Simple Pong 1: Bouncing Ball

Contents

Time required: 30 minutes

Comment each line of code as shown in the tutorials and other code examples.

Follow all directions carefully and accurately. Think of the directions as a minimum requirement.

Simple Pong Project Sequence

To give you an idea of what this project entails, here is the project sequence. If you are being creative with the project, you might want to wait until you get to that stage of the project.

1. Moving Ball
2. Bouncing Ball
3. Keyboard Input
4. Collision Detection

5. Scoring and Speed
6. Sound

Program Description

For several weeks, we are going to work on a long-term project to create a simple Pong style Visual C# game. The first step is to setup the form, and then get our ping pong ball moving.

Many computer games have a "Game Loop". This is the main repeating section of code that keeps the game moving. We will use an infinite while loop.

Please read this short article describing a game loop. https://gamedevelopment.tutsplus.com/articles/gamedev-glossary-what-is-the-game-loop--gamedev-2469

All controls and objects in this program have their own properties such as Location(x, y) and characteristics such as speed and direction. We use this information to inform our game play.

Create the Form

1. Create a Visual C# project called **Simple Pong**.
2. Set the following Form properties.
    a) **Size:** 800, 500
    b) **BackColor:** White
    c) **Text:** Simple Pong
3. Add a PictureBox control. Set the following properties.
    a) Add the **Ping Pong Ball**.jpg picture to the PictureBox control. This image is attached to the assignment.

b) **Name:** Ball

c) **Size Mode:** AutoSize

d) **Location:** 0, 0

Game Loop

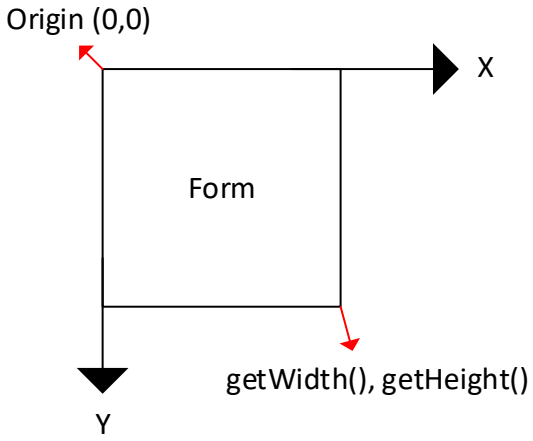
To setup our "Game Loop", we are going to intercept and change how our Windows Form program normally runs.

1. Do not run the program until this code is complete.
2. In Solution Explorer, open **Program.cs**.
3. Comment out any lines that start with **Application.Run**
4. Add the following lines as shown below the commented **Application.Run**

```
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    // Application.Run(new frmPong());
    Form1 gameForm = new Form1();
    gameForm.Show();
    gameForm.GameLoop();
}
```

5. Save and close **Program.cs**.

Move the Ball



To move our ping pong ball, we create some form level variables that will control how much our image will move in pixels for every timer tick. We move the ball 2 pixels at a time. We use x, y coordinates for movement. X is horizontal, Y is vertical.

1. Open Form1, go to the View menu, Click Code. Add the following form level variables to your program.

```
public partial class Form1 : Form
{
    const int TIME_DELAY = 10;  // How "fast" the game runs
    const int SPEED_X = 2;      // X movement speed constant
    const int SPEED_Y = 2;      // Y movement speed constant
    int MoveX = SPEED_X;        // Set horizontal movement/speed of the ball in pixels, based on SPEED_X
    int MoveY = SPEED_Y;        // Set vertical movement/speed of the ball in pixels, based on SPEED_Y
```

To move our ping pong ball, we use the **.Left** and **.Top** properties of the **PictureBox** to give us the current location of the top and left side of the ball. Every 10 milliseconds, we add 2 pixels to the current location values. This simulates motion by moving the ball 2 pixels diagonally every 10 ms.

More information on the DoEvents method.

