iOS 程式設計

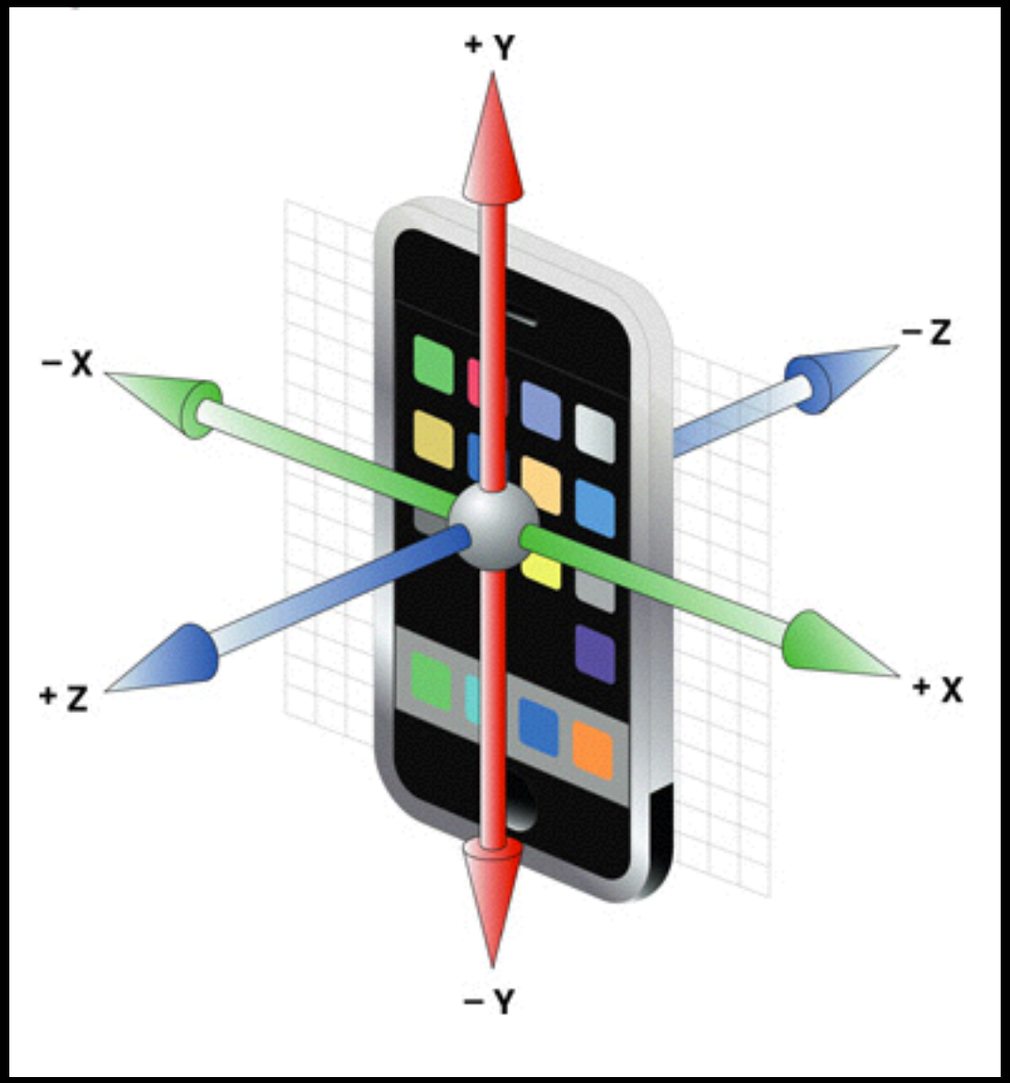
重力感應器與陀螺儀

Brian Wang 王昱景
brian.wang.frontline@gmail.com

何謂重力感應器

- 重力感應器可用來感應重力的方向與數值
- 提供三個軸項來判斷裝置的重力狀態
- 可用於判斷方向
  - 介面使用
  - 遊戲使用

重力感應器的方向



UIAccelerometer

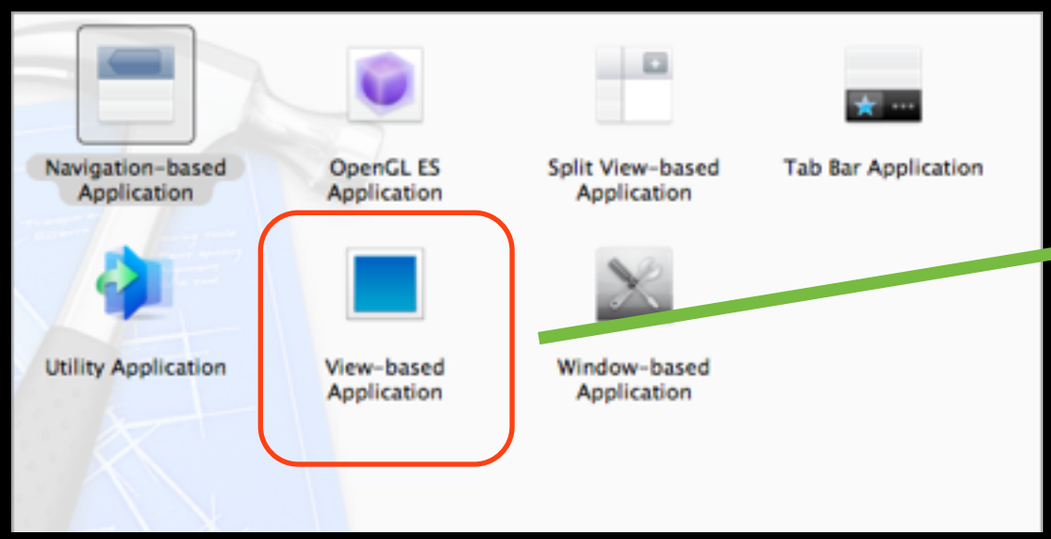
- 初始化重力感應器
  - [UIAccelerometer sharedAccelerometer]
- 屬性設定
