

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Drawing;
7 using System.Windows.Forms;
8
9 namespace Car_Soccer
10 {
11     class Sprite
12     {
13         internal double _speed;
14         internal float _x;
15         internal float _y;
16         internal double _angle;
17         internal Color _color;
18         internal float _size = 40;
19         internal Image _image;
20         public double Speed
21         {
22             get { return _speed; }
23         }
24         public float X
25         {
26             get { return _x; }
27         }
28         public float Y
29         {
30             get { return _y; }
31         }
32         public double Angle
33         {
34             get { return _angle; }
35         }
36         public float Size
37         {
38             get { return _size; }
39         }
40         public virtual void Draw(Graphics paper)
41         {
42             Brush br = new SolidBrush(_color);
43             Pen pen = new Pen(Color.Black);
44             //paper.FillEllipse(br, _x - _size/2, _y - _size/2, _size, _size);
45             paper.DrawImage(_image, _x - _size / 2, _y - _size / 2, _size,
46                 _size);
47             paper.DrawLine(pen, _x, _y, _x + (float)(_speed * 5F * Math.Cos
48                 (DegToRad(_angle - 90))), _y + (float)(_speed * 5F * Math.Sin
49                 (DegToRad(_angle - 90))));
50             paper.DrawLine(pen, _x, _y, _x + (float)(15 * Math.Cos(DegToRad
51                 (_angle - 90))), _y + (float)(15 * Math.Sin(DegToRad(_angle -
52                 90))));
53         }
54         public virtual void Advance()
55         {
56         }
57     }
58 }
```

```
52         Accelerate();
53
54         _x += (float)(_speed / 2.5F * Math.Cos(DegToRad(_angle-90)));
55         _y += (float)(_speed / 2.5F * Math.Sin(DegToRad(_angle-90)));
56         //_x += (float)(_speed/10F);
57     }
58     public virtual void Accelerate()
59     {
60
61     }
62     private double DegToRad(double degrees)
63     {
64         return degrees / 180 * Math.PI;
65     }
66     public virtual void Rotate()
67     {
68
69     }
70     public void WallCollide(PictureBox pictureBox)
71     {
72
73         RectangleF rect = new RectangleF(_x - _size / 2, _y - _size / 2, ↗
74             _size, _size);
75         if(rect.Intersects(new RectangleF(0, 0, pictureBox.Width, 0)))
76         {
77             if (_angle < 90 || _angle > 270)
78             {
79                 _angle = 270 - (_angle - 270);
80             }
81             else if (rect.Intersects(new RectangleF(60, 0, 0, ↗
82                 pictureBox.Height)))
83             {
84                 if (_angle > 180 && _angle < 360)
85                 {
86                     _angle = 180 - (_angle - 180);
87                 }
88             }
89             else if (rect.Intersects(new RectangleF(pictureBox.Width-60, ↗
90                 0, pictureBox.Width, pictureBox.Height)))
91             {
92                 if (_angle < 180 && _angle > 0)
93                 {
94                     _angle = 180 - (_angle - 180);
95                 }
96             }
97             else if (rect.Intersects(new RectangleF(0, pictureBox.Height, ↗
98                 pictureBox.Width, pictureBox.Height)))
99             {
100                 if (_angle > 90 && _angle < 270)
101                 {
102                     _angle = 270 - (_angle - 270);
103                 }
104             }
105         }
106         _angle %= 360;
107     }
```

```
104     public void If00B(PictureBox pictureBox)
105     {
106         if (_x < -20 || _x > pictureBox.Width + 20 || _y >
107             pictureBox.Height + 20 || _y < -20)
108         {
109             Reset(pictureBox);
110         }
111     public void Reset(PictureBox pictureBox)
112     {
113         _x = pictureBox.Width / 2;
114         _y = pictureBox.Height / 2;
115         _speed = 0;
116     }
117 }
118 }
119 }
```