

Informe de Laboratorio 05

Tema: Python

Nota

Estudiantes	Escuela	Asignatura
William Herderson Choquehuanca Berna wchoquehuancab@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Laboratorio de P.Web Semestre: III Código: 20233469

Laboratorio	Tema	Duración
05	Python	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 20 de mayo 2024	Al 24 de mayo 2024

1. Actividades

- 1. Instale Python!
- 2. Cree su entorno de trabajo:

```
mkdir lab04
cd lab04
virtualenv -p python3 .
mkdir src
cd src
git init .
```
- 3. Active el entorno virtual

```
source ../bin/activate
```
- 4. Ahora podrá instalar las bibliotecas que necesite usando pip

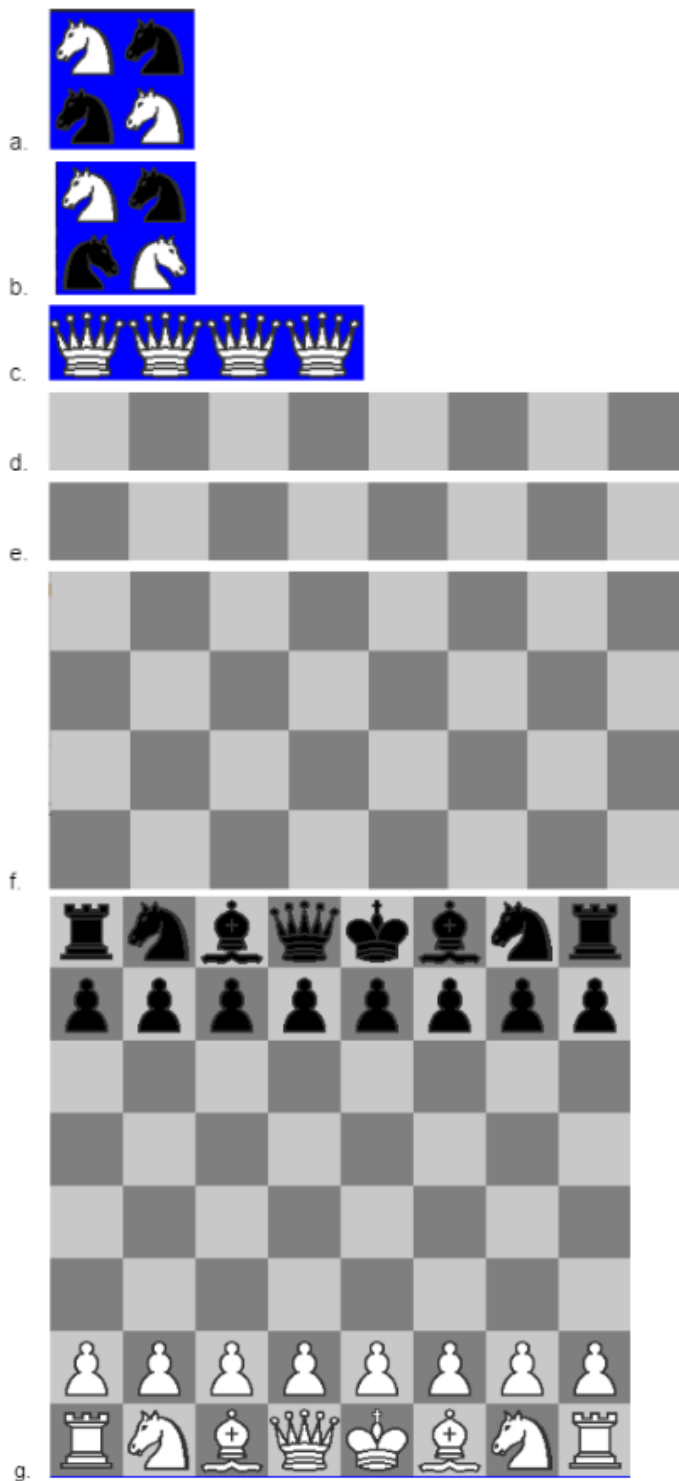
```
pip install pygame
```
- 5. Cuando quiera terminar de usar el entorno virtual ejecute

```
deactivate
```

