

```
1 using UDK;
2
3 namespace Grafico1
4 {
5     public enum CharacterType
6     {
7         PLAYER,
8         THIEF,
9         BOSS,
10        BOMB
11    }
12    public class Character
13    {
14        public Rectangle rectangle = new Rectangle();
15        public CharacterType type;
16        public double velocity = 1;
17
18        private double[] policeColor = new double[4]
19        {
20            0.2,
21            0.2,
22            1.0,
23            0.8
24        };
25
26        private double[] thiefColor = new double[4]
27        {
28            0.2,
29            0.8,
30            0.2,
31            0.8
32        };
33        private double[] bossColor = new double[4]
34        {
35            0.8,
36            0.2,
37            0.2,
38            0.8
39        };
40        private double[] bombColor = new double[4]
41        {
42            0.8,
43            0.8,
44            0.1,
45            0.8
46        };
47
48
49        #region Constructores
50        public Character()
51        {
52
53        }
```

```
54
55     public Character(CharacterType typeValue,
56                     double positionXValue, double positionYValue,
57                     double widthValue, double heightValue)
58     {
59         type = typeValue;
60         rectangle.X = positionXValue;
61         rectangle.Y = positionYValue;
62         rectangle.SetWidth(widthValue);
63         rectangle.SetHeight(heightValue);
64     }
65     #endregion
66
67
68     #region Metodos
69     public void Draw(ICanvas canvas)
70     {
71         double posX = rectangle.X;
72         double posY = rectangle.Y;
73         double characterWidth = rectangle.Width;
74         double characterHeight = rectangle.Height;
75
76         if (type == CharacterType.PLAYER)
77         {
78             canvas.FillShader.SetColor(policeColor[0], policeColor[1], policeColor[2], policeColor[3]);
79             canvas.DrawRectangle(posX, posY, characterWidth, characterHeight);
80         }
81
82         if (type == CharacterType.THIEF)
83         {
84             canvas.FillShader.SetColor(thiefColor[0], thiefColor[1], thiefColor[2], thiefColor[3]);
85             canvas.DrawRectangle(posX, posY, characterWidth, characterHeight);
86         }
87
88         if (type == CharacterType.BOSS)
89         {
90             canvas.FillShader.SetColor(bossColor[0], bossColor[1], bossColor[2], bossColor[3]);
91             canvas.DrawRectangle(posX, posY, characterWidth, characterHeight);
92         }
93
94         if (type == CharacterType.BOMB)
95         {
96             canvas.FillShader.SetColor(bombColor[0], bombColor[1], bombColor[2], bombColor[3]);
97             canvas.DrawRectangle(posX, posY, characterWidth, characterHeight);
98         }
```

```
99     }
100
101     public void MovePlayer(bool horizontal, bool vertical, double value) ➤
102     {
103         if (!horizontal && vertical)
104             rectangle.Y += value;
105
106         if (!horizontal && !vertical)
107             rectangle.Y -= value;
108
109         if (horizontal && vertical)
110             rectangle.X += value;
111
112         if (horizontal && !vertical)
113             rectangle.X -= value;
114     }
115     public void MoveIA()
116     {
117         velocity += 0.001;
118         rectangle.Y = (Math.Sin(velocity) * 4) + 5;
119     }
120
121     public void DeleteThief()
122     {
123         rectangle.SetWidth(0.01);
124     }
125     #endregion
126 }
127 }
128
```