Simple Pong 4: Collision Detection

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

Time to get our paddles to interact with the ping pong ball. It is not much of a game if we just sit and watch.

The Code

We will have the computer automatically move the paddle up and down at a set rate.

1. Add the following form level variable to the existing ones. This allows us to later control the computer paddle independently from the player paddle.

```
public partial class Form1 : Form
{
    const int DISTANCE_FROM_EDGE = 10;  // Distance of computer paddle from edge of playing field
    int ComputerPaddleSpeed = 5;        // Set the Computer paddle speed in pixels
    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
```

2. Add the DetectCollision method. This method checks the ball Bounds property to see if it intersects with the paddles Bounds properties. If it does, reverse direction.

```
// If either paddle collides with the ball, reverse the direction of the ball
private void detectCollision()
{
    // If ball hits the player or computer paddle
    if (Ball.Bounds.IntersectsWith(Player.Bounds) || Ball.Bounds.IntersectsWith(Computer.Bounds))
    {
        MoveX = -MoveX; // Reverse direction
    }
}
```

3. Add the following line to the MovePaddle method to move the computer paddle.

```
// Logic to control the paddles
private void movePaddle()
{
    // Move the computer paddle left or right if the screen is resized
    Computer.Left = ClientSize.Width - (Computer.Width + DISTANCE_FROM_EDGE);
    // Move the computer paddle up and down
    Computer.Top += ComputerPaddleSpeed;
    // If the computer paddle reached top or bottom of screen
    if (Computer.Top < 0 || Computer.Bottom > ClientSize.Height)
    {
        ComputerPaddleSpeed = -ComputerPaddleSpeed; // Change direction of computer paddle
    }
    // 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
    }
}
```

4. Add the **DetectCollision** method call to the **GameLoop** method.

```
// 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
        detectCollision();                  // Detect a collision
        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);
    }
}
```

5. Press F5 to debug and run the program.

Both paddles move, and interact with the ping pong ball. However, nothing happens if the paddles miss the ball. The ball just keeps bouncing. Coming up next!