Informe de Laboratorio 04

Tema: python

Nota

Estudiante	Escuela	Asignatura
Kevin Andree Llacma Quispe kllacma@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programacion web 2 Semestre: I Código: 20200585

Laboratorio	Tema	Duración
04	python	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	-	-

Programación Web
Laboratorio 05
Tema: python

5 de junio de 2024

1. Competencias

- General: C.c. Diseña responsablemente aplicaciones web, sus componentes o procesos para satisfacer necesidades dentro de restricciones realistas: económicas, medio ambientales, sociales,políticas, éticas, de salud, de seguridad, manufacturación y sostenibilidad.
- Específica: C.m. Construye responsablemente soluciones con tecnología web siguiendo un proceso adecuado llevando a cabo las pruebas ajustada a los recursos disponibles del cliente.
- Específica: C.p. Aplica de forma flexible técnicas, métodos, principios, normas, estándares y herramientas del desarrollo web necesarias para la construcción de aplicaciones web e implementación de estos sistemas en una organización.

2. Desarollo del lab

2.1. Descripcion

Ejercicio2a

para realziar el ejercicio se tuvo que modificar el archivo picture.py
.Usando los metodos negative, join y up

```
def join(self, p):

    """ Devuelve una nueva figura poniendo la figura del argumento
        al lado derecho de la figura actual """
    joined_img = []
    for row1, row2 in zip(self.img, p.img):
        joined_img.append(row1 + row2)
    return Picture(joined_img)
```

```
def up(self, p):
    """ Devuelve una nueva figura poniendo la figura p encima de la figura actual """
    return Picture(self.img + p.img)
```

```python
def negative(self):
    """ Devuelve un negativo de la imagen """
    negative_img = []
    for line in self.img:
        neg_line = "".join(self._invColor(char) for char in line)
        negative_img.append(neg_line)
    return Picture(negative_img)
```

Primero crea los caballos blanco y negro que seria negativo. Luego une horizontalmente el caballo blanco (knight white) y el caballo negro invertido (knight black). despues a lo contrario Finalemnte los une

Ejercicio2a.py > ...

```python
from interpreter import draw
from chessPictures import knight

# Crea los caballos
knight_white = knight
knight_black = knight.negative()

# Unir caballos horizontal yverticalmente
top_row = knight_white.join(knight_black)
bottom_row = knight_black.join(knight_white)

# Unir las dos fils de caballo verticalmene
four_knights = top_row.up(bottom_row)

# Dibujar la image
draw(four_knights)

```

Ejercicio2b Para este ejercicio. Se uso lo mismo que en anterior cambiando por vertical mirror los dos de abajo

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

Podemos observar que lo hacemos para que los caballos de abajo miren al contrario

Ejercicio2b.py > ...

```python
from interpreter import draw
from chessPictures import knight


knight_white = knight
knight_black = knight.negative()
knight_white_flipped = knight_white.verticalMirror()
knight_black_flipped = knight_black.verticalMirror()

top_row = knight_white.join(knight_black)
bottom_row = knight_black_flipped.join(knight_white_flipped)

four_knights = top_row.up(bottom_row)


draw(four_knights)

```

Ejercicio2c

Para este ejercicio se hace uso de horizontalRepeat:

- Toma la imagen actual (self.img), la expande horizontalmente n veces y devuelve esta nueva imagen como una instancia de la clase Picture.
- Cada fila de la imagen original se concatena n veces para formar la nueva imagen expandida horizontalmente.

```python
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_img = self.img
    for _ in range(n - 1):
        repeated_img = [row + row_part for row, row_part in zip(repeated_img, self.img)]
    return Picture(repeated_img)
```

lLe mandamos para que repita 4 veces la misma imagen

Ejercicio2c.py > ...

H:\djlnfsd\Ejercicio2c.py

```python
import draw
from chessPictures import queen

four_queens = queen.horizontalRepeat(4)

draw(four_queens)

```

Ejercicio2d

Combinamos lo que habuiamos hecho usando horizontalrepeat y join para imprimir cuadrados intercalando color y haciendolo horizontalmente

Ejercicio2d.py > ...

```python
from interpreter import draw
from chessPictures import square

white_square = square
black_square = square.negative()

row = white_square.join(black_square).horizontalRepeat(4)

draw(row)

```

Ejercicio2e

Lo mismo solo que ahora empieza con el color contario

Ejercicio2e.py > ...

```python
from interpreter import draw
from chessPictures import square

black_square = square.negative()
white_square = square

row = black_square.join(white_square).horizontalRepeat(4)

draw(row)

```

Ejercicio2f

Ahora hacemos uso de un nuevo metodo verticalrepeat

```python
def verticalRepeat(self, n):
    """ Devuelve una nueva figura repitiendo la figura actual hacia abajo
        la cantidad de veces que indique el valor de n """
    return Picture(self.img * n)
```

Ahora agregamos el metodo join para imprimir mas de 2 filas. Creamos 2 filas una empieza con blanco y la optra con negro y se unen. Finalmente se repiten 2 veces hacia abajo

```python
from interpreter import draw
from chessPictures import square

white_square = square
black_square = square.negative()

# fila intercalando 8 cuadros, comenzando con blanco
row_start_white = white_square.join(black_square).horizontalRepeat(4)

# comienza con negro
row_start_black = black_square.join(white_square).horizontalRepeat(4)

checkerboard = row_start_white.up(row_start_black).verticalRepeat(2)

draw(checkerboard)
```

Ejercicio2g

Hacemos uso de los que implementamos anterioremente

```python
from interpreter import draw
from chessPictures import square, rock, knight, bishop, queen, king, pawn

white_square = square
black_square = square.negative()


row_start_white = white_square.join(black_square).horizontalRepeat(4)
row_start_black = black_square.join(white_square).horizontalRepeat(4)

# filas alternando los colores
row1 = row_start_white
row2 = row_start_black
row3 = row_start_white
row4 = row_start_black
row5 = row_start_white
row6 = row_start_black
row7 = row_start_white
row8 = row_start_black

# filas con las piezas negras
row1_with_pieces = rock.join(knight).join(bishop).join(queen).join(king).join(bishop).join(knight).join(rock)
row2_with_pieces = pawn.horizontalRepeat(8)

# filas vacias
empty_row1 = row_start_white
empty_row2 = row_start_black
empty_row3 = row_start_white
empty_row4 = row_start_black

# filas con las piezas blancas
row7_with_pieces = pawn.horizontalRepeat(8)
row8_with_pieces = rock.join(knight).join(bishop).join(queen).join(king).join(bishop).join(knight).join(rock)

# Combinar todopara formar el tablero
chessboard = row1_with_pieces.up(row2_with_pieces).up(empty_row1).up(empty_row2).up(empty_row3).up(empty_row4).up(row7_with_pieces).up(row8_with_pieces)


draw(chessboard)
```