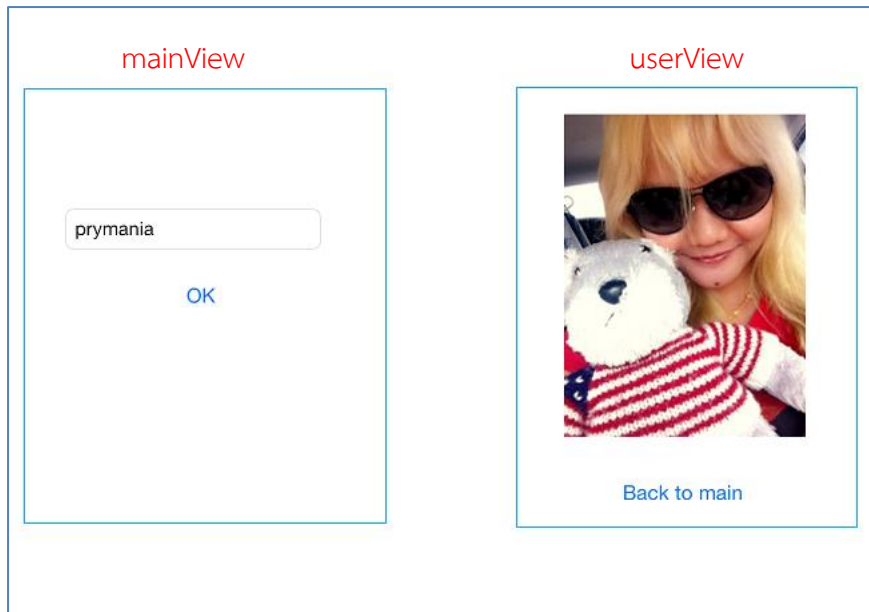
1. สร้างวิวใหม่เพิ่มในโปรแกรม
2. สร้างคลาสใหม่เป็น objective-c class เพื่อเป็นตัวจัดการวิวใหม่
3. สร้าง segue (อ่าน เซคเว - เส้นทางเชื่อมต่อระหว่างวิว) เพื่อเชื่อมต่อวิว
4. เพิ่ม code ในโปรแกรมเพื่อให้ทั้งสองวิวรู้จักกัน กรณีที่มีการส่งค่าระหว่างวิว

สร้างโปรเจคชื่อ C5_DemoTwoView

import รูปภาพที่ต้องการใช้เป็นชื่อ user เข้ามาในโปรเจค เช่น prymania.jpg และ jim.jpg

โปรแกรมนี้นี้มี 2 view คือ mainView และ userView การทำงานคือ

ใส่ชื่อ user ที่ mainView กดปุ่ม OK แสดงรูปภาพของ user นั้นที่ userView
ที่ userView กดปุ่ม Back to main เพื่อกลับมาที่ mainView



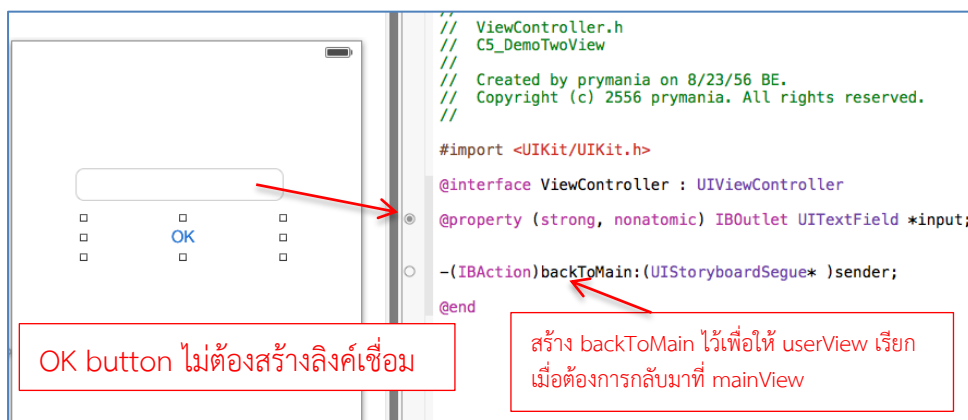
ขั้นตอนการทำงาน

1. ตกแต่งหน้าจอ mainView ตามที่ต้องการ

หน้าจอ mainView มีเพียง text field และ OK button

สร้างลิงค์เชื่อมเฉพาะ text field เป็น IBOutlet เท่านั้น ไม่ต้องสร้างลิงค์เชื่อม OK button

แต่สร้าง IBAction backToMain ทิ้งไว้ โดยไม่ต้องเชื่อมลิงค์กับออปเจกใดๆ เพื่อใช้เมธอดนี้ในการกลับมาที่ mainView



ไฟล์ ViewController.m ทำการ implement เมธอด backToMain ทิ้งไว้ โดยเมธอดนี้อาจยังไม่มีการทำงานใดๆก็ได้ แต่ต้องสร้างไว้เพื่อให้ userView สามารถเรียกใช้ได้