2. Ejercicios Propuestos

- Implemente los métodos de la clase Picture. Se recomienda que implemente la clase picture por etapas, probando realizar los dibujos que se muestran en la siguiente preguntas. Con el código proporcionado usted dispondrá de varios objetos de tipo Picture para poder realizar su tarea

- Usando únicamente los métodos de los objetos de la clase Picture dibuje las siguientes figuras (invoque a draw)



3. Equipos, materiales y temas utilizados

- Sistema operativo de 64 bits, procesador basado en x64.
- Latex.
- git version 2.41.0.windows.1
- Lenguaje Python.
- IDE Visual Studio Code.

4. URL Github, Video

- URL del Repositorio GitHub.
- <https://github.com/WilliamLawrence25/PWeb2/tree/main/Lab5>
- URL para el video flipgrid.
- <https://flip.com/s/QRG7DhggB2kt>

5. Capturas de los ejercicios propuestos

5.1. Ejercicio 1

```

1 from colors import *
2
3 class Picture:
4
5     def __init__(self, img):
6         self.img = img;
7
8     def __eq__(self, other):
9         return self.img == other.img
10
11     def _invColor(self, color):
12         if color not in inverter:
13             return color
14         return inverter[color]
15
16     def verticalMirror(self):
17         """ Devuelve el espejo vertical de la imagen """
18         vertical = []
19         vertical = [line[::-1] for line in self.img]
20         return Picture(vertical)
21
22     def horizontalMirror(self):
23         """ Devuelve el espejo horizontal de la imagen """
24         horizontal = self.img[::-1]
25         return Picture(horizontal)
26
27     def negative(self):
28         """ Devuelve un negativo de la imagen """
29         negative = [''.join(self._invColor(char) for char in line) for line in self.img]
30         return Picture(negative)
31
32     def join(self, p):
33         """ Devuelve una nueva figura poniendo la figura del argumento
34         || al lado derecho de la figura actual """
