Simple Pong 3: Keyboard Input

Time required: 30 minutes

Contents

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

The next step to a working game is to get input from the player by using the keyboard. We will also create two ping pong paddles using **PictureBox** controls.

Modify the Form

1. Add a **PictureBox** control with the following properties.

    a) **Name:** Player

    b) **BackColor:** Yellow

    c) **Location:** 10, 142

    d) **Size:** 27, 127

2. Add another **PictureBox** control with the following properties.

    a) **Name:** Computer

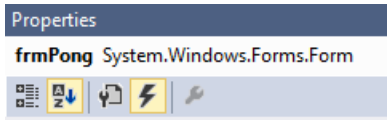
    b) **BackColor:** Fuchsia

    c) **Location:** 750,142

    d) **Size:** 27, 127

Add Event Handlers

To capture input from the keyboard, we will capture **KeyDown** and **KeyUp** events with two event handlers. One will capture the up arrow, one will capture the down arrow.



1. Click the background of the Form.
2. Under the Form Properties, Click the Lightning bolt.
3. To the right of KeyDown, type: **KeyIsDown**. Press Enter to create the event handler. This will take you to the code view.
4. Go back to the form Design view, and the form properties.
5. To the right of KeyUp, type: **KeyIsUp**. Press Enter to create the event handler. The will take you to the code view.

6. Type in the following code in the new event handlers.

```
// Logic for pressing a key
private void KeyIsDown(object sender, KeyEventArgs e)
{
    // If player presses up arrow key, change GoUp to true
    if (e.KeyCode == Keys.Up)
    {
        GoUp = true;
    }
    // If player presses down arrow key, change GoDown to true
    if (e.KeyCode == Keys.Down)
    {
        GoDown = true;
    }
}

// Logic for releasing a key
private void KeyIsUp(object sender, KeyEventArgs e)
{
    // If player releases up arrow key, change GoUp to false
    if (e.KeyCode == Keys.Up)
    {
        GoUp = false;
    }
    // If player releases down arrow key, change GoDown to false
    if (e.KeyCode == Keys.Down)
    {
        GoDown = false;
    }
}
```

7. At the form level, add the following Boolean variables to the ones already there.

```
public partial class Form1 : Form
{
    bool GoUp;      // Boolean used to store keyboard up arrow status
    bool GoDown;    // Boolean used to store keyboard down arrow status
    const int PLAYER_PADDLE_SPEED = 8;  // Set the Player paddle speed in pixels
    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
```

8. Create the **MovePaddle** method.

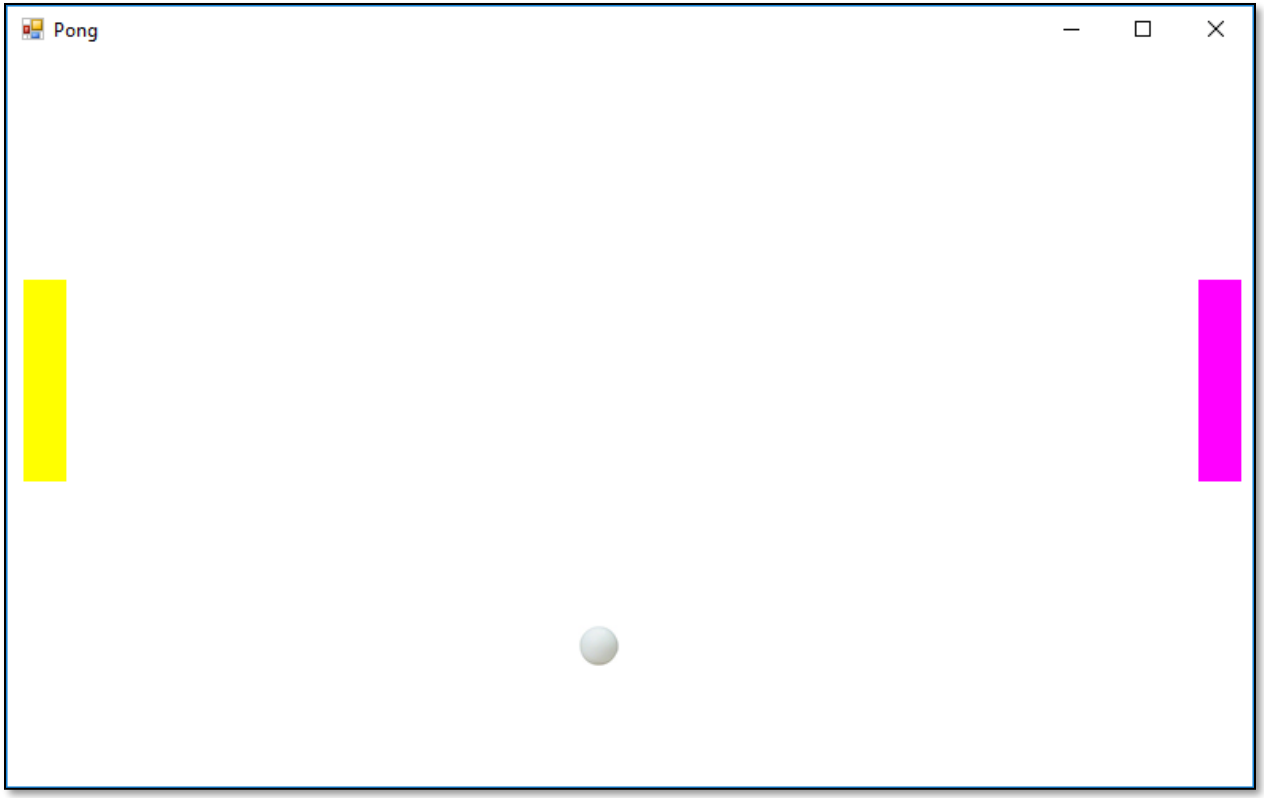
```
// Logic to control the paddle
private void movePaddle()
{
    // If the player has pressed the up arrow and is not at the top of the form
    if (GoUp == true && Player.Top > 0)
    {
        Player.Top -= PLAYER_PADDLE_SPEED;    // Move player paddle up
    }
    // If the player has pressed the down arrow and is not at the bottom of the form
    if (GoDown == true && Player.Bottom < ClientSize.Height)
    {
        Player.Top += PLAYER_PADDLE_SPEED;    // Move player paddle down
    }
}
```

9. Add the **movePaddle** method to the **GameLoop**.

```
// 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
        movePaddle();                       // Move the paddles
        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);
    }
}
```

10. Press F5 to debug and run the program.

In addition to the ping pong ball moving, the player ping pong paddle will move based on keyboard input. However, the ping pong ball and the paddles do not interact with the ball. That happens in the next tutorial.


Pong