

Chapter 17

iOS Multi-Touch & Gesture

Why Gesture? Why Touch

- ◎ การแตะ (touch) และการลากนิ้ว (Gesture) นั้น เป็นพัฒกรรมที่เราคุ้นเคยอยู่แล้ว เมื่อนำ touch และ gesture มาใช้กับอุปกรณ์เคลื่อนที่ การเรียนรู้เพื่อใช้งาน จึงเป็นไปอย่างง่ายดาย
- ◎ วิธีการที่ใช้งาน iOS device แบ่งออกเป็น
 - ◎ Tap แตะแล้วยกนิ้วออกไป
 - ◎ Pinch ใช้ 2 นิ้วแตะแล้วแยกนิ้วห่างออกจากกันหรือเข้าหากัน
 - ◎ Swipe แตะแล้วลากนิ้วไปในทิศทางใดทิศทางหนึ่งเร็วๆ แล้วยกนิ้วออก
 - ◎ Pan แตะด้วย 1-2 นิ้ว แล้วลากนิ้วไปในทิศทางใดทิศทางหนึ่งโดยไม่ยกนิ้วออก
 - ◎ Press-and-Hold แตะนิ้วค้างเอาไว้ระยะเวลาหนึ่ง

Easy to understand, but Difficult to implement

- ◉ แต่การ implement เพื่อให้รองรับการ touch และ gesture นั้น ไม่ใช่เรื่องง่าย สาเหตุแรก เพราะเราคุ้นเคยกับการเขียนโปรแกรมเพื่อรองรับ mouse และ keyboard และสาเหตุที่ 2 เพราะอุปกรณ์รุ่นใหม่ๆ รองรับการสัมผัสแบบหลายจุด (multi-touch) นั่นหมายความว่า วิธีจัดการกับ input ก็จะซุ่มยากและซับซ้อนมากยิ่งขึ้น
- ◉ การวิเคราะห์ถึงพฤติกรรมของ user ก็จะยิ่งซับซ้อนและกำกับ เช่น
 - ◉ เราจะแยกได้อย่างไรว่าการที่ user แตะ 2 ครั้ง ไม่ได้หมายความว่าเป็นการ Tap 2 ครั้ง แต่ user ตั้งใจ Double-tap
 - ◉ จะแยกได้อย่างไร Tap และปล่อยนิ้ว กับ Tap และค้างนิ้วไว้
 - ◉ จะแยกแยกการแตะ 2 นิ้วแล้วลาก กับการแตะ 2 นิ้วแล้วถ่างออก (pan) ได้ยังไง
 - ◉ Etc...
- ◉ ใน iOS SDK จะ implement ประเภทของสัมผัสแต่ละแบบไว้แล้ว หลักการก็คือ สร้าง object ที่ handle touch ที่ต้องการ และ binding object นั้นกับ UI Control ที่ต้องการให้รับ input แบบ touch ได้

Lab 1-4 : Basic Touch Handling (1/4)

◎ วัตถุประสงค์

- ◎ เขียน iOS App เพื่อจัดการกับการสัมผัสบน view ด้วย UITouch

◎ ขั้นตอน

- ◎ สร้าง project
- ◎ เพิ่ม property สำหรับรองรับ touch และอ่านค่า point ที่ touch
- ◎ Implement delegate method ของ view เพื่อจัดการกับรูปแบบของ touch ต่างๆ

(Optional)

Task : Create Project (2/4)

1. จาก Xcode สร้าง project ใหม่ โดยเลือก iOS > Application > Single View Application
2. ตั้งชื่อ project ว่า “BasicTouch” และเลือก Devices เป็น iPhone
3. Click “Next” เลือก folder ที่จะ save project แล้ว click ปุ่ม “Create”
4. เปิดไฟล์ “ViewController.h” เพิ่ม property ดังนี้

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

@interface ViewController : UIViewController

@property (nonatomic, strong) UITouch * trackedTouch;
@property (nonatomic) CGPoint startPoint;

@end
```

(Optional)

Task : Coding – Handle Touch (3/4)

5. เปิดไฟล์ “ViewController.m” เพิ่ม method ชื่อ “touchBegan:withEvent:” ดังนี้

```
- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
{
    if (self.trackedTouch == nil) {
        self.trackedTouch = [touches anyObject];
        self.startPoint = [self.trackedTouch locationInView:self.view];
        NSLog(@"Touch Begin");
    }
}
```

6. เพิ่ม constant variable เพื่อกำหนด threshold และ method “touchesMoved:withEvent:” สำหรับ handle การ swipe

