МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж



Лабораторна робота №3

з дисципліни

Спеціалізовані мови програмування

на тему

Розробка ASCII ART генератора для візуалізації текстових даних

Виконав:

ст. гр. ІТ-21сп

Олександр КОЗАК

Прийняв

доцент каф. ІСМ:

Сергій ЩЕРБАК

|  |  |
| --- | --- |
| **Балів** | **Дата** |
|  |  |

Львів-2023

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

**Хід роботи:**

**Завдання 1: Введення користувача**

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

**Завдання 2: Бібліотека ASCII-арту**

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

**Завдання 3: Вибір шрифту**

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

**Завдання 4: Колір тексту**

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

**Завдання 5: Форматування виводу**

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

**Завдання 6: Збереження у файл**

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

**Завдання 7: Розмір ARTу**

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

**Завдання 8: Вибір символів**

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

**Завдання 9: Функція попереднього перегляду**

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

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

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

**main.py**

from ascii\_art\_text import \*

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

runner = Ascii\_art\_runner()

runner.run\_ascii\_art\_gen()

**ascii\_art\_engine.py**

from ascii\_art\_engine import \*

class Ascii\_art\_runner:

def run\_ascii\_art\_gen(self):

ascii\_art\_gen = Ascii\_art\_generator()

user\_text, font, color, preview\_chosse, save\_chosse, user\_cahr, height, width = ascii\_art\_gen.user\_input()

ascii\_art = ascii\_art\_gen.creating\_ascii\_art(user\_text, font, color, user\_cahr, height, width)

ascii\_art\_gen.ascii\_art\_preview(ascii\_art, preview\_chosse)

if save\_chosse == "y":

ascii\_art\_gen.save\_ascii\_art\_to\_txt(ascii\_art, "ascci\_art")

**ascii\_art\_engine.py**

import pyfiglet

from termcolor import colored

from colorama import Fore, Style

from termcolor import COLORS

class Ascii\_art\_generator:

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

self.available\_fonts = ['standard', 'slant', 'alphabet', 'banner3-D', 'doh', 'bulbhead', 'letters', '6x10']

self.available\_colors = ['black', 'red', 'green', 'yellow', 'blue', 'magenta', 'cyan']

def user\_input(self):

user\_text = input("Введіть слово або фразу для перетворення в ASCII-арт: ")

self.show\_colors()

user\_color = self.get\_color("Виберіть номер доступного кольору: ")

self.show\_fonts()

user\_font = self.get\_font("Виберіть номер доступного шрифту: ")

width = self.get\_param("Введіть ширину ASCII-art-ту (кількість символів у ширину): ")

height = self.get\_param("Введіть висоту ASCII-art-ту (кількість символів у вистоу): ")

use\_custom\_chars = input("Ви хочете обрати спец-символ? (y/n): ").lower()

user\_char = self.get\_custom\_char() if use\_custom\_chars == 'y' else None

user\_preview\_chosse = input("Хочете попередньо переглянути арт перед тим як зберегти його?(y/n): ").lower()

user\_save\_chossoe = input("Хочете зберегти створений ASCII-art у текстовий файл?(y/n): ").lower()

return user\_text, user\_font, user\_color, user\_preview\_chosse, user\_save\_chossoe, user\_char, height, width

# Функція для вибору кольору користувачем

def show\_colors(self):

print("Доступні кольори:")

for index, color in enumerate(self.available\_colors, start=1):

print(f"{index}. {color}")

def get\_color(self,message):

while True:

try:

choice = int(input(message))

if 1 <= choice <= len(self.available\_colors):

return self.available\_colors[choice - 1]

else:

print("Виберіть правильний номер кольору.")

except ValueError:

print("Виберіть правильний номер кольору.")

# def set\_color(self):

def show\_fonts(self):

print("Доступні шрифти:")

for index, font in enumerate(self.available\_fonts, start=1):

print(f"{index}. {font}")

def get\_font(self, message):

while True:

try:

choice = int(input(message))

if 1 <= choice <= len(self.available\_fonts):

return self.available\_fonts[choice - 1]

else:

print("Виберіть правильний номер шрифту.")

except ValueError:

print("Виберіть правильний номер шрифту.")

def get\_param(self, message):

while True:

try:

number = int(input(message))

if number > 0:

return number

else:

print("Введіть число ібльше за 0.")

except ValueError:

print("Будь ласка введіть правильне число.")

def get\_custom\_char(self):

return input("Введіть символ яікий хочите використовувати ('@', '#', '\*', ...): ")

def creating\_ascii\_art(self, user\_text, user\_font, user\_color, user\_char, width, height ):

ascii\_art = pyfiglet.figlet\_format(user\_text, font=user\_font)

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

scaled\_ascii\_lines = []

char\_set\_length = len(user\_char) if user\_char else 0

for line in ascii\_lines:

scaled\_line = ""

for char in line:

if char == ' ':

scaled\_line += ' '

else:

if user\_char:

scaled\_line += user\_char[hash(char) % char\_set\_length]

else:

scaled\_line += char

scaled\_ascii\_lines.append(scaled\_line.center(width)[:width])

scaled\_ascii\_text = '\n'.join(scaled\_ascii\_lines[:height])

colored\_ascii\_art = colored(scaled\_ascii\_text, user\_color)

final\_ascii\_art = colored\_ascii\_art

return final\_ascii\_art

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

print(ascii\_art)

def save\_ascii\_art\_to\_txt(self, final\_ascii\_art, file\_name):

with open(f"{file\_name}.txt", "w") as file:

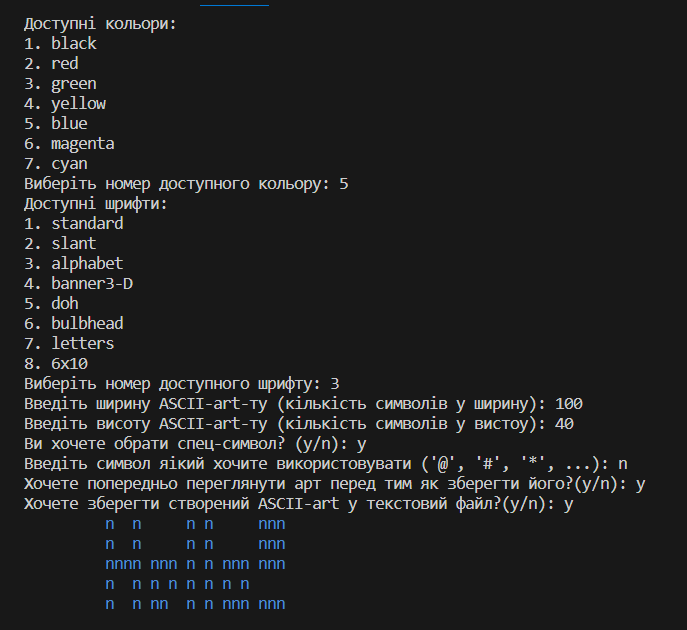
file.write(final\_ascii\_art)

print(f"Ваш ASCII-art успішно збережений у файл з ім'ям {file\_name}.txt! :D")

def ascii\_art\_preview(self, ascii\_art, preview\_chosse):

if preview\_chosse == 'y':

self.show\_ascii\_art(ascii\_art)



*Рис. 1 Результат виконання завдання*

**Висновок:** Виконуючи ці завдання, я створив універсальний Генератор ASCII-арту, який дозволить користувачам налаштовувати свої творіння з різними шрифтами, кольорами, розмірами та символами. Проект надав практичний досвід роботи з введенням користувача, зовнішніми бібліотеками, роботою з файлами та дизайном інтерфейсу користувача в Python.