

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace Fisher
12 {
13
14     public partial class Form1 : Form
15     {
16         // startup variables
17         private int score = 0;
18         private int fishCount = 20;
19         private bool hooking = false;
20         private int dir = 1;
21         System.Random r = new System.Random((int)
22             System.DateTime.Now.Ticks);
23         PictureBox[] DaSea = new PictureBox[20];
24         PictureBox[] bubbles = new PictureBox[4];
25         public Form1()
26         {
27             InitializeComponent();
28         }
29
30         private void label3_Click(object sender, EventArgs e)
31         {
32
33         }
34         //when the program starts
35         private void Form1_Load(object sender, EventArgs e)
36         {
37             // debugging
38             Console.WriteLine(Hook.Top);
39             Console.WriteLine(Chain1.Top);
40             Console.WriteLine(Chain2.Top);
41
42             // setting arrays for fish
43             DaSea[0] = Fish0;
44             DaSea[1] = Fish1;
45             DaSea[2] = Fish2;
46             DaSea[3] = Fish3;
47             DaSea[4] = Fish4;
48             DaSea[5] = Fish5;
49             DaSea[6] = Fish6;
```

```
50         DaSea[7] = Fish7;
51         DaSea[8] = Fish8;
52         DaSea[9] = Fish9;
53         DaSea[10] = Fish10;
54         DaSea[11] = Fish11;
55         DaSea[12] = Fish12;
56         DaSea[13] = Fish13;
57         DaSea[14] = Fish14;
58         DaSea[15] = Fish15;
59         DaSea[16] = Fish16;
60         DaSea[17] = Fish17;
61         DaSea[18] = Fish18;
62         DaSea[19] = Fish19;
63
64         // setting arrays for bubbles
65         bubbles[0] = bubble0;
66         bubbles[1] = bubble1;
67         bubbles[2] = bubble2;
68         bubbles[3] = bubble3;
69         // setting pos for fish
70         for (int n = 0; n < DaSea.Length; n++)
71         {
72             int randomx = r.Next(0, this.Width);
73             int randomy = r.Next(label1.Top, this.Height);
74             DaSea[n].Left = randomx;
75             DaSea[n].Top = randomy;
76         }
77     }
78     // update timer
79     private void timer1_Tick(object sender, EventArgs e)
80     {
81         // bubble code
82         for (int u = 0; u < bubbles.Length; u++)
83         {
84             bubbles[u].Top -= 15;
85             if (bubbles[u].Top <= label1.Top)
86             {
87                 int randomxbub = r.Next(0, this.Width);
88                 int randomybub = r.Next(label1.Top, label2.Top);
89                 bubbles[u].Left = randomxbub;
90                 bubbles[u].Top = randomybub;
91             }
92         }
93
94
95
96         //boat movement
97         if (dir == 0)
98         {
```

```
99         boat.Left -= 9;
100         boat.Image = imgboatL.Image;
101     }
102     else
103     {
104         boat.Left += 9;
105         boat.Image = imgboatR.Image;
106     }
107
108     if (boat.Left < 0)
109     {
110         dir = 1;
111     }
112     if (boat.Left > this.Width - 150)
113     {
114         dir = 0;
115     }
116     Hook.Left = boat.Left;
117     Chain1.Left = boat.Left;
118     Chain2.Left = boat.Left;
119     // hook func
120     if (hooking)
121     {
122         Hook.Top += 9;
123         Chain1.Top += 9;
124         Chain2.Top += 9;
125     }
126     else
127     {
128         Hook.Top = 106;
129         Chain1.Top = -64;
130         Chain2.Top = 24;
131     }
132
133     if (Hook.Top >= 286)
134     {
135         hooking = false;
136     }
137
138     // fish movement
139     int randomx = r.Next(0, 2);
140     int randomy = r.Next(0, 2);
141     for (int i = 0; i < DaSea.Length; i++)
142     {
143         randomx = r.Next(0, 2);
144         randomy = r.Next(0, 2);
145         if (DaSea[i].Image != imgFishDead.Image)
146         {
147
```