35         join_img = [self_line + p_line for self_line, p_line in zip(self.img, p.img)]
36         return Picture(join_img)
37
38     def up(self, p):
39         return Picture(self.img + p.img)
40
41     def under(self, p):
42         """ Devuelve una nueva figura poniendo la figura p sobre la
43         || figura actual """
44         max_height = max(len(self.img), len(p.img))
45         max_width = max(max(len(line) for line in self.img), max(len(line) for line in p.img))
46
47         new_img_self = [line.ljust(max_width) for line in self.img] + [' ' * max_width] * (max_height - len(self.img))
48         new_img_p = [line.ljust(max_width) for line in p.img] + [' ' * max_width] * (max_height - len(p.img))
49
50         combined_img = []
51         for line_self, line_p in zip(new_img_self, new_img_p):
52             combined_line = ''.join(char_p if char_p != ' ' else char_self for char_self, char_p in zip(line_self, line_p))
53             combined_img.append(combined_line)
54
55         return Picture(combined_img)
56
57     def horizontalRepeat(self, n):
58         """ Devuelve una nueva figura repitiendo la figura actual al costado
59         || la cantidad de veces que indique el valor de n """
60         repeat_img = [line * n for line in self.img]
61         return Picture(repeat_img)
62
63     def verticalRepeat(self, n):
64         repeat_img = self.img * n
65         return Picture(repeat_img)
66
67     #Extra: Sólo para realmente viciados
68     def rotate(self):
69         """Devuelve una figura rotada en 90 grados, puede ser en sentido horario
70         o antihorario"""
71         rotated = [''.join(row) for row in zip(*self.img)]
72         rotated = rotated[::-1]
73         return Picture(rotated)
74

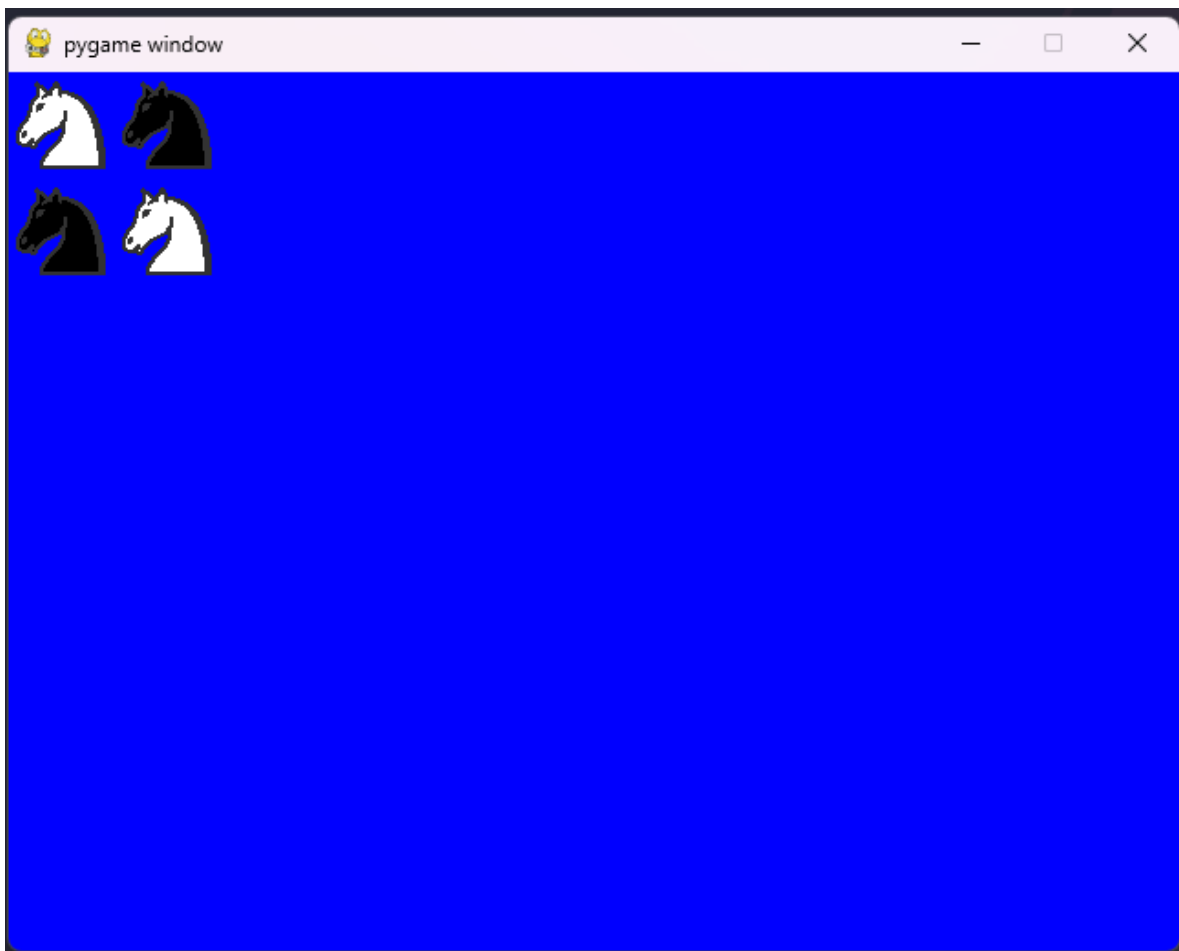
```

5.2. Ejercicio 2a

- Python

```
1 from interpreter import draw
2 from chessPictures import *
3
4 fila1 = knight.join(knight.negative())
5 fila2 = fila1.negative()
6 todo = fila1.up(fila2)
7 draw (todo)
```

