Міністерство освіти і науки України

Національний університет «Львівська політехніка»

лабораторної роботи №4

з дисципліни «Спеціалізовані мови програмування»

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

Виконав:

Кулявець В. Р.

Перевірив:

Щербак С.С.

Львів 2024

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

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

Завдання 1: Введення користувача

Створіть програму Python, яка отримує введення користувача щодо слова або фрази, яку вони хочуть перетворити в ASCII-арт.

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

user\_input = input()

Завдання 2: Набір символів

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

DENSITY\_MAP = " .:-=+\*#%@"

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

while True:

try:

width = int(input("Enter the desired width of the ASCII art (between 20 and 200): "))

if 20 <= width <= 200:

break

else:

print("Please enter a width between 20 and 200.")

except ValueError:

print("Invalid input. Please enter a numeric value.")

while True:

try:

height = int(input("Enter the desired height of the ASCII art (between 5 and 50): "))

if 5 <= height <= 50:

break

else:

print("Please enter a height between 5 and 50.")

except ValueError:

print("Invalid input. Please enter a numeric value.")

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.

def generate\_art(self, text):

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

# Limiting the height of the ASCII art

ascii\_art\_lines = ascii\_art.split('\n')[:self.height]

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

Завдання 5: Вирівнювання тексту

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

while True:

alignment = input("Enter text alignment (left, center, right): ").lower()

if alignment in ["left", "center", "right"]:

break

else:

print("Invalid alignment. Please enter 'left', 'center', or 'right'.")

Завдання 6: Відображення мистецтва

Відобразіть створений ASCII-арт на екрані за допомогою стандартних функцій друку Python.

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

Завдання 7: Збереження у файл

Додайте можливість зберігати створений ASCII-арт у текстовий файл, щоб користувачі могли легко завантажувати та обмінюватися своїми творіннями.

def save\_ascii\_art\_to\_file(art):

save\_option = input("Would you like to save the ASCII art to a text file? (yes/no): ").lower()

if save\_option == "yes":

file\_name = input("Enter the filename (without extension): ") + ".txt"

try:

with open(file\_name, 'w') as file:

file.write(art)

print(f"ASCII art successfully saved to {file\_name}")

except Exception as e:

print(f"An error occurred while saving the file: {e}")

Завдання 8: Варіанти кольорів

Дозвольте користувачам вибирати опції кольорів (чорно-білий, відтінки сірого) для свого ASCII-арту.

def map\_char\_to\_custom\_set(char, color\_symbols, color\_option):

if color\_option == "grayscale":

if char in DENSITY\_MAP:

index = DENSITY\_MAP.index(char) \* (len(color\_symbols['grayscale']) - 1) // (len(DENSITY\_MAP) - 1)

return color\_symbols['grayscale'][index]

elif color\_option == "black\_and\_white":

if char in '@#\*%':

return color\_symbols['black'][0]

elif char in "=-:. ":

return color\_symbols['white'][1]

else:

return color\_symbols['white'][0]

return char

Завдання 9: Функція попереднього перегляду

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

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

Завдання 10: Інтерфейс, зрозумілий для користувача

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

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: Створив Генератор ASCII-арту без використання зовнішніх бібліотек.