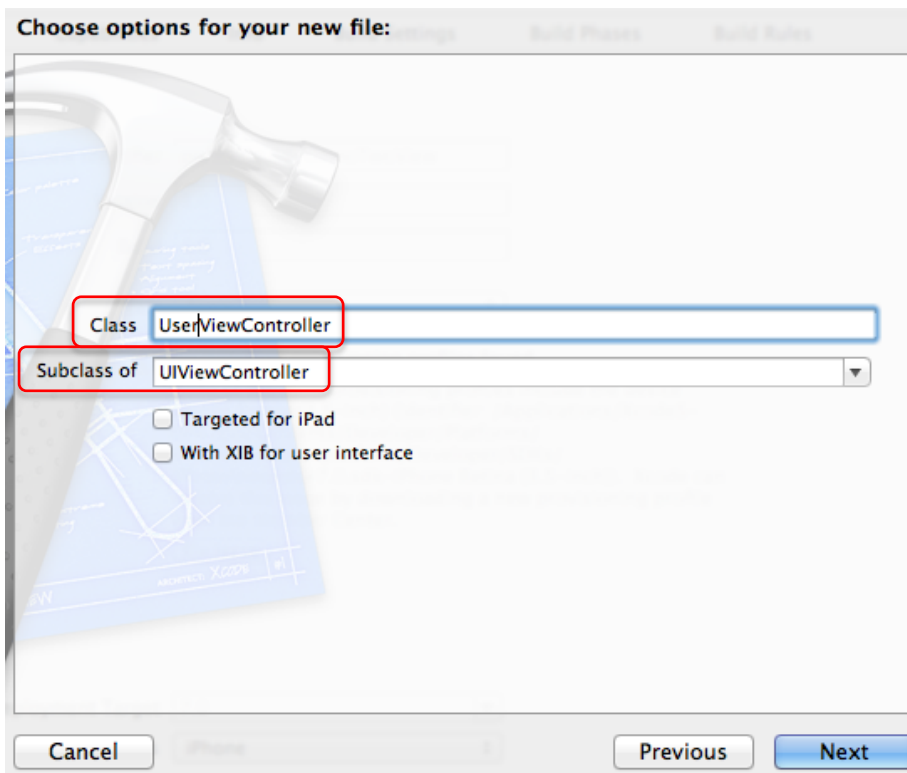
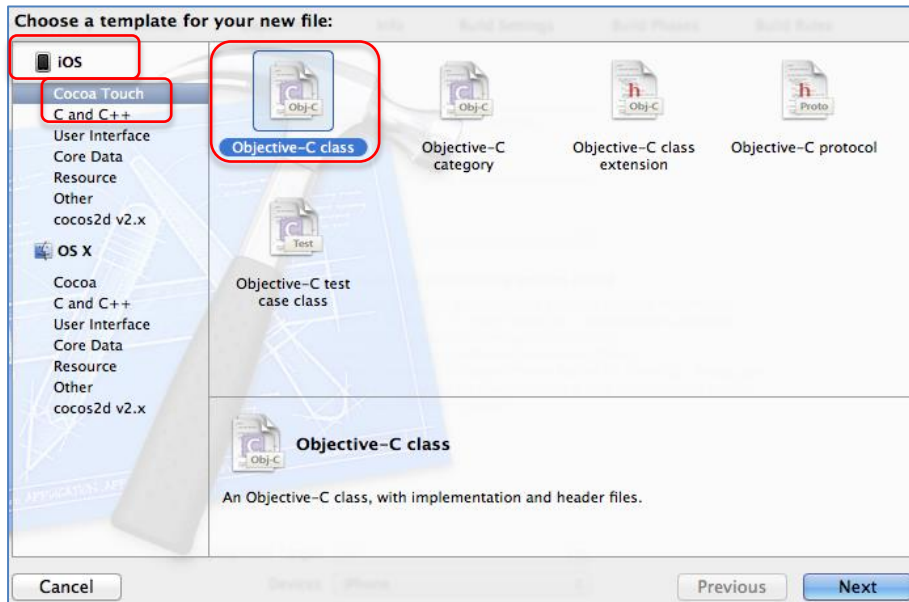
```
-(IBAction)backToMain:(UIStoryboardSegue*)sender{
}
```

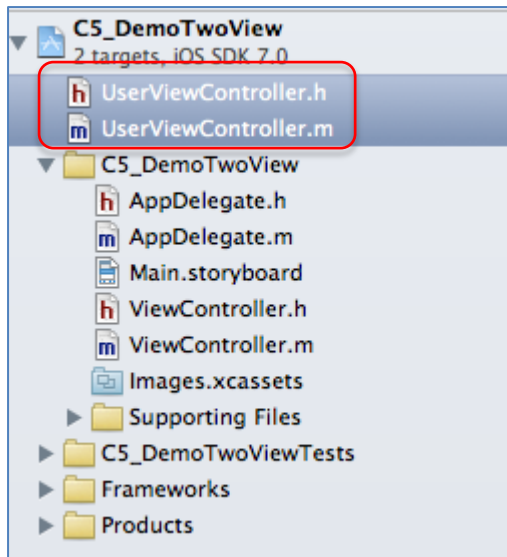
2. สร้างคลาสใหม่เพื่อใช้ในการทำงานของ userView

คลิกขวาที่ชื่อโปรเจกต์ / New File / ที่ iOS เลือก Cocoa Touch / Objective-C class

ตั้งชื่อ UserViewController / Subclass of เป็น UIViewController

ที่โปรเจกต์จะได้ไฟล์ใหม่ 2 ไฟล์คือ UserViewController.h และ UserViewController.m

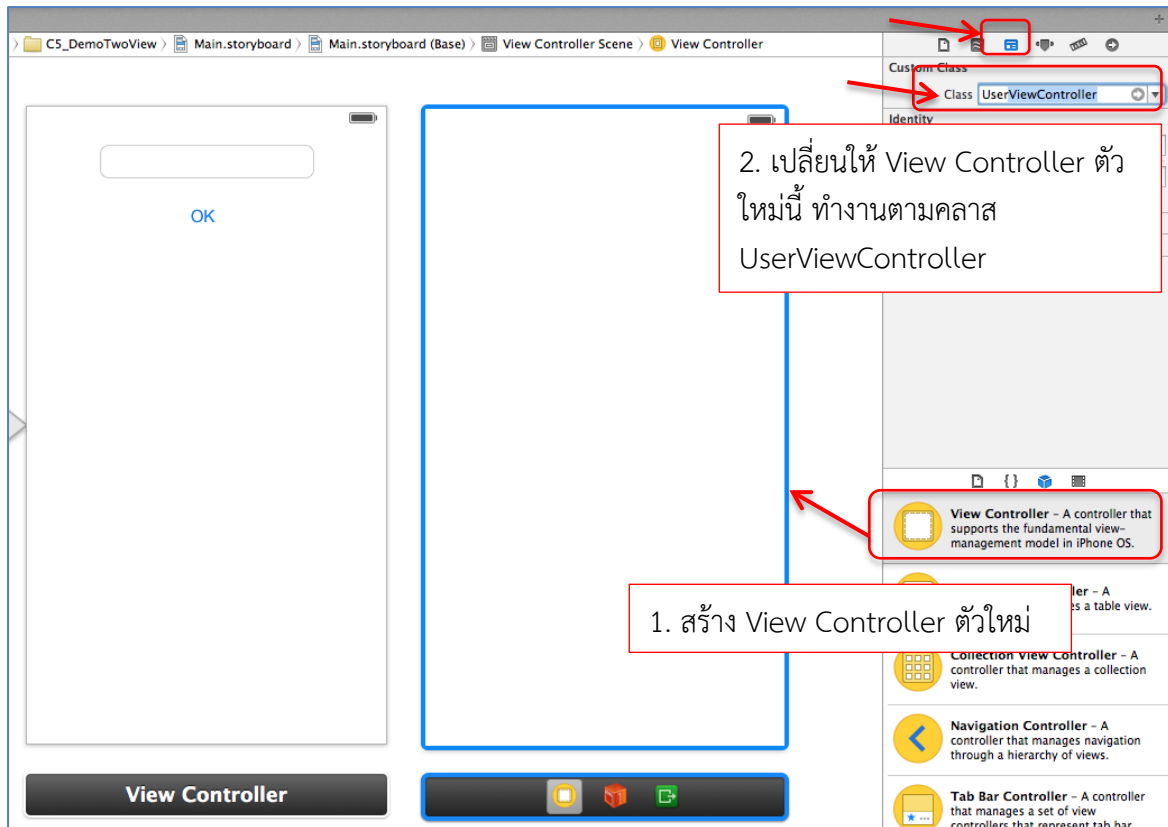




สร้าง View Controller สำหรับ userView

ที่ Main.Storyboard ลาก View Controller วางตรงที่ว่างเพื่อใช้เป็นวิว userView

คลิกที่ View Controller / คลิกเมนู Identity inspector / ที่ Custom Class เปลี่ยนชื่อคลาสเป็น
UserViewController