  - (float) updateInterval
- 委派方法
  - - (void)accelerometer:(UIAccelerometer *) accelerometer didAccelerate:(UIAcceleration *) acceleration

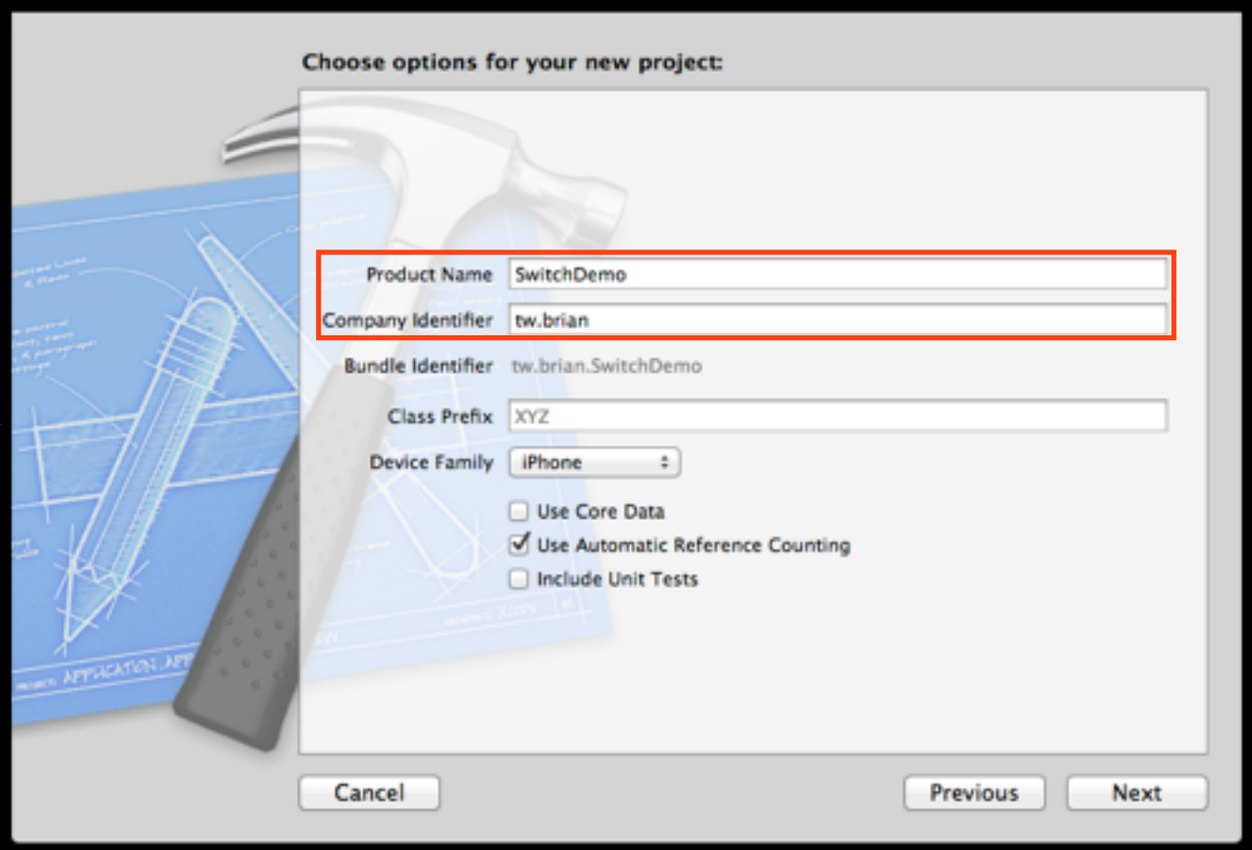
UIAccelerometer更新頻率

- 系統更新頻率範圍為10~100 Hz
  - 依照需求來進行設定
    - 不要更新的過於頻繁
- 建議數值
  - 遊戲輸入：30~60 Hz
  - 方向偵測：10~20 Hz
- 數值設定：1/50 = 50 Hz

CoreMotionSample01(1/10)

