МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

**Лабораторна робота №5**

**з дисципліни**

**СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ**

**на тему**

**«Розробка ASCII ART генератора для візуалізації 3D-фігур»**

Виконав:

ст. гр. ІТ-21сп

Одноріг Д.І.

Прийняв:

Щербак С.С.

Львів-2023

**Мета роботи:** Cтворення додатка для малювання 3D-фігур у ASCII-арті на основі об’єктно - орієнтованого підходу та мови Python

**Хід роботи**

**Завдання 1:** Проектування класів

Розробіть структуру класів для вашого генератора 3D ASCII-арту. Визначте основні компоненти, атрибути та методи, необхідні для програми.

**Завдання 2:** Введення користувача

Створіть методи у межах класу для введення користувача та вказання 3D-фігури, яку вони хочуть намалювати, та її параметрів (наприклад, розмір, кольори).

**Завдання 3:** Представлення фігури

Визначте структури даних у межах класу для представлення 3D-фігури. Це може включати використання списків, матриць або інших структур даних для зберігання форми фігури та її властивостей.

**Завдання 4:** Проектування з 3D в 2D

Реалізуйте метод, який перетворює 3D-представлення фігури у 2D-представлення, придатне для ASCII-арту.

**Завдання 5:** Відображення ASCII-арту

Напишіть метод у межах класу для відображення 2D-представлення 3D-фігури як ASCII-арту. Це може включати відображення кольорів і форми за допомогою символів ASCII.

**Завдання 6:** Інтерфейс, зрозумілий для користувача

Створіть зручний для користувача командний рядок або графічний інтерфейс користувача (GUI) за допомогою об'єктно-орієнтованих принципів, щоб дозволити користувачам спілкуватися з програмою.

**Завдання 7:** Маніпуляція фігурою

Реалізуйте методи для маніпулювання 3D-фігурою, такі масштабування або зміщення, щоб надавати користувачам контроль над її виглядом.

**Завдання 8:** Варіанти кольорів

Дозвольте користувачам вибирати варіанти кольорів для їхніх 3D ASCII-арт-фігур. Реалізуйте методи для призначення кольорів різним частинам фігури.

**Завдання 9:** Збереження та експорт

Додайте функціональність для зберігання згенерованого 3D ASCII-арту у текстовий файл

**Завдання 10:** Розширені функції

Розгляньте можливість додавання розширених функцій, таких як тінь, освітлення та ефекти перспективи, для підвищення реалізму 3D ASCII-арту.

Код програми:

cube.py:

from \_\_future\_\_ import division

from colorama import init, Fore, Style

init(autoreset=True)

class CubeDrawer:

def \_\_init\_\_(self, x\_size=1, y\_size=1, z\_size=1, color=Fore.WHITE):

self.x\_size = x\_size

self.y\_size = y\_size

self.z\_size = z\_size

self.color = color

self.width = x\_size \* 2

self.height = y\_size \* 2

self.cube = [[' ']\*self.width for \_ in range(self.height)]

def draw\_cube(self):

vertices = {

'tc': (self.width//2, 0),

'tl': (0, int(.25\*self.height)),

'tr': (self.width-1, int(.25\*self.height)),

'cc': (self.width//2, self.height//2),

'bl': (0, int(.75\*self.height)),

'br': (self.width-1, int(.75\*self.height)),

'bc': (self.width//2, self.height-1)

}

edges = (

('tc', 'tl'),

('tc', 'tr'),

('tl', 'cc'),

('tl', 'bl'),

('tr', 'cc'),

('tr', 'br'),

('bl', 'bc'),

('br', 'bc'),

('cc', 'bc')

)

for edge in edges:

v1 = vertices[edge[0]]

v2 = vertices[edge[1]]

x1 = v1[0]

y1 = v1[1]

x2 = v2[0]

y2 = v2[1]

if x1 > x2:

x1, x2 = x2, x1

y1, y2 = y2, y1

try:

m = (y2-y1)/(x2-x1)

except ZeroDivisionError:

c = '|'

for yy in range(min(y1, y2), max(y1, y2)+1):

self.cube[yy][x1] = self.color + c

else:

c = '.'

