



PONG – Add scoring

The game is fun but it is not easy to keep track of who is winning. The game should be able to do this for you. This is quite a big change. Firstly, the game needs to keep track of the score. I decided that the “Ball” object should keep track of the score because it knows when it leaves the court area. First, add these two lines before the first **class** declaration:

```
Player1Score = 0          # initialize player 1 score
Player2Score = 0          # initialize player 2 score

# Our main game class
class PiPong:
```

These two new variables are global because they are not declared within a class. Now change the **Ball “update”** function to increment the scores when the ball goes out of play. Note that Python insists that when a global variable is assigned to it, it must be declared as such.

```
elif self.rect.midleft[0] <= 0:
    # Hit left side, i.e. right side scores
    global Player2Score
    self.reset()
    Player2Score = Player2Score + 1
elif self.rect.midright[0] >= self.displaySize[0]:
    # Hit right side, i.e. left side scores
    global Player1Score
    self.reset()
    Player1Score = Player1Score + 1
```

There is enough code here to keep track of the scores.

P.T.O...



And now, the tricky bit, we need to display the scores. This is all carried out within the Background class. Add or modify the following code at the end of the **Background __Init__** function:

```
self.myfont = pygame.font.Font(None,60)
self.displayx = displaySize[0]

def draw(self, display):

    # Draw the background to the display that has been passed in
    display.blit(self.image, (0,0))
    # Scores on top
    scoretext = self.myfont.render(str(Player1Score), 1, (255,255,255))
    display.blit(scoretext, (self.displayx / 4 - 20, 20))
    scoretext = self.myfont.render(str(Player2Score), 1, (255,255,255))
    display.blit(scoretext, (self.displayx * 3 / 4 - 20, 20))
```

Info, to help explain some of the preceding code:

- **BLIT, BLock Image Transfer** - a computer graphics operation in which two bitmap patterns are combined.
- **Font** - is generally used to refer to a scalable set of digital shapes that may be printed at many different sizes.
- **Render** - To transform digital information into a display on a computer screen, or for other presentation to the user.

Run the program and confirm that the scoring is working correctly.