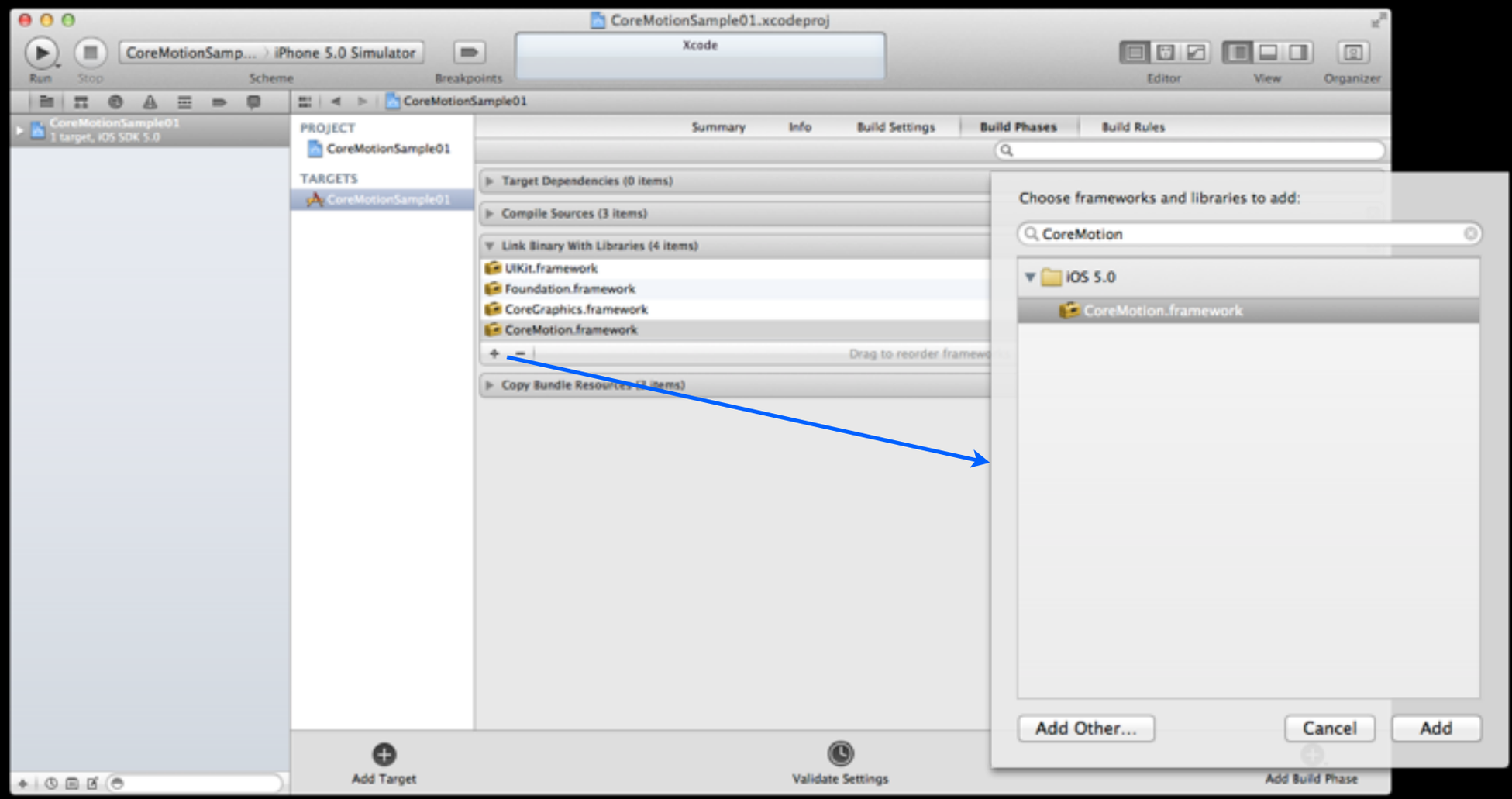
1.建置一個View-based 的專案，名稱『CoreMotionSample01』。