yy = y1

for xx in range(x1, x2+1):

self.cube[int(yy)][xx] = c

yy += m

if edge[0] in ('tc', 'tr', 'tl') and edge[1] in ('tc', 'tr', 'tl'):

self.cube[y1][x1] = c

cube\_str = '\n'.join(''.join(row) for row in self.cube)

return cube\_str

def draw\_square(self, length):

square\_width = int(length \* 2.5)

square = [[' ']\*square\_width for \_ in range(length)]

for i in range(length):

for j in range(square\_width):

if i == 0 or i == length-1:

square[i][j] = self.color + '-'

elif j == 0 or j == square\_width-1:

square[i][j] = self.color + '|'

square\_str = '\n'.join(''.join(row) for row in square)

return square\_str

def remove\_color\_codes(self, text):

while '\033[' in text:

start = text.find('\033[')

end = text.find('m', start)

if end != -1:

text = text[:start] + text[end+1:]

else:

break

return text

def save\_to\_file(self, content):

filename = input("Enter the file name to save: ")

with open(filename, 'w') as file:

file.write(self.remove\_color\_codes(content))

main.py:

from \_\_future\_\_ import division

from colorama import Fore

from Lab5.cube import CubeDrawer

def main():

try:

x\_size = int(input("Enter the cube size along the x-axis: "))

y\_size = int(input("Enter the cube size along the y-axis: "))

z\_size = int(input("Enter the cube size along the z-axis: "))

except ValueError:

print("Invalid input format. Using default sizes (1, 1, 1).")

x\_size, y\_size, z\_size = 1, 1, 1

selected\_color = input("Enter the color code (e.g., RED, GREEN, BLUE): ")

try:

color\_code = getattr(Fore, selected\_color.upper())

except AttributeError:

print("Invalid color code. Using white color by default.")

color\_code = Fore.WHITE

cube\_drawer = CubeDrawer(x\_size, y\_size, z\_size, color\_code)

cube\_str = cube\_drawer.draw\_cube()

print(cube\_str)

save\_cube = input("Do you want to save the cube to a file? (Yes/No): ").lower()

if save\_cube == "yes":

cube\_drawer.save\_to\_file(cube\_str)

generate\_square = input("Do you want to generate a square? (Yes/No): ").lower()

if generate\_square == "yes":

square\_str = cube\_drawer.draw\_square(y\_size)

print(square\_str)

save\_square = input("Do you want to save the square to a file? (Yes/No): ").lower()

if save\_square == "yes":

cube\_drawer.save\_to\_file(square\_str)

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

main()

Файл cube.py:

Цей файл містить клас CubeDrawer, який представляє собою 3D ASCII арт-генератор для куба та квадрата. Основні функції класу включають:

* draw\_cube(): Генерує ASCII-арт для куба з використанням ASCII-графіки та кольорів з бібліотеки colorama.
* draw\_square(length): Генерує ASCII-арт для квадрата зазначеного розміру, використовуючи ASCII-графіку та кольори.
* remove\_color\_codes(text): Видаляє коди кольорів з тексту, щоб отримати чистий ASCII-арт.
* save\_to\_file(content): Запитує у користувача ім'я файлу та зберігає ASCII-арт у вказаному файлі без кодів кольорів.

Файл main.py:

Цей файл використовує клас CubeDrawer з cube.py для створення та відображення 3D ASCII арт-куба:

* Запитує користувача про розміри куба та вибір кольору.
* Викликає метод draw\_cube() класу CubeDrawer для генерації ASCII-арт для куба.
* Відображає ASCII-арт куба у консолі.
* Запитує користувача, чи він хоче зберегти куб у файл.
* Якщо користувач бажає зберегти арт у файл, то програма запитує ім'я файлу та зберігає ASCII-арт куба у вказаному файлі.

На рис. 1 зображено генерацію куба.

*Зображення, що містить текст, знімок екрана, схема

Автоматично згенерований опис*

*Рис. 1. Результат генерації куба.*

Користувач також має змогу згенерувати проекцію грані куба, розміри якої відповідають розмірам грані згенерованого куба по осі y.

На рис 2 зображено проекцію грані куба.

*Зображення, що містить текст, знімок екрана

Автоматично згенерований опис*

*Рис. 2. Результат проекції грані куба.*

Усі згенеровані арти можна зберегти у текстовий файл.

На рис. 3, 4 зображено збережені фігури:

Зображення, що містить знімок екрана, текст, Графічний редактор

Автоматично згенерований опис

*Рис. 3. Збережена фігура куба у текстовому файлі.*

*Зображення, що містить знімок екрана, Прямокутник, квадрат

Автоматично згенерований опис*

*Рис. 4. Збережена фігура проекції грані куба у текстовому файлі.*

**Висновок:** виконуючи дану лабораторну роботу, я створив високорівневий об'єктно-орієнтований генератор 3D ASCII-арту, який дозволить користувачам проектувати, відображати та маніпулювати 3D-фігурами в ASCII-арті.