INFORME DE LABORATORIO

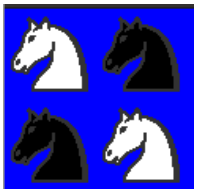
INFORMACIÓN BÁSICA					
ASIGNATURA:	**Programacion Web 2**				
TÍTULO DE LA PRÁCTICA:	***Laboratorio 05***				
NÚMERO DE PRÁCTICA:	***5***	**AÑO LECTIVO:**	***2025***	**NRO. SEMESTRE:**	***1***
FECHA DE PRESENTACIÓN	***13/05/2024***	**Repositorio**	*https://github.com/JesusFSP/Curso-PWeb2.git*		
INTEGRANTE (s): **Silva Pino Jesus Francisco**				**NOTA:**	
DOCENTE(s): **CARLO JOSE LUIS CORRALES DELGADO**					

SOLUCIÓN Y RESULTADOS

I. SOLUCIÓN DE EJERCICIOS/PROBLEMAS

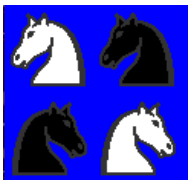
Implemente los métodos de la clase Picture. Se recomienda que implemente la clase picture por etapas, probando realizar los dibujos que se muestran en la siguiente preguntas.

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

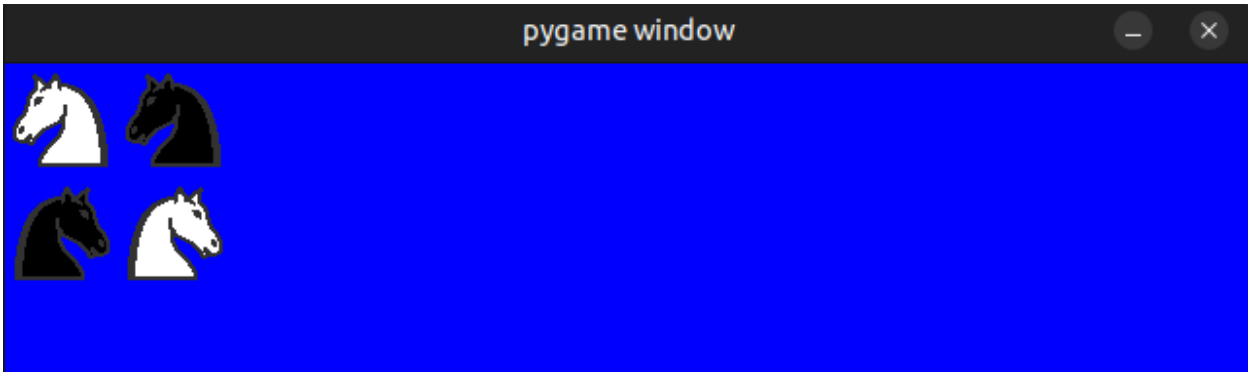
1. 

```
from chessPictures import *
from interpreter import draw


fig_a = (knight.negative().join(knight)).up(knight.join(knight.negative()))
```

draw(fig_a)

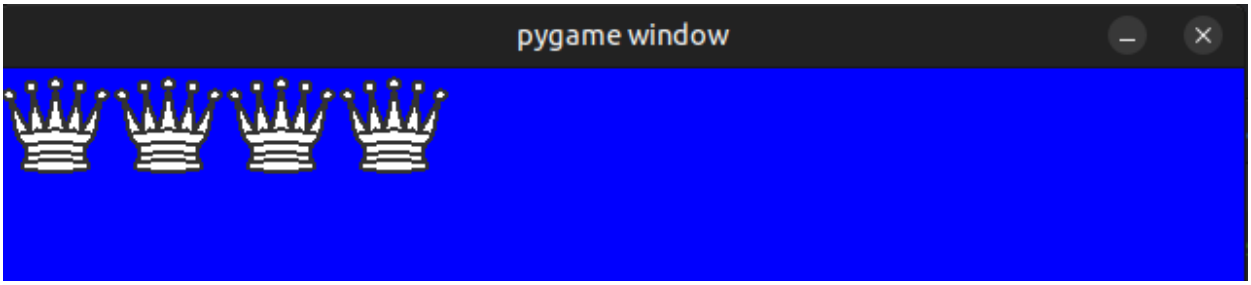
2. 

```
from chessPictures import *
from interpreter import draw

fig_b = ((knight.join(knight.negative())).verticalMirror()).up(knight.join(knight.negative()))
draw(fig_b)
```



