Міністерство освіти і науки України

Національний університет «Львівська політехніка»

лабораторної роботи №3

з дисципліни «Спеціалізовані мови програмування»

на тему «Розробка ASCII ART генератора для візуалізації текстових даних »

Виконав:

Кулявець В. Р.

Перевірив:

Щербак С.С.

Львів 2024

Мета: створення додатка Генератора ASCII-арту.

Git: <https://github.com/Pivinter/-.git>

Завдання 1: Введення користувача

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

print("Step 1: Enter a word or phrase to convert to ASCII art:")

user\_input = input()

Завдання 2: Бібліотека ASCII-арту

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

import pyfiglet

Завдання 3: Вибір шрифту

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

FONTS = pyfiglet.FigletFont.getFonts()

Завдання 4: Колір тексту

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

COLORS = {

"red": Fore.RED,

"green": Fore.GREEN,

"blue": Fore.BLUE,

"yellow": Fore.YELLOW,

"cyan": Fore.CYAN,

"magenta": Fore.MAGENTA,

"white": Fore.WHITE,

"reset": Style.RESET\_ALL,

}

Завдання 5: Форматування виводу

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

print(center\_text(colored\_art, get\_terminal\_width()))

Завдання 6: Збереження у файл

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

if input("Do you want to save this art? (yes/no): ").lower() == "yes":

filename = input("Enter filename to save: ")

save\_to\_file(filename, colored\_art)

Завдання 7: Розмір ARTу

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

ascii\_gen.set\_width(get\_terminal\_width())

Завдання 8: Вибір символів

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

def generate\_art(self, text):

ascii\_art = pyfiglet.figlet\_format(text, font=self.font, width=self.width)

return ascii\_art.replace('#', self.custom\_char)

Завдання 9: Функція попереднього перегляду

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

print(center\_text(colored\_art, get\_terminal\_width()))

Завдання 10: Інтерфейс, зрозумілий для користувача

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

print("Step 1: Enter a word or phrase to convert to ASCII art:")

user\_input = input()

print("\nStep 2: Choose a font from the list below:")

for index, font in enumerate(FONTS, start=1):

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

font\_choice\_index = get\_user\_input(f"Enter the number corresponding to your font choice (1-{len(FONTS)}): ",

[str(i) for i in range(1, len(FONTS) + 1)])

font\_choice = FONTS[int(font\_choice\_index) - 1]

print("\nStep 3: Choose a color:")

for color in COLORS.keys():

print(f" - {color}")

color\_choice = get\_user\_input("Enter your color choice: ", COLORS.keys())

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