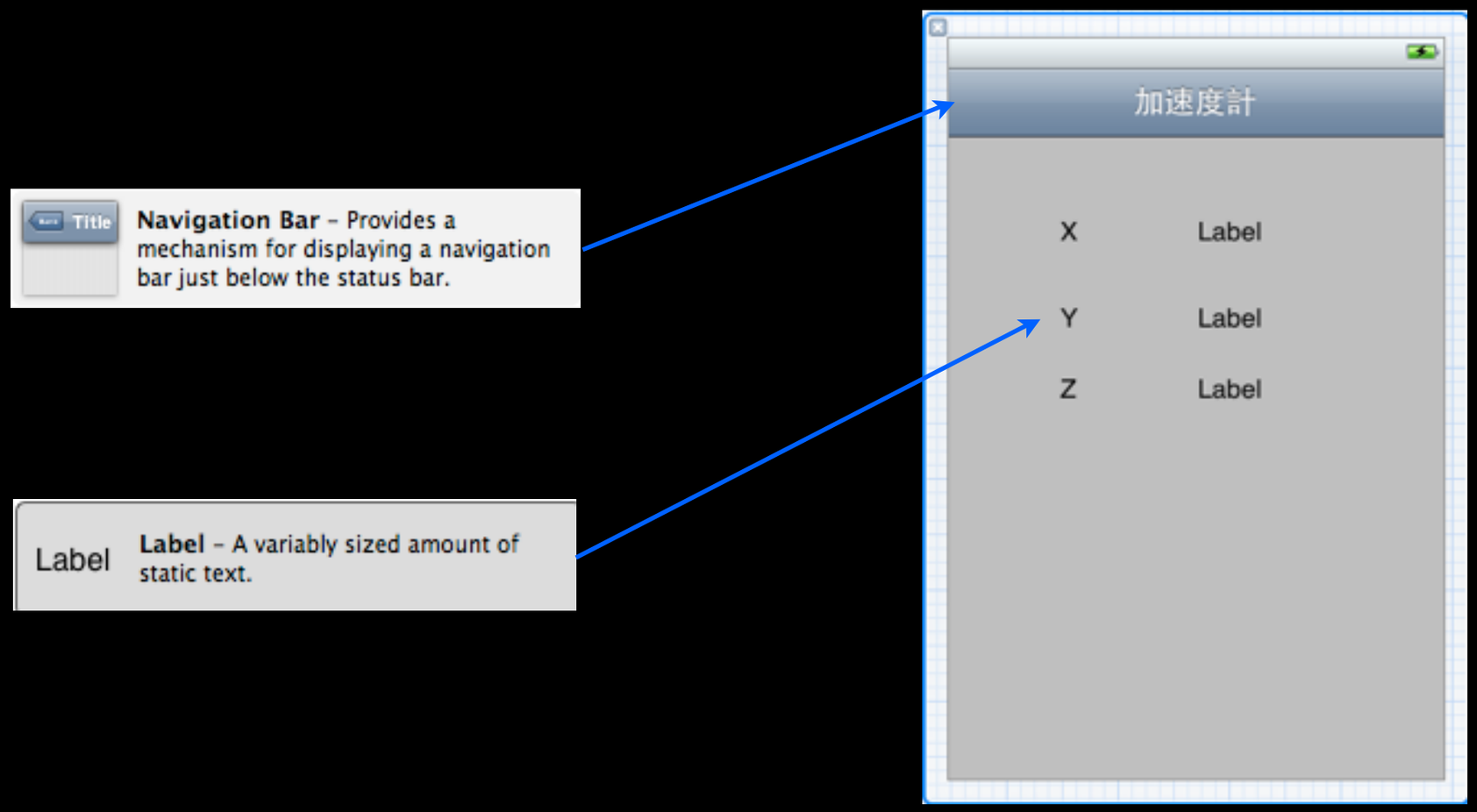
CoreMotionSample01(2/10)

2.從Targets > Build Phases > Link Binary With Libaries加入CoreMotion Framework



CoreMotionSample01(3/10)

3. 在View上加入一個Navigation Bar及6個元件 Label，並將View的配置完成如下圖所示。



CoreMotionSample01(4/10)

4.在 CoreMotionSample01ViewController.h 中加入3個 UILabel Outlet，名稱為如下所示並與 Label 元件建立參考。

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

@interface CoreMotionSample01ViewController : UIViewController {

    IBOutlet UILabel *accelerationx;
    IBOutlet UILabel *accelerationy;
    IBOutlet UILabel *accelerationz;

}

@property (nonatomic,retain) IBOutlet UILabel *accelerationx;
@property (nonatomic,retain) IBOutlet UILabel *accelerationy;
@property (nonatomic,retain) IBOutlet UILabel *accelerationz;

@end
```

CoreMotionSample01(5/10)

5. 加入 CoreMotion/CoreMotion.h 的 import，並增加 CMMotionManager 的記憶體指標。
6. 新增5個方法，名稱如下所示。

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

@interface CoreMotionSample01ViewController : UIViewController {

    CMMotionManager *motionManager;
    IBOutlet UILabel *accelerationx;
    IBOutlet UILabel *accelerationy;
    IBOutlet UILabel *accelerationz;

}

@property (nonatomic,retain) IBOutlet UILabel *accelerationx;
@property (nonatomic,retain) IBOutlet UILabel *accelerationy;
@property (nonatomic,retain) IBOutlet UILabel *accelerationz;

- (void)startGetAcceleration;
- (void)stopGetAcceleration;
- (void)getAcceleration;
- (void)showAcceleration;
- (void)autoGetAcceleration;

@end
```

CoreMotionSample01(6/10)

7. 在 CoreMotionSample01ViewController.m 中加入 double 的陣列，初始化大小為 3。

```
#import "CoreMotionSample01ViewController.h"

@implementation CoreMotionSample01ViewController

double accelerationData[3];

@synthesize accelerationx, accelerationy, accelerationz;
```

CoreMotionSample01(7/10)