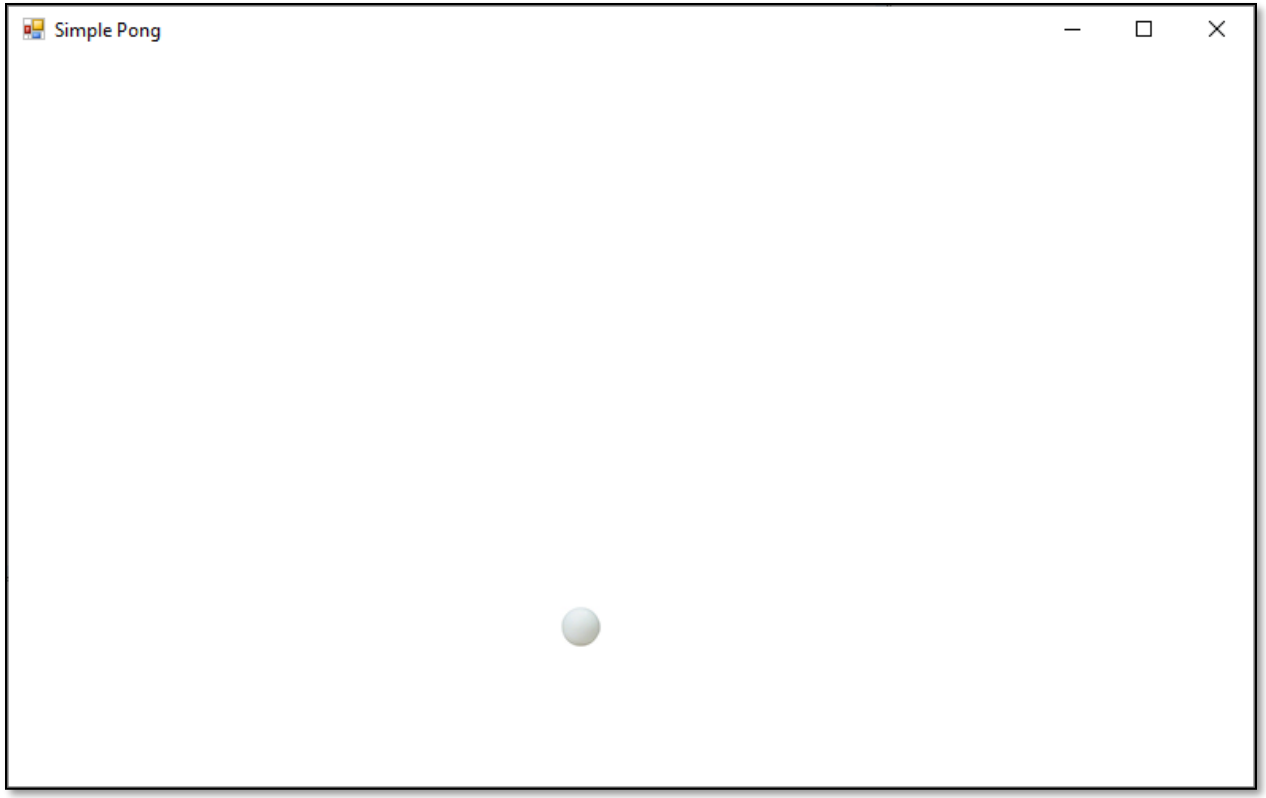
https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.application.doevents?view=netframework-4.7.2

2. Add the following methods to your code.

```
// Method with boolean (Created) that moves ball while this object/class is running (true)
public void GameLoop()
{
    // Main game loop, executes every 10 ms when program starts
    // An infinite "Game Loop" creates constant movement of the ball
    // "this" refers to this current Class (form)
    // "Created" refers to a boolean, which while tests for TRUE if "this" is running
    while (this.Created)
    {
        moveBall();                 // Move the ball
        Refresh();                  // Repaint the form
        // Processes all events in the Thread queue so the program animation doesn't stop during refresh
        Application.DoEvents();
        // Pause the foreground or program thread for 10 ms
        System.Threading.Thread.Sleep(TIME_DELAY);
    }
}

// Logic to move the ball on the playing field
private void moveBall()
{
    Ball.Left = Ball.Left + MoveX;  // Move current Ball Left position by adding X integer
    Ball.Top = Ball.Top + MoveY;    // Move current Ball Top position by adding Y integer
}
```

3. Press F5 to debug and run the program. The ping pong ball should move diagonally from the left to the right and run right off the program edge. We'll probably want to fix that in the next version of this program.


Simple Pong