Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра інформаційних систем та мереж

ЗВІТ

про виконання лабораторної роботи № 4

**“ Розробка ASCII ART генератора для візуалізації 2D-фігур ”**

з дисципліни **“** **Спеціалізовані мови програмування”**

Виконав:

студент групи ІТ-32

ОЛЕНЮК О. М.

Прийняв:

ЩЕРБАК С. С.

**Львів – 2023**

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

**План 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.

**Код програми:**

functions.py:

def load\_ascii\_art(file\_path):  
 ascii\_dict = {}  
 with open(file\_path, "r", encoding="utf-8") as file:  
 content = file.read().strip().split('@symbol::')  
 for item in content[0:]:  
 symbol, \*lines = item.split('\n')  
 ascii\_dict[symbol] = [line.strip('^$') for line in lines[0:]]  
 return ascii\_dict  
  
  
def select\_color():  
 color\_choice = input("Select text color (white, gray): ").lower()  
 if color\_choice in ['white', 'gray']:  
 return color\_choice  
 else:  
 return None  
  
  
def apply\_color(text, color):  
 color\_codes = {'white': '\u001b[97m', 'gray': '\u001b[90m'}  
 reset\_color = '\u001b[0m'  
 return color\_codes[color] + text + reset\_color  
  
  
def get\_valid\_text(ascii\_dict):  
 while True:  
 text = input("Enter a word or phrase: ")  
 if all(c in ascii\_dict or c == ' ' for c in text):  
 return text  
 else:  
 print("Error. The input contains invalid characters. Please enter a valid phrase.")  
  
  
def print\_ascii\_art(ascii\_dict, text, alignment, max\_width, symbol\_set, color):  
 output\_lines = [""] \* 6  
  
 for symbol in text:  
 if symbol == ' ':  
 for i in range(6):  
 output\_lines[i] += ' '  
 elif symbol in ascii\_dict:  
 symbol\_lines = ascii\_dict[symbol]  
 for i in range(6):  
 output\_lines[i] += symbol\_lines[i]  
  
 if len(output\_lines[0]) > max\_width:  
 raise ValueError("Error. Maximum width is too small for the input phrase.")  
  
 if alignment == 'center':  
 output\_lines = [line.center(max\_width) for line in output\_lines]  
 elif alignment == 'right':  
 output\_lines = [line.rjust(max\_width) for line in output\_lines]  
  
 colored\_output\_lines = [apply\_color(line, color) for line in output\_lines]  
  
 for line in colored\_output\_lines:  
 print(line)  
  
 return output\_lines  
  
  
def save\_to\_text\_file(output\_lines, output\_file\_name):  
 with open(output\_file\_name, 'w') as file:  
 for line in output\_lines:  
 file.write(line + '\n')

main.py:

from functions import \*  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 symbol\_set = {  
 'standart': "ascii\_art.txt",  
 '@': "ascii\_art\_dog.txt",  
 '#': "ascii\_art\_lattice.txt",  
 '\*': "ascii\_art\_star.txt",  
 }  
  
 while True:  
 selected\_symbol = input("Select a symbol (standart, @, #, \*): ")  
 if selected\_symbol not in symbol\_set:  
 print("Error! Invalid symbol.")  
 continue  
  
 file\_path = symbol\_set[selected\_symbol]  
 ascii\_dict = load\_ascii\_art(file\_path)  
  
 while True:  
 color = select\_color()  
  
 if color is None:  
 print("Error! Invalid color choice.")  
 continue  
  
 while True:  
 text = get\_valid\_text(ascii\_dict)  
  
 while True:  
 alignment = input("Select alignment (left, center, right): ").lower()  
 if alignment not in ['left', 'center', 'right']:  
 print("Error! Invalid alignment.")  
 else:  
 break  
  
 while True:  
 try:  
 max\_width = int(input("Enter maximum width: "))  
 if max\_width <= 0:  
 print("Error. Maximum width must be a positive integer.")  
 else:  
 break  
 except ValueError as e:  
 print(e)  
  
 while True:  
 try:  
 print("The result of ASCII art:")  
 output\_lines = print\_ascii\_art(ascii\_dict, text, alignment, max\_width, selected\_symbol, color)  
  
 save\_choice = input("Do you want to save the ASCII art to a text file? (yes/no): ").lower()  
 if save\_choice == "yes":  
 output\_file\_name = input("Enter the file name to save to: ")  
 save\_to\_text\_file(output\_lines, output\_file\_name)  
 print("ASCII art saved to", output\_file\_name)  
  
 continue\_choice = input("Do you want to continue drawing ASCII art? (yes/no): ").lower()  
 if continue\_choice != "yes":  
 exit() # Exit the program  
  
 break  
 except ValueError as e:  
 print(e)  
 max\_width = int(input("Enter maximum width: "))

**Результат програми:**

На рис.1, 2, 3 зображений результат виконання роботи у програмі.



Рис.1

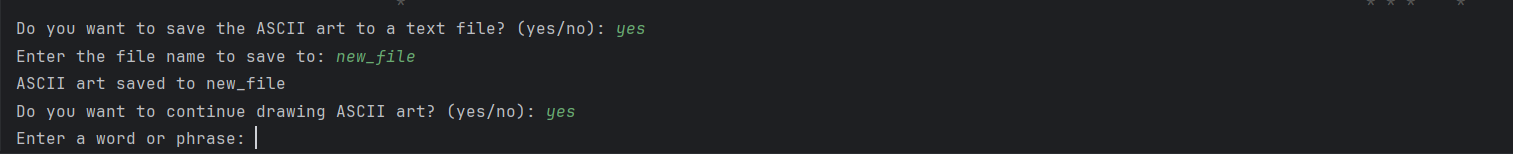


Рис.2

**Посилання на репозиторій**: <https://github.com/Oleksandr2004Oleniuk/SMP.git>

**Висновок**. у цій лабораторній роботі я створив генератор ASCII-арту з нуля, та надав можливість налаштовувати символи, розміри, вирівнювання та кольори, що дозволило мені глибше розібратися як створюється ASCII-арт.