Simple Pong 2: Bouncing Ball

Contents

Simple Pong 2: Bouncing Ball 1

Simple Pong Project Sequence 1

Program Description 2

MoveBall() Method....................2

Time required: 15 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

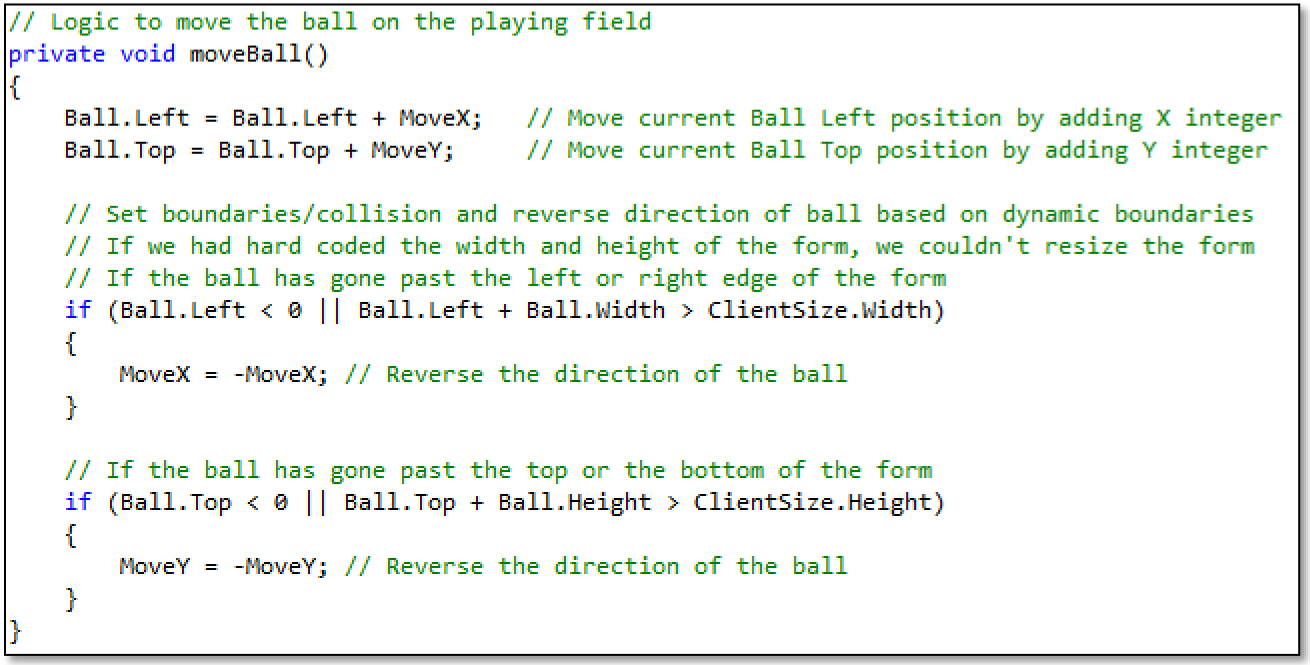
Program Description

Moving the ball across the screen is all well and good. To simulate a ping pong game, we want to simulate the physics of bouncing the ball against the side of the form. We need to check each time we go through the game loop to see if we have bumped up against the edge of the form and take appropriate action. This will simulate bouncing against the edge of the form.

The **.Height** and **.Width** property give us the height and width of our **PictureBox** and the form. **ClientSize** is the size of the form. We use this information to determine if the ball has moved outside the form area and take appropriate action. Putting a - (minus sign) in front of the variable changes it from positive to negative, or negative to positive, reversing the current direction.

MoveBall() Method

1. Change the MoveBall() method by adding the following code.



```
// 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

    // Set boundaries/collision and reverse direction of ball based on dynamic boundaries
    // If we had hard coded the width and height of the form, we couldn't resize the form
    // If the ball has gone past the left or right edge of the form
    if (Ball.Left < 0 || Ball.Left + Ball.Width > ClientSize.Width)
    {
        MoveX = -MoveX; // Reverse the direction of the ball
    }

    // If the ball has gone past the top or the bottom of the form
    if (Ball.Top < 0 || Ball.Top + Ball.Height > ClientSize.Height)
    {
        MoveY = -MoveY; // Reverse the direction of the ball
    }
}
```

2. Press F5 to debug and run the program. The ping pong ball should bounce around the form.

We have our ball bouncing. That just isn’t much fun yet, but we will work on that in the next tutorial.