3. ที่หน้าจอ userView วางออปเจตต่างๆตามที่ต้องการ

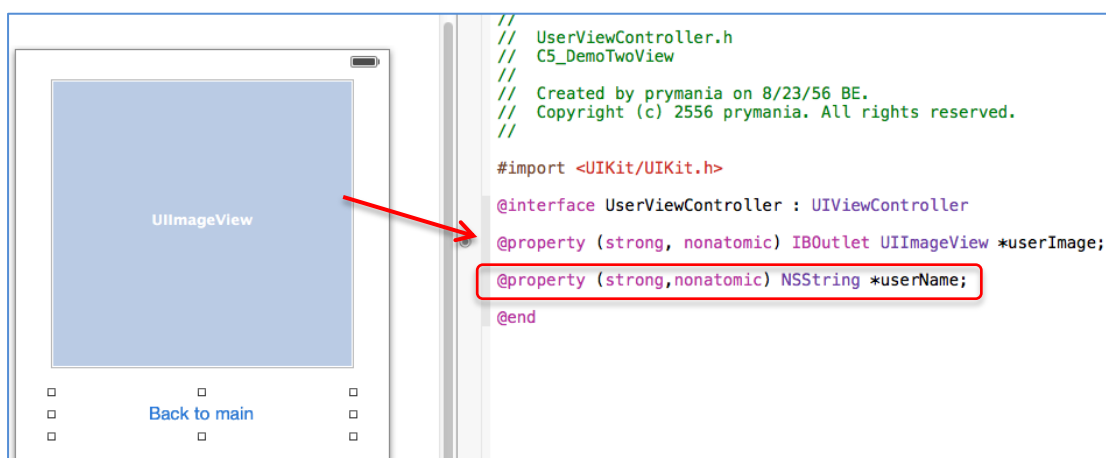
ไฟล์การทำงานของ userView คือ UserViewController.m และ UserViewController.h

กรณีนี้ userView มี 2 ออปเจตคือ UIImageView สำหรับแสดงรูปของ user และ button สำหรับกลับไป mainView

เปิด userView คู่กับไฟล์ UserViewController.h เพื่อสร้างลิงค์เชื่อม โดยสร้างเฉพาะ userImage เท่านั้น

ไม่ต้องทำลิงค์ของ button

นอกจากนั้น สร้างตัวแปร NSString* userName เพื่อไว้เก็บชื่อ user ที่จะส่งมาจาก mainView



4. การทำงานของ userView

ไฟล์ UserController.m

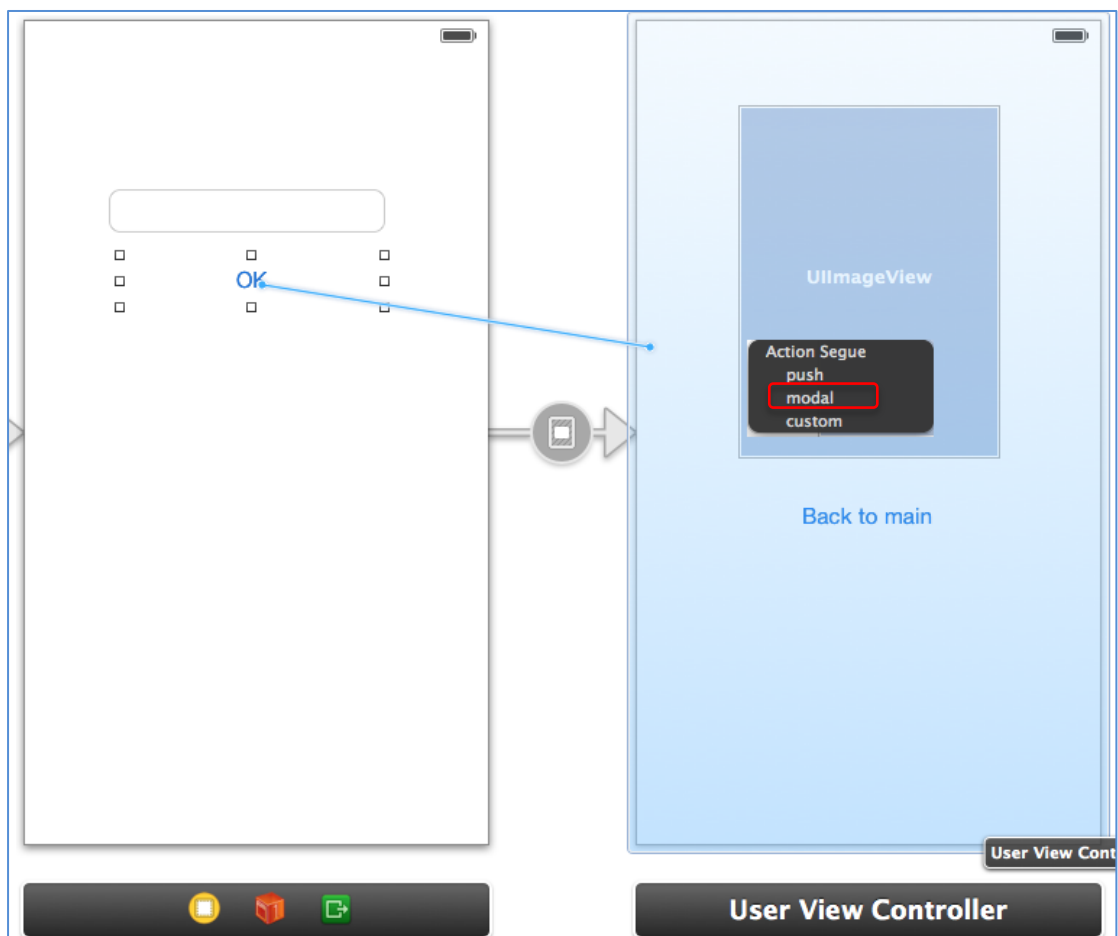
เพราะเราต้องการให้ เมื่อเปิด view นี้ แสดงรูปภาพของ user ตามชื่อที่ใส่ใน mainView
เพิ่ม code ในเมธอด viewDidLoad เพื่อแสดงรูปภาพตามตัวแปร userName

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view.