8.實作 startGetAcceleration 方法。

```
- (void)startGetAcceleration
{
    motionManager.accelerometerUpdateInterval = 0.01;
    [motionManager startAccelerometerUpdates];
}
```

9. 實作 stopGetAcceleration 方法。

```
- (void)stopGetAcceleration
{
    if (motionManager.accelerometerActive) {
        [motionManager stopAccelerometerUpdates];
    }

}
```

CoreMotionSample01(8/10)

9.實作 getAcceleration 方法。

```
- (void)getAcceleration
{
    CMAccelerometerData *newestAccel =
motionManager.accelerometerData;
    accelerationData[0] = newestAccel.acceleration.x;
    accelerationData[1] = newestAccel.acceleration.y;
    accelerationData[2] = newestAccel.acceleration.z;
}
```

10.實作 showAcceleration 方法。

```
- (void)showAcceleration
{
    [self getAcceleration];
    accelerationx.text = [NSString stringWithFormat:@"%f",accelerationData[0]];
    accelerationy.text = [NSString stringWithFormat:@"%f",accelerationData[1]];
    accelerationz.text = [NSString stringWithFormat:@"%f",accelerationData[2]];

}
```

CoreMotionSample01(9/10)

11.實作 autoAcceleration 方法。

```
- (void)autoGetAcceleration
{
    [motionManager startAccelerometerUpdatesToQueue:[NSOperationQueue currentQueue]
   withHandler: ^(CMAccelerometerData *acceleData, NSError *error)
     {
         CMAcceleration accel = acceleData.acceleration;
         NSLog(@"rotationrate:%f, %f, %f", accel.x, accel.y, accel.z);

     }];
}
```

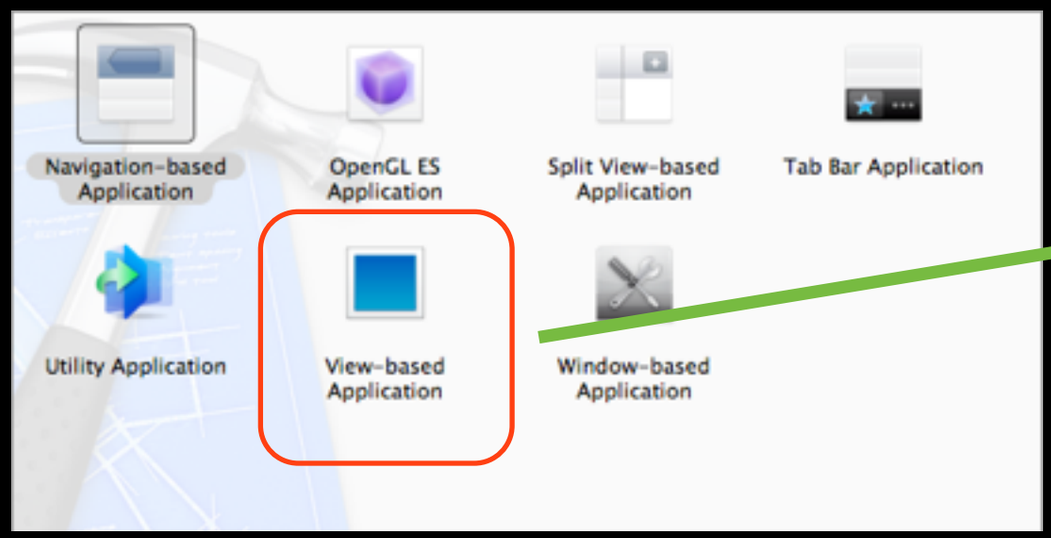
CoreMotionSample01(10/10)

