TTT Version 1 Reflection Activity

Q1 Answer the following questions about the given draw method in the Tile class of TTT Version 1:

Given Code

```
def draw(self):
      # Draw the Tile and its content
      # V1 draws the top left coordinates of each tile in the grid
      # - self is the Tile
      text_font = pygame.font.SysFont('', 24)
      text_str = str(self.rect.x) + ',' + str(self.rect.y)
      text_image = text_font.render(text_str, True, Tile.fg_color)
      location = (self.rect.x,self.rect.y)
      self.surface.blit(text_image,location )

      pygame.draw.rect(self.surface, Tile.fg_color, self.rect, self.border_width)
```

1a	**Which method in the Game class calls the draw method in the Tile class?**
	the draw method in the Game class is calling the draw method in the Tile class
1b	**Which argument in the function call pygame.draw.rect is optional?**
	self.border_width
1c	**Describe the rectangle that would be drawn for the following three cases :** **a. self.border_width > 0** **b. self..border_width = 0** **c. self..border_width < 0**
	a. color is used to fill in the border width which is specified in the 4th argument b. color is used to fill in the entire rectangle c. rectangle is not drawn

Q2 Trace the given code and record all changes to the values of the object that each identifier:i,j,number, row and matrix are bound to. Write the output of the given code in the output box.

Given Code

```
matrix = []
for i in range(0,3):
    row = []
    for j in range(0,3):
        number = i+j
        row.append(number)
    matrix.append(row)
print(matrix)
```

i	j	number	row	matrix
0	0	0	[]	[]
	1	1	[0]	[]
	2	2	[0,1]	[]
			[0,1,2]	[[0,1,2]]
1	0	1	[]	
	1	2	[1]	
	2	3	[1,2]	
			[1,2,3]	[[0,1,2],[1,2,3]]

Output	

Q3 For each method that an identifier in the given code segment (taken from the Game class), is bound to, identify the following:

- **Is the method user defined, built-in or imported?**
- **Specify the class the method is defined in.**

Given Code

```
class Game:
...

   def create_board(self):
      # Create the game board.
      # - self is the Game whose board is created
      width = self.surface.get_width() // self.board_size
      height = self.surface.get_height() // self.board_size
      for row_index in range(0, self.board_size):
         row = self.create_row(row_index, width, height)
         self.board.append(row)
```

	Method Name	Built-in or User-defined or Imported	Class Name
1.	create_board	user-defined	Game
2.	get_width	imported	pygame.Surface
3.	get_height	imported	pygame.Surface
4.	create_row	user-defined	Game
5.	append	built-in	list