    NSString* picName = [NSString stringWithFormat:@"%s.jpg", _userName];
    _userImage.image = [UIImage imageNamed:picName];
}
```

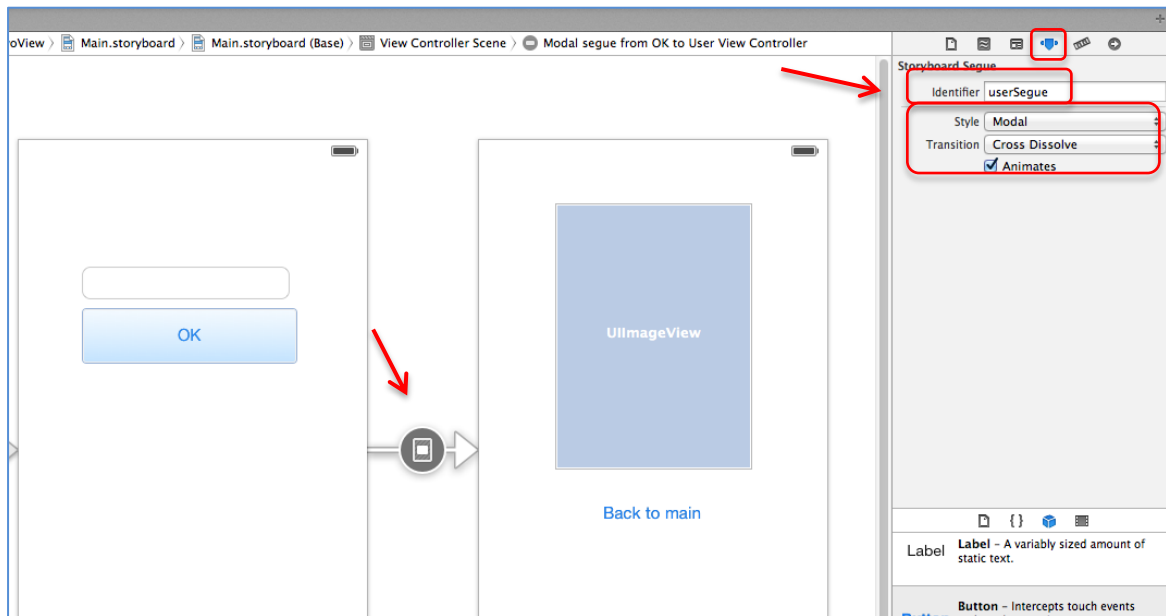
5. สร้างเส้นเชื่อมระหว่าง mainView และ userView

สร้างเส้นเชื่อมจาก OK button บน mainView ปุ่ม มาที่ userView โดย
กดที่ปุ่ม OK / กดปุ่ม control ค้างไว้ / ลากเส้นมาที่ userView / มี popup Action Segue ขึ้นมา เลือก
modal / จะได้เส้นเชื่อมระหว่าง mainView และ userView

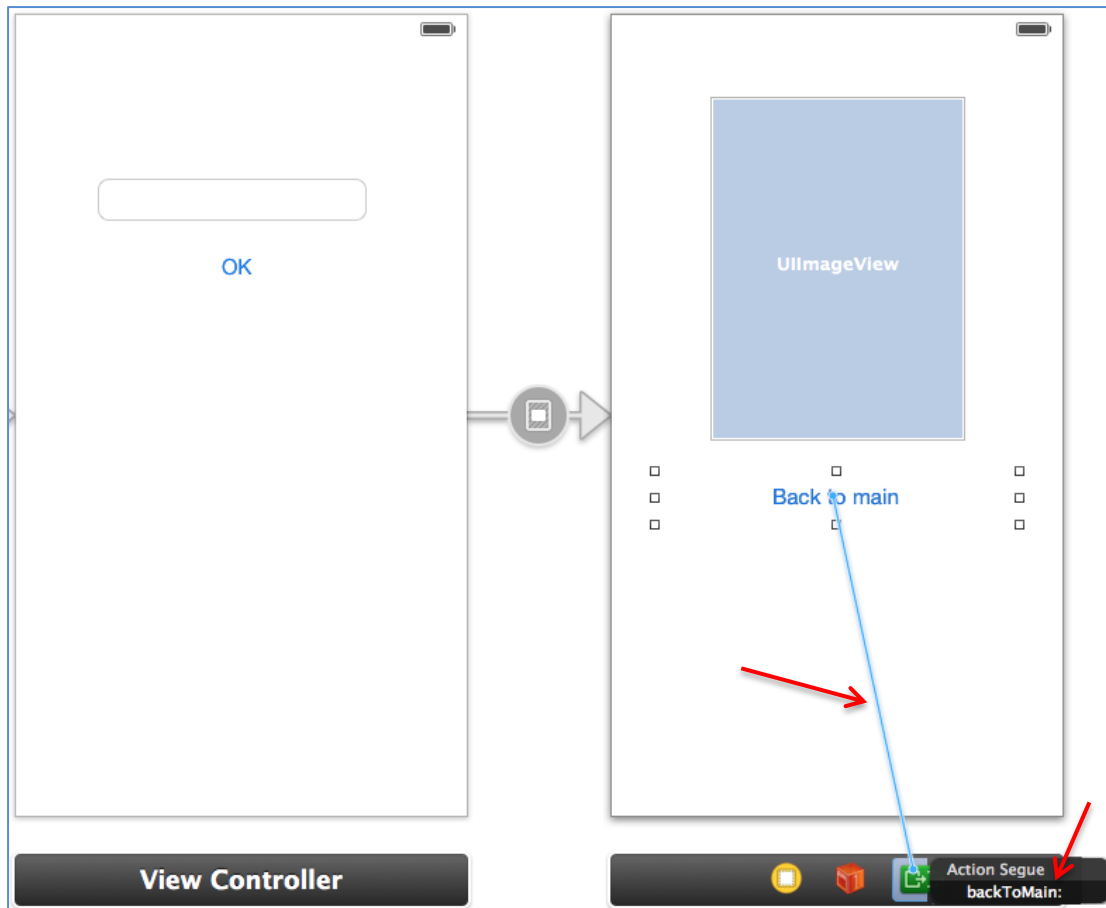


โปรแกรมนี้เราไม่จำเป็นต้องตั้งชื่อ segue เพราะมีเพียงเส้นเดียวเชื่อมไปหาวิวอื่นๆ แต่บางกรณีเราอาจจำเป็นต้องตั้งชื่อ segue ด้วย โปรแกรมนี้แม้ว่าจะไม่จำเป็นต้องตั้งชื่อ segue แต่เราจะพาดังชื่อเลย ทำได้โดย

คลิกที่เส้น segue / คลิกเมนู Attributes inspector / ที่ Identifier ตั้งชื่อเส้นเป็น userSegue / เราสามารถเปลี่ยน style และ รูปแบบ transition ได้ด้วย รูปแบบ transition คือลักษณะการเปลี่ยนวิว (หลังจากเขียนโปรแกรมเสร็จแล้ว ลองกลับมาแก้ไขรูปแบบ transition ที่นี้เพื่อดูลักษณะการทำงานของแต่ละรูปแบบ)



จาก userView ถ้ากดปุ่ม Back to main จะเปลี่ยนมาแสดง userView ดังนั้นสร้างเส้นเชื่อมจากปุ่ม Back to main แต่การสร้างเส้นเชื่อมครั้งนี้ เราจะเชื่อมไปที่ปุ่ม exit ของ userView โดยกดที่ปุ่ม Back to main / กดปุ่ม control ค้างไว้ / ลากเส้นมาที่ปุ่ม exit / มี popup Action Segue ขึ้นมาเลือก backToMain ซึ่งเป็นเมธอดที่เราสร้างไว้ใน mainView



6. การทำงานของ mainView

เราต้องทำให้ mainView รู้จัก ไฟล์ UserViewController.h ด้วยดังนั้นต้อง import UserViewController.h ในไฟล์ ViewController.m ด้วย

เปิดไฟล์ ViewController.m เพิ่ม code เพื่อ import UserViewController.h

เมธอด backToMain ที่เราสร้างทิ้งไว้ จะถูกเรียกให้ทำงานเมื่อมีการเปลี่ยนกลับมาที่ mainView กรณีนี้เราเพิ่ม code เพื่อส่งให้เคลียร์ค่าในช่อง input

สร้างเมธอด prepareForSegue เมธอดนี้จะถูกเรียกเมื่อมีการเปลี่ยนวิว ถ้าเรามีเส้น segue หลายเส้นจากวิวนี้ เราสามารถเช็คชื่อของเส้น segue ได้ว่าเส้นไหนทำงานและจะเปลี่ยนไปวิวใด โปรแกรมนี้เรามีเส้น segue เส้นเดียว ดังนั้นไม่จำเป็นต้องใส่ code ในการเช็คชื่อเส้น segue ก็ได้ เราใส่ comment code ส่วนนี้ไว้เพื่อให้ดูว่าถ้าเราต้องเช็คชื่อเส้น segue สามารถเช็คได้อย่างไร

ในเมธอดนี้ สิ่งที่ต้องทำคือ อ่านชื่อ user จากช่อง input ของ mainView เพื่อส่งชื่อนี้ให้กับ userView เมื่อแสดง userView เมธอด viewDidLoad ใน userViewController.m จะทำงาน โดยการแสดงรูป user ชื่อนี้นั่นเอง