```
#define MIN_SWIPE_X_THRESHOLD 30
#define MAX_SWIPE_Y_THRESHOLD 30

- (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event
{
    CGPoint currentPoint = [self.trackedTouch locationInView:self.view];
    if ((currentPoint.x - self.startPoint.x > MIN_SWIPE_X_THRESHOLD) &&
        (ABS(currentPoint.y - self.startPoint.y) < MAX_SWIPE_Y_THRESHOLD))
    {
        NSLog(@"Seem like swipe...");
    }
}
```

(Optional)

Task : Coding – Handle Touch (4/4)

7. เพิ่ม method “touchesEnded:withEvent:” เพื่อ handle การปล่อยนิ้ว

```
- (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event
{
    if (self.trackedTouch && [touches containsObject:self.trackedTouch]) {
        self.trackedTouch = nil;
        NSLog(@"Release track touch");
    }
}
```

8. Run โปรแกรมเพื่อดูผลลัพธ์

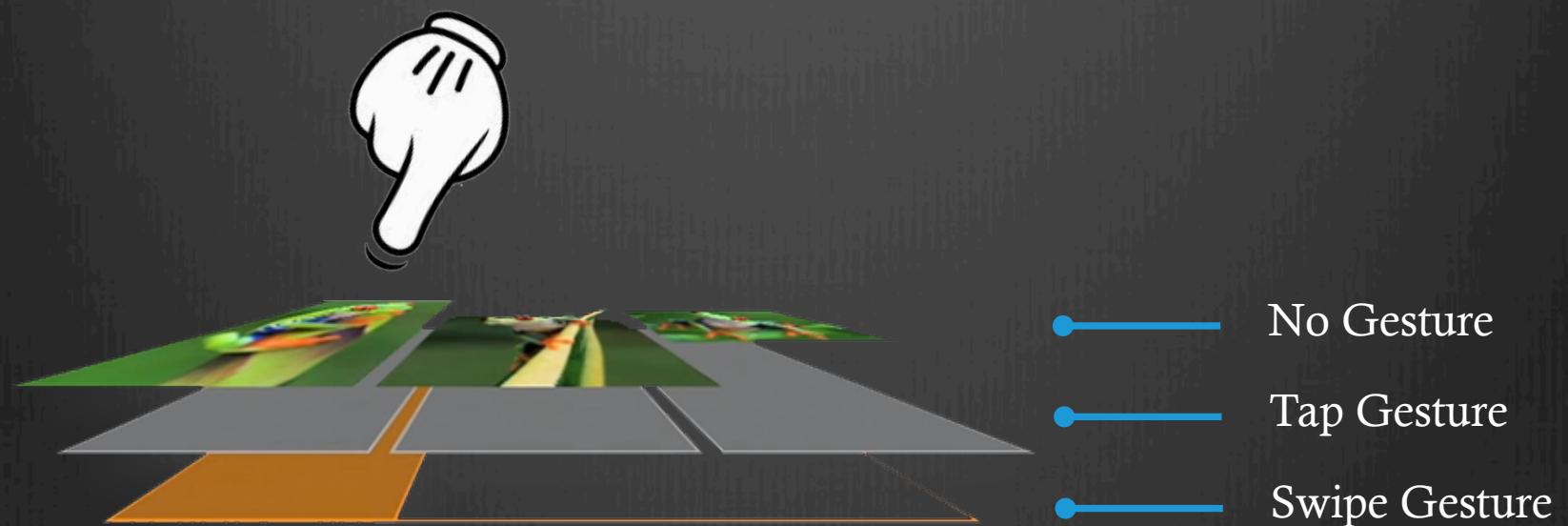
- แตะ (click) แล้วปล่อย
- แตะแล้วลากไปตาม view (ดูค่าจาก log)

(Optional)

Gesture Recognizer

- ◎ ตัวอย่างที่ผ่านมา เป็นการจัดการกับ Touch และ Gesture อย่างง่าย แต่ไม่ควรใช้วิธีนั้น เพราะการจัดการกับ gesture จะยุ่งยากมาก
- ◎ iOS SDK เตรียม class เพื่อใช้จัดการกับ Gesture ให้แล้ว ให้ใช้ผ่าน object ของ class UIGestureRecognizer แทน
- ◎ ขั้นตอนในการจัดการกับ gesture อย่างง่าย
 - ◎ สร้าง object ของ UIGestureRecognizer
 - ◎ Implement method สำหรับ handle gesture แต่ละแบบ
 - ◎ Add object ที่สร้างขึ้นมาเข้าไปใน UIView

How Gesture Recognizer Works



UIGestureRecognizer

- ⦿ รองรับ iOS 3.2 และ 4.0 ขึ้นไป
- ⦿ UIGestureRecognizer เป็น Abstract class
- ⦿ การ handle gesture ต้องใช้ subclass ต่างๆ ดังนี้
 - ⦿ UITapGestureRecognizer
 - ⦿ UIPinchGestureRecognizer
 - ⦿ UISwipeGestureRecognizer
 - ⦿ UIPanGestureRecognizer
 - ⦿ UILongPressGestureRecognizer
 - ⦿ UIRotationGestureRecognizer

Lab 2-4 : Basic Touch Handling (1/4)

◎ วัตถุประสงค์

- ◎ เขียน iOS App เพื่อจัดการกับการสัมผัสบน view ด้วย Gesture Recognizer

◎ ขั้นตอน

- ◎ สร้าง project ใหม่
- ◎ สร้าง Object “Swipe Gesture Recognizer” 2 ตัวเพื่อ handle up/down และ left/right
- ◎ เพิ่ม object ทั้ง 2 ตัวเข้าไปใน view ที่ต้องการให้รับ gesture

(Optional)

Task : Create Project (2/4)

1. จาก Xcode สร้าง project ใหม่ โดยเลือก iOS > Application > Single View Application
2. ตั้งชื่อ project ว่า “BasicGesture” และเลือก Devices เป็น iPhone
3. Click “Next” เลือก folder ที่จะ save project และ click ปุ่ม “Create”
4. เปิดไฟล์ “ViewController.m” เพิ่ม method “swipeRecognized:” ดังนี้

