1. We will use the skills/knowledge that we obtained from the first GUI Graphics project to do a short animation. Open a CLR/Windows Form Application as in Step 1 of the GUI1.htm document.

2. Drag a panel (in the Toolbox under Container) onto the form. Also drag a Timer (in the Toolbox under Components) The Timer will appear under the Form and will not show during runtime.

3. Select the timer and make sure that the Enabled Property (in the Property window) is set to true.
4. Select the panel, and in the Properties Window, either change the Background Color of the panel or place a picture into the panel using the BackgroundImage property. Here is one that you can use, or choose any .jpg file of your choice.
5. Use CTRL-F5 to run your code.

6. We are going to have a ball bounce around the panel (or park, if you used the picture above).
   To do this:
   - we will write the code to draw a filled-in circle (that’s just what we did in the previous
     GUI Lab!) in the topmost left corner of the panel
   - Then we will write code to refresh the panel with a new picture, where the ball is in a
     slightly different place (by changing the coordinates of the rectangle by adding to the x
     and y coordinates).
   - If the ball’s rectangle coordinates are moving off the panel, we will begin subtracting
     from x or y, instead of adding

7. Double click on the panel to get to the event handler for the panel. Above the event handler
   (and right under the `#pragma endregion`) line, type in:
   ```
   static int x = 0, y = 0, size = 20, yDiff = 5, xDiff = 5;
   ```

   We did not spend time on `static` in this class, but these are integers that are outside any of
   the functions, but can be used by all of the functions in this class.
8. Inside the Paint event handler for the panel type the following:

```csharp
    //rect is of type Rectangle with top left corner at (x,y) and
    //height of 20 and width of 20
    Rectangle rect(x,y,20,20);

    //new instance of a Brush
    Brush^ blueBrush = gcnew Drawing::SolidBrush(Color::Blue);

    //e is a parameter to this function and it is of type
    //System::Windows::Forms::PaintEventArgs^
    //This type can be used to create Graphics objects
    e->Graphics->FillEllipse(blueBrush,rect);
```
9. Press CTL-F5 to run your program. You should see a blue ball in the top left corner of your panel.

10. Double click on the timer to open the code for the event. Type the following into the body of the function:

   ```cpp
   //change the x and y coordinates
   x += xDiff;
y += yDiff;

   //check if the x coordinate is off the right or left edge
   if ((x + size >= panel1->Width) || (x <= 0))
       xDiff = -xDiff; //if yes, go in the other direction

   //check if the y coordinate is off the top or bottom edge
   if ((y + size >= panel1->Height) || (y <= 0))
       yDiff = -yDiff; //if yes go in the other direction

   //causes calls the paint event again
   // so the ball will be painted in the new x and y spot
   panel1->Invalidate();
   ```
11. If the ball is moving too slowly, change the Interval property of the timer, by selecting it, and going to the property window. The Interval is the number of milliseconds between ticks.
12. For your own interest: What if there are obstacles in the way? Place shapes in the panel, and see if you can get the ball to go around them or bounce off them.