МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж

Лабораторна робота №4

з дисципліни

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

на тему

Розробка ASCII ART генератора для візуалізації 2D-фігур

Виконав:

ст. гр. РІ-21сп

Наконечний В.Б.

Прийняв:

Щербак С.С.

Львів-2024

**Мета лабораторної роботи:** Створення Генератора ASCII-арту без використання зовнішніх бібліотек

**Хід роботи**

**Завдання 1: Введення користувача.**

Створіть програму Python, яка отримує введення користувача щодо слова або фрази, яку вони хочуть перетворити в ASCII-арт.

**Завдання 2: Набір символів.**

Визначте набір символів (наприклад, '@', '#', '\*', тощо), які будуть використовуватися для створення ASCII-арту. Ці символи будуть відображати різні відтінки.

**Завдання 3: Розміри Art-у.**

Запитайте у користувача розміри (ширина і висота) ASCII-арту, який вони хочуть створити. Переконайтеся, що розміри в межах керованого діапазону

**Завдання 4: Функція генерації Art-у.**

Напишіть функцію, яка генерує ASCII-арт на основі введення користувача, набору символів та розмірів. Використовуйте введення користувача, щоб визначити, які символи використовувати для кожної позиції в Art-у.

**Завдання 5: Вирівнювання тексту.**

Реалізуйте опції вирівнювання тексту (ліво, центр, право), щоб користувачі могли вибирати, як їх ASCII-арт розміщується на екрані.

**Завдання 6: Відображення мистецтва.**

Відобразіть створений ASCII-арт на екрані за допомогою стандартних функцій друку Python.

**Завдання 7: Збереження у файл.**

Додайте можливість зберігати створений ASCII-арт у текстовий файл, щоб користувачі могли легко завантажувати та обмінюватися своїми творіннями.

**Завдання 8: Варіанти кольорів.**

Дозвольте користувачам вибирати опції кольорів (чорно-білий, відтінки сірого) для свого ASCII-арту.

**Завдання 9: Функція попереднього перегляду.**

Реалізуйте функцію попереднього перегляду, яка показує користувачам попередній перегляд їх ASCII-арту перед остаточним збереженням

**Завдання 10: Інтерфейс, зрозумілий для користувача.**

Створіть інтерфейс для користувача у командному рядку, щоб зробити програму легкою та інтуїтивно зрозумілою для використання.

**Виконання роботи**

**Папка Classes:**

**Вміст файлу alignment\_manager.py:**

class AlignmentManager:

def \_\_init\_\_(self):

self.alignment = "center"

def choose\_alignment(self):

print("Select text alignment:")

print("1. Left")

print("2. Center")

print("3. Right")

alignment\_option = input("Select option 1, 2 or 3: ").strip()

if alignment\_option == '1':

self.alignment = "left"

elif alignment\_option == '2':

self.alignment = "center"

elif alignment\_option == '3':

self.alignment = "right"

else:

print("Incorrect option: Default center alignment (center) is set.")

self.alignment = "center"

def \_align\_line(self, line, width):

if self.alignment == 'left':

return line.ljust(width)

elif self.alignment == 'center':

return line.center(width)

elif self.alignment == 'right':

return line.rjust(width)

else:

return line

**Вміст файлу art\_operations.py:**

import os

from Classes.alignment\_manager import AlignmentManager

from Classes.file\_manager import FileManager

from Classes.color\_manager import ColorManager

class ArtOperations:

def \_\_init\_\_(self):

self.color\_manager = ColorManager()

self.char\_set = {}

self.load\_characters()

self.width = 0

self.height = 0

self.text = ""

self.alignment\_manager = AlignmentManager()

def load\_characters(self):

for char in "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!@#\_+-:,. ":

if char.isupper():

filename = f"/Users/admin/Desktop/ДЗ/3 семестр/Спеціалізовані мови програмування/Lab4/Assets/letters/{char}{char}.txt"

elif char.islower():

filename = f"/Users/admin/Desktop/ДЗ/3 семестр/Спеціалізовані мови програмування/Lab4/Assets/letters/{char}.txt"

elif char.isdigit():

filename = f"/Users/admin/Desktop/ДЗ/3 семестр/Спеціалізовані мови програмування/Lab4/Assets/numbers/{char}.txt"

else:

filename = f"/Users/admin/Desktop/ДЗ/3 семестр/Спеціалізовані мови програмування/Lab4/Assets/symbols/{char}.txt"

if os.path.exists(filename):

with open(filename, 'r') as file:

self.char\_set[char] = [line.rstrip('\n') for line in file.readlines()]

else:

self.char\_set[char] = [' ' \* 10] \* 10

def choose\_alignment(self):

self.alignment\_manager.choose\_alignment()

def choose\_color(self):

self.color\_manager.choose\_color()

def apply\_color(self, text):

return self.color\_manager.apply\_color(text)

def display\_art(self, ascii\_art):

for line in ascii\_art:

colored\_line = self.apply\_color(line)

aligned\_line = self.alignment\_manager.\_align\_line(colored\_line, self.width \* len(self.text))

print(aligned\_line)

def save\_art\_to\_file(self, ascii\_art):