```
- (void)swipeRecognized:(UIGestureRecognizer *)recognizer
{
    NSLog(@"Seem like swipe...");
}
```

(Optional)

Task : Create Project (3/4)

5. เพิ่ม code ใน method “viewDidLoad:” เพื่อสร้าง object swipeGestureRecognizer ดังนี้

```
- (void)viewDidLoad
{
    [super viewDidLoad];

    UISwipeGestureRecognizer *swipeLeftRight = [[UISwipeGestureRecognizer alloc]
                                                initWithTarget:self
                                              action:@selector(swipeRecognized:)];
    [swipeLeftRight setDirection:(UISwipeGestureRecognizerDirectionRight |
                             UISwipeGestureRecognizerDirectionLeft )];
    [self.view addGestureRecognizer:swipeLeftRight];

    UISwipeGestureRecognizer *swipeUpDown = [[UISwipeGestureRecognizer alloc]
                                             initWithTarget:self
                                           action:@selector(swipeRecognized:)];
    [swipeUpDown setDirection:(UISwipeGestureRecognizerDirectionUp |
                           UISwipeGestureRecognizerDirectionDown )];
    [self.view addGestureRecognizer:swipeUpDown];
}
```

(Optional)

Task : Create Project (4/4)

- Run โปรแกรมเพื่อดูผลลัพธ์

Note : เราไม่สามารถสร้าง Swipe Gesture Recognizer เพื่อให้ handle
ทั้ง Up/Down และ Left/Right พร้อมกันได้ (by-design)

(Optional)

Lab 3-4 : Image Viewer (1/8)

◎ วัตถุประสงค์

- ◎ เขียน iOS App เพื่อแสดงภาพและสามารถ zoom ได้โดยใช้ Gesture Recognizer

◎ ขั้นตอน

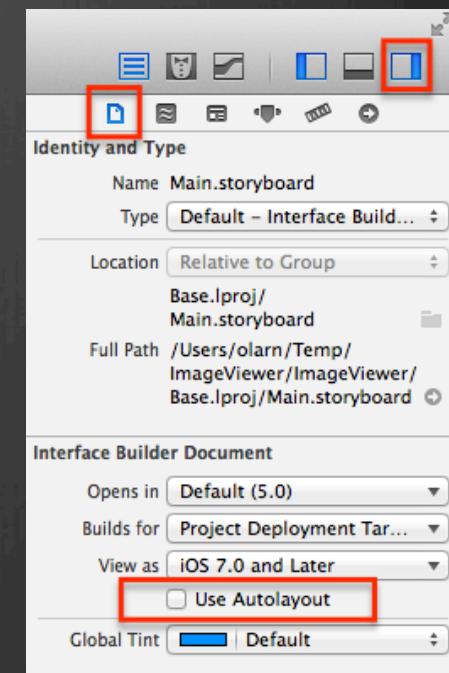
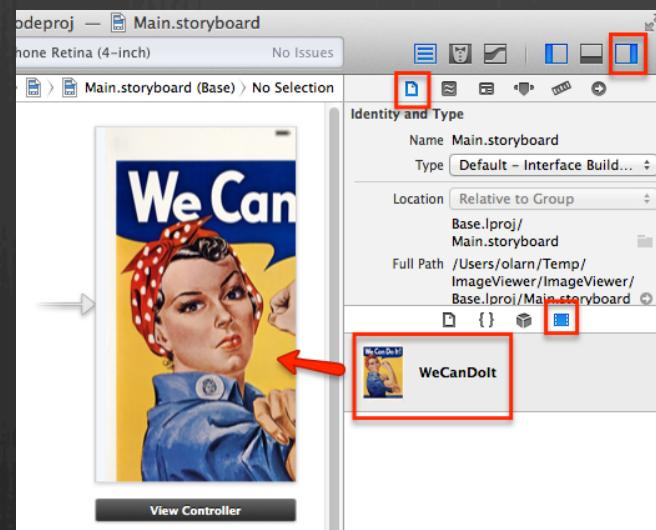
- ◎ สร้าง project ใหม่
- ◎ เพิ่ม Scroll View ลงบน view และเพิ่ม Image View ใน Scroll view โดยให้ขนาดของ image view ใหญ่กว่า scroll view
- ◎ เขียน code เพื่อ implement delegate ของ scroll view
- ◎ เขียน code เพื่อ handle Gesture Recognizer

Task : Create Project (2/8)

1. จาก Xcode สร้าง project ใหม่ โดยเลือก iOS > Application > Single View Application
2. ตั้งชื่อ project ว่า “ImageViewer” และเลือก Devices เป็น iPhone
3. Click “Next” เลือก folder ที่จะ save project แล้ว click ปุ่ม “Create”