```

#import "ViewController.h"
#import "UserViewController.h"

@interface ViewController ()

@end

@implementation ViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

- (IBAction)backToMain:(UIStoryboardSegue *)sender{
    _input.text = NULL;
}

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender{
    // if([segue identifier] isEqualToString:@"userSegue"]){
        UserViewController *view = [segue destinationViewController];
        view.userName = _input.text;
    // }
}

@end

```

ทดสอบโปรแกรม ใส่ชื่อ user กดปุ่ม OK จะแสดงรูปภาพของ user ตามชื่อที่ใส่ (ต้องมีรูปที่ตรงกับชื่อ user ในโปรเจกต์ด้วย)

สรุปการทำงาน

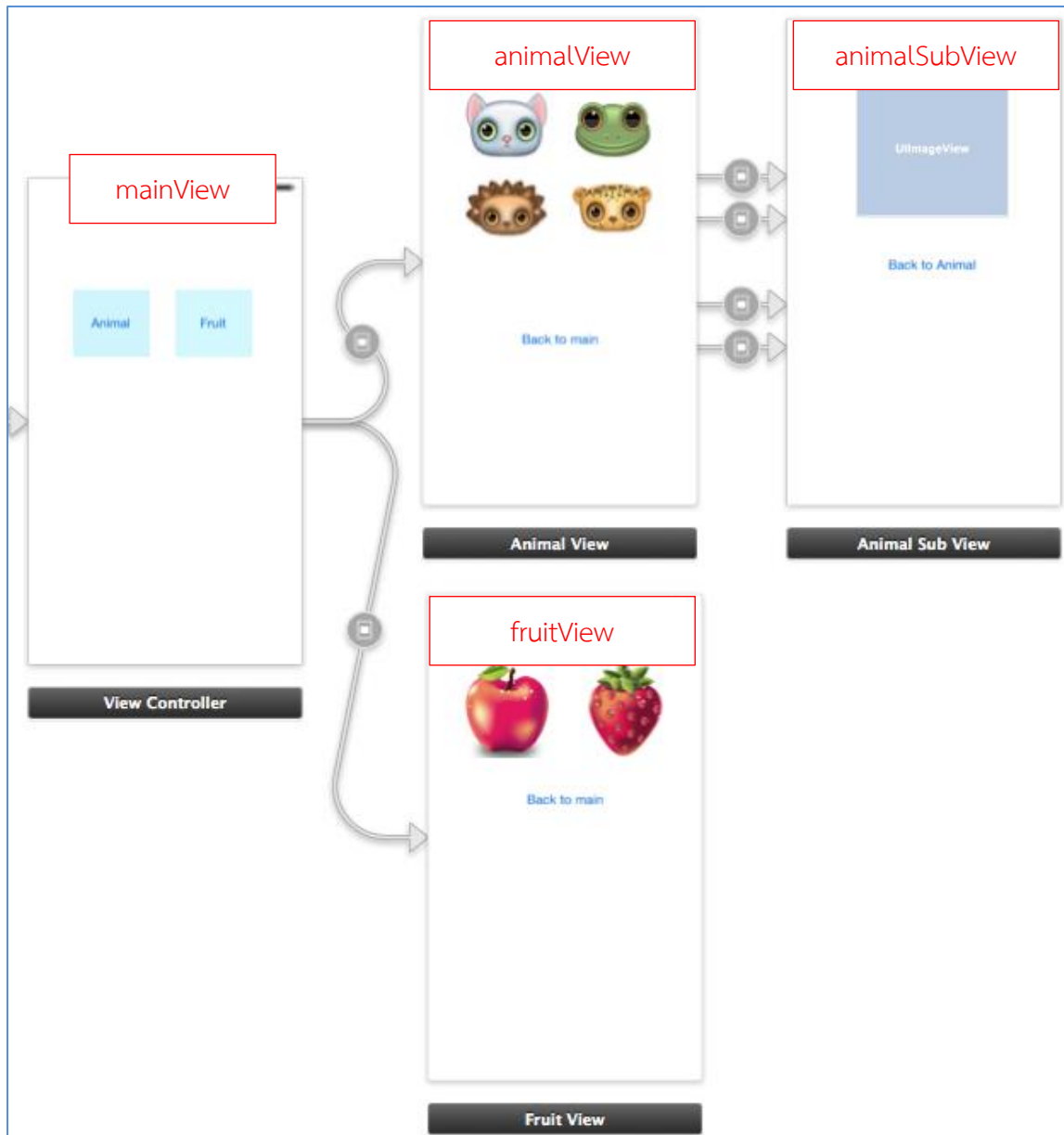
1. ที่ mainView สร้าง segue เชื่อมไป userView
2. mainView สร้าง IBAction backToMain ที่ไว้เพื่อให้ userView เรียกเพื่อถอยกลับมาที่ mainView
