```
1  using Microsoft.VisualBasic.Devices;
2  using System.Windows.Forms.Design.Behavior;
3
4  //Programmer: Aaron Yoo
5  //Project: Basic AI
6  //date: 2/28/23
7  namespace BasicAI
8  {
9      public partial class Form1 : Form
10     {
11         int heavyMoveSpeed = 20;
12         double deltaX = 1;
13         double deltaY = 1;
14         Boolean bulletFired = false;
15         Boolean gameEnd = false;
16
17         PictureBox[] lives = new PictureBox[3];
18
19         public Form1()
20         {
21             InitializeComponent();
22         }
23
24         private void Form1_Load(object sender, EventArgs e)
25         {
26             snailTimer.Enabled = false;
27
28             lives[0] = life1;
29             lives[1] = life2;
30             lives[2] = life3;
31         }
32
33         private void moveHeavy(double x, double y)
34         {
35             picHeavy.Left = (int)x - (int)(picHeavy.Width * 1);
36             picHeavy.Top = (int)y - (int)(picHeavy.Height / 2);
37             picHeavy.Image = picHeavy.Image;
38         }
39
40         private void moveSnail(double x, double y)
41         {
42             picSnail.Left = (int)x - picSnail.Width / 2;
43             picSnail.Top = (int)y - picSnail.Height / 2;
44
45         }
46         private void picHeavy_MouseClick(object sender, MouseEventArgs e)
47         {
48
49         }
```

```
50
51          private void Form1_MouseClick(object sender, MouseEventArgs e)
52          {
53              deltaX = getDeltaX(false);
54              deltaY = getDeltaY(false);
55              bulletFired = true;
56              bulletTimer.Enabled = true;
57          }
58
59          private void heavyTimer_Tick(object sender, EventArgs e)
60          {
61
62          }
63
64          private void fireProj(double deltaX, double deltaY)
65          {
66              //change the rounding to a serate variable
67
68              double slopeX = deltaY / deltaX;
69
70              double x = lblProj.Left - (0.1 * deltaX);
71              double y = lblProj.Top - (0.1 * deltaY);
72
73              lblProj.Left = (int)x;
74              lblProj.Top = (int)y;
75
76          }
77
78          private double getDeltaX(Boolean snail)
79          {
80              int heavyX = picHeavy.Left + (picHeavy.Width / 2);
81              int heavyY = picHeavy.Top + (picHeavy.Height / 2);
82
83              int snailX = picSnail.Left + (picSnail.Width / 2);
84              int snailY = picSnail.Top + (picSnail.Height / 2);
85
86              int mouseX = Cursor.Position.X;
87              int mouseY = Cursor.Position.Y;
88
89              if (!snail)
90              {
91                  double mouseDeltaX = heavyX - mouseX;
92                  double mouseDeltaY = heavyY - mouseY;
93
94                  return mouseDeltaX;
95              }
96              else
97              {
98                  double snailDeltaX = heavyX - snailX;
```

```
 99                 double snailDeltaY = heavyY - snailY;
100
101                 return snailDeltaX;
102             }
103         }
104         private double getDeltaY(Boolean snail)
105         {
106             int heavyX = picHeavy.Left + (picHeavy.Width / 2);
107             int heavyY = picHeavy.Top + (picHeavy.Height / 2);
108
109             int snailX = picSnail.Left + (picSnail.Width / 2);
110             int snailY = picSnail.Top + (picSnail.Height / 2);
111
112             int mouseX = Cursor.Position.X;
113             int mouseY = Cursor.Position.Y;
114
115             if (!snail)
116             {
117                 double mouseDeltaX = heavyX - mouseX;
118                 double mouseDeltaY = heavyY - mouseY;
119
120                 return mouseDeltaY;
121             }
122             else
123             {
124                 double snailDeltaX = heavyX - snailX;
125                 double snailDeltaY = heavyY - snailY;
126
127                 return snailDeltaY;
128             }
129         }
130         private void Form1_KeyDown(object sender, KeyEventArgs e)
131         {
132             int whichKey = e.KeyValue;
133             //lblDebug.Text = whichKey.ToString();
134
135             if (!gameEnd)
136             {
137                 if (whichKey == 87) //w
138                 {
139                     picHeavy.Top -= heavyMoveSpeed;
140                 }
141
142                 else if (whichKey == 65) //a
143                 {
144                     picHeavy.Left -= heavyMoveSpeed;
145                 }
146
147                 else if (whichKey == 83) //s
```

```
148                 {
149                     picHeavy.Top += heavyMoveSpeed;
150                 }
151 
152                 else if (whichKey == 68) //d
153                 {
154                     picHeavy.Left += heavyMoveSpeed;
155                 }
156 
157                 if (!bulletFired)
158                 {