4. ทำการ import รูปเข้ามาใน project โดยเปิดไฟล์ “Images.xcassets” แล้ว click ขวาที่พื้นที่ว่างๆ ได้คำว่า LaunchImage แล้วเลือกเมนู “Import...” เปิดไปที่ folder “../Resources/Day5 - Lab16/” แล้วเลือกไฟล์ “WeCanDoIt.png” แล้ว click “Open”

Task : Create Project (3/8)

5. เปิดไฟล์ Main.storyboard และเปิด File Inspector บน Inspector Pane และเลือก check box “Use Auto layout” ออก
6. เพิ่ม control “Scroll View” ลงบน view โดยให้ขนาดของ scroll view เต็มพื้นที่ของ view พอดี
7. เปิด Media Library บน Library Pane และลากรูป WeCanDoIt มาวางลงบน scroll view (ขนาดของ image จะใหญ่เกินกว่า scroll view มาก)



Task : Create Project (4/8)

8. เปลี่ยน editor mode เป็น Assistant editor และผูก Outlet โดยที่

- Scroll View ตั้งชื่อ property ว่า “imageScrollView”
- Image View ตั้งชื่อ property ว่า “imageView”

9. เปิดไฟล์ “ViewController.h” เพิ่ม import UI Scroll View Delegate ดังนี้

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

@interface ViewController : UIViewController <UIScrollViewDelegate>
@property (weak, nonatomic) IBOutlet UIScrollView *imageScrollView;
@property (weak, nonatomic) IBOutlet UIImageView *imageView;
@end
```

10. เปิดไฟล์ “ViewController.m” เพิ่ม static variable เพื่อกำหนด tag id ของ imageView และ step สำหรับ zoom

```
...
#define ZOOM_VIEW_TAG 100
#define ZOOM_STEP 1.5
```

```
@implementation ViewController
```

Task : Create Project (5/8)

11. เพิ่ม code ใน method “viewDidLoad” ดังนี้

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    self.imageViewScrollView.delegate = self;

    float minimumScale = [self.imageViewScrollView frame].size.width /
        [self.imageView frame].size.width;
    [self.imageViewScrollView setMinimumZoomScale:minimumScale];
    [self.imageViewScrollView setZoomScale:minimumScale];
}
```

12. เพิ่ม method “” เพื่อ implement delegate method ของ UI Scroll View

```
- (UIView *)viewForZoomingInScrollView:(UIScrollView *)scrollView {
    return [scrollView viewWithTag:ZOOM_VIEW_TAG];
}
```

13. Run เพื่อดูผลลัพธ์ (สังเกตว่าเราสามารถ zoom ได้แล้ว)



Task : Create Project (6/8)

14. เพิ่ม method “zoomRectForScale:” เพื่อช่วยคำนวณ scale ในการ zoom เข้า/ออก เมื่อ tap และ double tap บนภาพ

