

```
//
//  RatPressButtonViewController.m
//  RatShaping
//
//  Created by Student Worker Catherine Urbano on 5/10/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "RatPressButtonViewController.h"

#define SERVER_ADDRESS @"138.237.137.139"
#define SERVER_PORT 1024
@interface RatPressButtonViewController()
@property (nonatomic) BOOL buttonIsPressed;
@end


static NSString* format_date(){
    NSDate* date = [NSDate date];
    NSString* format = @"MM-dd-yyyy HH:mm:ss";
    NSDateFormatter* formatter = [NSDateFormatter new];
    [formatter setDateFormat: format];
    NSString* dateFormat = [formatter stringFromDate: date];
    return dateFormat;
}

static NSString* socket_read_message(NSInputStream* inputStream){
    uint8_t buffer[1024];
    int len;
    while ([inputStream hasBytesAvailable]) {
        len = [inputStream read:buffer maxLength:sizeof(buffer)];
        if (len > 0) {

            NSString *output = [[NSString alloc] initWithBytes:buffer
                                length:len encoding:NSUTF8StringEncoding];

            if (nil != output) {
                return output;
            }
        }
    }
    return @"";
}

@interface RatPressButtonViewController()<NSStreamDelegate>{
    NSInputStream *inputStream;
    NSOutputStream *outputStream;
    BOOL isConnected;
    NSString* serverId;
}
```

```
@end

@implementation RatPressButtonViewController
@synthesize buttonIsPressed = _buttonIsPressed;

-(void) awakeFromNib{
}

-(void) viewWillDisappear:(BOOL)animated{
    [self closeConnection];
}

-(void) viewDidAppear:(BOOL)animated{
    [self openConnection];
}

- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)
    interfaceOrientation
{
    return (interfaceOrientation == UIInterfaceOrientationLandscapeLeft);
}

-(void) closeConnection{
    [inputStream close];
    [outputStream close];
    isConnected = NO;
}

-(void) openConnection{
    CFReadStreamRef readStream;
    CFWriteStreamRef writeStream;
    CFStreamCreatePairWithSocketToHost(NULL, (CFStringRef)SERVER_ADDRESS,
        SERVER_PORT, &readStream, &writeStream);
    inputStream = (__bridge NSInputStream *)readStream;
    outputStream = (__bridge NSOutputStream *)writeStream;
    [inputStream setDelegate:self];
    [outputStream setDelegate:self];
    [inputStream scheduleInRunLoop:[NSRunLoop currentRunLoop] forMode:
        NSDefaultRunLoopMode];
    [outputStream scheduleInRunLoop:[NSRunLoop currentRunLoop] forMode:
        NSDefaultRunLoopMode];
    [inputStream open];
    [outputStream open];
}

-(void) sendMessage: (NSString*) message{
    if(isConnected == NO){
        [self openConnection];
        return;
    }

    NSString *response = [NSString stringWithFormat:@"%s|%@",serverId ,
        message];
```

```

    NSData *data = [[NSData alloc] initWithData:[response dataUsingEncoding:
        NSASCIIStringEncoding]];
    [outputStream write:[data bytes] maxLength:[data length]];
}

- (void)stream:(NSStream *)theStream handleEvent:(NSStreamEvent)streamEvent
{
    switch (streamEvent) {

        case NSStreamEventOpenCompleted:
            isConnected = YES;
            break;

        case NSStreamEventHasBytesAvailable:
            if (theStream == inputStream) {
                NSString* output = socket_read_message(inputStream);
                serverId = output;
                deviceID.text = [NSString stringWithFormat: @"Device ID:\n%
                    @" , serverId];
            }
            break;

        case NSStreamEventErrorOccurred:
            isConnected = NO;
            [outputStream close];
            [inputStream close];
            break;

        case NSStreamEventEndEncountered:
            isConnected = NO;
            serverId = @"";
            deviceID.text = @"Not Connected";
            [outputStream close];
            [inputStream close];
            break;

    }

}

- (void) viewDidLoad{
    [super viewDidLoad];
    for(int i = 0; i <= 100; i++)
    {
        self.buttonIsPressed = NO;
        CGPoint midpoint;
        UIImage *rand_img;
        midpoint.y = self.view.bounds.origin.y + (self.view.bounds.size.
            width/2);
        int imgvalue = (rand() % 3) + 1;
        // NSLog(@"Image # %i", value);
        if (imgvalue == 1)
            rand_img = [UIImage imageNamed:@"cross_100px.png"];

        if (imgvalue == 2)

```

```

        rand_img = [UIImage imageNamed:@"triangle_100px.png"];

    if (imgvalue == 3)
        rand_img = [UIImage imageNamed:@"circle100px.png"];

    int xvalue;
    xvalue = (rand() % 3) + 1;
    if (xvalue == 1)

        midpoint.x = self.view.bounds.size.height/4;
    if (xvalue == 2)

        midpoint.x = self.view.bounds.size.height/2;
    if (xvalue == 3)

        midpoint.x = (self.view.bounds.size.height*3)/4;
    NSLog(@"Image %@", rand_img);
    NSLog(@"Position %f", midpoint.x);
    CGFloat xpos = midpoint.x - 50;
    CGFloat ypos = midpoint.y - 50;
    UIButton *targetButton = [UIButton buttonWithType:
        UIButtonTypeRoundedRect];
    targetButton.frame = CGRectMake(xpos, ypos, 100, 100);
    [targetButton setBackgroundImage:rand_img forState:
        (UIControlStateNormal)];
    [self.view addSubview:targetButton];
    [targetButton addTarget:self action:@selector(onPressButton:)
        forControlEvents:UIControlEventTouchUpInside];
    while (self.buttonIsPressed == YES) {
        [targetButton removeFromSuperview];
    }

    }

}

- (IBAction) onPressButton: (id) sender{
    UIButton* btn = (UIButton*) sender;
    [self sendMessage: [NSString stringWithFormat: @"%d|%@", btn.tag,
        format_date()]];
    NSLog(@"%d|%@", btn.tag, format_date());
    self.buttonIsPressed = YES;
}

```

@end