



UNIVERSIDAD NACIONAL DE SAN AGUSTIN
FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y
SERVICIOS ESCUELA PROFESIONAL DE
INGENIERÍA DE SISTEMA-



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 1

INFORME DE LABORATORIO

(formato estudiante)

INFORMACION BASICA

ASIGNATURA:	Programacion web 2				
TITULO DE LA PRACTICA:	Python				
NÚMERO DE PRÁCTICA:	5	AÑO LECTIVO:	2024	NRO. SEMESTRE:	3
FECHA DE PRE-SENTACIÓN:	2024/6/1	HORA DE PRE-SENTACIÓN:	16:35		
INTEGRANTE (s): HUAMANI CONDORI JEANPIERO SIXTO			NOTA(0-20):	Nota colocada por el docente	
DOCENTE(s): LINO JOSE PINTO OPPE					

RESULTADOS Y PRUEBAS

-github Lab 5: [click aquí](#)

-codigo latex: [click aquí](#)

I. EJERCICIOS RESUELTOS:

- horizontalMirror: Devuelve el espejo horizontal de la imagen
- negative: Devuelve un negativo de la imagen
- join: Devuelve una nueva figura poniendo la figura del argumento al lado derecho de la figura actual
- up: Devuelve una nueva figura poniendo la figura recibida como argumento, encima de la figura actual
- under: Devuelve una nueva figura poniendo la figura recibida como argumento, sobre la figura actual



• **horizontalRepeat**: Devuelve una nueva figura repitiendo la figura actual al costado la cantidad de veces que indique el valor de n

• **verticalRepeat**: Devuelve una nueva figura repitiendo la figura actual debajo, la cantidad de veces que indique el valor de n

Ejercicios:

• Para resolver los siguientes ejercicios sólo está permitido usar ciclos, condicionales, definición de listas por comprensión, sublistas, map, join, (+), lambda, zip, append, pop, range.

• 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.

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

Commits:

Commits on Jun 1, 2024
ultimo ejercicio resuelto
jhuamaniCond committed 13 hours ago
Commits on May 31, 2024
ejercicio d,e,f
jhuamaniCond committed 16 hours ago
se arreglo codigo que daba errores
jhuamaniCond committed 16 hours ago
se hizo todos los metodos de la clase picture
jhuamaniCond committed 17 hours ago
se agrego las funciones horizontalMirror negative y join
jhuamaniCond committed 18 hours ago
copie estructura del lab
jhuamaniCond committed 19 hours ago

Clase Picture implementada:

```
from colors import *
class Picture:
    def __init__(self, img):
        self.img = img;

    def __eq__(self, other):
        return self.img == other.img

    def _invColor(self, color):
        if color not in inverter:
            return color
        return inverter[color]

    def verticalMirror(self):
        """Devuelve el espejo vertical de la imagen"""
        vertical = []
        for value in self.img:
            vertical.append(value[::-1])
        return Picture(vertical)

    def horizontalMirror(self):
        """Devuelve el espejo horizontal de la imagen"""
        horizontal = []
        for i in range(len(self.img) - 1, -1, -1):
            horizontal.append(self.img[i])
        return horizontal

    def negative(self):
        """Devuelve un negativo de la imagen"""
        negative = []
        for value in self.img:
            linea = ''
            for char in value:
                linea += self._invColor(char)
            negative.append(linea)
        return negative
```

```
def join(self, p):
    """Devuelve una nueva figura poniendo la figura del argumento
    al lado derecho de la figura actual"""
    try:
        imagen = p.img
    except:
        imagen = p

    joined=[]
    for index in range(len(self.img)):
        joined.append(self.img[index]+imagen[index])
    return joined

def up(self, p):
    try:
        imagen = p.img
    except:
        imagen = p

    return p.img+imagen

def under(self, p):
    try:
        imagen = p.img
    except:
        imagen = p

    return self.img+imagen

def horizontalRepeat(self, n):
    """Devuelve una nueva figura repitiendo la figura actual al costado
    la cantidad de veces que indique el valor de n"""
    repeated = self.img
    for i in range(n):
        repeated=self.join(repeated)
    return repeated

def verticalRepeat(self, n):
    repeated = self.img
    for i in range(n):
        repeated= self.under(repeated)

    return repeated
```

```
#Extra: Slo para realmente viciosos
def rotate(self):
    """Devuelve una figura rotada en 90 grados, puede ser en sentido horario
    o antihorario"""
    longitud= len(self.img[0])
    rotated=[]
    for i in range(longitud):
        line=''
        for values in range(len(self.img) - 1, -1, -1):
            line+=self.img[values][i]
        rotated.append(line)

    return rotated
```

