

```
1 namespace tickTacToe
2 {
3     public partial class Form1 : Form
4     {
5
6         PictureBox[,] grid = new PictureBox[3, 3];
7
8         int[,] XPlays = new int[3, 3];
9         int[,] OPlays = new int[3, 3];
10
11         int xWins = 0;
12         int oWins = 0;
13         int ties = 0;
14
15         String turn = "X";
16
17         public Form1()
18         {
19             InitializeComponent();
20         }
21
22         private void Form1_Load(object sender, EventArgs e)
23         {
24             grid[0, 0] = pic00;
25             grid[0, 1] = pic01;
26             grid[0, 2] = pic02;
27             grid[1, 0] = pic10;
28             grid[1, 1] = pic11;
29             grid[1, 2] = pic12;
30             grid[2, 0] = pic20;
31             grid[2, 1] = pic21;
32             grid[2, 2] = pic22;
33         }
34
35         private void pictureBox1_Click(object sender, EventArgs e)
36         {
37             confirmation(0, 2);
38         }
39
40
41         private void pic12_Click(object sender, EventArgs e)
42         {
43             confirmation(1, 2);
44         }
45
46         private void pic00_Click(object sender, EventArgs e)
47         {
48             confirmation(0, 0);
49         }
```

```

50
51     public void confirmation(int x, int y)
52     {
53         DialogResult resultAnswer = new DialogResult();
54         resultAnswer = MessageBox.Show("You sure?", "Confirm",
55         MessageBoxButtons.YesNo);
56
57         if (resultAnswer == DialogResult.Yes)
58         {
59             if (turn == "X")
60             {
61                 grid[x, y].Image = picX.Image;
62                 grid[x, y].Enabled = false;
63                 XPlays[x, y] = 1;
64                 turn = "O";
65             }
66             else if (turn == "O")
67             {
68                 grid[x, y].Image = picCircle.Image;
69                 grid[x, y].Enabled = false;
70                 OPlays[x, y] = 1;
71                 turn = "X";
72             }
73
74             checkWinner();
75         }
76     }
77     //hello mr salesky this is your concious. if you like to eat      ↗
       burritos and salad bowls come to chipotle on march 1st and order ↗
       100 salad bowls for the mubotics fundraiser. this is not a      ↗
       request.
78     public void checkWinner()
79     {
80         // 0 2 | 1 2 | 2 2
81         // ----|-----|-----
82         // 0 1 | 1 1 | 2 1
83         // ----|-----|-----
84         // 0 0 | 1 0 | 2 0
85
86
87
88         for(int row = 0; row <= 2; row++)
89         {
90             int xHor = 0;
91             int xVer = 0;
92
93             int oHor = 0;
94             int oVer = 0;
95

```

```
196         for (int col = 0; col <= 2; col++)
197         {
198
199             xHor += XPlays[row, col];
200
201             if(xHor == 3)
202             {
203                 DialogResult resultAnswer = new DialogResult();
204                 resultAnswer = MessageBox.Show("X wins Vertically", "winner winner",
205                                     MessageBoxButtons.OK);
206
207                 if(resultAnswer == DialogResult.OK) {
208                     xWins += 1;
209                     reset();
210                 }
211             }
212
213             oHor += OPlays[row, col];
214
215             if (oHor == 3)
216             {
217                 DialogResult resultAnswer = new DialogResult();
218                 resultAnswer = MessageBox.Show("O wins Vertically", "winner winner",
219                                     MessageBoxButtons.OK);
220
221                 if (resultAnswer == DialogResult.OK)
222                 {
223                     oWins += 1;
224                     reset();
225                 }
226             }
227
228             xVer += XPlays[col, row];
229
230             if(xVer == 3)
231             {
232                 DialogResult resultAnswer = new DialogResult();
233                 resultAnswer = MessageBox.Show("X wins Horizontally", "winner winner",
234                                     MessageBoxButtons.OK);
235
236                 if (resultAnswer == DialogResult.OK)
237                 {
238                     reset();
239                     xWins += 1;
240                 }
241             }
```

```
142
143         oVer += OPlays[col, row];
144
145         if (oVer == 3)
146         {
147             DialogResult resultAnswer = new DialogResult();
148             resultAnswer = MessageBox.Show("O wins      ↗
Horizontally", "winner winner",
149             MessageBoxButtons.OK);
150
151             if (resultAnswer == DialogResult.OK)
152             {
153                 reset();
154                 oWins += 1;
155             }
156         }
157     }
158 }
159
160 int xDiagSumPos = 0;
161 int xDiagSumNeg = 0;
162
163 int oDiagSumPos = 0;
164 int oDiagSumNeg = 0;
165
166 for (int diag = 0; diag <= 2; diag++)
167 {
