

```
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;
7
8 namespace GameDemo.Shared
9 {
10     class BackgroundSprite
11     {
12         Texture2D texture;
13         Vector2 position;
14         Vector2 size;
15         float speed;
16         Color color;
17         bool loops;
18         string name;
19
20
21         public bool Loops
22         {
23             get { return loops; }
24         }
25
26         public int FrameWidth
27         {
28             get { return (int)size.X; }
29         }
30
31         public string Name
32         {
33             get { return name; }
34         }
35         public BackgroundSprite(Texture2D texture, Vector2 position, Vector2 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)
```

```
54     {
55         for(int i = 0; i < screenWidth / FrameWidth+1;i++)
56         {
57             spriteBatch.Draw(texture, new Rectangle((int)position.X
58                 +(i*FrameWidth), (int)position.Y, (int)size.X, (int)size.Y),
59                 color);
60             spriteBatch.Draw(texture, new Rectangle((int)position.X + (i *
61                 FrameWidth) + screenWidth, (int)position.Y, (int)size.X,
62                 (int)size.Y), color);
63         }
64     }
65     public virtual void Update(GameTime gametime, int screenWidth)
66     {
67         // if the player moves forwards the screen should go backwards
68         position.X += speed;
69         if (position.X <= -screenWidth)
70         {
71             position.X = 0;
72         }
73     }
74 }
```