3. 

```
from chessPictures import *
from interpreter import draw

fig_c = queen.horizontalRepeat(4)
draw(fig_c)
```

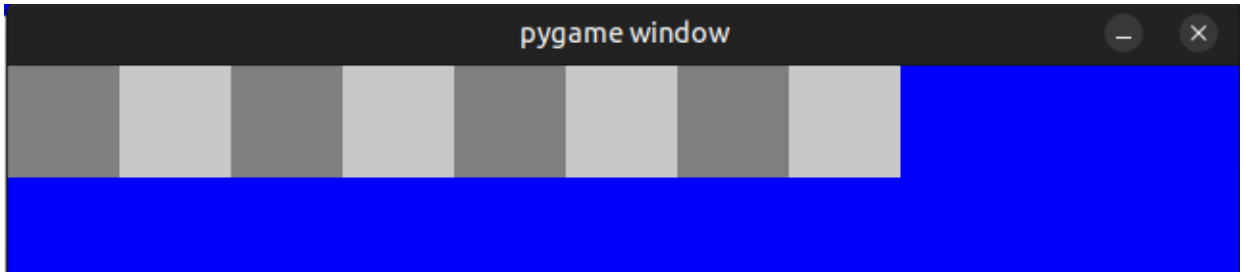


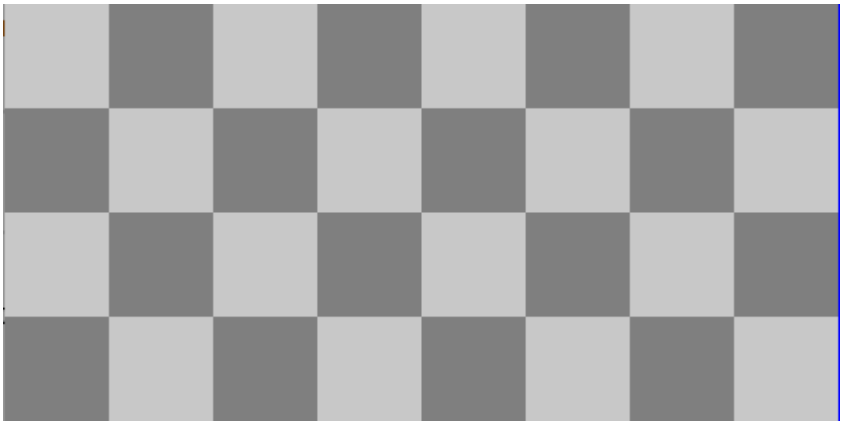
4. 

```
from chessPictures import *
from interpreter import draw

fig_d = square.join(square.negative()).horizontalRepeat(4)
draw(fig_d)
```

