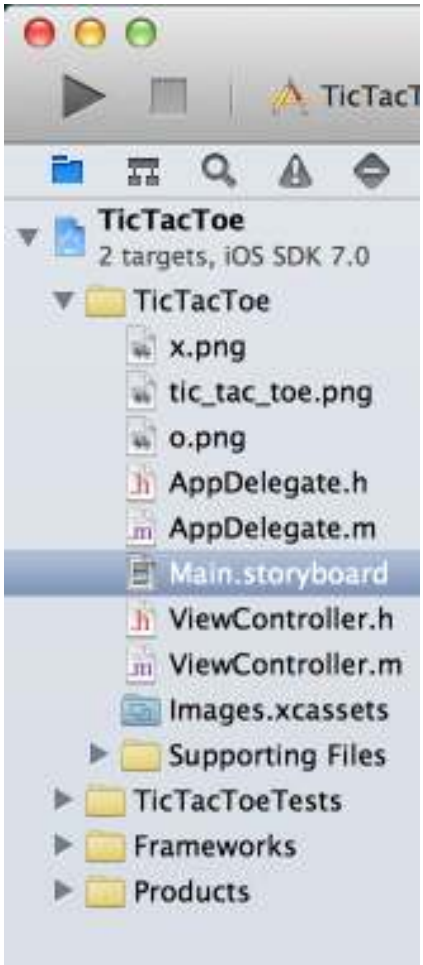
Tic-Tac-Toe - Beginner iOS Application

(1) Create a new Xcode project. Use the "Single View" template.

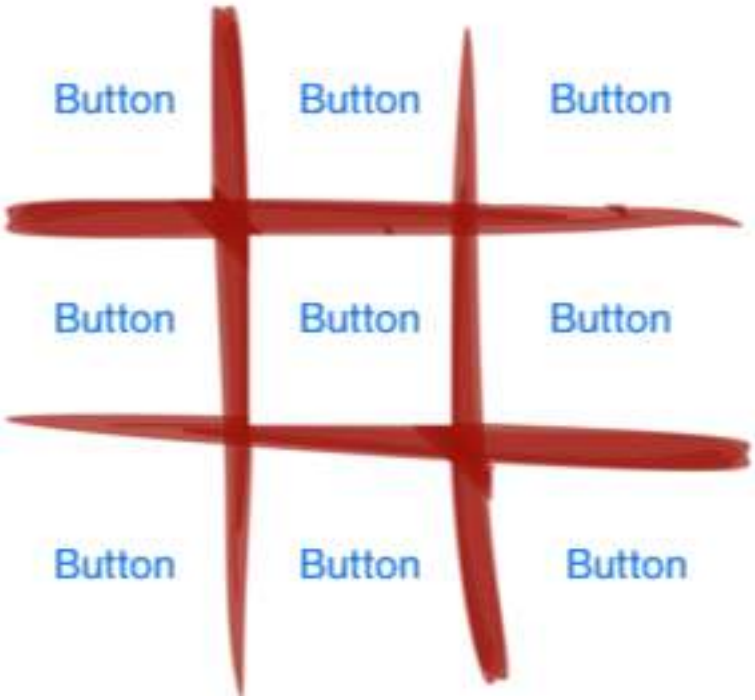
(2) Drag a background board, X and O images into a project folder.



(3) Click on **Main.storyboard** and Drag an **Image View** to the window. This will hold the background image. Change the property to show the background image you are using.

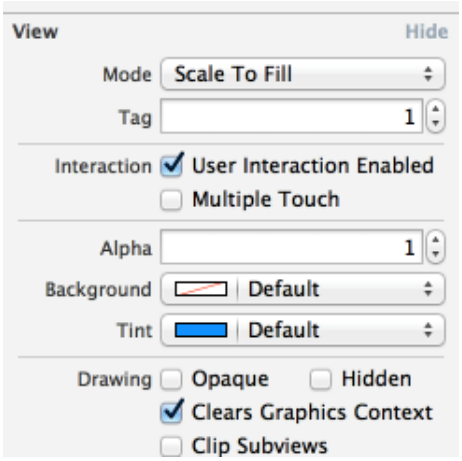
(4) Drag 6 buttons and place them in the squares on top of the background image.