3. mainView ต้องทำตัวเองให้รู้จัก userView โดยการ import UserViewController.m
4. mainView สร้างเมธอด prepareForSegue ในเมธอดนี้จัดการเรื่องการส่งค่าต่างๆที่จำเป็นต้องส่งไปให้ userView กรณีนี้คือส่งชื่อ user
5. ที่ userView สร้างตัวถอยกลับไปที่ mainView โดยการเชื่อมกับปุ่ม exit / เมธอด backToMain ซึ่งต้องสร้างไว้ใน mainView
6. userView เมธอด viewDidLoad ต้องสั่งงานที่ต้องการให้ทำเมื่อแสดง userView

ตัวอย่างที่ 3 สร้าง storyboard หลายวิว

สร้างโปรเจค C5_DemoFourView

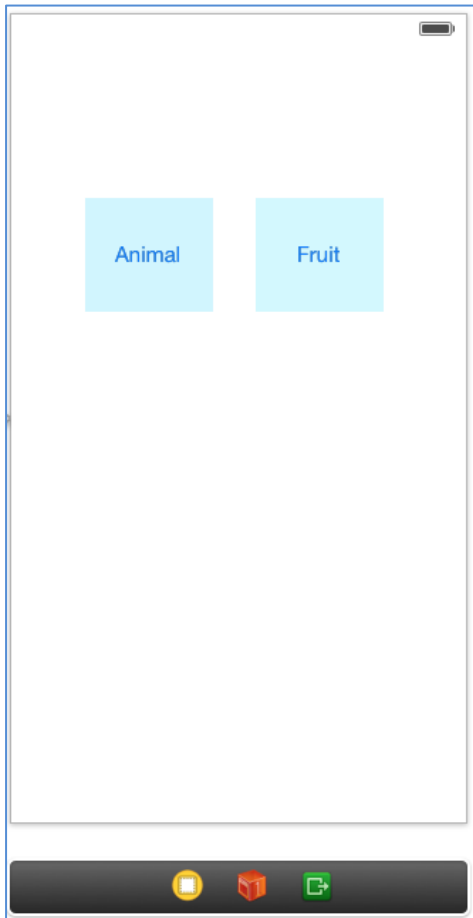
import ไฟล์รูปภาพจาก animalIcon และ fruitIcon เพื่อไว้ในโปรแกรม

เราต้องการทำงานประกอบด้วย 4 วิว คือดังนี้

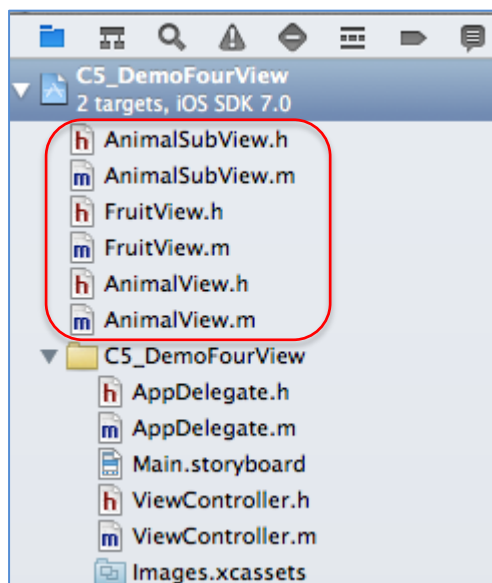


1. สร้าง mainView วาง button 2 ตัว เปลี่ยน text เป็น Animal และ Fruit
ไฟล์ ViewController.h สร้าง IBAction backToMain ที่ไว้
