

```

1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  using Microsoft.Xna.Framework;
6  using Microsoft.Xna.Framework.Content;
7  using Microsoft.Xna.Framework.Graphics;
8
9  namespace Demo_Animation_Explosion
10 {
11     /// <summary>
12     /// a class to generate text on the screen
13     /// </summary>
14     public class ScreenText
15     {
16         #region Fields
17
18         // sprite font
19         private SpriteFont _textFont;
20
21         #endregion
22
23         #region PROPERTIES
24
25         // boolean to set status through the game loop
26         public bool Active { get; set; }
27
28         #endregion
29
30         #region Constructors
31
32         /// <summary>
33         /// construct a new screen text object
34         /// </summary>
35         /// <param name="contentManager">the content manager</param>
36         public ScreenText(ContentManager contentManager)
37         {
38             // load the font
39             _textFont = contentManager.Load<SpriteFont>("Text");
40
41             // initialize objects status as not active
42             Active = false;
43         }
44
45         #endregion
46
47         #region Public methods
48
49         /// <summary>
50         /// draw the text
51         /// </summary>
52         /// <param name="spriteBatch">the spriteBatch</param>
53         /// <param name="location">the Vector2 location of the text on the screen</param>
54         public void Draw(SpriteBatch spriteBatch, Vector2 location, string message)
55         {
56             spriteBatch.DrawString(_textFont, message, new Vector2(100, 100),
57                                     Color.Black);
58         }
59
60         #endregion
61     }
62 }

```