Now your UI should look something like this:

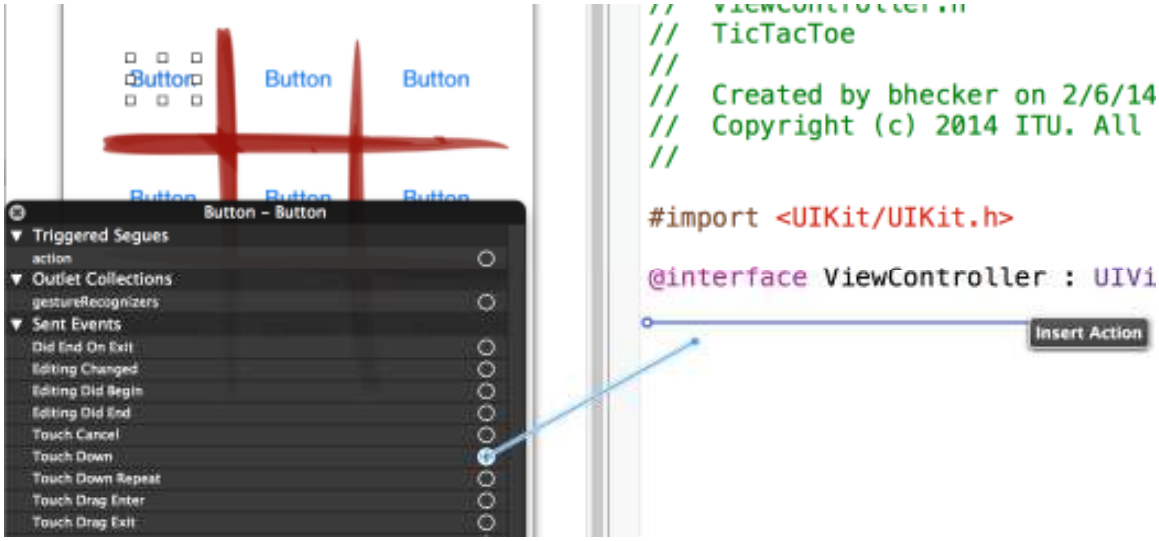


Eventually you are going to want to get rid of the "Button" labels on the buttons since they are going to show images instead.

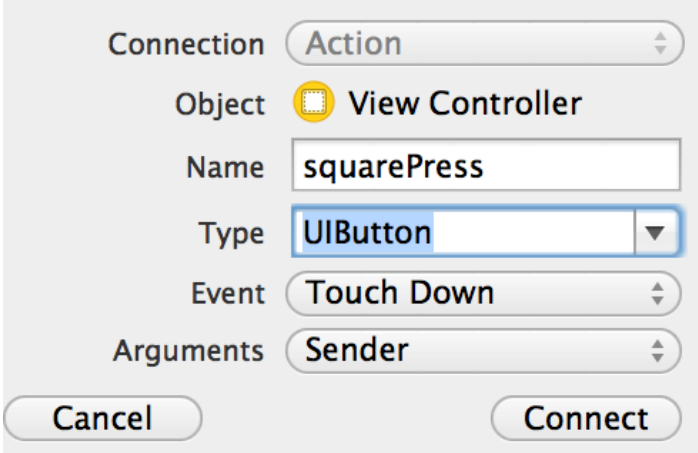
(5) Change the tag property of all of the buttons to be 1 through 9. Look in the view section of the properties window under "Mode."



(6) Wire the first button to **ViewController.h**, in between the @interface and the @end.