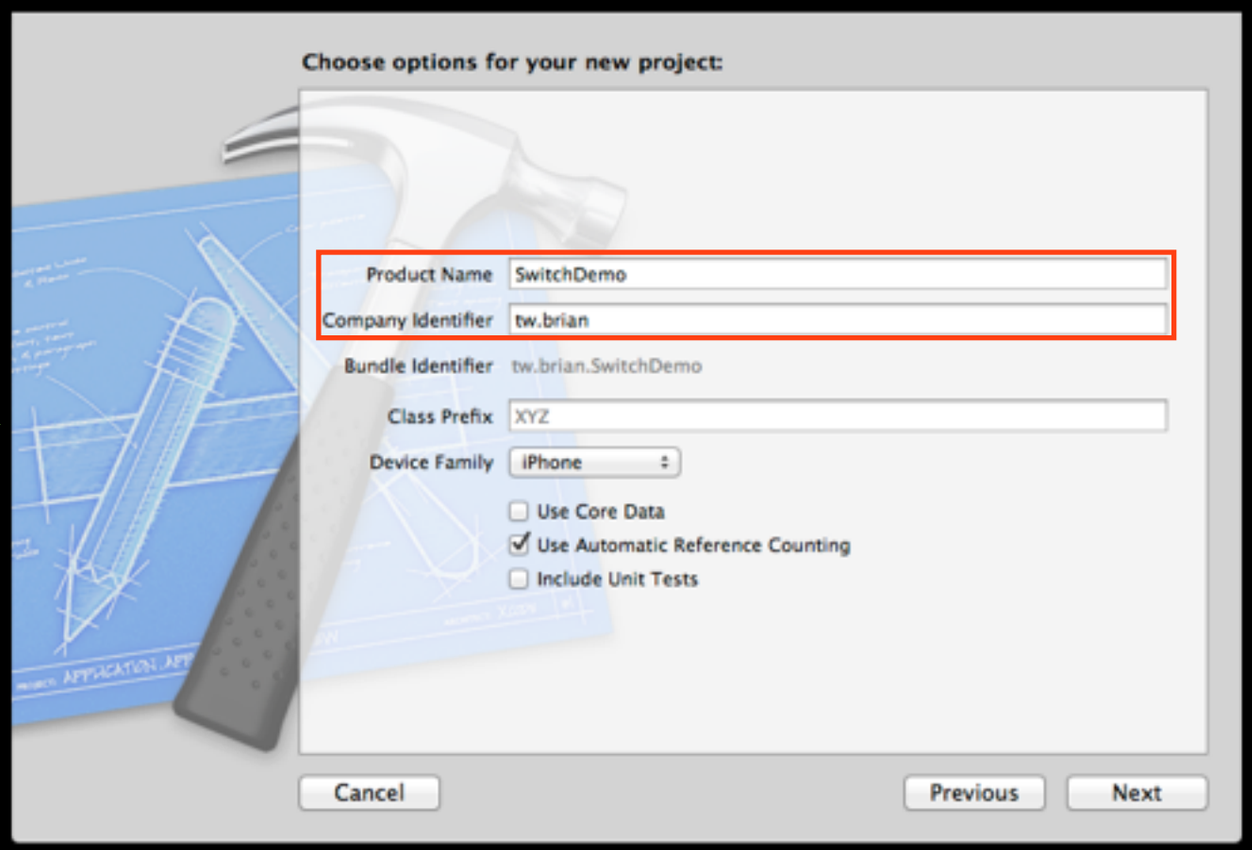
12.在viewDidLoad方法中加入以下程式碼後執行。

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

    motionManager = [[CMMotionManager alloc] init];

    if (motionManager.accelerometerAvailable) {
        memset(accelerationData, 0, sizeof(accelerationData));

        NSTimer *aTimer = [[NSTimer alloc] init];
        aTimer = [NSTimer scheduledTimerWithTimeInterval:1.0 target:self
          selector:@selector(showAcceleration) userInfo:nil repeats:NO];