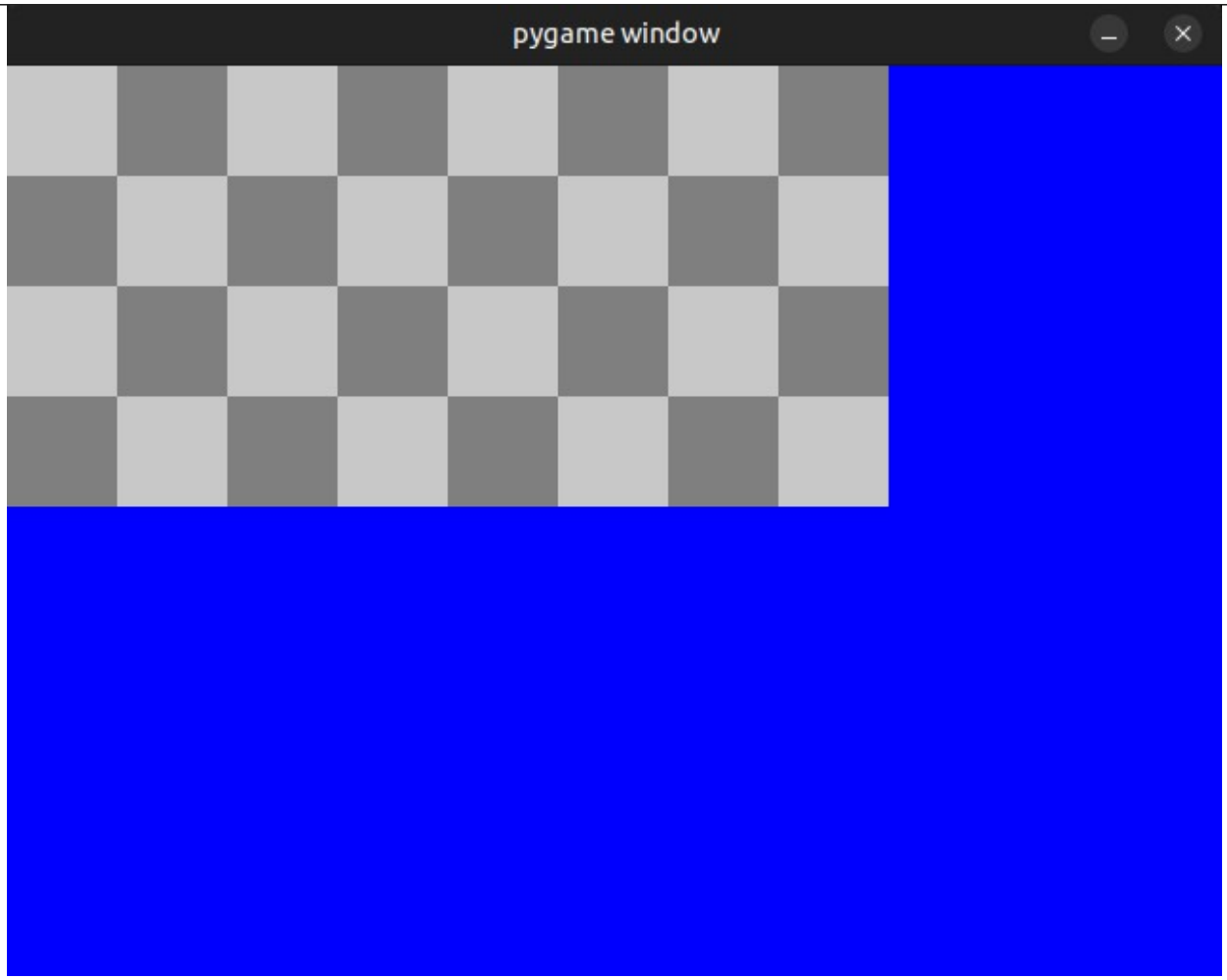
5. 

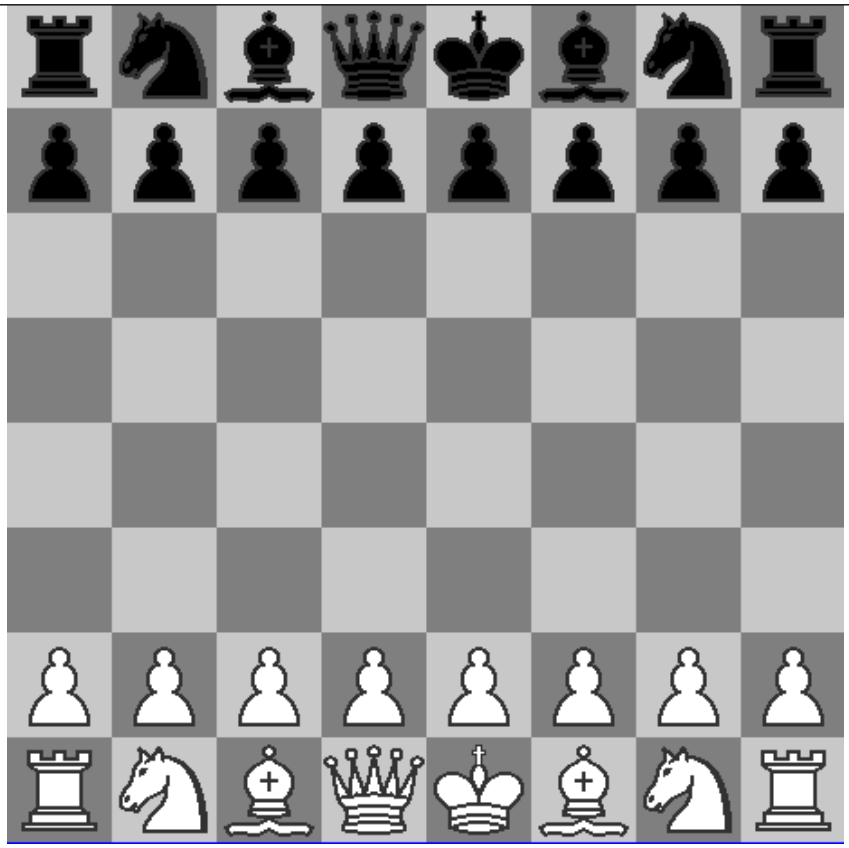
```
from chessPictures import *
from interpreter import draw

fig_e = square.negative().join(square).horizontalRepeat(4)
draw(fig_e)
```



6. 

