```
1 using System;
 2 using System.Collections.Generic;
 3 using System.Text;
 4 using Android.Views;
 5 using Microsoft.Xna.Framework;
 6 using Microsoft.Xna.Framework.Graphics;
 8 namespace GameDemo.Shared
 9
10
        class BackgroundSprite
11
12
            Texture2D texture;
13
            Vector2 position;
14
            Vector2 size;
            float speed;
15
16
            Color color;
17
            bool loops;
18
            string name;
19
20
21
            public bool Loops
22
23
                get { return loops; }
24
            }
25
            public int FrameWidth
26
27
            {
28
                get { return (int)size.X; }
29
30
31
            public string Name
32
            {
33
                get { return name; }
34
            public BackgroundSprite(Texture2D texture, Vector2 position, Vector2
35
              size, float speed, Color color,bool loops)
36
37
38
                this.texture = texture;
39
                this.name = texture.Name;
40
                this.position = position;
41
                this.size = size;
42
                this.speed = speed;
43
                this.color = color;
44
                this.loops = loops;
45
            }
46
47
            public virtual void Draw(GameTime gametime,SpriteBatch spriteBatch)
48
            {
49
                spriteBatch.Draw(texture, new Rectangle((int)position.X, (int)
                  position.Y, (int)size.X, (int)size.Y), color);
50
51
52
            // for looping background
53
            public virtual void Draw(GameTime gametime, SpriteBatch spriteBatch,
              int screenWidth)
```

```
...ursova)\BloodLetter\GameDemo.Shared\BackgroundSprite.cs
54
55
                for(int i = 0; i < screenWidth / FrameWidth+1;i++)</pre>
56
57
                    spriteBatch.Draw(texture, new Rectangle((int)position.X
                      +(i*FrameWidth), (int)position.Y, (int)size.X, (int)size.Y), →
                      color);
                    spriteBatch.Draw(texture, new Rectangle((int))position.X + (i * →
58
                      FrameWidth) + screenWidth, (int)position.Y, (int)size.X,
                      (int)size.Y), color);
59
                }
60
61
62
            }
            public virtual void Update(GameTime gametime, int screenWidth)
63
64
                // if the player moves forwards the screen should go backwards
65
66
                position.X += speed;
67
                if (position.X <= -screenWidth)</pre>
68
                {
69
                    position.X = 0;
70
                }
71
            }
72
        }
```

73 }74