In-Class BONUS Lab (2 pts)   Submit via Canvas by noon

Student Group

You are encouraged to work in pairs. Both students must be present in class during this lab activity. In order to earn BONUS points, both students should submit a compressed solution folder via Canvas by noon of the class activity day. Turn this sheet back to the instructor during class to initiate grading activity.

Student #1 Name: _Devin Waldon___________Student #2 Name: __________________________

Purpose Explore using the Graphics Design Interface (GDI+) with WinForms. Brushes, line objects, filled objects, custom colors.

Instructions

1. Begin by creating a project and a blank form. Give the form a title "Bounce"

2. Add data members to the Form1 class as follows:

```
 const int iTimerInterval = 25;    // In milliseconds
const int iBallSize = 12;        // As fraction of client area
const int iMoveSize = 4;         // As fraction of ball size

 Bitmap bitmap;                  // created on the fly
int xCenter,  yCenter;           // center of ball
int cxRadius, cyRadius;          // radius of the ball
int cxMove,   cyMove;            // horizontal & vertical distance to move
int cxTotal,  cyTotal;           // width and height of the ball image
```

3. Add the following to the constructor of Form1 class:

```
   // Set form's background color to white    this.BackColor = _______________;

   // Create a timer, register a timer event handler, and set tick interval
```

```
Timer timer = new Timer();

timer.Interval = iTimerInterval;

timer.Tick += new EventHandler(TimerOnTick);


   // Start the timer running...

timer.Start();


   // force redraw on form resize

this.ResizeRedraw = true;


   // calculate form's center

 xCenter = ______________________;

yCenter = ______________________;
```

4. Create a TimerOnTick event handler taking object o and EventArgs ea as parameters. Add the following code:

```
   // draw the ball on the form's drawing canvas, get Graphics object from

// the form

 Graphics g = ______________________________
```

In-Class BONUS Lab (2 pts)   Submit via Canvas by noon

```
  // Draw the bitmap on the drawing canvas (Use MSDN for syntax help)

g.DrawImage(bitmap, ________, _________ ,_________, ________);
```

```
// Keep moving the ball to the next spot
xCenter += ______;
yCenter += ______;

  // Do we need to reverse direction on the next move?
// i.e., change cxMove, cyMove
if (xCenter+cxRadius >= DisplayRectangle.Width || _________________)        cxMove = -cxMove;
if (yCenter+cyRadius >= DisplayRectangle.Height || yCenter-cyRadius <= 0)        cyMove = -cyMove;
```

5. Create on Paint event handler and add the following code:

```
 Graphics g = _______________; // get form Graphics object from PaintEventArgs
g.Clear(BackColor);

 // Make ball's radius proportional to form's size, use display canvas width
 cxRadius = ______________________ / iBallSize;
 cyRadius = ______________________ / iBallSize;

 // Make the amount of movement of the ball proportional to the ball's radius
 // or 1 whichever is greater. Use a method of the Math class from System
// namespace
cxMove = ________(1, cxRadius / iMoveSize);
cyMove = ________(1, cyRadius / iMoveSize);

 // Make bitmap size take into account the ball size and extra space around it
// This is done to avoid erasing previous ball image
cxTotal = 2 * (cxRadius + cxMove);
cyTotal = 2 * (cyRadius + cyMove);
```

```
   // create a bitmap; MSDN lookup for syntax
 bitmap = new Bitmap(________,_________);


   // Get the Graphics object for alteration from bitmap, MSDN lookup
 g = Graphics.___________________;


   g.Clear(BackColor);
// Draw circle on the bitmap in your custom color. Use Paint to
// select the custom color's RGB components.
g.FillEllipse(____________________,
              new Rectangle(cxMove, cyMove, 2*cxRadius, 2*cyRadius));


   ________________ // dispose of the Graphics object
```

6. Experiment by making these changes. Run your application after each change then back it out (0.2 point each)

Change iMoveSize to 10 ➔ what is the effect of doing that? The ball moves slower.

Change iBallSize to 5 ➔ what is the effect of doing that? The ball gets bigger.

Change iTimerInterval to 5 ➔ what is the effect of doing that? The ball bounces back and forth faster.

Set a break-point on line xCenter += cxMove; in TimerOnTick.

Press F5 several times, what is the effect? The ball froze and program force closed.

Set a break-point on line cxMove = -cxMove; in TimerOnTick. Effect? The ball flew off the screen then program force closed.