МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем та мереж

Лабораторна робота №3

з курсу

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

Виконав студент

групи ІТ-21сп

**Гузар А.І.**

Прийняв

**Щербак С.С.**

Львів - 2023

**Мета:** створення додатка Генератора ASCII-арту.

**План роботи**

**Завдання 1: Введення користувача.**

Створити Python-програму, яка приймає введення користувача для слова або фрази, яку треба перетворити в ASCII-арт.

**Завдання 2: Бібліотека ASCII-арту.**

Інтегрувати бібліотеку ASCII-арту (наприклад, pyfiglet або art) у програму для генерації ASCII-арту з введення користувача.

**Завдання 3: Вибір шрифту.**

Дозволити користувачам вибирати різні стилі шрифтів для свого ASCII-арту. Надати список доступних шрифтів та дозвольте їм вибрати один.

**Завдання 4: Колір тексту.**

Реалізувати опцію вибору користувачем кольору тексту для їхнього ASCII-арту. Підтримати основний вибір кольорів (наприклад, червоний, синій, зелений).

**Завдання 5: Форматування виводу.**

Переконатися, що створений ASCII-арт правильно відформатований та вирівнюється на екрані для зручності читання.

**Завдання 6: Збереження у файл.**

Додати функціональність для збереження створеного ASCII-арту у текстовому файлі, щоб користувачі могли легко завантажувати та обмінюватися своїми творіннями.

**Завдання 7: Розмір ARTу.**

Дозволити користувачам вказувати розмір (ширина і висота) ASCII-арту, який вони хочуть створити. Масштабувати текст відповідно.

**Завдання 8: Вибір символів.**

Дозволити користувачам вибирати символи, які вони хочуть використовувати для створення ASCII-арту (наприклад, '@', '#', '\*', тощо).

**Завдання 9: Функція попереднього перегляду.**

Реалізувати функцію попереднього перегляду, яка показує користувачам попередній перегляд їхнього ASCII-арту перед остаточним збереженням.

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

Створити зручний для користувача інтерфейс командного рядку для додатка, щоб зробити його інтуїтивно зрозумілим та легким у використанні.

Код виконаних завдань представлено нижче.

import psutil

import pyfiglet

from colorama import init, Fore

class ASCIIArtGenerator:

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

self.text = ""

self.font = "standard"

self.color = Fore.WHITE

self.character = "@"

def get\_user\_input(self):

self.text = input("Enter a word or phrase: ")

def choose\_font(self):

fonts = ["standard", "banner", "big", "block", "doom", "slant"]

print("Available fonts:")

for font in fonts:

print(font)

font\_1 = input("Choose a font: ").lower()

if font\_1 not in fonts:

print("Invalid font choice. Using default (standard).")

else:

self.font = font\_1

def choose\_color(self):

print("Available colors: RED, BLUE, GREEN, YELLOW, WHITE, etc.")

color\_choice = input("Choose a color: ").upper()

if hasattr(Fore, color\_choice):

self.color = getattr(Fore, color\_choice)

print(f"Selected color: {color\_choice}")

else:

print(f"Invalid color choice: {color\_choice}. Using default (WHITE).")

def choose\_art\_size(self):

while True:

try:

self.width = int(input("Enter the width of the ASCII art: "))

self.height = int(input("Enter the height of the ASCII art: "))

if self.width <= 0 or self.height <= 0:

print("Width and height must be positive integers.")

else:

break

except ValueError:

print("Invalid input. Please enter a valid integer for width and height.")

def generate\_ascii\_art(self):

ascii\_art = pyfiglet.Figlet(font=self.font).renderText(self.text)

formatted\_ascii\_art = self.format\_ascii\_art(ascii\_art)

return formatted\_ascii\_art

def format\_ascii\_art(self, ascii\_art):

lines = ascii\_art.split('\n')

formatted\_art = ""

for line in lines:

formatted\_art += line.center(self.width) + "\n"

return formatted\_art

def preview\_ascii\_art(self, ascii\_art):

print("Preview of the ASCII art:")

print(self.color + ascii\_art)

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

filename = input("Enter the file name for saving (with .txt extension): ")

available\_memory = psutil.virtual\_memory().available

ascii\_art\_size = len(ascii\_art.encode('utf-8'))

if ascii\_art\_size > available\_memory:

print("Not enough memory to save the ASCII art. Please reduce the size of your art.")

else:

with open(filename, "w") as file:

file.write(ascii\_art)

print(f"ASCII art saved to the file {filename}")

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

init(autoreset=True) # Initializes colorama to automatically reset the color

generator = ASCIIArtGenerator()

generator.get\_user\_input()

generator.choose\_font()

generator.choose\_color()

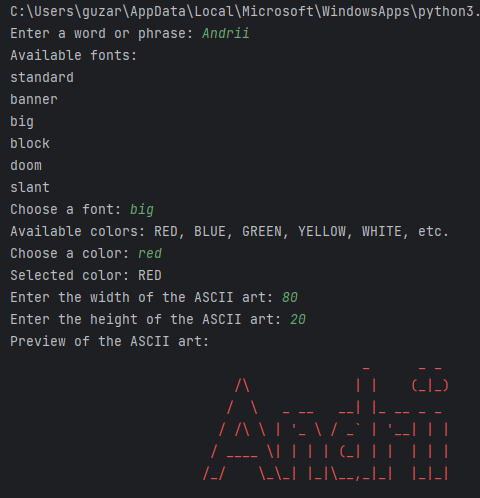
generator.choose\_art\_size()

ascii\_art = generator.generate\_ascii\_art()

generator.preview\_ascii\_art(ascii\_art)

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

На рисунку 1 зображено результат виконання програми.



*Рис.1 Робота ASCII ART генератора*

**Висновок:** під час виконання лабораторної роботи я навчився розробляти ASCII ART генератор для візуалізації текстових даних.