Міністерство освіти і науки України

Національний університет «Львівська політехніка»



**ЗВІТ**

про виконання лабораторної роботи № 3

з курсу:

«Спеціалізовані мови програмування»

**Виконала:**

студентка гр. ІТ-31

Ірина ПЕРХУН

**Прийняв:**

Сергій ЩЕРБАК

Львів 2023

**Тема:** Розробка ASCII ART генератора для візуалізації текстових даних

**Мета роботи:** 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.

*import* os  *# Import the module for interacting with the operating system*

*import* pyfiglet  *# Import the module for generating ASCII art*

*from* termcolor *import* colored  *# Import the module for colored text in the terminal*

*from* tkinter *import* filedialog, Tk  *# Import the module for a file dialog to choose a file*

def list\_available\_fonts():

*return* pyfiglet.Figlet().getFonts()  *# Function to get a list of available fonts for ASCII art*

def generate\_ascii\_art(text, font="standard", color="white", special\_symbols=None):

*try*:

*# Get the size of the terminal*

        columns, \_ = os.get\_terminal\_size()

*# Check if there are special symbols to add at the end of the text*

*if* special\_symbols:

            special\_symbols\_str = ' '.join(special\_symbols)

            text = f"{text} {special\_symbols\_str}"

*# Generate ASCII art using pyfiglet*

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

*# Split the ASCII art into lines*

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

*# Determine the number of characters in the longest line*

        max\_line\_length = max(len(line) *for* line *in* lines)

*# Calculate padding for center alignment*

        padding = (columns - max\_line\_length) // 2

*# Add padding for center alignment*

        centered\_ascii\_art = '\n'.join(' ' \* padding + line *for* line *in* lines)

*# Color the ASCII art with the selected color*

        colored\_ascii\_art = colored(centered\_ascii\_art, color)

*return* colored\_ascii\_art

*except* Exception *as* e:

*return* str(e)

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

    print("Preview of ASCII art:")

    print(ascii\_art)

    input("Press Enter to continue...")  *# Display a preview of the ASCII art and wait for Enter key press*

def save\_ascii\_art(ascii\_art, filename):

*try*:

*# Save the ASCII art to the specified file*

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

            file.write(ascii\_art)

        print(f"ASCII art saved to {filename}")  *# Notify about successful file save*

*except* Exception *as* e:

        print(f"Error saving ASCII art: {e}")  *# Print an error message if there's an issue with saving*

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

    available\_fonts = list\_available\_fonts()

*if* not available\_fonts:

        print("No fonts found.")  *# Display a message if there are no available fonts*

*else*:

        print("Available fonts:")

*for* font *in* available\_fonts:

            print(font)  *# Print a list of available fonts*

*# Prompt the user to enter text for conversion to ASCII art*

        user\_input = input("Enter a word or phrase to convert to ASCII art: ")

        selected\_font = input("Enter the name of the font (choose from the list above): ")

*if* selected\_font not in available\_fonts:

            print("Invalid font. Using the default font.")

            selected\_font = "standard"  *# Use the default font if the selected one is invalid*

        available\_colors = {

            "red": "red",

            "green": "green",

            "blue": "blue",

            "yellow": "yellow",

            "white": "white"

        }

        print("Available colors:")

*for* color *in* available\_colors:

            print(color)  *# Print a list of available colors*

        selected\_color = input("Enter the color for the text (choose from the list above): ")

*if* selected\_color not in available\_colors:

            print("Invalid color. Using white as the default color.")

            selected\_color = "white"  *# Use white color as default if the selected one is invalid*

*# Ask the user if they want to use special symbols*

        use\_special\_symbols = input("Do you want to use special symbols in the end of your phrase like '@', '#', or '\*'? (y/n): ")

        special\_symbols\_after = []

*if* use\_special\_symbols.lower() == 'y':

*while* True:

                symbol\_choice = input("Choose a special symbol: 1 for '@', 2 for '#', 3 for '\* (q to quit): ")

*if* symbol\_choice == '1':

                    special\_symbols\_after.append('@')

*elif* symbol\_choice == '2':

                    special\_symbols\_after.append('#')

*elif* symbol\_choice == '3':

                    special\_symbols\_after.append('\*')

*elif* symbol\_choice.lower() == 'q':

*break*

*# Generate ASCII art*

        ascii\_art\_result = generate\_ascii\_art(user\_input, font=selected\_font, color=selected\_color, special\_symbols=special\_symbols\_after)

*# Display the generated ASCII art for preview*

        preview\_ascii\_art(ascii\_art\_result)

        root = Tk()