Name the IBAction **squarePress** and change the type to **UIButton**:

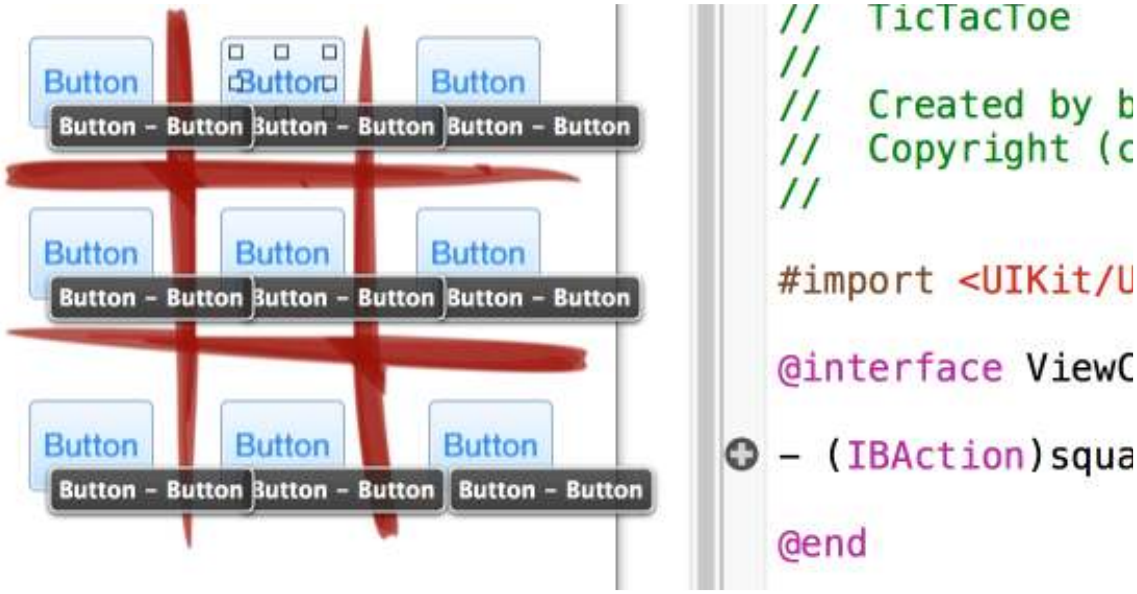


The finished method looks like this:

```
- (IBAction)squarePress:(UIButton *)sender;
```

Drag the other 8 buttons to this same method.

Check to make sure they are all connected by hovering your mouse over the "connection dot" next to the method.



Important notes:

- Wire to **ViewController.h** and not the ViewController.m.
- Add everything in between the @interface and @end.

(7) Lets write some source code now! First let's test the buttons. Add the following code and run the project. Look at the lower console window and the button tag numbers should show as you click on the different buttons.

```
NSInteger i = [sender tag];
NSLog(@"test %d", i);
```

Fix any problems now otherwise your game functionality isn't going to work right.

(8) Add a counter to **ViewController.h** to keep track of how many turns have taken place and some chars to remember what was picked.

```
{
    int turns;
}
```

(9) Set the number of turns to 0 in the **ViewDidLoad** method:

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    turns = 0;
}
```

(10) We are going to play Tic-Tac-Toe by changing the image when the user clicks a button. The user goes first and will be X. The computer will be O and will automatically go next. Copy the following button code into the **squarePress** method and make sure the button images are named appropriately.

```
- (IBAction)squarePress:(UIButton *)sender {
    // the button tag testing
    NSInteger i = [sender tag];
    NSLog(@"test %d", i);

    // the button that was clicked gets an x
   UIButton *tmpButton = (UIButton *)[self.view
viewWithTag:i];
   [tmpButton setImage:[UIImage
imageNamed:@"x.png"]
forState:UIControlStateNormal];
   [tmpButton setTag:i+10]; // item can't be selected by
computer
    turns++;

    if([self checkForWin]) // see for a winner
    {
        UIAlertView *alert = [[UIAlertView alloc]
initWithTitle:@"You won!" message:@"Great job, you
```

```
beat the computer1" delegate:nil
cancelButtonTitle:@"OK" otherButtonTitles:nil];
        [alert show];
        turns=10;
    }

