

```
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 using static
    System.Windows.Forms.VisualStyles.VisualStyleElement.StartPanel;
11
12 namespace staircase
13 {
14     //Programmer: Aaron Yoon
15     //Date 12/2/22
16     //Project: Stair case simulation
17     public partial class Form1 : Form
18     {
19         int totalSteps = 0;
20         int ghostDoor = 0;
21         int monsterDoor = 0;
22
23         System.Random r = new System.Random((int)
            System.DateTime.Now.Ticks);
24         bool onOff = false;
25
26         PictureBox[] berdman = new PictureBox[8];
27         int berdMove = 0;
28         int berdPos = 5;
29         public Form1()
30         {
31             InitializeComponent();
32         }
33
34         private void Form1_Load(object sender, EventArgs e)
35         {
36             berdman[0] = picStair1;
37             berdman[1] = picStair2;
38             berdman[2] = picStair3;
39             berdman[3] = picStair4;
40             berdman[4] = picStair5;
41             berdman[5] = picStair6;
42             berdman[6] = picStair7;
43             berdman[7] = picStair8;
44
45
46         }
47     }
```

```
48     private void btnClose_Click(object sender, EventArgs e)
49     {
50         this.Close();
51     }
52
53     private void move()
54     {
55         //int berdBeforePos = berdPos;
56         berdMove = r.Next(-1, 2);
57
58         if (berdMove != 0)
59             berdman[berdPos].Image = null;
60
61
62
63         if (berdMove == 1 && berdPos != 7)
64         {
65             berdPos += berdMove;
66             moveUp();
67             totalSteps++;
68         }
69         else if (berdMove == -1 && berdPos != 0)
70         {
71             berdPos += berdMove;
72             moveDown();
73             totalSteps++;
74         }
75     }
76
77     private void moveUp()
78     {
79         if (berdPos < 7)
80         {
81             picDoorDown.Image = picDoorClosed.Image;
82             picDoorUp.Image = picDoorClosed.Image;
83             berdman[berdPos].Image = picBerdRight.Image;
84             txtDebug.Text = berdPos.ToString();
85         }
86         else if (berdPos == 7)
87         {
88             picDoorUp.Image = picDoorGhost.Image;
89             ghostDoor ++;
90
91         }
92
93     }
94
95     private void moveDown()
96     {
```

```
197         if (berdPos > 0)
198         {
199             txtDebug.Text = berdPos.ToString();
200             picDoorDown.Image = picDoorClosed.Image;
201             picDoorUp.Image = picDoorClosed.Image;
202             berdman[berdPos].Image = picBerdLeft.Image;
203         }
204         if (berdPos == 0)
205         {
206             picDoorDown.Image = picDoorMonster.Image;
207             monsterDoor ++;
208         }
209     }
210 }
211
212 private void btnStart_Click(object sender, EventArgs e)
213 {
214     move();
215 }
216
217 private void timer_Tick(object sender, EventArgs e)
218 {
219     move();
220 }
221
222 private void btnStats_Click(object sender, EventArgs e)
223 {
224     MessageBox.Show(
225         "total Steps: " + totalSteps.ToString() + "\n"
226         + "Times Ghost Door opened: " + ghostDoor.ToString() +
227         "\n"
228         + "Times Monster Door opened: " + monsterDoor.ToString()
229         );
230 }
231
232 private void btnAuto_Click(object sender, EventArgs e)
233 {
234     if (timer.Enabled == false)
235         timer.Enabled = true;
236     else if (timer.Enabled == true)
237         timer.Enabled = false;
238 }
239 }
240 }
```