Ejercicios implementados:

Ejercicio (a)

```
from interpreter import draw
from chessPictures import *

caballosArriba=knight.join(knight.negative())
caballosAbajo=Picture(knight.negative()).join(knight)
figura=Picture(caballosArriba).under(caballosAbajo)
draw(figura)
```

Ejercicio (b)

```
from interpreter import draw
from chessPictures import *
caballosArriba=Picture(knight.join(knight.negative()))
caballosAbajo=Picture(Picture(knight.negative()).join(knight)).verticalMirror()
figura=caballosArriba.under(caballosAbajo)
draw(figura)
```

Ejercicio (c)

```
from interpreter import draw
from chessPictures import *

reinas=queen.horizontalRepeat(3)
draw(reinas)
```

Ejercicio (d)

```
from interpreter import draw
from chessPictures import *

fila=square
for i in range(7):
    if(i%2==0):
        fila=Picture(fila.join(square.negative()))
    else:
        fila=Picture(fila.join(square))

draw(fila)
```

Ejercicio (e)

```
from interpreter import draw
from chessPictures import *

fila=Picture(square.negative())
for i in range(7):
    if(i%2==0):
        fila=Picture(fila.join(square))
    else:
        fila=Picture(fila.join(square.negative()))

draw(fila)
```

Ejercicio (f)

```
from interpreter import draw
from chessPictures import *

def generarFilaEmpiezaConNegro():
    fila=Picture(square.negative())
    for i in range(7):
        if(i%2==0):
            fila=Picture(fila.join(square))
        else:
            fila=Picture(fila.join(square.negative()))
    return fila

def generarFilaEmpiezaConBlanco():
    fila=square
    for i in range(7):
        if(i%2==0):
            fila=Picture(fila.join(square.negative()))
        else:
            fila=Picture(fila.join(square))
    return fila

tablero= generarFilaEmpiezaConBlanco()
for i in range(3):
    if(i%2==0):
        tablero=Picture(tablero.under(generarFilaEmpiezaConNegro()))
    else :
        tablero=Picture(tablero.under(generarFilaEmpiezaConBlanco()))
draw(tablero)
```

Ejercicio (g)

```
from interpreter import draw
from chessPictures import *
def generarFilaEmpiezaConNegro():
    fila=Picture(square.negative())
    for i in range(7):
        if(i%2==0):
```

```
        fila=Picture(fila.join(square))
    else:
        fila=Picture(fila.join(square.negative()))
    return fila

def generarFilaEmpiezaConBlanco():
    fila=square
    for i in range(7):
        if(i%2==0):
            fila=Picture(fila.join(square.negative()))
        else:
            fila=Picture(fila.join(square))
    return fila

def printArr(arr):
    for i in arr:
        print(i)

def generarTablero():
    tablero= generarFilaEmpiezaConBlanco()
    for i in range(7):
        if(i%2==0):
            tablero=Picture(tablero.under(generarFilaEmpiezaConNegro()))
        else :
            tablero=Picture(tablero.under(generarFilaEmpiezaConBlanco()))
    return tablero

def generarCuadradoVacio():
    cuadrado=[]
    for i in range(58):
        fila=''
        for j in range(58):
            fila+='□'
        cuadrado.append(fila)
    return cuadrado

def generarFilaSinPiezas():
    cuadrado=Picture(generarCuadradoVacio())
    filaVacía=cuadrado.horizontalRepeat(7)
    return filaVacía

def ponerPiezas(tablero,piezas):
    try:
        imagen = piezas.img
    except:
        imagen = piezas
```



```
try:
    tabImagen = tablero.img
except:
    tabImagen = tablero
newTab=[]
for i in range(len(tabImagen)):
    newFila=''
    for j in range(len(tabImagen[i])):
        try:
            if(imagen[i][j]=="□"):
                newFila+=tabImagen[i][j]
            else:
                newFila+=imagen[i][j]
        except:
            newFila+=tabImagen[i][j]
    newTab.append(newFila)
return newTab

tablero=generarTablero()
filaSinPiezas=generarFilaSinPiezas()
fila1=Picture(Picture(Picture(Picture(Picture(Picture(Picture(Picture(rock.join(knight)).join(bishop))
.join(queen)).join(king)).join(bishop)).join(knight)).join(rock)).negative())
fila2=Picture(pawn.horizontalRepeat(7)).negative()
fila7=Picture(pawn.horizontalRepeat(7))
fila8=Picture(Picture(Picture(Picture(Picture(Picture(Picture(rock.join(knight)).join(bishop))
.join(queen)).join(king)).join(bishop)).join(knight)).join(rock))
piezas=Picture(Picture(Picture(Picture(Picture(Picture(Picture(fila1.under(fila2)).under(filaSinPiezas))
.under(filaSinPiezas)).under(filaSinPiezas)).under(filaSinPiezas)).under(fila7)).under(fila8))

tablero=ponerPiezas(tablero,piezas)
draw(tablero)
```