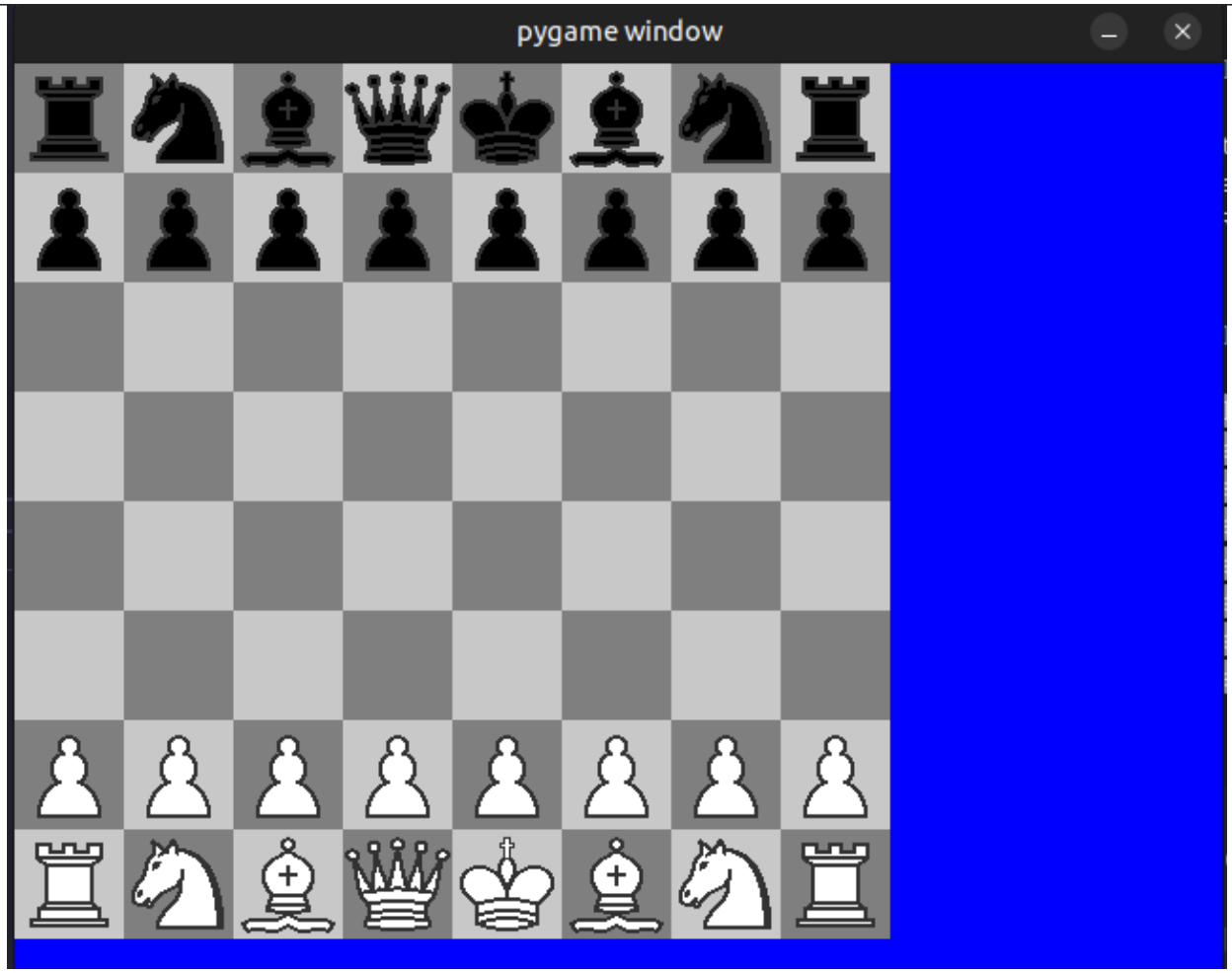
```
from chessPictures import *
from interpreter import draw

base = square.negative().join(square).up(square.join(square.negative()))
fig_f = base.horizontalRepeat(4).verticalRepeat(2)
draw(fig_f)
```


