

```
1 using BasicAI;
2 using Microsoft.VisualBasic.Devices;
3 using System.Drawing.Text;
4 using System.Windows.Forms.Design.Behavior;
5
6 //Programmer: Aaron Yoon
7 //Project: Basic AI
8 //date: 2/28/23
9 namespace BasicAI
10 {
11     public partial class Form1 : Form
12     {
13         int heavyMoveSpeed = 20;
14         double deltaX = 1;
15         double deltaY = 1;
16         Boolean bulletFired = false;
17         Boolean gameEnd = false;
18
19         private static Form1 instance = new Form1();
20
21         public static PictureBox[] lives = new PictureBox[3];
22         Snail snail1;
23         Snail snail2;
24         Snail snail3;
25         Snail snail4;
26         Snail snail5;
27         Snail snail6;
28         Snail snail7;
29         Snail snail8;
30         Snail snail9;
31         Snail snail10;
32
33         Snail[] snails = new Snail[10];
34
35         int points = 0;
36
37         public Form1()
38         {
39             InitializeComponent();
40             snail1 = new Snail(picSnail, snailTimer, healthBar, picHeavy, ↗
                lblProj);
41             snail2 = new Snail(picSnail2, snailTimer2, healthBar2, ↗
                picHeavy, lblProj);
42             snail3 = new Snail(picSnail3, snailTimer3, healthBar3, ↗
                picHeavy, lblProj);
43             snail4 = new Snail(picSnail4, snailTimer4, healthBar4, ↗
                picHeavy, lblProj);
44             snail5 = new Snail(picSnail5, snailTimer5, healthBar5, ↗
                picHeavy, lblProj);
```

...	eso\Desktop\GITA 1\projects\pursuit tracking\Form1.cs	2
-----	---	---

---

```

45         snail6 = new Snail(picSnail6, snailTimer6, healthBar6,
                                picHeavy, lblProj);
46         snail7 = new Snail(picSnail7, snailTimer7, healthBar7,
                                picHeavy, lblProj);
47         snail8 = new Snail(picSnail8, snailTimer8, healthBar8,
                                picHeavy, lblProj);
48         snail9 = new Snail(picSnail9, snailTimer9, healthBar9,
                                picHeavy, lblProj);
49         snail10 = new Snail(picSnail10, snailTimer10, healthBar10,
                                picHeavy, lblProj);

50
51         snails[0] = snail1;
52         snails[1] = snail2;
53         snails[2] = snail3;
54         snails[3] = snail4;
55         snails[4] = snail5;
56         snails[5] = snail6;
57         snails[6] = snail7;
58         snails[7] = snail8;
59         snails[8] = snail9;
60         snails[9] = snail10;
61
62
63
64     }
65
66     public static Form1 getInstance()
67     {
68         return instance;
69     }
70
71     private void Form1_Load(object sender, EventArgs e)
72     {
73
74         for(int i = 0; i < 10; i++)
75         {
76             snails[i].reset();
77         }
78
79         lives[0] = life1;
80         lives[1] = life2;
81         lives[2] = life3;
82     }
83
84     private void moveHeavy(double x, double y)
85     {
86         picHeavy.Left = (int)x - (int)(picHeavy.Width * 1);
87         picHeavy.Top = (int)y - (int)(picHeavy.Height / 2);
88         picHeavy.Image = picHeavy.Image;