```
- (CGRect)zoomRectForScale:(float)scale withCenter:(CGPoint)center
{
    CGRect zoomRect;

    zoomRect.size.height = [self.scrollView frame].size.height / scale;
    zoomRect.size.width  = [self.scrollView frame].size.width / scale;
    zoomRect.origin.x   = center.x - (zoomRect.size.width / 2.0);
    zoomRect.origin.y   = center.y - (zoomRect.size.height / 2.0);

    return zoomRect;
}
```

Task : Create Project (7/8)

15. เพิ่ม object ของ UITapGestureRecognizer เพื่อรับ event ของ touch ใน method “viewDidLoad” ของ ViewController.m

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    self.imageViewScrollView.delegate = self;

    self.imageView.tag = ZOOM_VIEW_TAG;

    UITapGestureRecognizer *doubleTap = [[UITapGestureRecognizer alloc]
                                         initWithTarget:self
                                         action:@selector(handleDoubleTap:)];
    UITapGestureRecognizer *twoFingerTap = [[UITapGestureRecognizer alloc]
                                         initWithTarget:self
                                         action:@selector(handleTwoFingerTap:)];
    [doubleTap setNumberOfTapsRequired:2];
    [twoFingerTap setNumberOfTouchesRequired:2];

    [self.imageView addGestureRecognizer:doubleTap];
    [self.imageView addGestureRecognizer:twoFingerTap];

    float minimumScale = [self.imageViewScrollView frame].size.width /
                           [self.imageView frame].size.width;
    [self.imageViewScrollView setMinimumZoomScale:minimumScale];
    [self.imageViewScrollView setZoomScale:minimumScale];
}
```

Task : Create Project (8/8)

16. เพิ่ม method “handleDoubleTap:” และ “handleTwoFingerTap” เพื่อจัดการกับ การ zoom เข้าเมื่อแตะ 2 ครั้ง และ zoom ออกเมื่อแตะ 2 นิ้วพร้อมกัน

```
- (void)handleDoubleTap:(UIGestureRecognizer *)gestureRecognizer
{
    float newScale = [self.scrollView zoomScale] * ZOOM_STEP;
    CGRect zoomRect = [self zoomRectForScale:newScale
                           withCenter:[gestureRecognizer locationInView:gestureRecognizer.view]];
    [self.scrollView zoomToRect:zoomRect animated:YES];
}

- (void)handleTwoFingerTap:(UIGestureRecognizer *)gestureRecognizer
{
    float newScale = [self.scrollView zoomScale] / ZOOM_STEP;
    CGRect zoomRect = [self zoomRectForScale:newScale
                           withCenter:[gestureRecognizer
                           locationInView:gestureRecognizer.view]];
    [self.scrollView zoomToRect:zoomRect animated:YES];
}
```

17. Run โปรแกรมเพื่อดูผลลัพธ์

Lab 4-4 : Image Viewer (1/11)

◎ วัตถุประสงค์

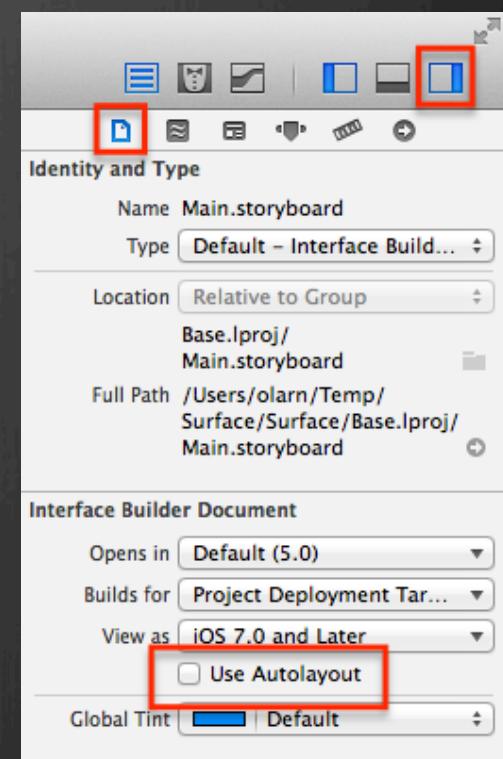
- ◎ เขียน iOS App เพื่อแสดงภาพและสามารถ zoom ได้โดยใช้ Gesture Recognizer

◎ ขั้นตอน

- ◎ สร้าง project ใหม่
- ◎ เพิ่ม Scroll View ลงบน view และเพิ่ม Image View ใน Scroll view โดยให้ขนาดของ image view ใหญ่กว่า scroll view
- ◎ เขียน code เพื่อ implement delegate ของ scroll view
- ◎ เขียน code เพื่อ handle Gesture Recognizer