ไฟล์ ViewController.m implement เมธอด backToMain

ไม่ต้องสร้างลิงค์เชื่อมตัวแปรใดๆ

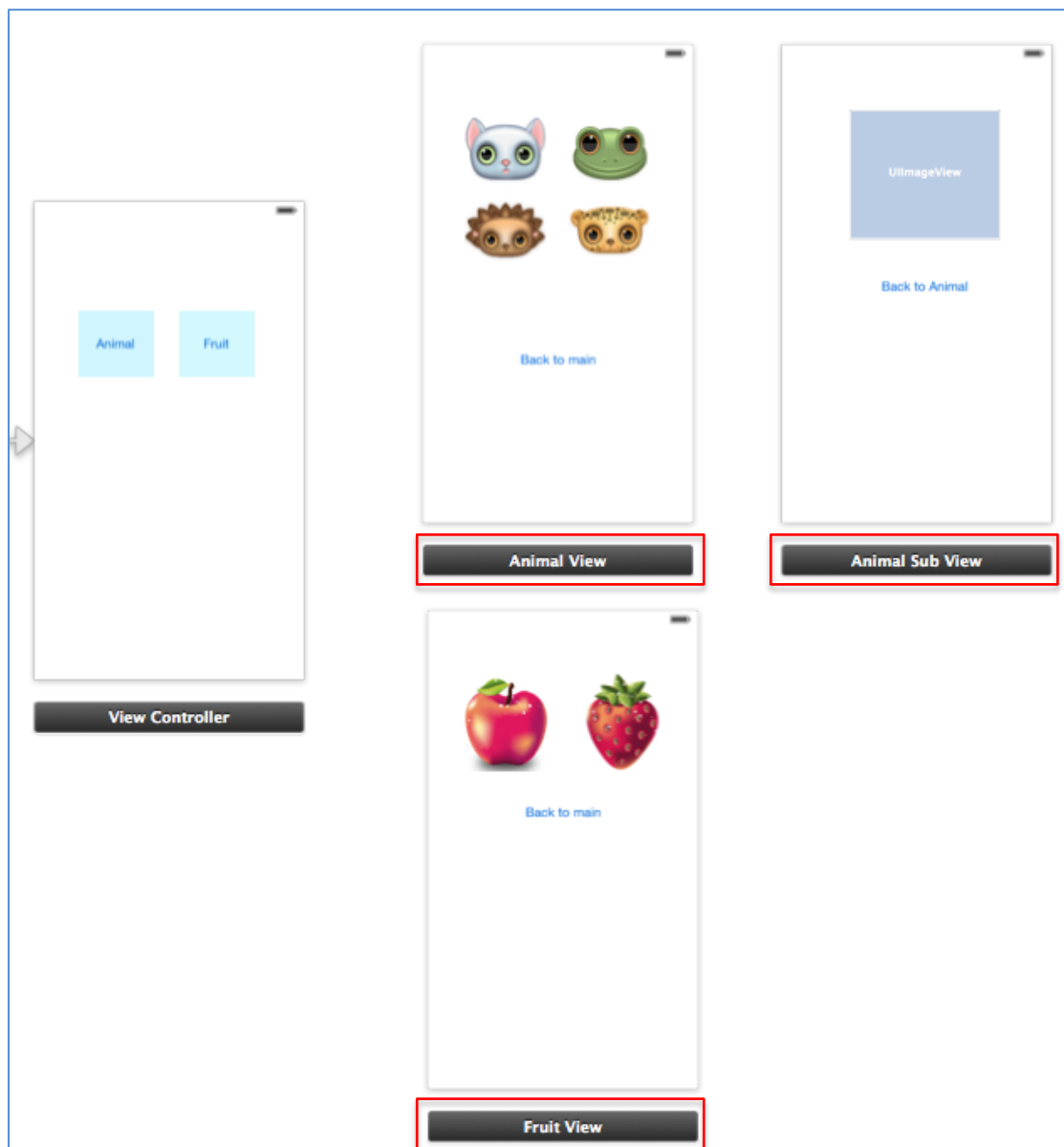


2. สร้าง Objective-C class เพื่อทำงานกับ view แต่ละตัว ดังนั้นเราต้องสร้าง 3 คลาส ตั้งชื่อเป็น AnimalView, FruitView และ AnimalSubView



วาง View Controller บน storyboard เพื่อใช้เป็น view ต่างๆ เปลี่ยนชื่อคลาสของแต่ละ view ให้ถูกต้อง

ที่ view ต่างๆ วางออบเจกต์ และเปลี่ยนรูปภาพเป็น สัตว์และผลไม้ตามที่ต้องการ
ที่ AnimalView ต้องใช้เป็น button เพราะเราต้องการกดแล้วแสดง AnimalSubView
แต่ที่ FruitView สามารถใช้เป็น UIImageView เพื่อแสดงรูปภาพได้เลย



ที่ไฟล์ AnimalView.h เราต้องสร้าง เมธอดไว้เพื่อให้เป็นตัวกลับจาก AnimalSubView

ไฟล์ AnimalView.h สร้าง IBAction backToAnimal ทิ้งไว้

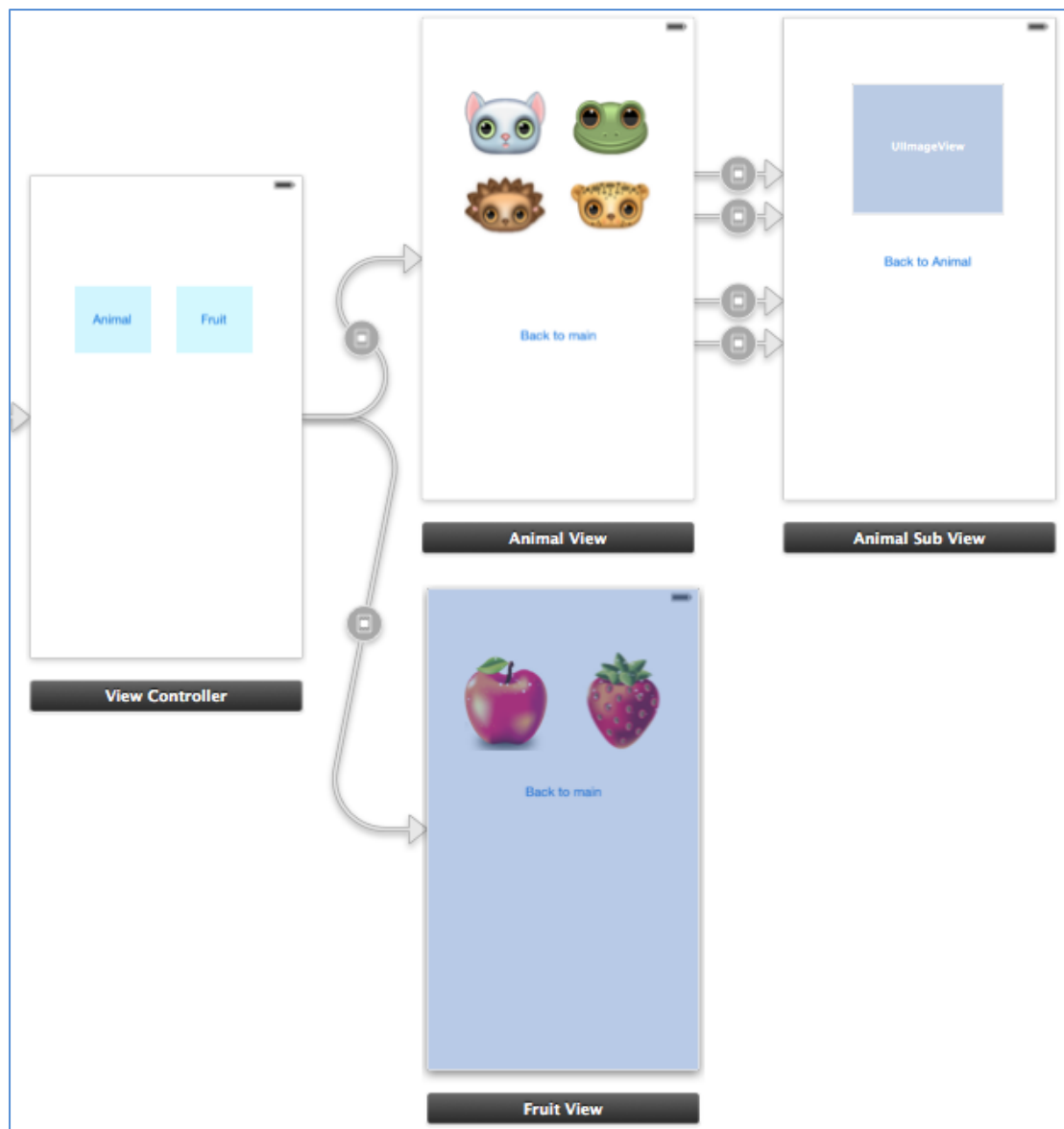
ไฟล์ AnimalView.m implement เมธอด backToAnimal

