

## Coding a square

I am playing around with this section.

```
//TO DO: Fill this comment in
else
{
    if (e.getActionCommand ().equals ("a"))
        updateSquare (0, 0, a);
    else if (e.getActionCommand ().equals ("b"))
        updateSquare (0, 1, b);
    else if (e.getActionCommand ().equals ("c"))
        updateSquare (0, 2, c);
```

I also removed the flip turn call from the bottom of because I want it to only happen if they have finished a turn.

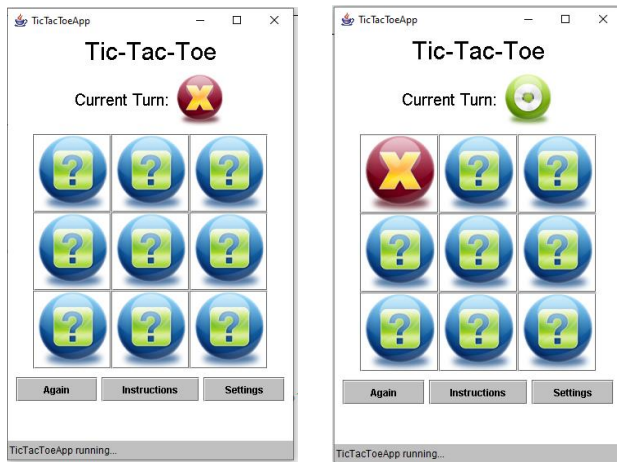
```
        else if (e.getActionCommand ().equals ("h"))
            updateSquare (2, 1, h);
        else if (e.getActionCommand ().equals ("i"))
            updateSquare (2, 2, i);
    //flipTurn ();
}
```

The 'a' button would work something like this:

```
//TO DO: Fill this comment in
else
{
    if (e.getActionCommand ().equals ("a"))
    {
        //updateSquare (0, 0, a);

        //if empty (b), put the current turn in it
        if (board [0] [0] == 'b')
        {
            board [0] [0] = turn;
            a.setIcon (createImageIcon (turn + ".png"));
            flipTurn ();
        }
        else
        { //error, space is filled
        }
    }
    else if (e.getActionCommand ().equals ("b"))
        updateSquare (0, 1, b);
```

If you coded that, the code should allow you to place a piece, but only in the a position:



For your information, the board's co-ordinates work like this:

[0][0]	[0][1]	[0][2]
[1][0]	[1][1]	[1][2]
[2][0]	[2][1]	[2][2]

You can cut and paste this code into all of the other buttons and change the position in board and button.

```
//if empty (b), put the current turn in it
if (board [0] [0] == 'b')
{
    board [0] [0] = turn;
    a.setIcon (createImageIcon (turn + ".png"));
    flipTurn ();
}
else
{ //error, space is filled
}
```

However, DO NOT DO THAT. Cutting and pasting is bad. Remember the Pizza Party? Remember Reusability?

Instead, fill in the update method. Its parameters are set up well.

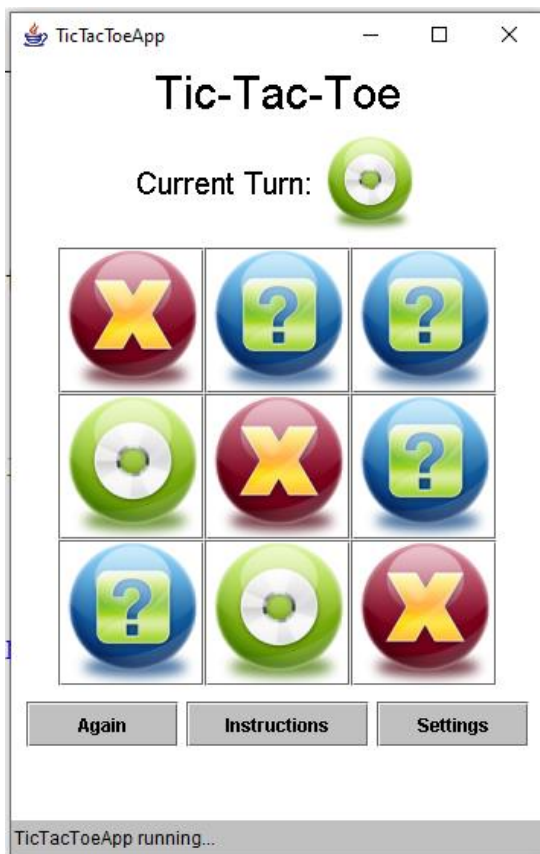
```
public void updateSquare (int x, int y, JButton square)
```

Look at the code for button A and figure out how to adapt it.

And use the method calls as originally written:

```
if (e.getActionCommand ().equals ("a"))
    updateSquare (0, 0, a);
else if (e.getActionCommand ().equals ("b"))
    updateSquare (0, 1, b);
else if (e.getActionCommand ().equals ("c"))
    updateSquare (0, 2, c);
else if (e.getActionCommand ().equals ("d"))
    updateSquare (1, 0, d);
else if (e.getActionCommand ().equals ("ee"))
    updateSquare (1, 1, ee);
```

Then all buttons should work.



There is a checkpoint on the Google classroom for this.