```
using System;
using System.Collections.Generic;
using System.Text;

using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.Graphics;

using Demo_Animation_Explosion;

namespace Demo_Animation_Explosion
{
    /// <summary>
    /// the animated explosion class
    /// </summary>
    public class Explosion
    {
        #region Fields

        // sprite strip info
        private Texture2D _spriteStrip;
        private const int _ROWS = 3;
        private const int _COLUMNS = 3;
        private const int _NUMBER_OF_FRAMES = 9;

        // explosion location
        private Rectangle _drawRectangle;

        // frame location on sprite strip
        private Rectangle _sourceRectangle;

        // frame size on sprite strip
        private int _frameWidth;
        private int _frameHeight;

        // fields used to track and draw animation frames
        private int _currentFrame;
        private int _frameTime;
        private int _elapsedFrameTime;

        #endregion

        #region PROPERTIES

        // Boolean to set status through the game loop
        public bool Active { get; set; }

        #endregion

        #region Constructors

        /// <summary>
        /// Construct a new explosion object
        /// </summary>
        /// <param name="contentManager">the content manager</param>
        public Explosion(ContentManager contentManager)
        {
            // initialize animation
            _currentFrame = 0;
            _elapsedFrameTime = 0;
            _frameTime = 50;

            LoadContent(contentManager);

            // initialize objects status as not active
            Active = false;
        }

        #endregion

        #region Public methods

        /// <summary>
        /// Updates the explosion. This only has an effect if the explosion animation is playing
        /// </summary>
        /// <param name="gameTime">the game time</param>
        public void Update(GameTime gameTime)
        {
            if (Active)
            {
                // check for advancing animation frame
                _elapsedFrameTime += gameTime.ElapsedGameTime.Milliseconds;
                if (_elapsedFrameTime > _frameTime)
                {
                    // reset frame timer
                    _elapsedFrameTime = 0;

                    // advance the animation
                    if (_currentFrame < _NUMBER_OF_FRAMES - 1)
                    {
                        _currentFrame++;
                    }
                    else
                    {
                        // reached the end of the animation
                        // set the objects status to inactive
                        Active = false;
                    }
                }
            }
        }

        /// <summary>
        /// Draws the explosion. This only has an effect if the explosion animation is playing
        /// </summary>
        /// <param name="spriteBatch">the spritebatch</param>
        public void Draw(SpriteBatch spriteBatch, Vector2 location)
        {
            // calculate frame size
            _frameWidth = _spriteStrip.Width / _ROWS;
            _frameHeight = _spriteStrip.Height / _COLUMNS;

            // set the source rectangle for the current frame on the sprite strip
            _sourceRectangle = new Rectangle(0, 0, _frameWidth, _frameHeight);

            _sourceRectangle.X = (_currentFrame % _ROWS) * _frameWidth;
            _sourceRectangle.Y = (_currentFrame / _COLUMNS) * _frameHeight;

            // set the draw rectangle for the current frame on the screen
            _drawRectangle = new Rectangle(0, 0, _frameWidth, _frameHeight);
            _drawRectangle.X = (int)location.X;
            _drawRectangle.Y = (int)location.Y;

            spriteBatch.Draw(_spriteStrip, _drawRectangle, _sourceRectangle, Color.White);
        }

        #endregion

        #region Private methods

        /// <summary>
        /// Load the content for the explosion
        /// </summary>
        /// <param name="contentManager">the content manager</param>
        private void LoadContent(ContentManager contentManager)
        {
            // load the animation strip
            _spriteStrip = contentManager.Load<Texture2D>("explosion");
        }

         #endregion

    }
}
```