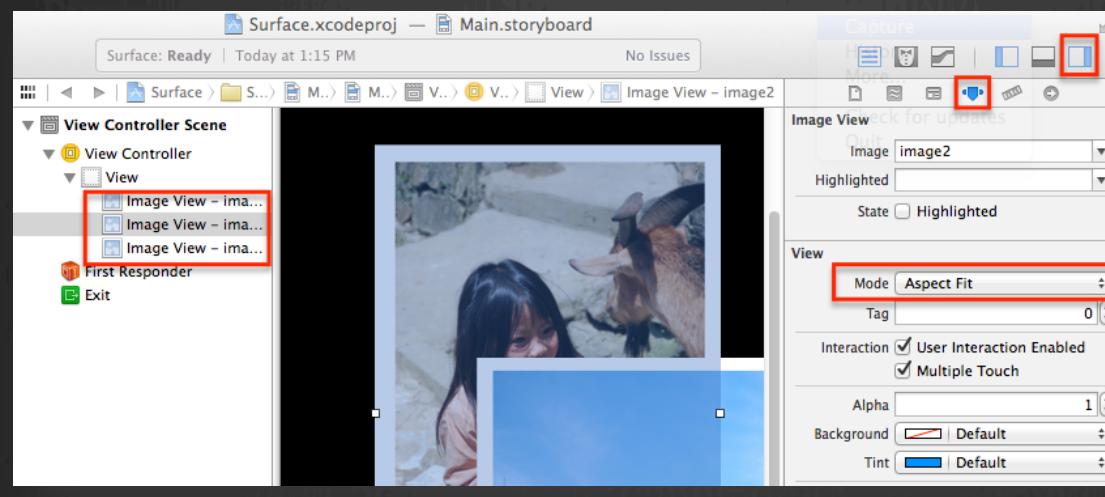
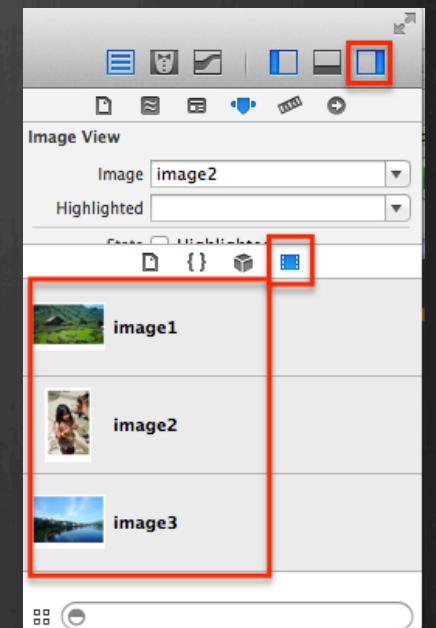
Task : Create Project (2/11)

1. จาก Xcode สร้าง project ใหม่ โดยเลือก iOS > Application > Single View Application
2. ตั้งชื่อ project ว่า “Surface” และเลือก Devices เป็น iPad
3. Click “Next” เลือก folder ที่จะ save project แล้ว click ปุ่ม “Create”
4. ทำการ import รูปเข้ามาใน project โดยเปิดไฟล์ “Images.xcassets” และ click ขวาที่พื้นที่ว่างๆ ได้คำว่า LaunchImage แล้วเลือกเมนู “Import...” เปิดไปที่ folder “../Resources/Day5 - Lab16/” และเลือกไฟล์ “image1.png”, “image2.png”, “image3.png” แล้ว click “Open”
5. เปิดไฟล์ Main.storyboard และเปิด File Inspector บน Inspector Pane และเอา check box “Use Auto layout” ออกร



Task : Create Project (3/11)

6. เปลี่ยน background ของ view เป็นสีดำ โดย click เลือก “View” บน Document outline และเปิด Attributes inspector จากนั้นเปลี่ยน property “Background Color” เป็น “Black Color”
7. เปิด Media Library บน Library pane และลากกรุปทั้ง 3 รูปมาวางบน view
8. เปิด Attribute Inspector และเปลี่ยน property “View > Mode” ของ Image view ทั้ง 3 object เป็น “Aspect Fit”



Task : Create Project (4/11)

9. เปลี่ยน editor mode เป็น Assistant editor และผูก Outlet ให้ Image view ทั้ง 3 ตัว และตั้งชื่อ property ว่า “imageView1”, “imageView2”, “imageView3”
10. เปิดไฟล์ “ViewController.h” เพิ่ม import UI Gesture Recognizer Delegate และเพิ่ม method “addGestureRecognizerToImageView:” ดังนี้

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

@interface ViewController : UIViewController <UIGestureRecognizerDelegate>

@property (weak, nonatomic) IBOutlet UIImageView *imageView1;
@property (weak, nonatomic) IBOutlet UIImageView *imageView2;
@property (weak, nonatomic) IBOutlet UIImageView *imageView3;

@property (weak, nonatomic) UIView *imageForReset;

- (void)addGestureRecognizerToImageView:(UIView *)imageView;

