# Creating a simple particle explosion in C#

Create a new Windows Forms solution. We are going to make a particle explosion so let’s call our project “Explosion”.

## Making the particle class

Let’s start by creating a particle class. Select ***Project >add class*** from the menu. Call the new class Particle.cs

We need to give it a start location and a direction and speed. The latter two will be supplied in a vector. So let’s add the two attributes first.

Visual Studio is going to complain about the vector class we’re using because we haven’t got the correct using directive. Add using System.Numerics; to the namespace directives. You may not have the Numerics package installed but VS should offer to install it for you. Once done the red underline should go away.

Now let’s write the constructor for the new class. We need an X and Y location for the particle and the two float values needed by the vector. Notice how we use the x and y location to create a new point. This is used as our particle starting point.

A white background with black text

Description automatically generated

Now I need a class method to update the particle position. Here it is. Notice we do some kind of vector maths by adding the location point to the vector to calculate a new position.

A white background with black text

Description automatically generated

That’s all we need for the Particle class. We can now build our form interface. See the next page.

## Creating our form

Go to the form designer and add a button (this will be button1) and a timer (timer1).

Change the text on the button to something like BOOM or BANG or whatever you like.

Make sure the time is set to an interval of 100mS and Enabled is set to false. These are the default values but best to check.

Double click the button to open the Form1 code window and create the button handler then go back to the Form designer and do the same with the Timer. You should have two handlers now.

Now we are going to set up an array to hold 100 particles. This is a class attribute so it goes in under the form class definition as shown below.

A close-up of a computer code

Description automatically generated

Next, we are going to instantiate our array of particle individually using a loop.

### The Form constructor

This goes in the constructor for the Form under the InitializeComponent( ) command.

A computer code with text

Description automatically generated with medium confidence

Here I’m seeding two Random objects but Random on gives values that are positive integers and I want positive and negative floats between -2.5 and 2.5 so I had to get a bit constructive. The X and Y locations passed to the constructor are the centre of the form on the screen (no matter what size the form is.

### Timer code

Next here the code for the Timer handler. It just calls Update( ) on each particle in the array (it does this every 100mS while the timer is running.

### A white background with black and green text Description automatically generated

### The Button Code

For the button we just need to set the timer to enabled.



### The Paint Event

Finally, we need to paint the particles to the Form. We’ll need to create a Form\_Paint handler for this. Go back to the form designer, select the form and then in the form properties select the events button. It looks like a lightning bolt. Scroll down the list of events until you see Paint and click on it.

You should now have a Form\_Paint handler. Her’s the code to paint the particles to the form. Add the following code to paint each particle in turn.