```

```
89     }
90     private void picHeavy_MouseClick(object sender, MouseEventArgs e)
91     {
92     }
93     }
94
95     private void Form1_MouseClick(object sender, MouseEventArgs e)
96     {
97         deltaX = getDeltaX();
98         deltaY = getDeltaY();
99         bulletFired = true;
100        bulletTimer.Enabled = true;
101    }
102
103    private void heavyTimer_Tick(object sender, EventArgs e)
104    {
105    }
106    }
107
108    private void fireProj(double deltaX, double deltaY)
109    {
110        //change the rounding to a serate variable
111
112        double slopeX = deltaY / deltaX;
113
114        double x = lblProj.Left - (0.1 * deltaX);
115        double y = lblProj.Top - (0.1 * deltaY);
116
117        lblProj.Left = (int)x;
118        lblProj.Top = (int)y;
119    }
120    }
121
122    private double getDeltaX()
123    {
124        int heavyX = picHeavy.Left + (picHeavy.Width / 2);
125        int heavyY = picHeavy.Top + (picHeavy.Height / 2);
126
127        int mouseX = Cursor.Position.X;
128        int mouseY = Cursor.Position.Y;
129
130
131        double mouseDeltaX = heavyX - mouseX;
132        double mouseDeltaY = heavyY - mouseY;
133
134        return mouseDeltaX;
135    }
136    }
137    private double getDeltaY()
```

```
138     {
139         int heavyX = picHeavy.Left + (picHeavy.Width / 2);
140         int heavyY = picHeavy.Top + (picHeavy.Height / 2);
141
142         int mouseX = Cursor.Position.X;
143         int mouseY = Cursor.Position.Y;
144
145
146         double mouseDeltaX = heavyX - mouseX;
147         double mouseDeltaY = heavyY - mouseY;
148
149         return mouseDeltaY;
150     }
151 }
152 private void Form1_KeyDown(object sender, KeyEventArgs e)
153 {
154     int whichKey = e.KeyValue;
155     //lblDebug.Text = whichKey.ToString();
156
157     if (!gameEnd)
158     {
159         if (whichKey == 87) //w
160         {
161             picHeavy.Top -= heavyMoveSpeed;
162         }
163
164         else if (whichKey == 65) //a
165         {
166             picHeavy.Left -= heavyMoveSpeed;
167         }
168
169         else if (whichKey == 83) //s
170         {
171             picHeavy.Top += heavyMoveSpeed;
172         }
173
174         else if (whichKey == 68) //d
175         {
176             picHeavy.Left += heavyMoveSpeed;
177         }
178
179         if (!bulletFired)
180         {
181             lblProj.Left = picHeavy.Left + (picHeavy.Width / 2);
182             lblProj.Top = picHeavy.Top + (picHeavy.Height / 2);
183         }
184     }
185 }
186
```

```
187     private void bulletTimer_Tick(object sender, EventArgs e)
188     {
189
190
191
192         if (snail1.autoDamage()
193             || snail2.autoDamage()
194             || snail3.autoDamage()
195             || snail4.autoDamage()
196             || snail5.autoDamage()
197             || snail6.autoDamage()
198             || snail7.autoDamage()
199             || snail8.autoDamage()
200             || snail9.autoDamage()
201             || snail10.autoDamage()
202         )
203         {
204             resetBullet();
205             points += 100;
206             label1.Text = "Score: " + points.ToString();
207         }
208
209
210
211         if (lblProj.Left > this.Width || lblProj.Right < 0 ||
212             lblProj.Top > this.Height || lblProj.Bottom < 0 )
213         {
214             resetBullet();
215
216         }
217
218
219
220         fireProj(deltaX, deltaY);
221     }
222
223     public void resetBullet()
224     {
225         bulletTimer.Enabled = false;
226         bulletFired = false;
227
228         lblProj.Left = picHeavy.Left + (picHeavy.Width / 2);
229         lblProj.Top = picHeavy.Top + (picHeavy.Height / 2);
230
231     }
232
233
234     private void picSnail_Click(object sender, EventArgs e)
```

```
235     {
236
237     }
238
239     private void snailTimer_Tick(object sender, EventArgs e)
240     {
241         for(int i = 0; i < snails.Length; i++)
242         {
243             snails[i].pursuit();
244         }
245
246         for (int i = 0; i < snails.Length; i++)
247         {
248             if (snails[i].snailTouch())
249             {
250                 takeDamage();
251                 snails[i].moveSnail(0, 0);
252             }
253         }
254
255     }
256
257     public void takeDamage()
258     {
259         if (lives[2].Image != null)
260         {
261             lives[2].Image = null;
262         }
263         else if (lives[1].Image != null)
264         {
265             lives[1].Image = null;
266         }
267         else if (lives[0].Image != null)
268         {
269             lives[0].Image = null;
270             picHeavy.Image = picHeavyDead.Image;
271             snailTimer.Enabled = false;
272             gameEnd = true;
273             lblEnd.Text = "GAME OVER!";
274             lblEnd.Visible = true;
275             lblEnd.BringToFront();
276
277             btnReset.Visible = true;
278             btnReset.Enabled = true;
279         }
280     }
281
282
283     private void button1_Click(object sender, EventArgs e)
```

```
284     {
285
286         snailTimer.Interval = 100;
287         if (snailTimer.Enabled)
288         {
289             snailTimer.Enabled = false;
290         }
291         else
292         {
293             snailTimer.Enabled = true;
294         }
295         btnStart.Visible = false;
296         btnStart.Enabled = false;
297         btnStart.SendToBack();
298     }
299
300     private void btnReset_Click(object sender, EventArgs e)
301     {
302         btnReset.Visible = false;
303         btnReset.Enabled = false;
304
305         snailTimer.Enabled = true;
306
307         lives[2].Image = picLife.Image;
308         lives[1].Image = picLife.Image;
309         lives[0].Image = picLife.Image;
310
311         bulletFired = false;
312
313         picHeavy.Image = heavyBank.Image;
314         gameEnd = false;
315         lblEnd.Text = "VICTORY!";
316         lblEnd.Visible = false;
317         for(int i = 0; i < snails.Length; i++)
318         {
319             snails[i].reset();
320         }
321         points = 0;
322     }
323
324     private void button1_Click_1(object sender, EventArgs e)
325     {
326         for(int i = 0; i < snails.Length; i++)
327         {
328             snails[i].spawn();
329         }
330     }
331
332     private void pictureBox3_Click(object sender, EventArgs e)
```

---

```
333     {
334
335     }
336
337     private void label6_Click(object sender, EventArgs e)
338     {
339
340     }
341 }
342 }
```