- Tablero

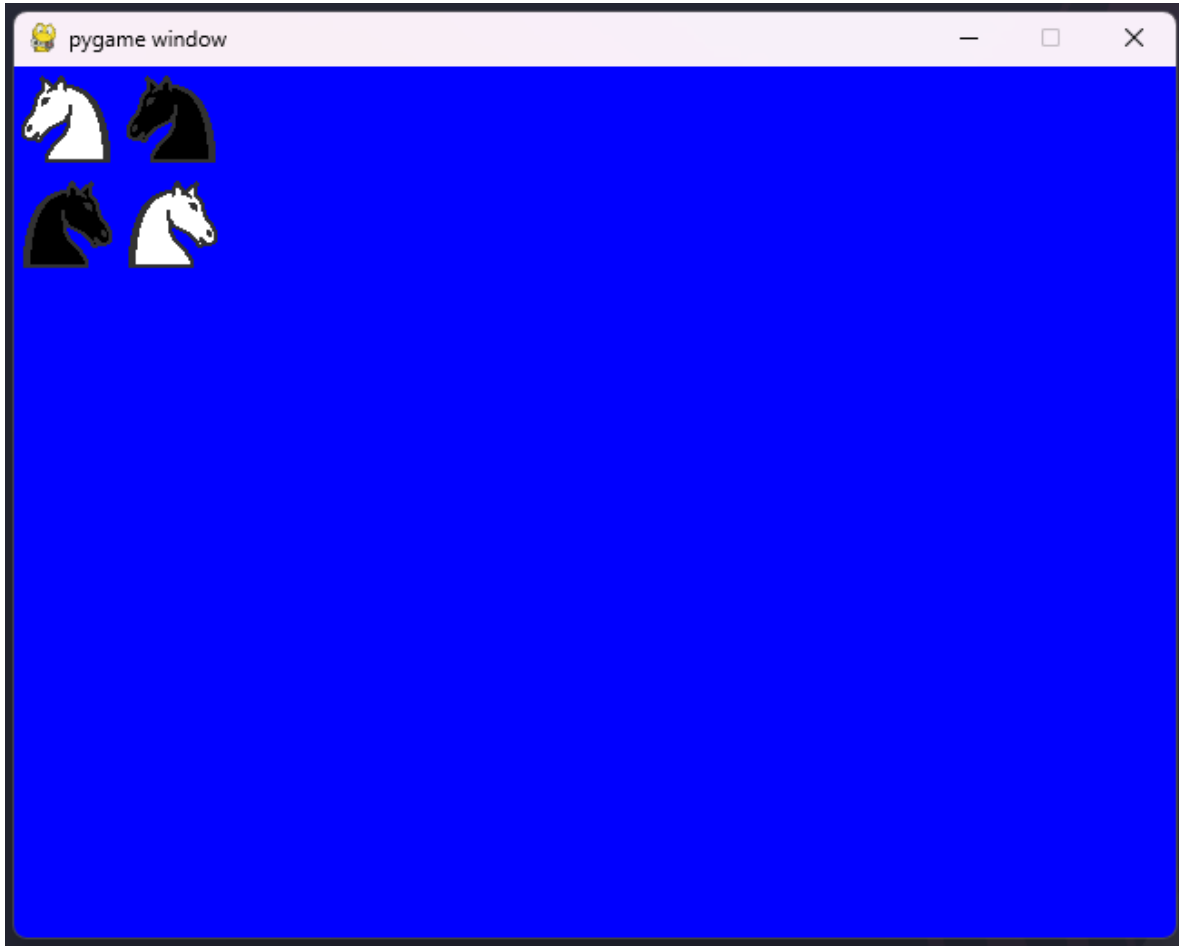


5.3. Ejercicio 2b

- Python

```
1 from interpreter import draw
2 from chessPictures import *
3
4 fila1 = knight.join(knight.negative())
5 fila2 = (fila1).verticalMirror()
6 todo = fila1.up(fila2)
7 draw (todo)
```