II. PRUEBAS

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 10

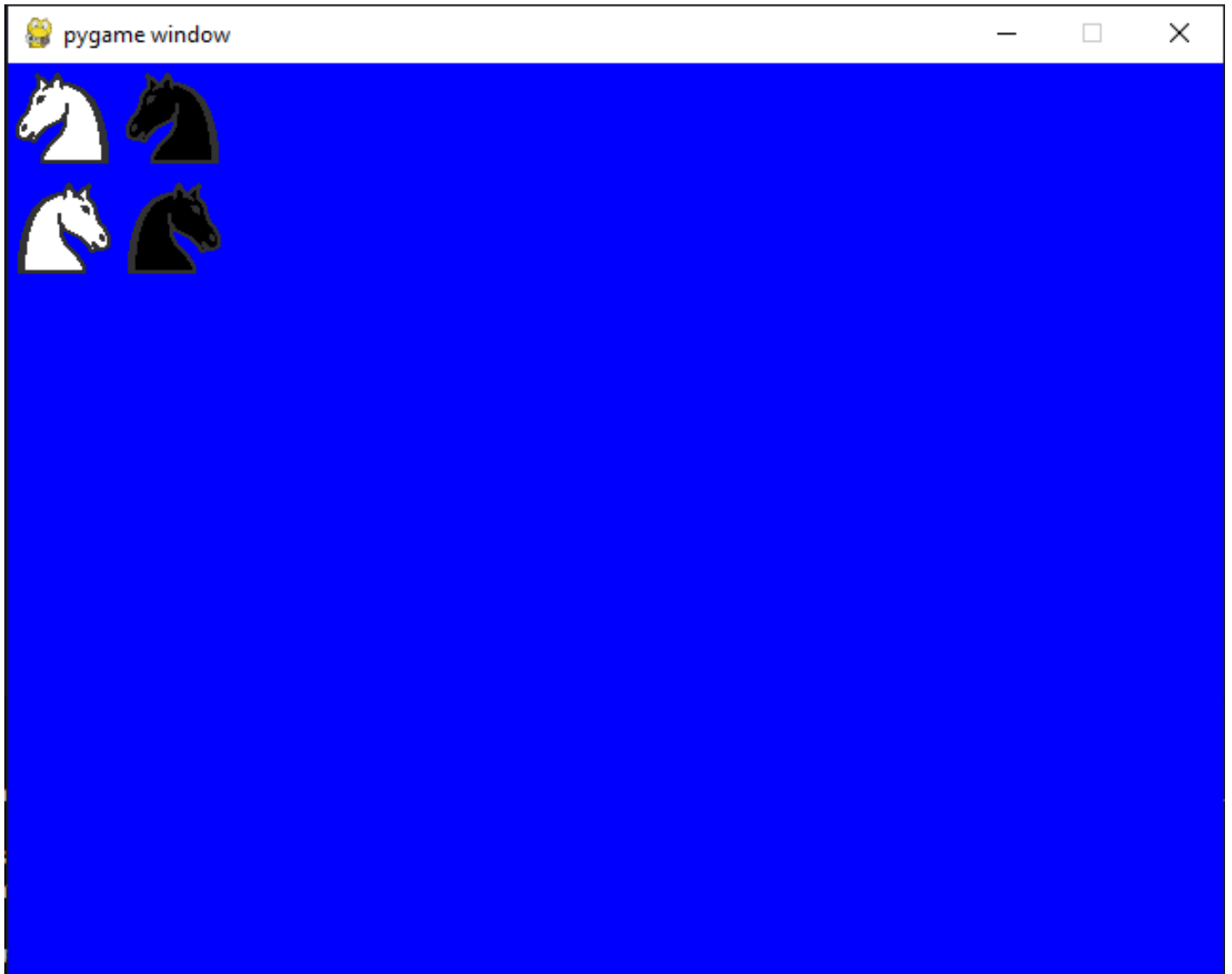
Ejecucion (a) pygame window

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 11


Ejecucion (b)

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 12

Ejecucion (c) pygame window



UNIVERSIDAD NACIONAL DE SAN AGUSTIN
FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y
SERVICIOS ESCUELA PROFESIONAL DE
INGENIERÍA DE SISTEMA-