```
148         if (randomx == 1)
149         {
150
151             DaSea[i].Image = imgFishAliveR.Image;
152             DaSea[i].Left += 9;
153
154         }
155         else
156         {
157
158             DaSea[i].Image = imgFishAliveL.Image;
159             DaSea[i].Left -= 9;
160
161         }
162
163         if (randomy == 1)
164         {
165             DaSea[i].Top += 9;
166         }
167         else
168         {
169             DaSea[i].Top -= 9;
170         }
171         // fix OOB
172         if (DaSea[i].Top < label1.Top)
173         {
174             DaSea[i].Top = label1.Top;
175         }
176         if (DaSea[i].Top > label2.Top)
177         {
178             DaSea[i].Top = label2.Top;
179         }
180
181         if (DaSea[i].Left < 0)
182         {
183             DaSea[i].Left = 0;
184         }
185         if (DaSea[i].Left > this.Width - 150)
186         {
187             DaSea[i].Left = this.Width - 150;
188         }
189         // collision func
190         if (DaSea[i].Bounds.Intersects(Hook.Bounds))
191         {
192             if (hooking)
193             {
194                 DaSea[i].Visible = false;
195                 DaSea[i].Image = imgFishDead.Image;
196                 fishCount -= 1;
```

```
197             score++;
198             lblscore.Text = score.ToString();
199         }
200     }
201     if (DaSea[i].Bounds.Intersects(Shark.Bounds))
202     {
203         DaSea[i].Top = label2.Top;
204         DaSea[i].Image = imgFishDead.Image;
205         fishCount -= 1;
206     }
207 }
208
209
210 // shark movement
211 if (randomx == 1)
212 {
213
214     Shark.Image = imgSharkR.Image;
215     Shark.Left += 9;
216
217 }
218 else
219 {
220
221     Shark.Image = imgSharkL.Image;
222     Shark.Left -= 9;
223
224 }
225
226 if (randomy == 1)
227 {
228     Shark.Top += 9;
229 }
230 else
231 {
232     Shark.Top -= 9;
233 }
234 // fix OOB
235 if (Shark.Top < label1.Top)
236 {
237     Shark.Top = label1.Top;
238 }
239 if (Shark.Top > label2.Top)
240 {
241     Shark.Top = label1.Top;
242 }
243
244 if (Shark.Left < 0)
245 {
```

```
246         Shark.Left = 0;
247     }
248     if (Shark.Left > this.Width - 150)
249     {
250         Shark.Left = this.Width - 150;
251     }
252     // message box func
253     if(fishCount <= 0)
254     {
255         timer1.Enabled = false;
256         DialogResult resultAnswer = DialogResult;
257         string messageString = null;
258         string msg = null;
259         messageString = "Game Over ";
260         msg = score.ToString();
261         resultAnswer = MessageBox.Show("Your Score is " + msg + "\n      ↗
                Would You Like To Play Again?",
262         messageString,
263         MessageBoxButtons.YesNo,
264         MessageBoxIcon.Question);
265         if (resultAnswer == DialogResult.Yes)
266         {
267             timer1.Enabled = true;
268             score = 0;
269
270             lblscore.Text = "0";
271
272             for (int n = 0; n < DaSea.Length; n++)
273             {
274                 randomx = r.Next(0, this.Width);
275                 randomy = r.Next(label1.Top, this.Height);
276                 DaSea[n].Left = randomx;
277                 DaSea[n].Top = randomy;
278                 DaSea[n].Image = imgFishAliveL.Image;
279                 fishCount += 1;
280             }
281
282
283         }
284         if (resultAnswer == DialogResult.No)
285         {
286             this.Close();
287         }
288     }
289
290 }
291
292 private void Fish9_Click(object sender, EventArgs e)
293 {
```

```
294
295     }
296
297     private void Hook_Click(object sender, EventArgs e)
298     {
299
300     }
301
302     private void Form1_KeyPress(object sender, KeyPressEventArgs e)
303     {
304
305
306     }
307     // keyboard controlls
308     private void Form1_KeyDown(object sender, KeyEventArgs e)
309     {
310         int whichKey = e.KeyValue;
311         if (whichKey == 32)
312         {
313             if (hooking)
314             {
315                 hooking = false;
316             }
317             else
318             {
319                 hooking = true;
320             }
321         }
322     }
323 }
324 }
325
```