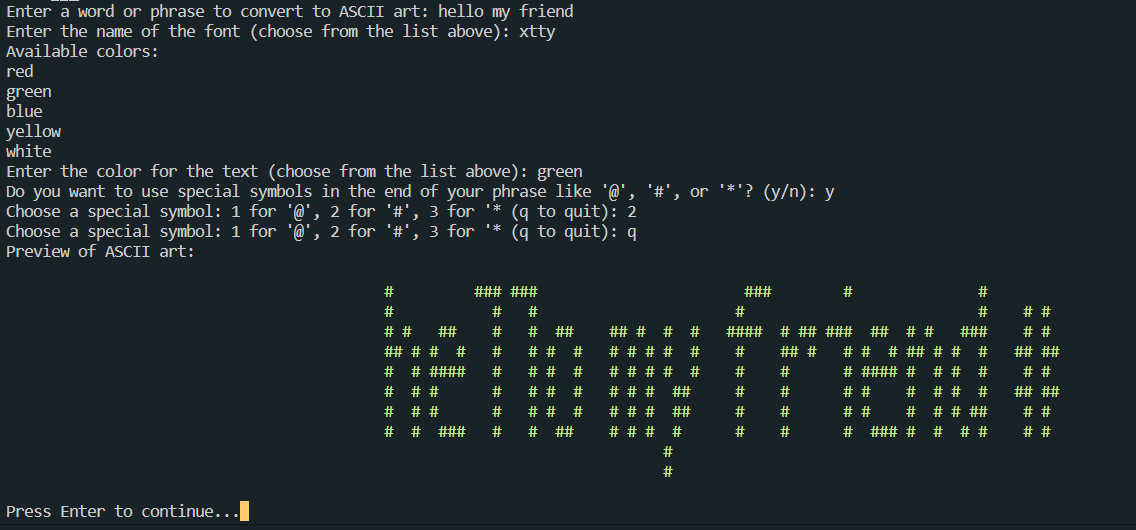
        root.withdraw()  *# Hide the main Tkinter window*

*# Ask the user for a path to save the ASCII art*

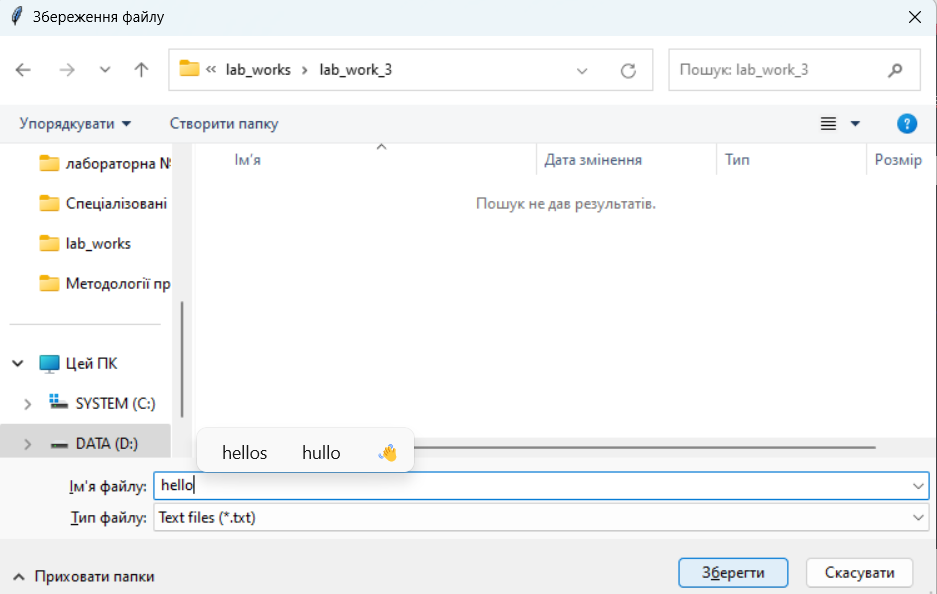
        file\_path = filedialog.asksaveasfilename(defaultextension=".txt", filetypes=[("Text files", "\*.txt")])

*if* file\_path:

            save\_ascii\_art(ascii\_art\_result, file\_path)  *# Save the ASCII art to the specified file*



*Рис. 1 – Результат створення арту*



*Рис. 2 – Збереження результату*

**Висновок:** на даній лабораторній роботі я створила універсальний Генератор ASCII-арту, який дозволить користувачам налаштовувати свої творіння з різними шрифтами, кольорами, розмірами та символами.