Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Лабораторна роботи №5

з дисципліни «Спеціалізовані мови програмування»

на тему

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

Виконав:

Ложовський В. А.

Перевірив:

Щербак С. С.

Львів 2024

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

**План 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.

**Реалізація:**

**Папка utils, файл shapes.py:**  
class Shape:

    def \_\_init\_\_(self, size):

        self.size = size

        self.color = None

    def draw(self):

        raise NotImplementedError("This method should be overridden by subclasses")

    def set\_color(self, color):

        self.color = color

class Cube(Shape):

    def draw(self):

        s = self.size

        cube = [

            f"  \_\_\_\_{'\_' \* s}",

            f" /\_\_\_/|{'\_' \* s}|",

            f"|   |{'|' \* s}|",

            f"|\_\_\_|{'|' \* s}/",

            f"    {'|' \* s}/"

        ]

        return "\n".join(cube)

class Sphere(Shape):

    def draw(self):

        s = self.size

        sphere = [

            f"  ,{'\_' \* s},  ",

            f" /{'   ' \* s}\\ ",

            f"(  {'O' \* s}  )",

            f" \\{'   ' \* s}/ ",

            f"  `{'-' \* s}`  "

        ]

        return "\n".join(sphere)

class AdvancedShape(Shape):

    def draw(self):

        return "Advanced representation of the shape"

**Файл main.py:**

from lab5.input.user\_input import get\_shape\_input

from lab5.utils.shapes import Cube, Sphere

from lab5.utils.renderer import Renderer

from lab5.utils.manipulator import Manipulator

from lab5.colors.color import Color

from lab5.output.file\_handler import save\_art\_to\_file

def main():

    shape\_type, size = get\_shape\_input()

    if shape\_type == "cube":

        shape = Cube(size)

    elif shape\_type == "sphere":

        shape = Sphere(size)

    else:

        print("Invalid shape type!")

        return

    color = Color("white")  # Default color

    shape.set\_color(color)

    ascii\_art = Renderer.render(shape)

    print(ascii\_art)

    Manipulator.scale(shape, 2)

    print("Scaled shape:")

    ascii\_art = Renderer.render(shape)

    print(ascii\_art)

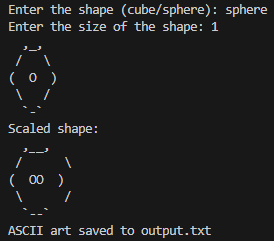
    save\_art\_to\_file(ascii\_art, "lab5/output.txt")

    print("ASCII art saved to output.txt")

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

    main()

**Результат виконання:**

****

**Висновок:** на цій лабораторній роботі я створив високорівневий об'єктно-орієнтований генератор 3D ASCII-арту, який дозволить користувачам проектувати, відображати та маніпулювати 3D-фігурами в ASCII-арті