- Tablero

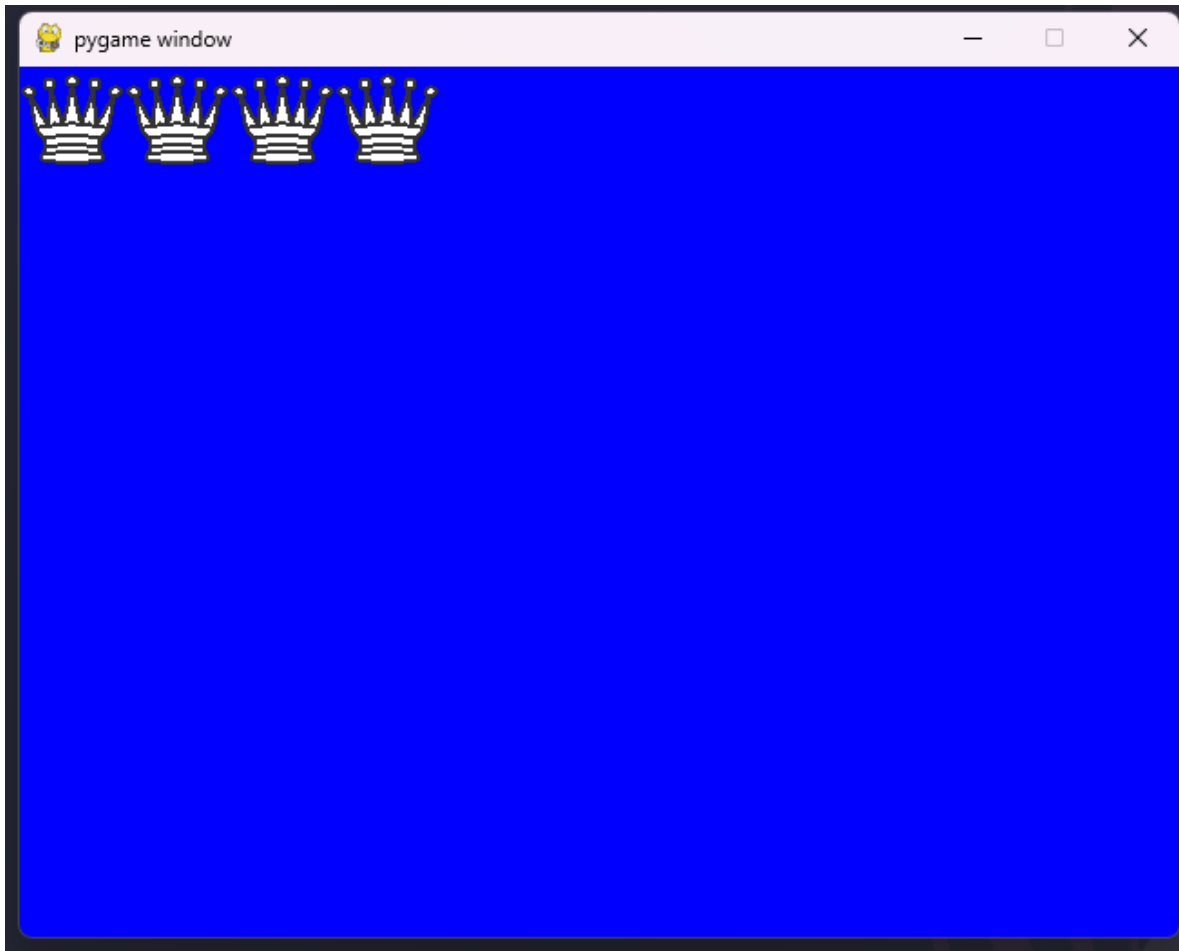


5.4. Ejercicio 2c

- Python

```
1 from interpreter import draw
2 from chessPictures import *
3
4 draw (queen.horizontalRepeat(4))
```