168
169     int diagNeg = (diag * -1) + 2;
170
171     xDiagSumPos += XPlays[diag, diag];
172     xDiagSumNeg += XPlays[diag, diagNeg];
173
174     //MessageBox.Show(diagSumPos.ToString());
175     if(xDiagSumPos == 3)
176     {
177         DialogResult resultAnswer = new DialogResult();
178         resultAnswer = MessageBox.Show("X wins Diagonally",      ↗
"winner winner",
179         MessageBoxButtons.OK);
180
181         if (resultAnswer == DialogResult.OK)
182         {
183             reset();
184             xWins += 1;
185         }
186     }
187 }
188
```

```
189         if(xDiagSumNeg == 3)
190         {
191             DialogResult resultAnswer = new DialogResult();
192             resultAnswer = MessageBox.Show("X wins Diagonally",
193             "winner winner",
194             MessageBoxButtons.OK);
195
196             if (resultAnswer == DialogResult.OK)
197             {
198                 reset();
199                 xWins += 1;
200             }
201         }
202
203         oDiagSumPos += OPlays[diag, diag];
204         oDiagSumNeg += OPlays[diag, diagNeg];
205
206         //MessageBox.Show(diagSumPos.ToString());
207         if (oDiagSumPos == 3)
208         {
209             DialogResult resultAnswer = new DialogResult();
210             resultAnswer = MessageBox.Show("O wins Diagonally",
211             "winner winner",
212             MessageBoxButtons.OK);
213
214
215             if (resultAnswer == DialogResult.OK)
216             {
217                 reset();
218                 oWins += 1;
219             }
220         }
221
222         if (oDiagSumNeg == 3)
223         {
224             DialogResult resultAnswer = new DialogResult();
225             resultAnswer = MessageBox.Show("O wins Diagonally",
226             "winner winner",
227             MessageBoxButtons.OK);
228
229             if (resultAnswer == DialogResult.OK)
230             {
231                 reset();
232                 oWins += 1;
233             }
234         }
```

```
235     }
236
237     if(grid[0, 0].Image != null &&
238     grid[0, 1].Image != null &&
239     grid[0, 2].Image != null &&
240     grid[1, 0].Image != null &&
241     grid[1, 1].Image != null &&
242     grid[1, 2].Image != null &&
243     grid[2, 0].Image != null &&
244     grid[2, 1].Image != null &&
245     grid[2, 2].Image != null)
246     {
247         MessageBox.Show("No Winners!");
248         ties += 1;
249         reset();
250     }
251
252
253 }
254
255 public void reset()
256 {
257     turn = "X";
258
259     pic00.Image = null;
260     pic01.Image = null;
261     pic02.Image = null;
262     pic10.Image = null;
263     pic11.Image = null;
264     pic12.Image = null;
265     pic20.Image = null;
266     pic21.Image = null;
267     pic22.Image = null;
268
269
270     XPlays[0, 0] = 0;
271     XPlays[0, 1] = 0;
272     XPlays[0, 2] = 0;
273     XPlays[1, 0] = 0;
274     XPlays[1, 1] = 0;
275     XPlays[1, 2] = 0;
276     XPlays[2, 0] = 0;
277     XPlays[2, 1] = 0;
278     XPlays[2, 2] = 0;
279
280     OPlays[0, 0] = 0;
281     OPlays[0, 1] = 0;
282     OPlays[0, 2] = 0;
283     OPlays[1, 0] = 0;
```

```
284         OPlays[1, 1] = 0;
285         OPlays[1, 2] = 0;
286         OPlays[2, 0] = 0;
287         OPlays[2, 1] = 0;
288         OPlays[2, 2] = 0;
289
290         grid[0, 0].Enabled = true;
291         grid[0, 1].Enabled = true;
292         grid[0, 2].Enabled = true;
293         grid[1, 0].Enabled = true;
294         grid[1, 1].Enabled = true;
295         grid[1, 2].Enabled = true;
296         grid[2, 0].Enabled = true;
297         grid[2, 1].Enabled = true;
298         grid[2, 2].Enabled = true;
299
300     }
301
302     private void btnReset_Click(object sender, EventArgs e)
303     {
304         reset();
305     }
306
307     private void btnQuit_Click(object sender, EventArgs e)
308     {
309         this.Close();
310     }
311
312     private void pic10_Click(object sender, EventArgs e)
313     {
314         confirmation(1, 0);
315     }
316
317     private void pic20_Click(object sender, EventArgs e)
318     {
319         confirmation(2, 0);
320     }
321
322     private void pic01_Click(object sender, EventArgs e)
323     {
324         confirmation(0, 1);
325     }
326
327     private void pic11_Click(object sender, EventArgs e)
328     {
329         confirmation(1, 1);
330     }
331
332     private void pic21_Click(object sender, EventArgs e)
```

```
333     {
334         confirmation(2, 1);
335     }
336
337     private void pic22_Click(object sender, EventArgs e)
338     {
339         confirmation(2, 2);
340     }
341
342     private void btnStats_Click(object sender, EventArgs e)
343     {
344         MessageBox.Show(
345             "X Wins " + xWins.ToString() + "\n" +
346             "O Wins " + oWins.ToString() + "\n" +
347             "Ties " + ties.ToString()
348         );
349     }
350 }
351 }
```