Counting Neighbours

Make a new method.

```
public void neighbours ()
{
    for (int i = 0 ; i < row ; i++)
    {
        for (int j = 0 ; j < col ; j++)
        {
            if (b [i] [j] != 10)
            {
                int count = 0;
                if (i - 1 >= 0 && j - 1 >= 0 && b [i - 1] [j - 1] == 10)
                    count++;
                if (i - 1 >= 0 && b [i - 1] [j] == 10)
                    count++;
                if (i - 1 >= 0 && j + 1 < col && b [i - 1] [j + 1] == 10)
                    count++;
                //add other directions here

                b [i] [j] = count;
            }
        }
    }
}
```

Make them ifs, do not use else ifs

Add all of the directions

Up, Left

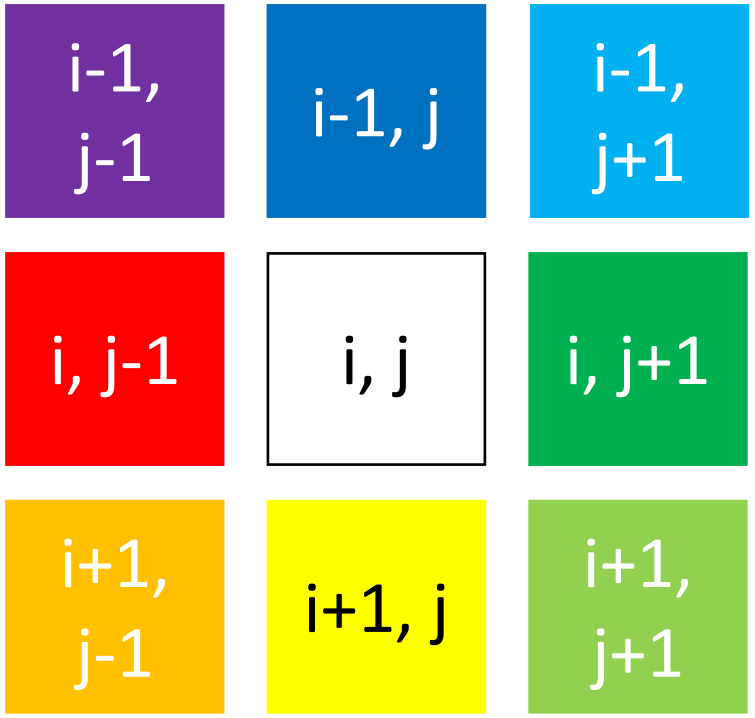
Guard:
i-1>=0
j-1>=0

Up

Guard:
i-1>=0

Up, Right

Guard:
i-1>=0
j+1<col



Left

Guard:
j-1>=0

Right

Guard:
j+1<col

```
public void neighbours ()
{
    for (int i = 0 ; i < row ; i++)
    {
        for (int j = 0 ; j < col ; j++)
        {
            if (b [i] [j] != 10)
            {
                int count = 0;
                if (i - 1 >= 0 && j - 1 >= 0 && b [i - 1] [j - 1] == 10)
                    count++;
                if (i - 1 >= 0 && b [i - 1] [j] == 10)
                    count++;
                if (i - 1 >= 0 && j + 1 < col && b [i - 1] [j + 1] == 10)
                    count++;
                //add other directions here

                b [i] [j] = count;
            }
        }
    }
}
```

Down, Left

Guard:
i+1<row
j-1>=0

Down

Guard:
i+1<row

Down, Right

Guard:
i+1<row
j+1<col

Update Temporary Add for Testing

```
//TO DO: Fill this comment in
else if (e.getActionCommand ().equals ("reset"))
{
    addMines (20);
    neighbours();
    reveal ();
}
```

Test your code to make sure that it works. Run it multiple times.