МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем та мереж



Лабораторна робота 4

з дисципліни «Спеціалізовані мови програмування»

на тему «Розробка ASCII ART генератора для візуалізації 2D-фігур»

Виконав студент

групи РІ-32:

Фігура Т.Я

Прийняв:

Щербак С.С

Львів - 2024

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

**Хід виконання роботи:**

**Завдання 1:** Введення користувача

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

if choice == '1':  
 text = input("Enter text for ASCII Art: ") *#Завдання 1: Введення користувача*  
generator = ASCIIArtGenerator(text, selected\_font, selected\_color)  
 art = generator.generate\_art()  
 print(art)

**Завдання 2:** Набір символів

Визначте набір символів (наприклад, '@', '#', '\*', тощо), які будуть використовуватися для створення ASCII-арту. Ці символи будуть відображати різні відтінки.

art\_lines = [''] \* len(next(iter(self.font.values()))) *# Порожні рядки для рядків шрифту*for char in self.text:  
 if char in self.font:  
 char\_lines = self.font[char]  
 for i, line in enumerate(char\_lines):  
 art\_lines[i] += line.replace('#', self.symbol) + ' '  
 else:  
 for i in range(len(art\_lines)):  
 art\_lines[i] += ' ' \* 6 *# Пропуск для невідомих символів*return '\n'.join(art\_lines)

**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

def \_\_init\_\_(self, text, symbol='\*', color='', width=80, height=10, alignment='left'):  
 self.text = text.upper()  
 self.symbol = symbol  
 self.color = color  
 self.width = width  
 self.height = height  
 self.alignment = alignment  
 self.font = get\_font()

**Завдання 4:** Колір тексту

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

def apply\_color(text, color='white'): *#Завдання 4: Колір тексту*  
if is\_color\_supported():  
 try:  
 return colored(text, color)  
 except KeyError:  
 return colored(text, 'white')  
 else:  
 return text *# Якщо кольори не підтримуються, просто повертаємо текст без кольору*

**Завдання 5:** Вирівнювання тексту

Реалізуйте опції вирівнювання тексту (ліво, центр, право), щоб користувачі могли вибирати, як їх ASCII-арт розміщується на екрані.

def align\_text(art, alignment='center', width=80):  
 if alignment == 'left':  
 return [line.ljust(width) for line in art]  
 elif alignment == 'right':  
 return [line.rjust(width) for line in art]  
 else: *# center* return [line.center(width) for line in art]

**Завдання 6:** Збереження у файл

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

def save\_to\_file(self, file\_path, art): *#Завдання 6: Збереження у файл*  
with open(file\_path, 'w') as f:  
 f.write(art)

**Завдання 7:** Розмір ARTу

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

def generate\_art\_with\_size(self, width): *#Завдання 7: Розмір ARTу*  
figlet = pyfiglet.Figlet(font=self.font, width=width)  
 ascii\_art = figlet.renderText(self.text)  
 return self.apply\_color(ascii\_art, self.color)

**Завдання 8:** Вибір символів

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

symbols = input("Enter symbols to use (e.g., '@#\*'): ")  
ascii\_art = generator.generate\_with\_symbols(text, symbols)

**Завдання 9:** Функція попереднього перегляду

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

def preview\_art(generator): *#Завдання 9: Функція попереднього перегляду*  
art = generator.generate\_art()  
 print(art)

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

Створіть інтерфейс для користувача у командному рядку, щоб зробити програму легкою та інтуїтивно зрозумілою для використання.

def menu():  
 text = ''  
 symbol = '\*'  
 color = 'reset'  
 width, height = 80, 10  
 alignment = 'left'  
  
 while True:  
 print("\n1. Enter Text for ASCII Art")  
 print("2. Choose Symbol for ASCII Art")  
 print("3. Choose Text Color")  
 print("4. Set Width and Height")  
 print("5. Set Text Alignment")  
 print("6. Preview ASCII Art")  
 print("7. Save ASCII Art to File")  
 print("8. Exit")  
  
 choice = input("Select an option: ")  
  
 if choice == '1':  
 text = input("Enter the text: ")  
  
 elif choice == '2':  
 symbol = input("Enter the symbol to use (e.g., '@', '#', '\*'): ")  
  
 elif choice == '3':  
 print("Available colors: red, green, blue")  
 color = input("Enter the color: ").lower()  
  
 elif choice == '4':  
 width = int(input("Enter the width (max 100): "))  
 height = int(input("Enter the height (max 20): "))  
 width = min(max(10, width), 100)  
 height = min(max(5, height), 20)  
  
 elif choice == '5':  
 print("Alignment options: left, center, right")  
 alignment = input("Choose alignment: ").lower()  
  
 elif choice == '6':  
 if text:  
 generator = ASCIIArtGenerator(text, symbol, color, width, height, alignment)  
 art = generator.generate\_art()  
 print("\nPreview of ASCII Art:\n")  
 print(generator.apply\_color(art))  
 else:  
 print("Please enter text first.")  
  
 elif choice == '7':  
 if text:  
 file\_path = input("Enter file name (with .txt extension): ")  
 generator = ASCIIArtGenerator(text, symbol, color, width, height, alignment)  
 generator.save\_to\_file(file\_path)  
 else:  
 print("Please enter text first.")  
  
 elif choice == '8':  
 print("Goodbye!")  
 break  
  
 else:  
 print("Invalid choice. Please try again.")

**Посилання на GitHub:** [***https://github.com/FihuraTaras/SPL\_Python\_Project.git***](https://github.com/FihuraTaras/SPL_Python_Project.git)

**Висновок:** навчився створювати універсальний Генератор ASCII-арту, який дозволив налаштовувати свої творіння з створеної бібліотеки шрифтів, кольорами, розмірами та символами. Проект дав практичний досвід роботи з введенням користувача, зовнішніми бібліотеками, роботою з файлами та дизайном інтерфейсу користувача в Python.