        [self autoGetAcceleration];
    } else {
        NSLog(@"Dont't support accelerometer");
        [motionManager release];
    }
}
```

CoreMotionSample02(1/10)

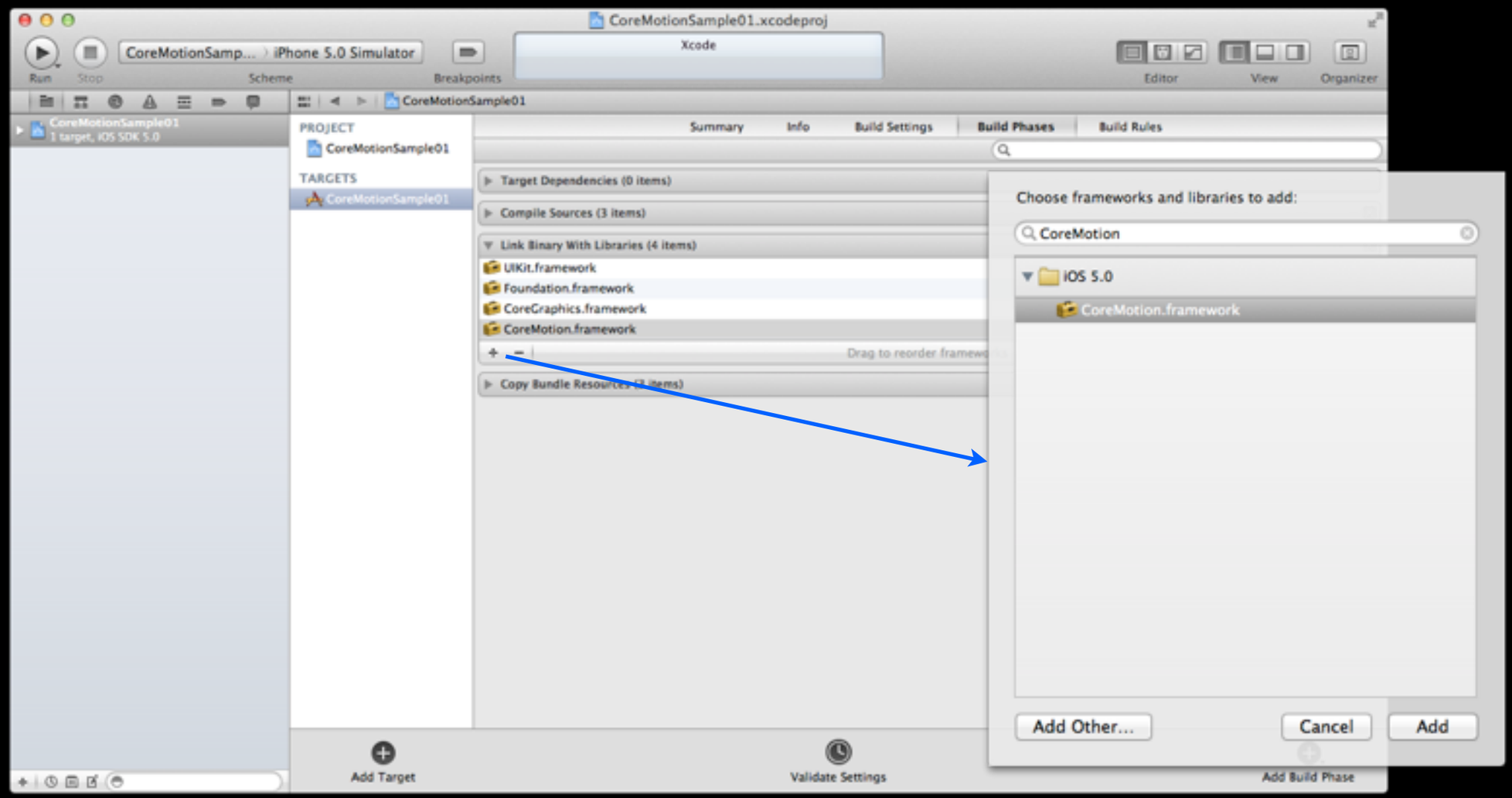
1.建置一個View-based 的專案，名稱『CoreMotionSample02』。





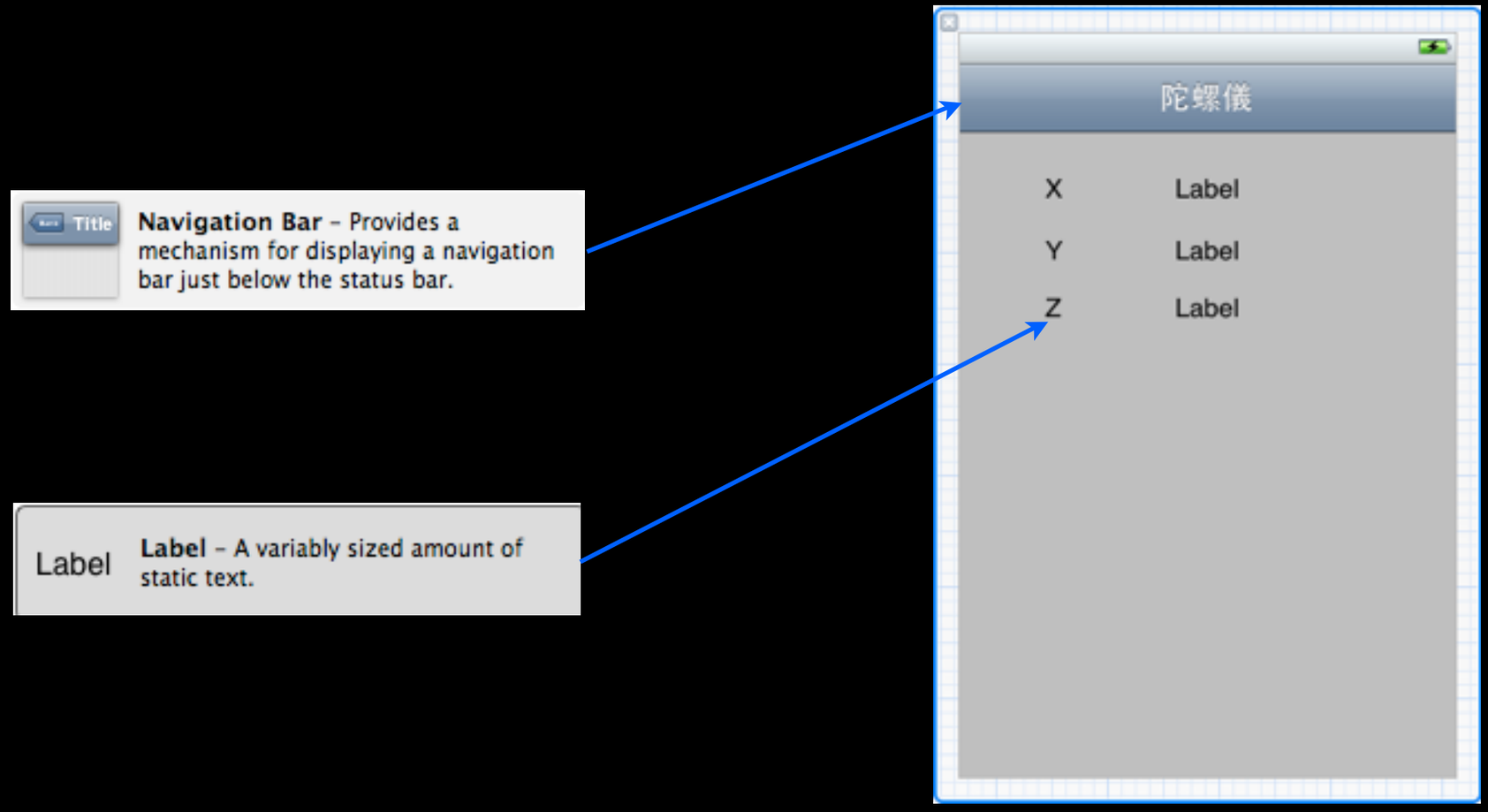
CoreMotionSample02(2/10)

2.從Targets > Build Phases > Link Binary With Libaries加入CoreMotion Framework



CoreMotionSample02(3/10)

3. 在View上加入一個Navigation Bar及6個元件 Label，並將View的配置完成如下圖所示。



CoreMotionSample02(4/10)

4.在 CoreMotionSample02ViewController.h 中加入3個 UILabel Outlet，名稱為如下所示並與 Label 元件建立參考。

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

@interface CoreMotionSample02ViewController : UIViewController {
    IBOutlet UILabel *gyroX;
    IBOutlet UILabel *gyroY;
    IBOutlet UILabel *gyroZ;
}

@property (nonatomic,retain) IBOutlet UILabel *gyroX;
@property (nonatomic,retain) IBOutlet UILabel *gyroY;
@property (nonatomic,retain) IBOutlet UILabel *gyroZ;

@end
```