    // check to see if there are turns left
    if (turns < 9)
    {
       // randomly select another button for the
computer's turn
        int r = arc4random() % 9; // randomly select a
number between 0 and 9
        UIButton *compButton = (UIButton *)[self.view
viewWithTag:r]; // get the square

    // see if the square is selectable. If not, select
another one
     while (![[compButton.superview viewWithTag:r]
isKindOfClass:[UIButton class]])
     {
         r = arc4random() % 9;
         compButton = (UIButton *)[self.view
viewWithTag:r];
         NSLog(@"random %d", r);
     }

    [compButton setImage:[UIImage
imageNamed:@"o.png"]
forState:UIControlStateNormal];
    [compButton setTag:r+10];
```

```
    turns++;

    if([self checkForWin]) // see for a winner
        {
            UIAlertView *alert = [[UIAlertView alloc]
initWithTitle:@"Computer Won!" message:@"You got
beat by the computer. Try again!" delegate:nil
cancelButtonTitle:@"OK" otherButtonTitles:nil];
            [alert show];
            turns=10;
        }
    }
}
```

(11) There needs to be a check for a winner. Add this method to your **ViewController.m** file:

```
// method that will check to see if someone has won
returns TRUE if someone wins
-(BOOL) checkForWin {

    UIButton *Button1 = (UIButton *)[self.view
viewWithTag:11];
    UIButton *Button2 = (UIButton *)[self.view
viewWithTag:12];
    UIButton *Button3 = (UIButton *)[self.view
viewWithTag:13];
    UIButton *Button4 = (UIButton *)[self.view
viewWithTag:14];
    UIButton *Button5 = (UIButton *)[self.view
viewWithTag:15];
```

```
    UIButton *Button6 = (UIButton *)[self.view
viewWithTag:16];
    UIButton *Button7 = (UIButton *)[self.view
viewWithTag:17];
    UIButton *Button8 = (UIButton *)[self.view
viewWithTag:18];
    UIButton *Button9 = (UIButton *)[self.view
viewWithTag:19];


    // HORIZONTAL WINS
        if((Button1.currentImage ==
Button2.currentImage) & (Button2.currentImage ==
Button3.currentImage) & (Button1.currentImage !=
NULL))
        {
            return YES;
        }
        if((Button4.currentImage ==
Button5.currentImage) & (Button5.currentImage ==
Button6.currentImage) & (Button4.currentImage !=
NULL))
        {
            return YES;
        }
        if((Button7.currentImage ==
Button8.currentImage) & (Button8.currentImage ==
Button9.currentImage) & (Button7.currentImage !=
NULL))
        {
            return YES;
```

```
        }
        // VERTICAL WINS
        if((Button1.currentImage ==
Button4.currentImage) & (Button4.currentImage ==
Button7.currentImage) & (Button1.currentImage !=
NULL))
        {
            return YES;
        }
        if((Button2.currentImage ==
Button5.currentImage) & (Button5.currentImage ==
Button8.currentImage) & (Button2.currentImage !=
NULL))
        {
            return YES;
        }
        if((Button3.currentImage ==
Button6.currentImage) & (Button6.currentImage ==
Button9.currentImage) & (Button3.currentImage !=
NULL))
        {
            return YES;
        }
        // DIAGONAL WINS
        if((Button1.currentImage ==
Button5.currentImage) & (Button5.currentImage ==
Button9.currentImage) & (Button1.currentImage !=
NULL))
        {
            return YES;
        }
```

```
        if((Button3.currentImage ==
Button5.currentImage) & (Button5.currentImage ==
Button7.currentImage) & (Button3.currentImage !=
NULL))
        {
            return YES;
        }

        return NO;
}
```

(12) You are done! Run and enjoy!

I left out the check for a tie game. You could also add the functionality to reset the board. Maybe a reset button would be a good idea? You could also change the code to update the status of a label with who won or keep track of how many times you and the computer beat each other. There are a lot of simple modification you could make with this program to continue your learning process. Have fun!