- Tablero

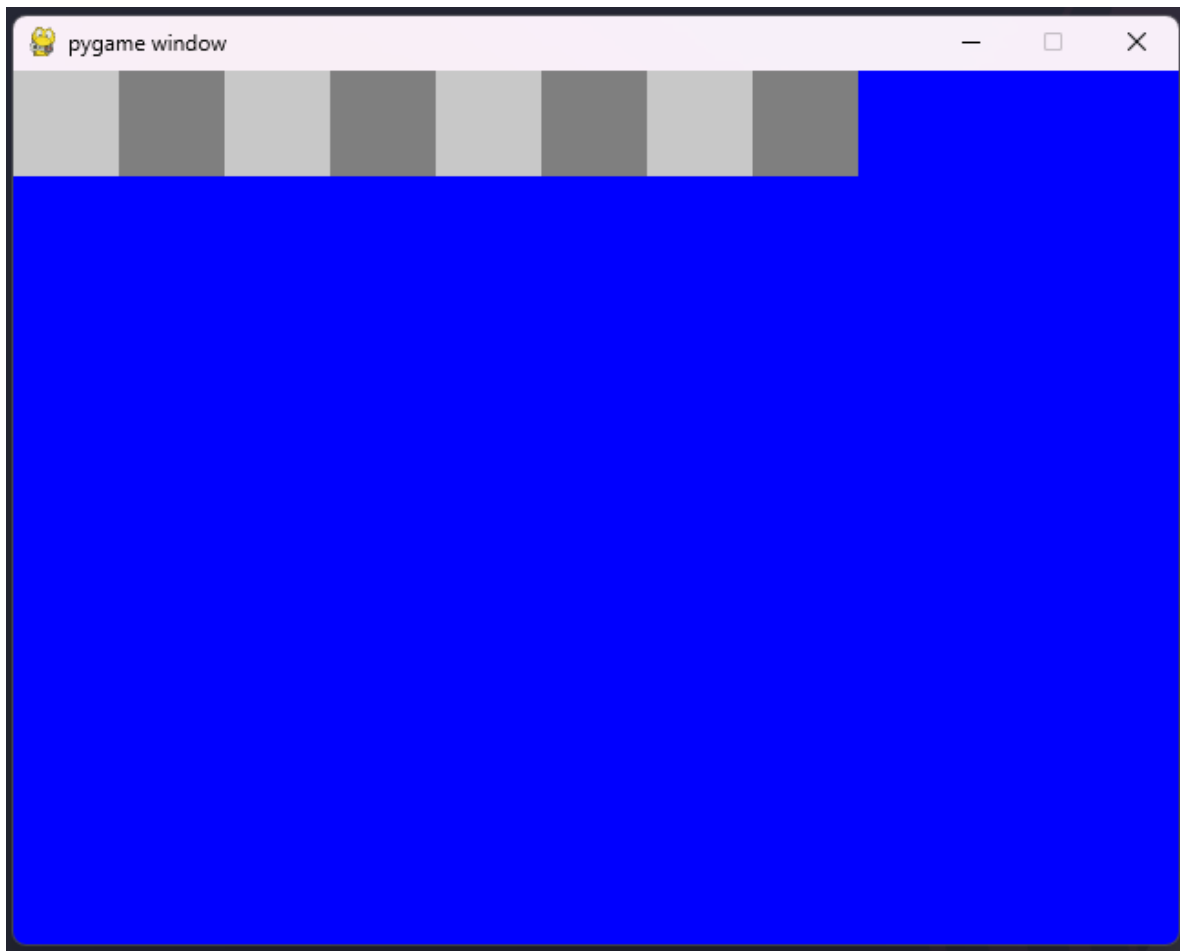


5.5. Ejercicio 2d

- Python

```
1 from interpreter import draw
2 from chessPictures import *
3 |
4 draw (square.join(square.negative()).horizontalRepeat(4))
```