159                     lblProj.Left = picHeavy.Left + (picHeavy.Width / 2);
160                     lblProj.Top = picHeavy.Top + (picHeavy.Height / 2);
161                 }
162             }
163         }
164 
165         private void bulletTimer_Tick(object sender, EventArgs e)
166         {
167             Boolean bulletHitD = false;
168             if (bulletHit())
169             {
170                 bulletHitD = true;
171             }
172 
173             if (lblProj.Left > this.Width || lblProj.Right < 0 ||
                  lblProj.Top > this.Height || lblProj.Bottom < 0 || bulletHit
                  ())
174             {
175                 if (bulletHitD)
176                 {
177                     if (healthBar.Width > 0)
178                     {
179                         enrageSnail();
180                         healthBar.Width -= 50;
181                     }
182                     if (healthBar.Width <= 0)
183                     {
184                         snailTimer.Enabled = false;
185                         picSnail.Image = deadPic.Image;
186                         gameEnd = true;
187                         lblEnd.Text = "VICTORY!";
188                         lblEnd.Visible = true;
189                         lblEnd.BringToFront();
190 
191                         btnReset.Visible = true;
192                         btnReset.Enabled = true;
193                     }
194                 }
```

```
195
196
197                 bulletFired = false;
198                 bulletTimer.Enabled = false;
199                 lblProj.Left = picHeavy.Left + (picHeavy.Width / 2);
200                 lblProj.Top = picHeavy.Top + (picHeavy.Height / 2);
201
202
203             }
204
205
206
207             fireProj(deltaX, deltaY);
208         }
209
210         private Boolean bulletHit()
211         {
212             return ((lblProj.Left > picSnail.Left &&
213                      lblProj.Right < picSnail.Right &&
214                      lblProj.Top > picSnail.Top &&
215                      lblProj.Bottom < picSnail.Bottom));
216
217         }
218
219
220         private Boolean snailTouch()
221         {
222             return ((Math.Abs(picSnail.Left - picHeavy.Left) <
                      picHeavy.Width  &&
223                      Math.Abs(picSnail.Right - picHeavy.Right) <
                     picHeavy.Width &&
224                      Math.Abs(picSnail.Top - picHeavy.Top) <
                     picHeavy.Height &&
225                      Math.Abs(picSnail.Bottom - picHeavy.Bottom) <
                     picHeavy.Height));
226
227         }
228
229         private void enrageSnail()
230         {
231             picSnail.Image = enragePic.Image;
232             picSnail.Left -= (int)(getDeltaX(true) * .25);
233             picSnail.Top -= (int)(getDeltaY(true) * .25);
234         }
235
236         private void picSnail_Click(object sender, EventArgs e)
237         {
238
239         }
```

```
240
241         private void snailTimer_Tick(object sender, EventArgs e)
242         {
243             double x = picSnail.Left + (0.05 * getDeltaX(true));
244             double y = picSnail.Top + (0.05 * getDeltaY(true));
245
246             picSnail.Left = (int)x;
247             picSnail.Top = (int)y;
248
249             if(snailTouch())
250             {
251                 takeDamage();
252                 picSnail.Left = this.Width;//this.Width - picSnail.Left;
253                 picSnail.Top = this.Height; //this.Height - picSnail.Top;
254             }
255
256         }
257
258         private void takeDamage()
259         {
260             if (lives[2].Image != null)
261             {
262                 lives[2].Image = null;
263             }
264             else if (lives[1].Image != null)
265             {
266                 lives[1].Image = null;
267             }
268             else if (lives[0].Image != null)
269             {
270                 lives[0].Image = null;
271                 picHeavy.Image = picHeavyDead.Image;
272                 snailTimer.Enabled = false;
273                 gameEnd = true;
274                 lblEnd.Text = "GAME OVER!";
275                 lblEnd.Visible = true;
276                 lblEnd.BringToFront();
277
278                 btnReset.Visible = true;
279                 btnReset.Enabled = true;
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             picSnail.Image = snailBank.Image;
314             picHeavy.Image = heavyBank.Image;
315             gameEnd = false;
316             lblEnd.Text = "VICTORY!";
317             lblEnd.Visible = false;
318             picSnail.Left = this.Width;
319             picSnail.Top = this.Height;
320 
321             healthBar.Width = 1123;
322 
323             picHeavy.Top = 500;
324             picHeavy.Left = 500;
325 
326 
327 
328 
329         }
330     }
331 }
```