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

Виконала:

ст. гр. РІ-31

Оксана ЛЕСЮК

Прийняв:

Сергій Сергійович ЩЕРБАК

Львів-2024

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

**Завдання:**

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.

Виконуючи ці завдання, студенти мають створити генератор ASCII-арту з нуля, та надати можливість налаштовувати символи, розміри, вирівнювання та кольори, що дозволить їм глибше розібратися як створюється ASCII-арт

**Хід роботи**

**AsciiArtGenerator.py:**

class AsciiArtGenerator:

"""Main class for generating ASCII art."""

def \_\_init\_\_(self, settings, language):

self.settings = settings

self.language = language

self.font = font8x8

def \_str\_to\_ascii\_list(self, text):

"""Convert a string to a list of ASCII or font indices."""

return [ord(char) for char in text] if self.language == 'en' else self.\_map\_ukr\_to\_font(text)

@staticmethod

def \_map\_ukr\_to\_font(text):

"""Map Ukrainian characters to font indices."""

ukr\_map = {'І': 0, 'і': 0, 'Д': 1, 'д': 1}

return [ukr\_map.get(char, ord(char)) for char in text]

@staticmethod

def \_add\_shadow(row):

"""Add shadow to a row by replacing spaces with shadow symbols."""

return [SHADOW if row[i] == SYMBOL and row[i + 1] == SPACE else row[i] for i in range(len(row) - 1)] + [row[-1]]

@staticmethod

def \_row\_to\_string(row, regular\_symbol, shadow\_symbol):

"""Convert row data to a string representation."""

return ''.join(regular\_symbol if item == SYMBOL else shadow\_symbol if item == SHADOW else ' ' for item in row)

@staticmethod

def \_align(alignment, width, row\_length):

"""Calculate left padding for alignment."""

if alignment == 'center':

return (width - row\_length) // 2

if alignment == 'right':

return width - row\_length

return 0

def render(self, text):

"""Generate ASCII art for the given text."""

char\_list = self.\_str\_to\_ascii\_list(text)

width, height = self.settings.size

lines = [char\_list[i:i + width // 8] for i in range(0, len(char\_list), width // 8)]

art = ""

for line in lines:

for row\_index in range(8):

row = []

for char\_index in line:

bitmap = self.font[char\_index]

row\_data = [(bitmap[row\_index] >> bit) & 1 for bit in range(8)]

if self.settings.is\_3d:

row\_data = [SPACE] \* row\_index + row\_data

if self.settings.symbols[1]:

row\_data = self.\_add\_shadow(row\_data)

row.extend(row\_data)

row\_str = self.\_row\_to\_string(row, \*self.settings.symbols)

padding = self.\_align(self.settings.alignment, width, len(row\_str))

art += ' ' \* padding + row\_str + '\n'

return self.settings.color + art

**AsciiArtSettings.py**:

class AsciiArtSettings:

"""Class to store settings for ASCII art generation."""

def \_\_init\_\_(self, color, symbols, size, alignment, is\_3d):

self.color = color

self.symbols = symbols # (regular\_symbol, shadow\_symbol)

self.size = size # (width, height)

self.alignment = alignment # 'left', 'center', 'right'

self.is\_3d = is\_3d # True or False

**Посилання на гіт-репозиторій:**

https://github.com/lesiukoksana/SMP-labs

**Висновок:**

У ході виконання лабораторної роботи було розроблено генератор ASCII-арту на Python, який дозволяє користувачам перетворювати текст у зображення, використовуючи набір символів для різних відтінків, а також дозволяє налаштовувати розміри, вирівнювання, кольори та інші параметри.