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

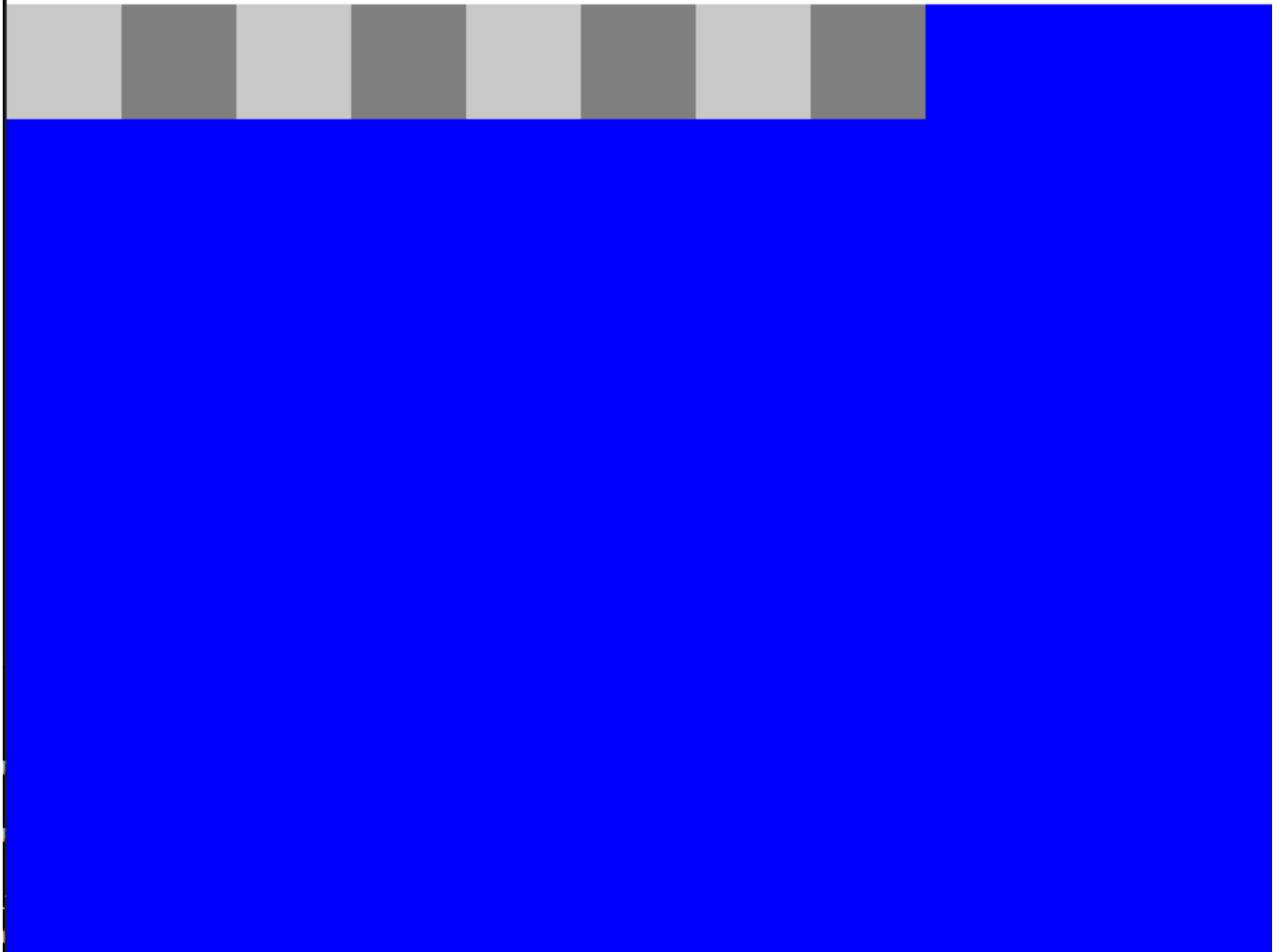
Código: GUIA-PRLE-001

Página: 13

Ejecucion (d)



pygame window





UNIVERSIDAD NACIONAL DE SAN AGUSTIN
FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y
SERVICIOS ESCUELA PROFESIONAL DE
INGENIERÍA DE SISTEMA-