FileManager.save\_art\_to\_file(ascii\_art)

**Вміст файлу ASCIIart\_generator.py:**

class ArtGenerator:

def \_\_init\_\_(self, char\_set, width, height, text):

self.char\_set = char\_set

self.width = width

self.height = height

self.text = text

def generate\_art(self):

ascii\_lines = ['' for \_ in range(self.height)]

for char in self.text:

char\_lines = self.char\_set.get(char, [' ' \* self.width] \* self.height)

if len(char\_lines) < self.height:

char\_lines += [' ' \* self.width] \* (self.height - len(char\_lines))

for i in range(self.height):

ascii\_lines[i] += char\_lines[i] + ' '

return ascii\_lines

**Вміст файлу color\_manager.py:**

import json

class ColorManager:

def \_\_init\_\_(self):

self.color\_code = ""

self.colors = self.load\_colors()

def load\_colors(self):

with open('/Users/admin/Desktop/ДЗ/3 семестр/Спеціалізовані мови програмування/Lab4/Config/colors.json', 'r', encoding='utf-8') as file:

colors = json.load(file)

return colors["colors"]

def choose\_color(self):

print("Select color for ASCII-art:")

for key in self.colors.keys():

print(key)

color\_option = input("Select option: ").strip()

self.color\_code = self.colors.get(color\_option, "0")

def apply\_color(self, text):

return f"\033[{self.color\_code}m{text}\033[0m"

**Вміст файлу file\_manager.py:**

import os

class FileManager:

@staticmethod

def save\_art\_to\_file(ascii\_art):

if not os.path.exists('/Users/admin/Desktop/ДЗ/3 семестр/Спеціалізовані мови програмування/Lab4/Data'):

os.makedirs('/Users/admin/Desktop/ДЗ/3 семестр/Спеціалізовані мови програмування/Lab4/Data')

filename = input("Enter a file name to save ASCII-art with extension .txt: ")

filepath = os.path.join('/Users/admin/Desktop/ДЗ/3 семестр/Спеціалізовані мови програмування/Lab4/Data', filename)

with open(filepath, 'w') as file:

for line in ascii\_art:

file.write(line + '\n')

print(f"ASCII-art was saved in file {filepath}")

**Папка Config:**

**Вміст файлу colors.json:**

{

"colors": {

"Black": "30",

"Red": "31",

"Green": "32",

"Yellow": "33",

"Blue": "34",

"Fiolet": "35",

"Cyan": "36",

"Grey": "37",

"Dark-grey": "90",

"White": "97"

}

}

**Папка UI:**

**Вміст файлу user\_interface.py:**

from Classes.ASCIIart\_generator import ArtGenerator

from Classes.art\_operations import ArtOperations

def main():

while True:

print("Generator ASCII-arts")

print("1. Create new ASCII-art")

print("2. Exit")

choice = input("Select option, 1 or 2: ").strip()

if choice == '1':

generator = ArtOperations()

generator.text = input("Enter a word to convert to ASCII-art: ")

generator.width = int(input("Enter width (default 5): ") or 5)

generator.height = int(input("Enter the height (default is 5): ") or 5)

generator.choose\_alignment()

generator.choose\_color()

art\_generator = ArtGenerator(generator.char\_set, generator.width, generator.height, generator.text)

ascii\_art = art\_generator.generate\_art()

generator.display\_art(ascii\_art)

print("Want to save ASCII-art in file?")

print("1. Yes")

print("2. No")

save\_option = input("Select option 1 or 2: ").strip()

if save\_option == '1':

generator.save\_art\_to\_file(ascii\_art)

elif save\_option == '2':

print("ASCII-art is not saved.")

else:

print("Incorrect option: ASCII art is not saved.")

elif choice == '2':

print("Exit from generator!")

break

else:

print("Incorrect option: Please, try again!")

**Вміст файлу runner.py:**

from UI.user\_interface import main

if \_\_name\_\_ == "\_\_main\_\_":

main()

Результат виконання, а саме генерація ASCII-арту та збереження його у файлі зображено на рис. 1 та рис. 2.



Рис. 1 – Генерація ASCII-арту



Рис. 2 – Зображення арту у файлі

**Висновок:** У ході виконання ЛР я створив генератор ASCII-арту з нуля, який надає можливість налаштовувати символи, розміри, вирівнювання та кольори для арту, а також має можливість зберігати арт у файлі.