```
-(void)backToAnimal:(UIStoryboardSegue *)sender{  
}
```

3. สร้างเส้นเชื่อม segue ต่างๆ

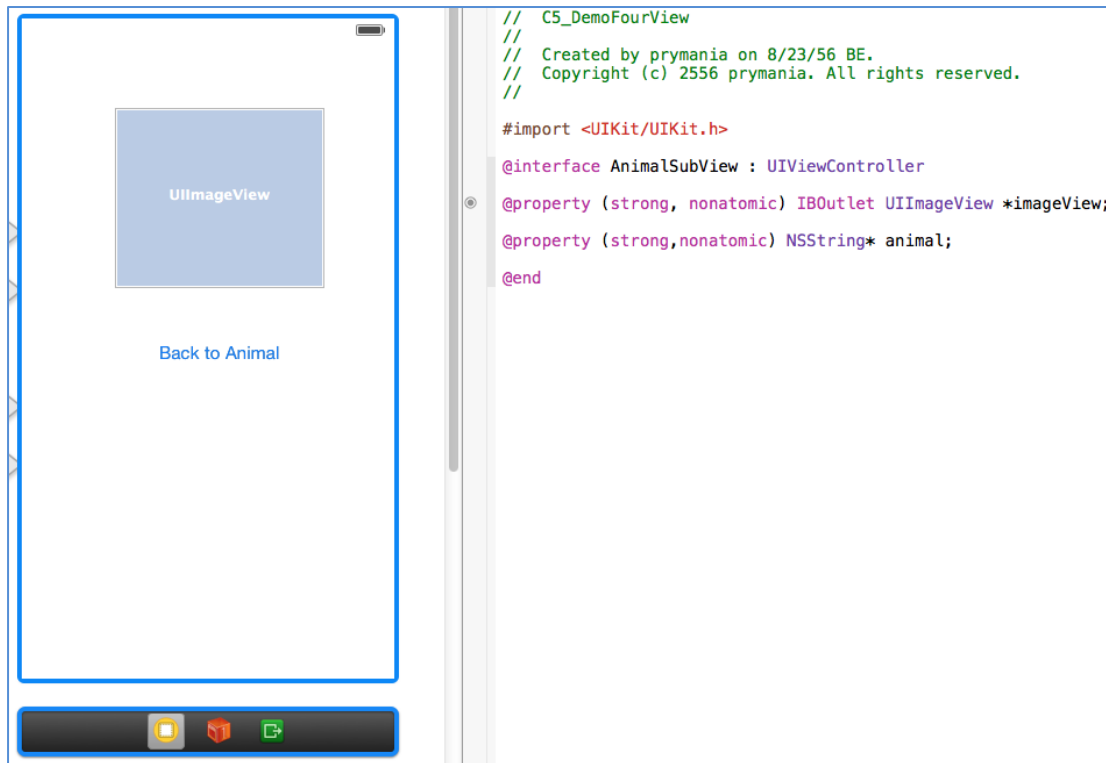
จาก mainView ปุ่ม Animal	สร้างไป AnimalView	ตั้งชื่อ animalSegue
จาก mainView ปุ่ม Fruit	สร้างไป FruitView	ตั้งชื่อ fruitSegue
จาก AnimalView ปุ่ม รูปแมว	สร้างไป AnimalSubView	ตั้งชื่อ catSegue
จาก AnimalView ปุ่ม รูปกบ	สร้างไป AnimalSubView	ตั้งชื่อ frogSegue
จาก AnimalView ปุ่ม รูปเห็ดฮอก	สร้างไป AnimalSubView	ตั้งชื่อ hedgeSegue
จาก AnimalView ปุ่ม รูปเสือดาว	สร้างไป AnimalSubView	ตั้งชื่อ leoSegue

จาก AnimalSubView ปุ่ม back to Animal	สร้างไปปุ่ม exit	เลือก backToAnimal
จาก AnimalView ปุ่ม back to main	สร้างไปปุ่ม exit	เลือก backToMain
จาก FruitView ปุ่ม back to main	สร้างไปปุ่ม exit	เลือก backToMain



4. ไฟล์ AnimalSubView.h

ไฟล์ AnimalSubView.h เชื่อมลิงก์กับ UIImageView เพื่อไว้แสดงรูปภาพ และสร้างตัวแปร NSString* animal เพื่อรับค่าชื่อสัตว์ที่เลือกจาก animalView



เมื่อ viewDidLoad เขียน code เพื่อแสดงภาพนั้นๆ สัตว์ที่เลือกจาก animalView

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view.

    NSString* pictName = [NSString stringWithFormat:@"%s.png", _animal];
    _imageView.image = [UIImage imageNamed:pictName];
}
```

5. ไฟล์ AnimalView.m

เพิ่ม code #import "AnimalSubView.h"

สร้างเมธอด prepareForSegue เพื่อเช็คค่าเส้น segue ที่ทำงานคือเส้นใด เพื่อจะได้ส่งชื่อรูปให้กับ animalSubView ได้ถูกต้อง

```
-(void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender{
    AnimalSubView *view = [segue destinationViewController];
    if([[segue identifier] isEqualToString:@"catSegue"])
        view.animal = @"cat";
    else if([[segue identifier] isEqualToString:@"frogSegue"])
        view.animal = @"frog";
    else if([[segue identifier] isEqualToString:@"hedgeSegue"])
        view.animal = @"hedgehog";
    else if([[segue identifier] isEqualToString:@"leoSegue"])
        view.animal = @"leopard";
}
```

ส่วนไฟล์ ViewController.m ซึ่งเป็นไฟล์ที่รับผิดชอบ mainScreen เราไม่จำเป็นต้องสร้างเมธอด prepareForSegue เพราะจาก mainScreen ไม่มีการส่งค่าใดๆไปให้ AnimalView หรือ FruitView เลย