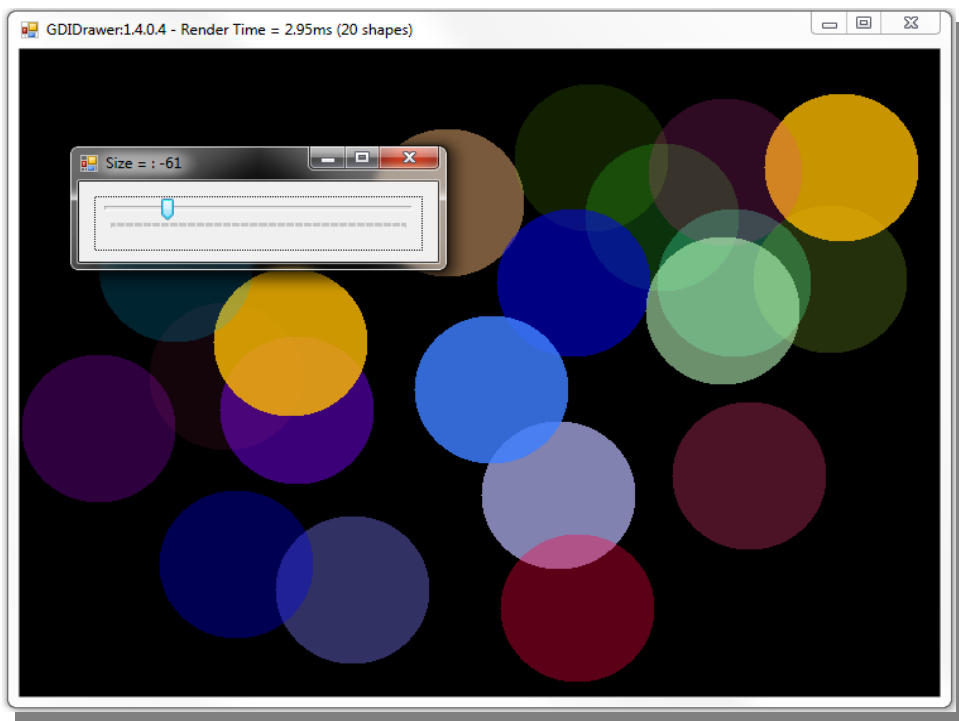


CMPE2300 - ICA03 - Static Balls of Fun

In this ICA you implement a simple class including of some static methods and a static constructor to act as helpers

Create a new Windows form project for this ice, add the reference for the CDrawer
Construct a UI for your form fashioned after the following capture:



Using Add...new Class, then create a class named **Ball** in a new file.

Your new class will contain the following instance and static fields:

- a **static** field of type Random, use an initializer - this will be shared amongst all the Balls
- a **static** field of type CDrawer, initialize to null - this will be shared amongst all the Balls
- a **static** field of type int, representing the Radius of the ball - **ADD** a public property as well, set only, take the absolute value of the value and set the Radius
- an **instance** field of type Color for the ball color
- an **instance** field of type Point for the ball location
- an **instance** field of type int for X, Y velocity (like the last ica)
- an **instance** field of type int called `_iAlive` representing the lifespan of this ball before "Reincarnation"

Add a **static** constructor to your class

- Use your static Random object to initialize :
 - Your CDrawer member to a random width between 600-900 and height between 500, 800 with continuous update off
 - your static Radius member to a random number between 10 and 80

Add an **instance** constructor, no arguments – it's a Default Constructor

- initialize the color to a random color
- initialize the velocities to a random value between -10 and 10 (use your static Random object)
- initialize the location to a random location but ensuring the it is entirely visible (!! Radius)

Add a public **instance** method called **ShowBall**, returns nothing, accepts nothing (Balls "own" a shared CDrawer now) - this only adds a CenteredEllipse to the drawer using the instance location. Use the `_iAlive` member as the Opacity value for your Color and the static Radius

Add a public **instance** method called **MoveBall**, returns nothing, accepts nothing - this moves our ball similar to our last ica, but with a twist.

- Decrement your `_iAlive` member, if it has fallen below 1, set a new random location and set `_iAlive` to a random value between 50 and 127. This causes our "Reincarnation"
- Like before, determine if we are going out of bounds, if so negate our velocity AND set our potentially offending coordinate to be in-bounds – ie. If we are going off the right, negate our X velocity and set X to the CDrawer Width – Radius, thereby ensuring we will not go out of bounds

Add a public **static** property of type bool - a virtual one, not tied to an actual field at all, called **Loading**

- supply set only, no get
- this property acts like a "switch", the user will set Loading to true, then using `MoveBall/ShowBall()` add their Balls, then set Loading to false, at which time the Balls are shown
- this will be implemented by checking the Load "value", if the user set it to true, clear the drawer, if he set it to false, render the drawer ** This will be contingent on the continuous update of the drawer being false

In the main form : add yourself a List of your Ball type.

○ Add the trackbar (minimum = -100, max = 100), and Add a handler for the track bar for Scroll, set the Text of the form as shown, and set the static Ball Radius property to the trackbar value. (Note : use of class name)

○ Add a key down handler (remember the KeyPreview Property)

- if it was the Add key : add 5 Balls to our list
- if it was the Escape key : clear our list

○ Add a thread method to run always upon startup. This thread will show our balls.

To do this : first set the Loading property to true (in our property this clears the drawer), then iterate through our list of balls invoking `Move/Show`, after iteration, finally set Loading to false (in our property this Renders our drawer) and your balls should all now be visible. Let the thread nap for 25ms between iterations. Would any data synchronization be required here ? On what ?