```
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        /// the Apple class
13        /// </summary>
14        public class Apple
15        {
16            #region Fields
17
18            // sprite image
19            private Texture2D _sprite;
20
21            // sprite location on screen
22            private Rectangle _drawRectangle;
23
24            // set sprite size
25            private const int _SPRITE_WIDTH = 64;
26            private const int _SPRITE_HEIGHT = 64;
27
28            #endregion
29
30            #region PROPERTIES
31
32            // Boolean to set status through the game loop
33            public bool Active { get; set; }
34
35            #endregion
36
37            #region Constructors
38
39            /// <summary>
40            /// constructs a new explosion object
41            /// </summary>
42            /// <param name="contentManager">the content manager</param>
43            public Apple(ContentManager contentManager)
44            {
45                // load the sprite
46                _sprite = contentManager.Load<Texture2D>("apple");
47
48                // initialize objects status as not active
49                Active = false;
50            }
51
52            #endregion
53
54            #region Public methods
55
```

```
56          /// <summary>
57          /// draw the apple
58          /// </summary>
59          /// <param name="spriteBatch">the spritebatch</param>
60          /// <param name="location">the Vector2 location of the apple on the screen<
            /param>
61          public void Draw(SpriteBatch spriteBatch, Vector2 location)
62          {
63              // set the draw rectangle for the sprite on the screen
64              _drawRectangle = new Rectangle(0, 0, _SPRITE_WIDTH, _SPRITE_HEIGHT);
65              _drawRectangle.X = (int)location.X;
66              _drawRectangle.Y = (int)location.Y;
67
68              spriteBatch.Draw(_sprite, _drawRectangle, Color.White);
69          }
70
71          #endregion
72
73      }
74  }
```