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

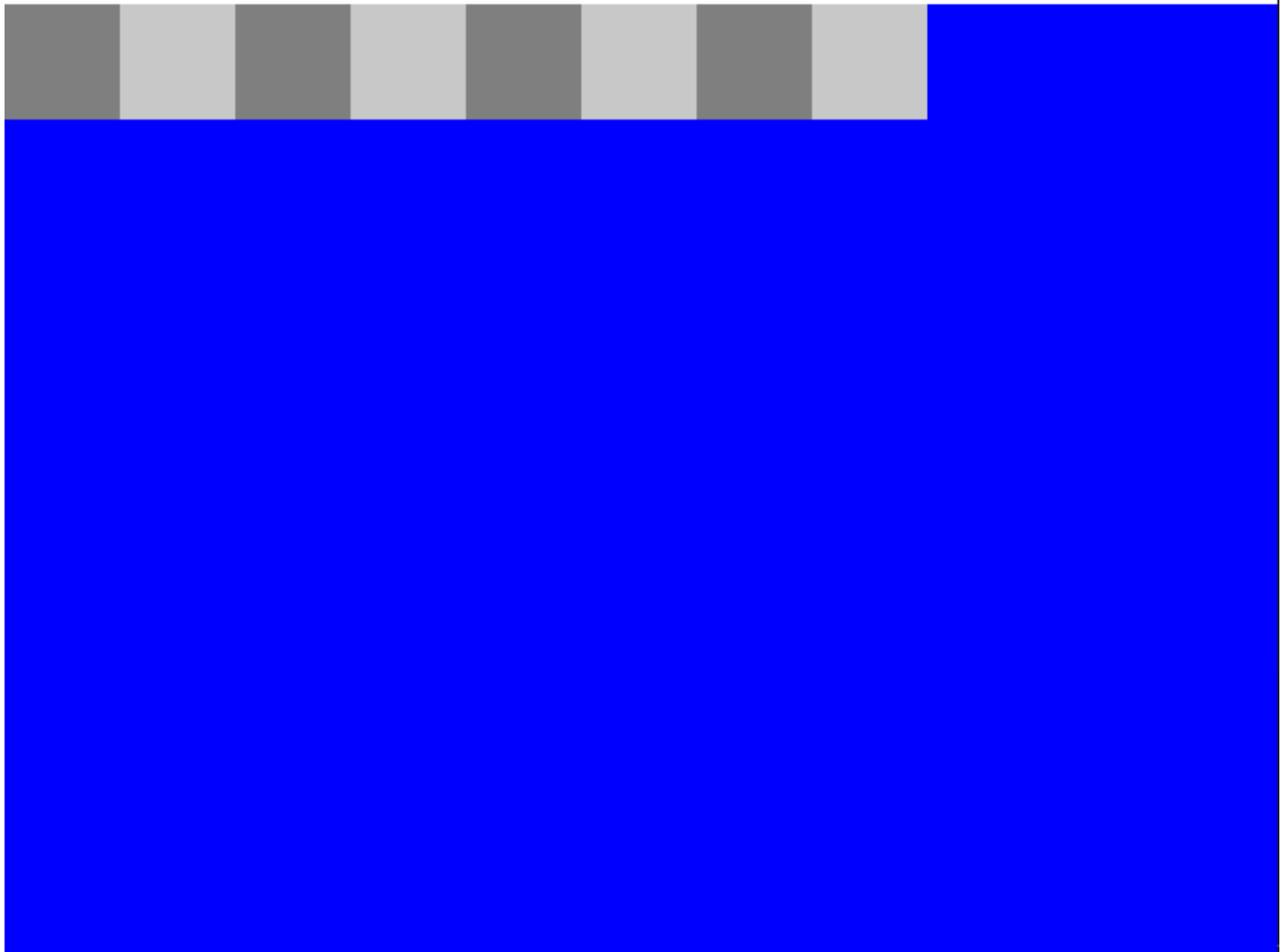
Código: GUIA-PRLE-001

Página: 14

Ejecucion (e)