@end
```

Task : Create Project (5/11)

11. เปิดไฟล์ “ViewController.m” เพิ่ม implement code ใน method “addGestureRecognizerToImageView:” ดังนี้

```
- (void)addGestureRecognizerToImageView:(UIImageView *) imageView
{
    UITapGestureRecognizer *tapGesture =
        [[UITapGestureRecognizer alloc] initWithTarget:self action:@selector(tapImageView:)];
    UIRotationGestureRecognizer *rotationGesture =
        [[UIRotationGestureRecognizer alloc] initWithTarget:self action:@selector(rotateImageView:)];
    UIPinchGestureRecognizer *pinchGesture =
        [[UIPinchGestureRecognizer alloc] initWithTarget:self action:@selector(scaleImageView:)];
    [pinchGesture setDelegate:self];

    UIPanGestureRecognizer *panGesture =
        [[UIPanGestureRecognizer alloc] initWithTarget:self action:@selector(panImageView:)];
    [panGesture setMaximumNumberOfTouches:2];
    [panGesture setDelegate:self];

    UILongPressGestureRecognizer *longPressGesture =
        [[UILongPressGestureRecognizer alloc] initWithTarget:self action:@selector(showResetMenu:)];

    [imageView addGestureRecognizer:tapGesture];
    [imageView addGestureRecognizer:rotationGesture];
    [imageView addGestureRecognizer:pinchGesture];
    [imageView addGestureRecognizer:panGesture];
    [imageView addGestureRecognizer:longPressGesture];
}
```

Task : Create Project (6/11)

12. เพิ่ม code ใน method “viewDidLoad:” เพื่อผูก gesture recognizer เข้ากับ Image View แต่ละตัว

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    [self addGestureRecognizerToImageView:self.imageView1];
    [self addGestureRecognizerToImageView:self.imageView2];
    [self addGestureRecognizerToImageView:self.imageView3];
}
```

13. เพิ่ม method adjustAnchorPointForGestureRecognizer: สำหรับเป็น utility method เพื่อคำนวณตำแหน่งของ view ที่ gesture recognizer เกาะอยู่

```
- (void)adjustAnchorPointForGestureRecognizer:(UIGestureRecognizer *)gestureRecognizer
{
    if ( gestureRecognizer.state == UIGestureRecognizerStateBegan ) {
        UIView *itsView = gestureRecognizer.view;
        CGPoint locationInView = [gestureRecognizer locationInView:itsView];
        CGPoint locationInSuperview = [gestureRecognizer locationInView:itsView.superview];
        itsView.layer.anchorPoint = CGPointMake(locationInView.x / itsView.bounds.size.width,
                                                locationInView.y / itsView.bounds.size.height);
        itsView.center = locationInSuperview;
    }
}
```

Task : Create Project (7/11)

14. เพิ่ม code ของ method showResetMenu: เพื่อแสดง pop up เมื่อเราแตะรูปนั้นนานระยะเวลาหนึ่ง

```
- (void)showResetMenu:(UILongPressGestureRecognizer *)gestureRecognizer
{
    if ([gestureRecognizer state] == UIGestureRecognizerStateBegan) {
        UIMenuController *menuController = [UIMenuController sharedMenuController];
        UIMenuItem *resetMenuItem = [[UIMenuItem alloc] initWithTitle:@"Reset"
                                                               action:@selector(resetImageView:)];
        CGPoint location = [gestureRecognizer locationInView:[gestureRecognizer view]];

        [self becomeFirstResponder];
        [menuController setMenuItems:[NSArray arrayWithObject:resetMenuItem]];
        [menuController setTargetRect:CGRectMake(location.x, location.y, 0, 0)
                               inView:[gestureRecognizer view]];
        [menuController setMenuVisible:YES animated:YES];

        self.imageViewForReset = [gestureRecognizer view];
    }
}
```

Task : Create Project (8/11)

15. เพิ่ม code ของ method resetImageView: เพื่อ reset ขนาดและมุมของ image ให้เหมือนเริ่มต้นเมื่อแตะ pop up “Reset”

```
- (void)resetImageView:(UIViewControllerAnimated *)controller
{
    CGPoint locationInSuperview =
        [self.imageForReset convertPoint:CGPointMake(
            CGRectGetMidX(self.imageForReset.bounds),
            CGRectGetMidY(self.imageForReset.bounds))
         toView:[self.imageForReset superview]];

    [[self.imageForReset layer] setAnchorPoint:CGPointMake(0.5, 0.5)];
    [self.imageForReset setCenter:locationInSuperview];

    [UIView beginAnimations:nil context:nil];
    [self.imageForReset setTransform:CGAffineTransformIdentity];
    [UIView commitAnimations];
}
```