pygame window

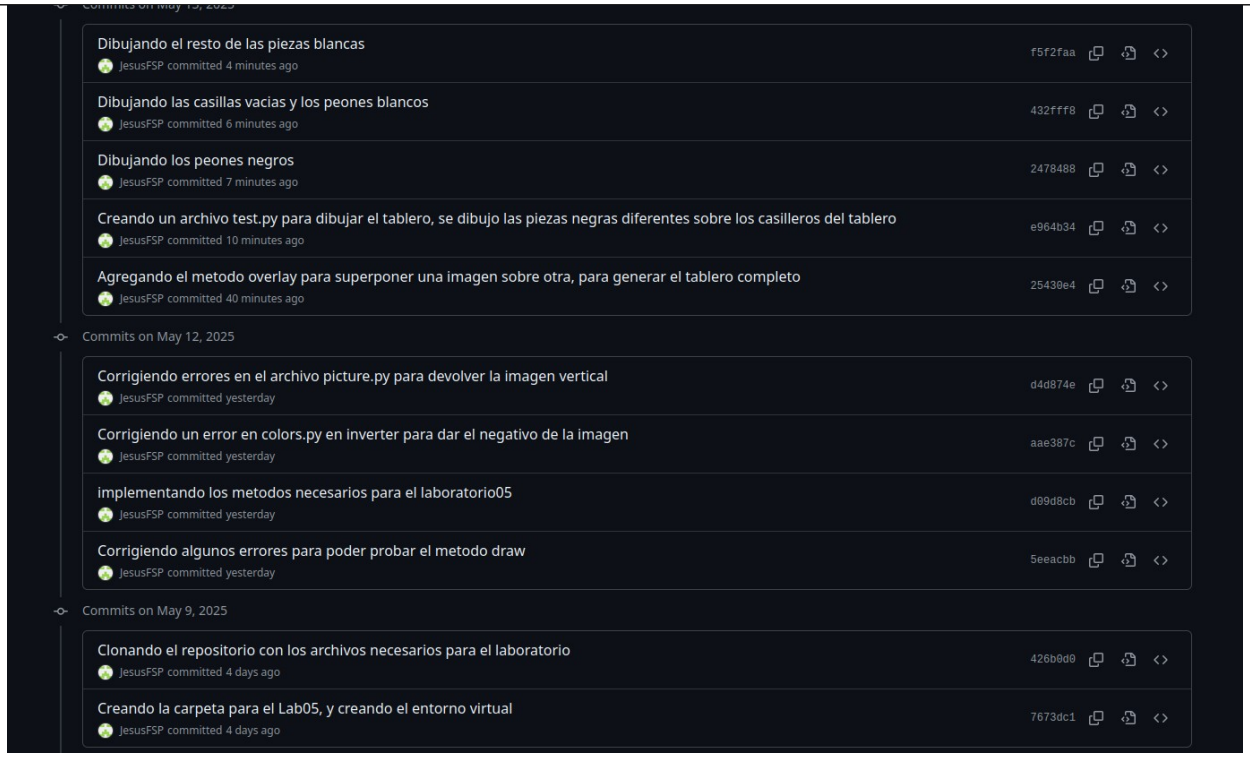
7. 

```
from chessPictures import *
from interpreter import draw

draw(
    # piezas negras superpuestos en los casilleros
    square.negative().overlay(rock.negative())
    .join(square.overlay(knight.negative()))
    .join(square.negative().overlay(bishop.negative()))
    .join(square.overlay(queen.negative()))
    .join(square.negative().overlay(king.negative()))
    .join(square.overlay(bishop.negative()))
    .join(square.negative().overlay(knight.negative()))
    .join(square.overlay(rock.negative()))
```

```
        .under(
            square.overlay(pawn.negative())
            .join(square.negative().overlay(pawn.negative()))
            .horizontalRepeat(4)
        )
        .under(
            square.join(square.negative()).horizontalRepeat(4)
            .up(square.negative().join(square).horizontalRepeat(4))
            .verticalRepeat(2)
        )
        .under(
            square.negative().overlay(pawn)
            .join(square.overlay(pawn))
            .horizontalRepeat(4)
        )
        .under(
            square.overlay(rock)
            .join(square.negative().overlay(knight))
            .join(square.overlay(bishop))
            .join(square.negative().overlay(queen))
            .join(square.overlay(king))
            .join(square.negative().overlay(bishop))
            .join(square.overlay(knight))
            .join(square.negative().overlay(rock))
        )
    )
```



Historial de commits



II. SOLUCIÓN DEL CUESTIONARIO

III. CONCLUSIONES

RETROALIMENTACIÓN GENERAL

REFERENCIAS Y BIBLIOGRAFÍA