pygame window



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

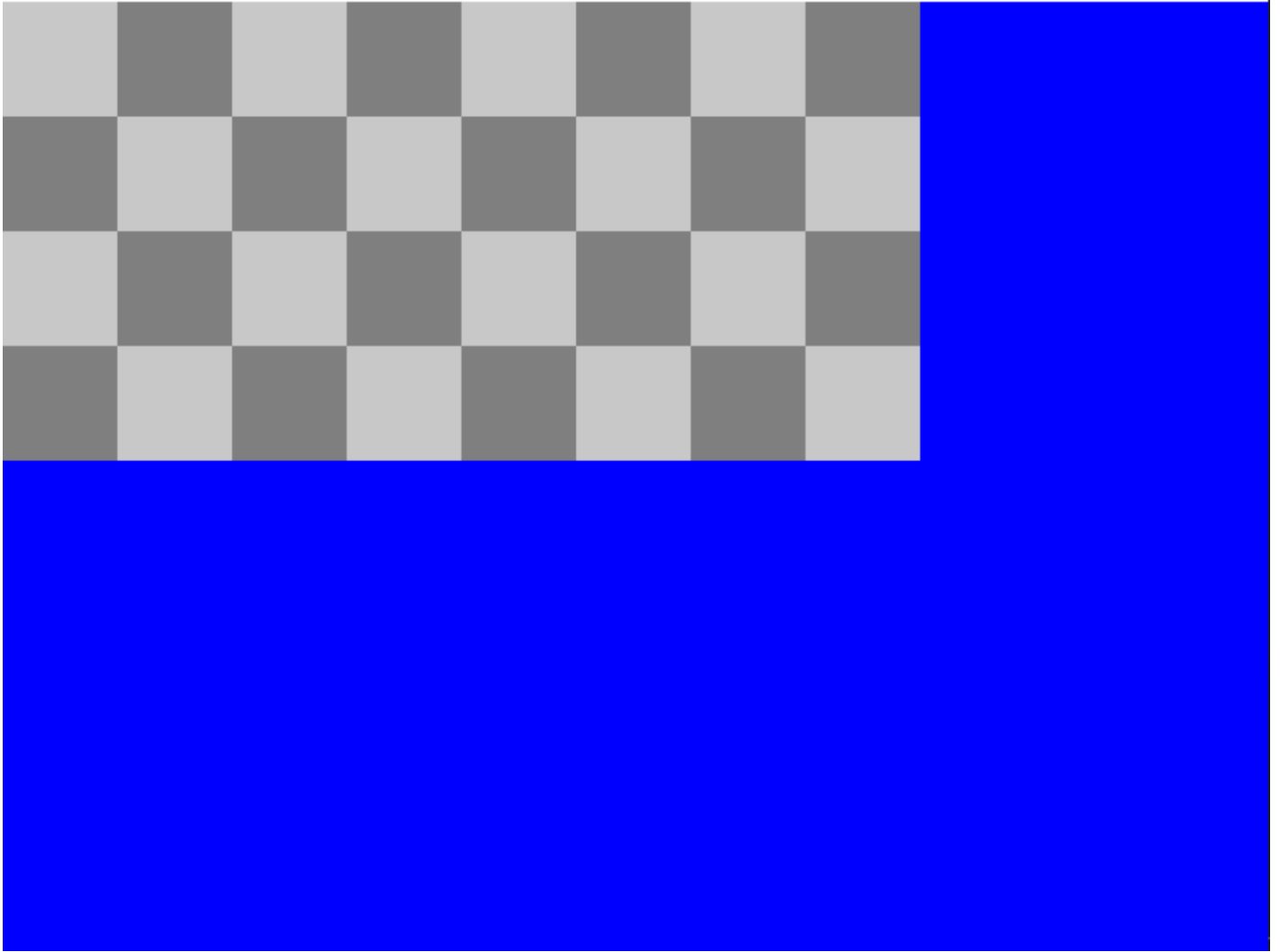
Aprobación: 2022/03/01

Código: GUIA-PRLE-001


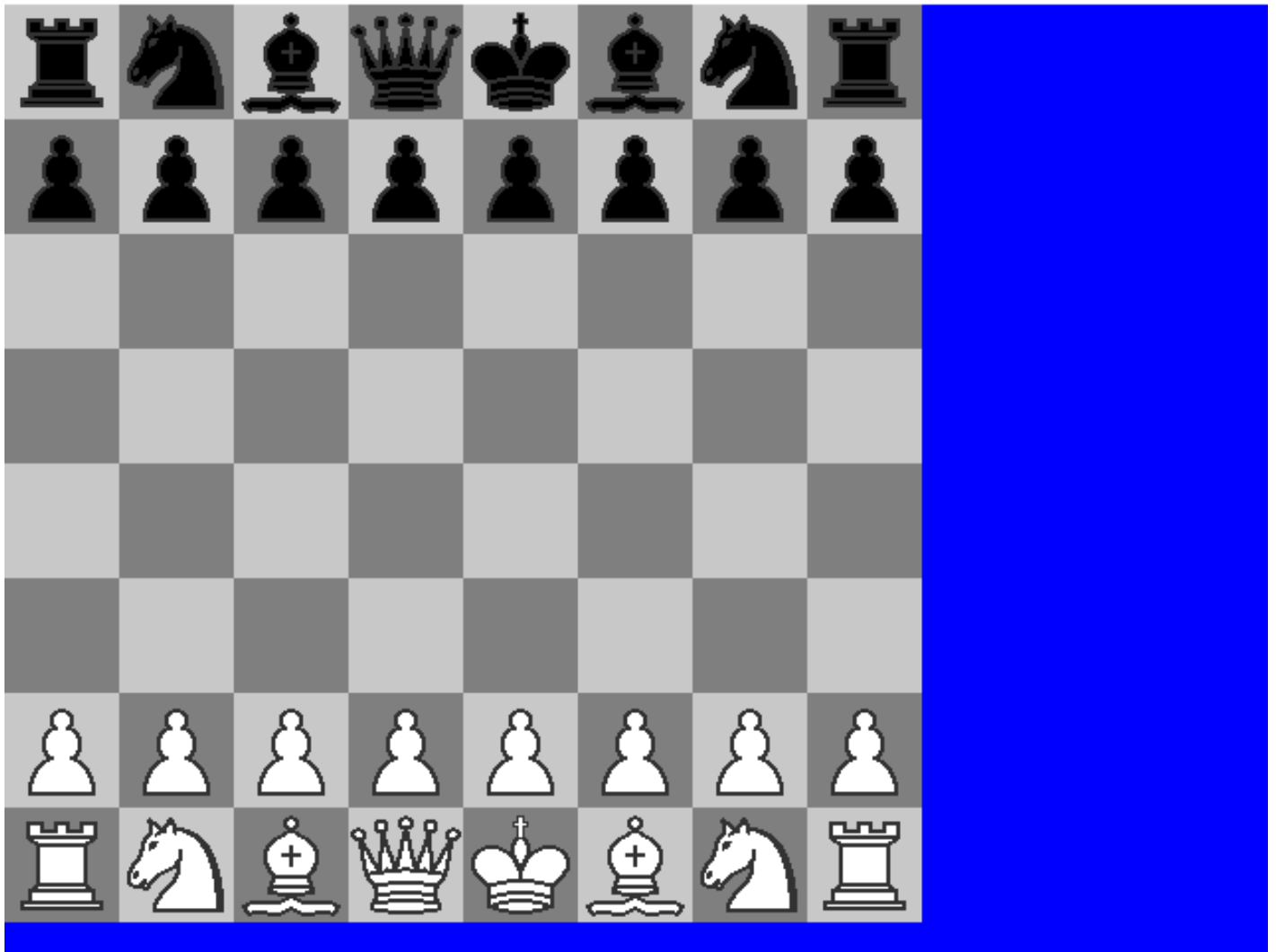
Página: 15

Ejecucion (f)

pygame window



Ejecucion (g)

 pygame window



III. CUESTIONARIO:

Explique: ¿Para qué sirve el directorio pycache?

El directorio pycache en Python es utilizado para almacenar los archivos bytecode compilados de los módulos Python.

Estos archivos tienen la extensión .pyc (o a veces .pyo si se crean en un modo de optimización). La razón principal de la existencia de este directorio y los archivos que contiene es mejorar la eficiencia en la ejecución de los programas Python.

Compilación:

Cuando ejecutas un programa Python, el intérprete primero compila el código fuente (.py) en bytecode, un formato intermedio que es más fácil y rápido de ejecutar por la máquina virtual de Python (PVM). Este bytecode compilado es lo que se almacena en los archivos .pyc dentro del directorio pycache.

Ejecución:

La próxima vez que ejecute el mismo módulo, si no ha cambiado, el intérprete de Python puede cargar directamente el bytecode compilado desde el archivo .pyc en lugar de volver a compilar el código fuente.