- Tablero

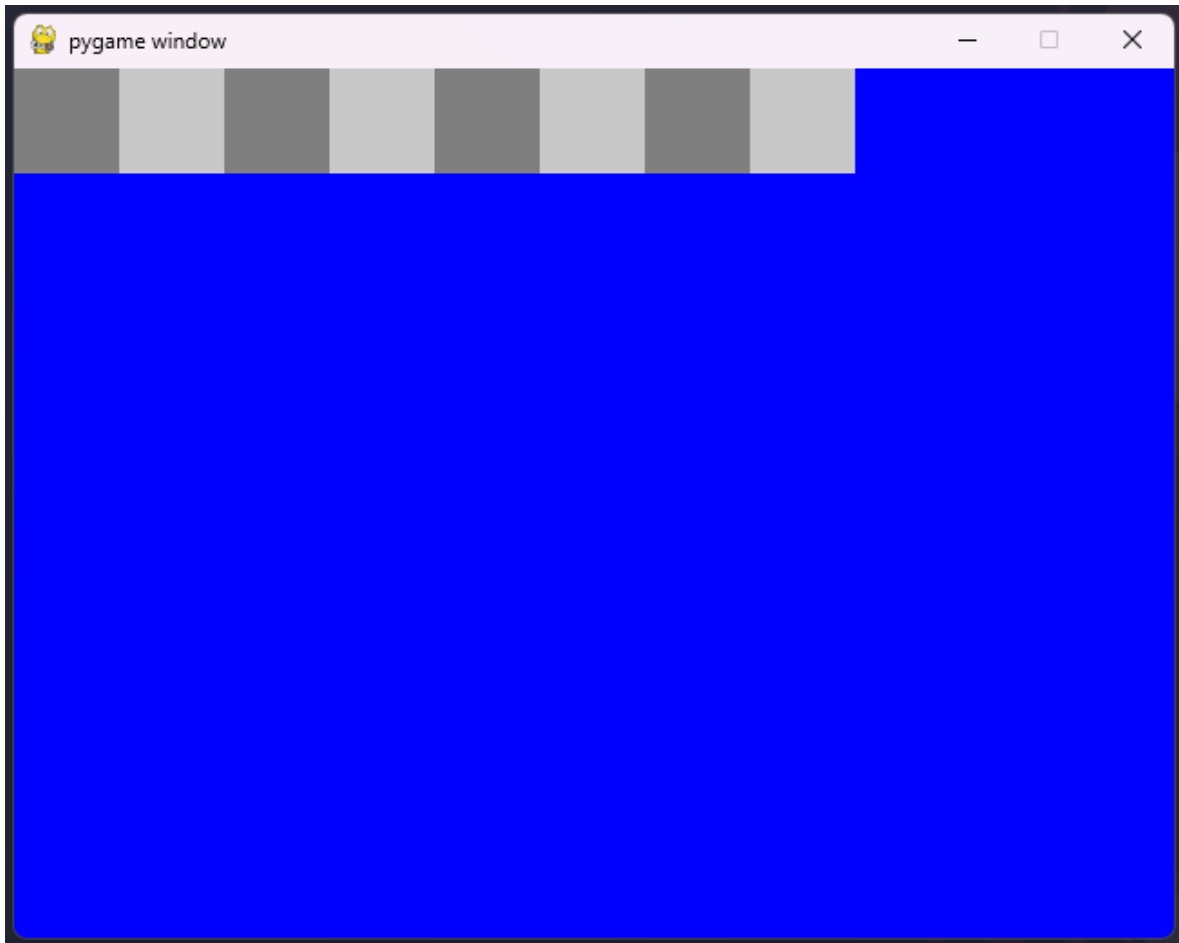


5.6. Ejercicio 2e

- Python

```
1 from interpreter import draw
2 from chessPictures import *
3
4 draw (square.negative().join(square).horizontalRepeat(4))
```