CoreMotionSample02(5/10)

5. 加入 CoreMotion/CoreMotion.h 的 import，並增加 CMMotionManager 的記憶體指標。
6. 新增5個方法，名稱如下所示。

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

@interface CoreMotionSample02ViewController : UIViewController {
    CMMotionManager *motionManager;
    IBOutlet UILabel *gyroX;
    IBOutlet UILabel *gyroY;
    IBOutlet UILabel *gyroZ;
}

@property (nonatomic,retain) IBOutlet UILabel *gyroX;
@property (nonatomic,retain) IBOutlet UILabel *gyroY;
@property (nonatomic,retain) IBOutlet UILabel *gyroZ;

- (void)startGetGyro;
- (void)stopGetGyro;
- (void)getGyroData;
- (void)showGyroData;
- (void)autoGetGyroData;

@end
```

CoreMotionSample02(6/10)

7. 在 CoreMotionSample02ViewController.m 中加入 double 的陣列，初始化大小為 3。

```
#import "CoreMotionSample02ViewController.h"

double gyrodata[3];

@implementation CoreMotionSample02ViewController

@synthesize gyroX,gyroY,gyroZ;
```

CoreMotionSample02(7/10)

8.實作 startGetGyro 方法。

```
- (void)startGetGyro
{
    motionManager.gyroUpdateInterval = 1.0 / 60.0;
    [motionManager startGyroUpdates];
}
```

9. 實作 stopGetGyro 方法。

```
- (void)stopGetGyro
{
    if (motionManager.gyroActive) {
        [motionManager stopGyroUpdates];
    }
}
```

CoreMotionSample02(8/10)

9.實作 getGyroData 方法。

```
- (void)getGyroData
{
    CMGyroData *newgydodata = motionManager.gyroData;
    gyrodata[0] = newgydodata.rotationRate.x;
    gyrodata[1] = newgydodata.rotationRate.y;
    gyrodata[2] = newgydodata.rotationRate.z;
}
```

10.實作 showGyroData 方法。

```
- (void)showGyroData{
    [self getGyroData];
    NSLog(@"%f %f %f",gyrodata[0],gyrodata[1],gyrodata[2]);
    gyroX.text = [NSString stringWithFormat:@"%f",gyrodata[0]];
    gyroY.text = [NSString stringWithFormat:@"%f",gyrodata[1]];
    gyroZ.text = [NSString stringWithFormat:@"%f",gyrodata[2]];
}
```

CoreMotionSample02(9/10)

11.實作 autoGetGyroData 方法。

```
- (void)autoGetGyroData{
    [motionManager startGyroUpdatesToQueue:
    [NSOperationQueue currentQueue] withHandler: ^(CMGyroData
    *gyroData, NSError *error)
     {
         CMRotationRate rotation = gyroData.rotationRate;
         NSLog(@"rotationrate:%f, %f, %f",
      rotation.x, rotation.y, rotation.z);
     }];

}
```

CoreMotionSample02(10/10)

12.在viewDidLoad方法中加入以下程式碼後執行。

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

    motionManager = [[CMMotionManager alloc] init];

    if (motionManager.gyroAvailable) {
        [self startGetGyro];

        NSTimer *aTimer = [[NSTimer alloc] init];
        aTimer = [NSTimer scheduledTimerWithTimeInterval:1.0 target:self
      selector:@selector(showGyroData) userInfo:nil repeats:30];

        [self autoGetGyroData];
    } else {
        [motionManager release];
    }
}
```