Esto ahorra tiempo de ejecución, ya que la compilación puede ser un proceso costoso en términos de tiempo, especialmente para programas grandes o complejos.

CONCLUSIONES

Para las conclusiones tenemos que python es un lenguaje poderoso, en este laboratorio usamos la librería pygame para convertir un string, en este caso fichas del juego ajedrez a interfaz gráfica. También utilizamos las clases en python y me di cuenta que es parecido a java

METODOLOGÍA DE TRABAJO

Para la metodología utilizada, fue primero llenar la clase picture para poder, realizar todos sus métodos sobre las pictures (las fichas de ajedrez), luego de implementar la clase picture se procedió a realizar los ejercicios de interfaz gráfica que pedía el laboratorio



Estructura de directorios lab 5:

```
jhuamanicond/  
|-- pw2-24a  
|   |-- lab05  
|       |-- Tarea-del-Ajedrez  
|       |   |-- .gitignore  
|       |   |-- Ejercicio2a.py  
|       |   |-- Ejercicio2b.py  
|       |   |-- Ejercicio2c.py  
|       |   |-- Ejercicio2d.py  
|       |   |-- Ejercicio2e.py  
|       |   |-- Ejercicio2f.py  
|       |   |-- Ejercicio2g.py  
|       |   |-- chessPictures.py  
|       |   |-- colors.py  
|       |   |-- interpreter.py  
|       |   |-- picture.py  
|       |   |-- pieces.py  
|       |-- latex  
|       |   |-- lab05.pdf  
|       |   |-- lab05.tex
```



	Contenido y demostración	Puntos	Checklist	Estudiante	Profesor
1. GitHub	Repositorio se pudo clonar y se evidencia la estructura adecuada para revisar los entregables. (Se descontará puntos por error o omisión)	4	✓	4	
2. Commits	Hay porciones de código fuente asociado a los commits planificados con explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	✓	3	
3. Ejecución	Se incluyen comandos para ejecuciones y pruebas del código fuente explicadas gradualmente que permitirían replicar el proyecto. (Se descontará puntos por cada omisión)	4	✓	3	
4. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	✓	2	
7. Ortografía	El documento no muestra errores ortográficos. (Se descontará puntos por error encontrado)	2	✓	1	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente con explicaciones puntuales pero precisas, agregando diagramas generados partir del código fuente y refleja un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	✓	3	
	Total			16	

REFERENCIAS

https://www.w3schools.com/python/python_reference.asp

<https://docs.python.org/3/tutorial/>