- Tablero

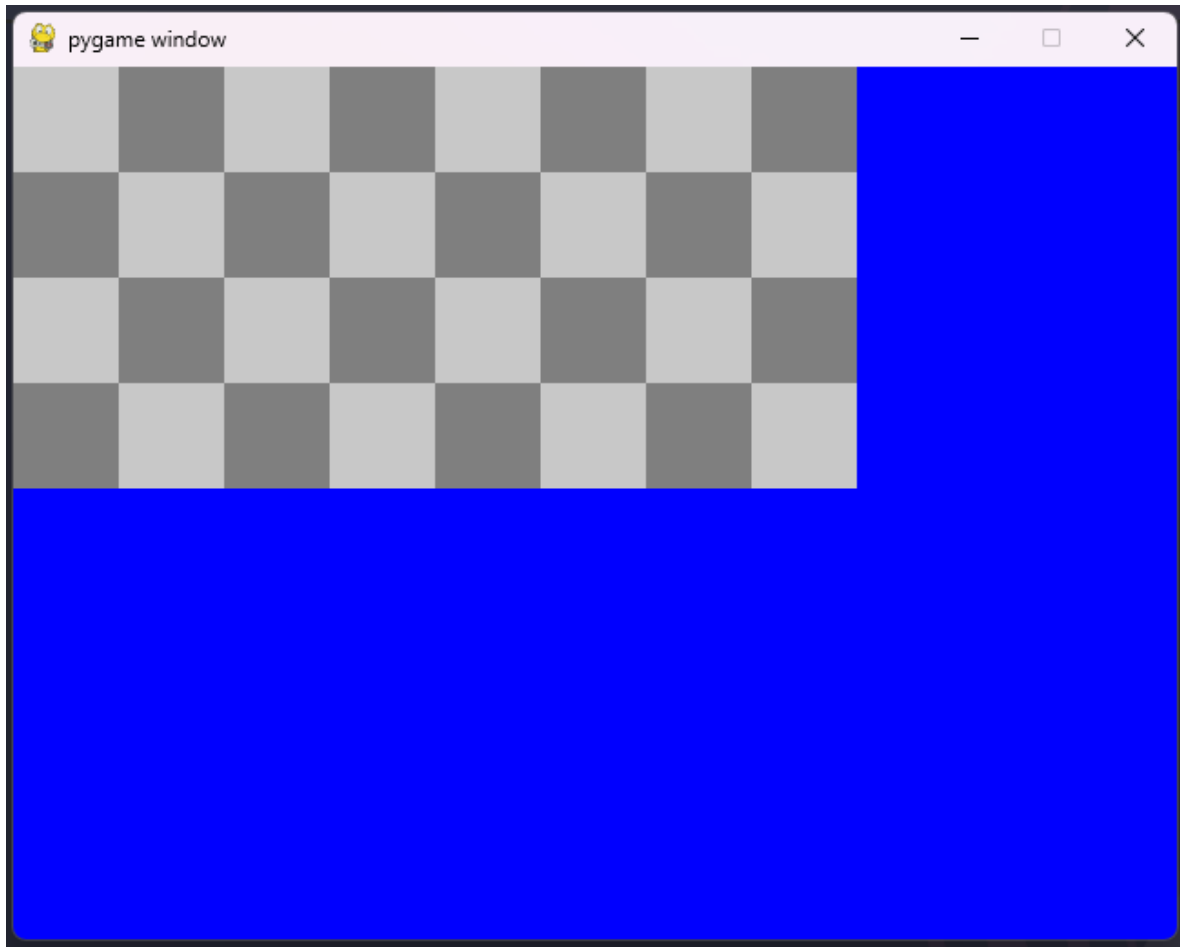


5.7. Ejercicio 2f

- Python

```
1 from interpreter import draw
2 from chessPictures import *
3
4 type1 = (square.join(square.negative()).horizontalRepeat(4))
5 type2 = (square.negative().join(square).horizontalRepeat(4))
6 total = type1.up(type2).verticalRepeat(2)
7
8 draw (total)
```

- Tablero



5.8. Ejercicio 2g

■ Python

```

1  from interpreter import draw
2  from chessPictures import *
3
4  type1 = (square.join(square.negative()).horizontalRepeat(4))
5  type2 = (square.negative().join(square).horizontalRepeat(4))
6
7  chess1s = type2.under(rock.join(knight).join(bishop).join(queen).join(king).join(bishop).join(knight).join
8  (rock)).negative()
9  chess2s = type1.under(pawn.horizontalRepeat(8)).negative()
10 table = chess1s.up(chess2s.up(type1.up(type2).verticalRepeat(2))).up(chess2s.negative()).up(chess1s.negative
11 ())
12 draw (table)

```

■ Tablero



6. Referencias

- <https://github.com/rescobedoq/pw2/tree/main/labs/lab04/Tarea-del-Ajedrez>