16. เพิ่ม code ของ method canBecomeFirstResponder เพื่อให้ popup menu สามารถแสดงขึ้นมาได้

```
- (BOOL)canBecomeFirstResponder
{
    return YES;
}
```

Task : Create Project (9/11)

17. เพิ่ม code ของ method tapImage: เพื่อดึง imageView ที่ถูกแตะขึ้นมาเป็น layer บนสุด
(กรณีที่รูปนั้นอยู่ใต้รูปอื่น รูปนั้นจะถูกดึงขึ้นมาอยู่บนรูปอื่น)

```
- (void)tapImageView:(UITapGestureRecognizer *)gestureRecognizer
{
    UIView *imageView = [gestureRecognizer view];
    [self.view bringSubviewToFront:imageView];
}
```

18. เพิ่ม code ของ method panImageView: เพื่อขยาย image ตามนิ้วเมื่อแตะแล้วลากนิ้วไปมา

```
- (void)panImageView:(UIPanGestureRecognizer *)gestureRecognizer
{
    UIView *imageView = [gestureRecognizer view];
    [self.view bringSubviewToFront:imageView];

    [self adjustAnchorPointForGestureRecognizer:gestureRecognizer];

    if ([gestureRecognizer state] == UIGestureRecognizerStateBegan ||
        [gestureRecognizer state] == UIGestureRecognizerStateChanged)
    {
        CGPoint translation =
            [gestureRecognizer translationInView:[imageView superview]];

        [imageView setCenter:CGPointMake([imageView center].x + translation.x,
                                         [imageView center].y + translation.y)];
        [gestureRecognizer setTranslation:CGPointZero inView:[imageView superview]];
    }
}
```

Task : Create Project (10/11)

19. เพิ่ม code ของ method rotateImageView: เพื่อหมุนรูปตามนิ้ว

```
- (void)rotateImageView:(UIRotationGestureRecognizer *)gestureRecognizer
{
    [self adjustAnchorPointForGestureRecognizer:gestureRecognizer];

    if ([gestureRecognizer state] == UIGestureRecognizerStateBegan ||
        [gestureRecognizer state] == UIGestureRecognizerStateChanged)
    {
        [gestureRecognizer view].transform =
            CGAffineTransformRotate([[gestureRecognizer view] transform],
                                  [gestureRecognizer rotation]);
        [gestureRecognizer setRotation:0];
    }
}
```

20. เพิ่ม code ของ method scaleImageView: เพื่อขยายหรือย่อรูปเมื่อแตะ 2 นิ้วแล้วลากออก / เข้าหากัน

```
- (void)scaleImageView:(UIPinchGestureRecognizer *)gestureRecognizer
{
    [self adjustAnchorPointForGestureRecognizer:gestureRecognizer];

    if ([gestureRecognizer state] == UIGestureRecognizerStateBegan ||
        [gestureRecognizer state] == UIGestureRecognizerStateChanged)
    {
        [gestureRecognizer view].transform =
            CGAffineTransformScale([[gestureRecognizer view] transform],
                                 [gestureRecognizer scale],
                                 [gestureRecognizer scale]);
        [gestureRecognizer setScale:1];
    }
}
```

Task : Create Project (11/11)

19. เพิ่ม code ของ method “shouldRecognizeSimultaneouslyWithGestureRecognizer:” เพื่อให้ gesture recognizer สามารถทำงานพร้อมๆ กันได้

```
- (BOOL)gestureRecognizer:(UIGestureRecognizer *)gestureRecognizer  
shouldRecognizeSimultaneouslyWithGestureRecognizer:(UIGestureRecognizer *)otherGestureRecognizer  
{  
    if (gestureRecognizer.view != self.imageView1 &&  
        gestureRecognizer.view != self.imageView2 &&  
        gestureRecognizer.view != self.imageView3)  
    {  
        return NO;  
    }  
  
    if (gestureRecognizer.view != otherGestureRecognizer.view)  
        return NO;  
  
    if ([gestureRecognizer isKindOfClass:[UILongPressGestureRecognizer class]] ||  
        [otherGestureRecognizer isKindOfClass:[UILongPressGestureRecognizer class]])  
    {  
        return NO;  
    }  
    return YES;  
}
```

20. Run เพื่อดูผลลัพธ์

หมายเหตุ:

- กดปุ่ม *alt (Option)* บน *keyboard* และ *click* เพื่อจำลองการแตะ 2 นิ้ว
- ในระหว่างที่กด *Option* ถ้ากด *Shift* บน *keyboard* จะสามารถเลื่อน *cursor* ทั้ง 2 จุดไปมาได้