

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж

Лабораторна робота №5
з дисципліни
СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ
на тему
Розробка ASCII ART генератора для візуалізації 3D-фігур

Виконав:
ст. гр. РІ-21сп
Рак В.П.
Прийняв:
Щербак С.С.

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

Завдання

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.

Виконання роботи

Текст програмної реалізації:

"""

Основна програма для рендерингу 3D куба та взаємодії з користувачем через командний інтерфейс.

Програма дозволяє користувачу виконувати операції над кубом, такі як обертання, зміна кольору, масштабування та збереження куба в ASCII-форматі в файл.

Імпортуються модулі:

- `Cube`: для створення та маніпуляцій з кубом.
- `Renderer`: для проектування та рендерингу куба.
- `UserInput`: для отримання введення від користувача.
- `os`: для очищення консолі.
- `time`: для затримки між кадрами.

Основні етапи роботи:

1. Ініціалізація куба з початковими розмірами.
2. Використання рендерера для відображення куба.
3. Взаємодія з користувачем для виконання операцій.
4. Оновлення зображення куба після кожної дії користувача.
5. Можливість зберегти куб у файл або завершити програму.

Функція:

- `main()`: основна функція, яка керує всім процесом: отримує введення користувача, керує обертанням куба, його кольором, масштабом та збереженням файлу.

"""

```
import os
```

```
import time
```

```
from models.cube import Cube
```

```
from render.renderer import Renderer
```

```
from controllers.user_input import UserInput
```

```
def main():
```

```
    user_input = UserInput(None)
```

```
    # Отримання початкових розмірів куба
```

```
    size_x, size_y, size_z = user_input.get_initial_size()
```

```
    cube = Cube(size_x, size_y, size_z)
```

```
    user_input.cube = cube # Прив'яжемо куб до вводу користувача
```

```
    renderer = Renderer(resolution=40, foco=40, y_distorter=1.1, left_right=1.5, up_down=0.5)
```

```
    try:
```

```
        while True:
```

```
            os.system('cls' if os.name == 'nt' else 'clear')
```

```
            rotated_cube = cube.rotate()
```

```
            projection = renderer.project(rotated_cube)
```

```
            lines = renderer.get_lines(projection)
```

```
            rendered_ascii = renderer.render(projection, lines, cube.color)
```

```
            time.sleep(0.5)
```

```
            action = user_input.get_next_action()
```

```
            if action == "1":
```

```
                user_input.get_rotation_input()
```

```
            elif action == "2":
```

```
                user_input.change_color()
```

```
            elif action == "3":
```

```
                user_input.get_scale_input()
```

```
            elif action == "4":
```

```
                filename = input("Enter filename to save the cube: ")
```

```
                cube.save_to_file(filename, rendered_ascii)
```

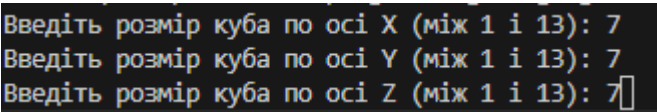
```
            elif action == "5":
```

```
                print("Exiting...")
```

```
        break
    else:
        print("Invalid choice. Try again.")
except KeyboardInterrupt:
    print("Program interrupted.")

if __name__ == "__main__":
    main()
```

Результат роботи програми:



```
Введіть розмір куба по осі X (між 1 і 13): 7
Введіть розмір куба по осі Y (між 1 і 13): 7
Введіть розмір куба по осі Z (між 1 і 13): 7
```

Рис. 1 - Параметри для куба

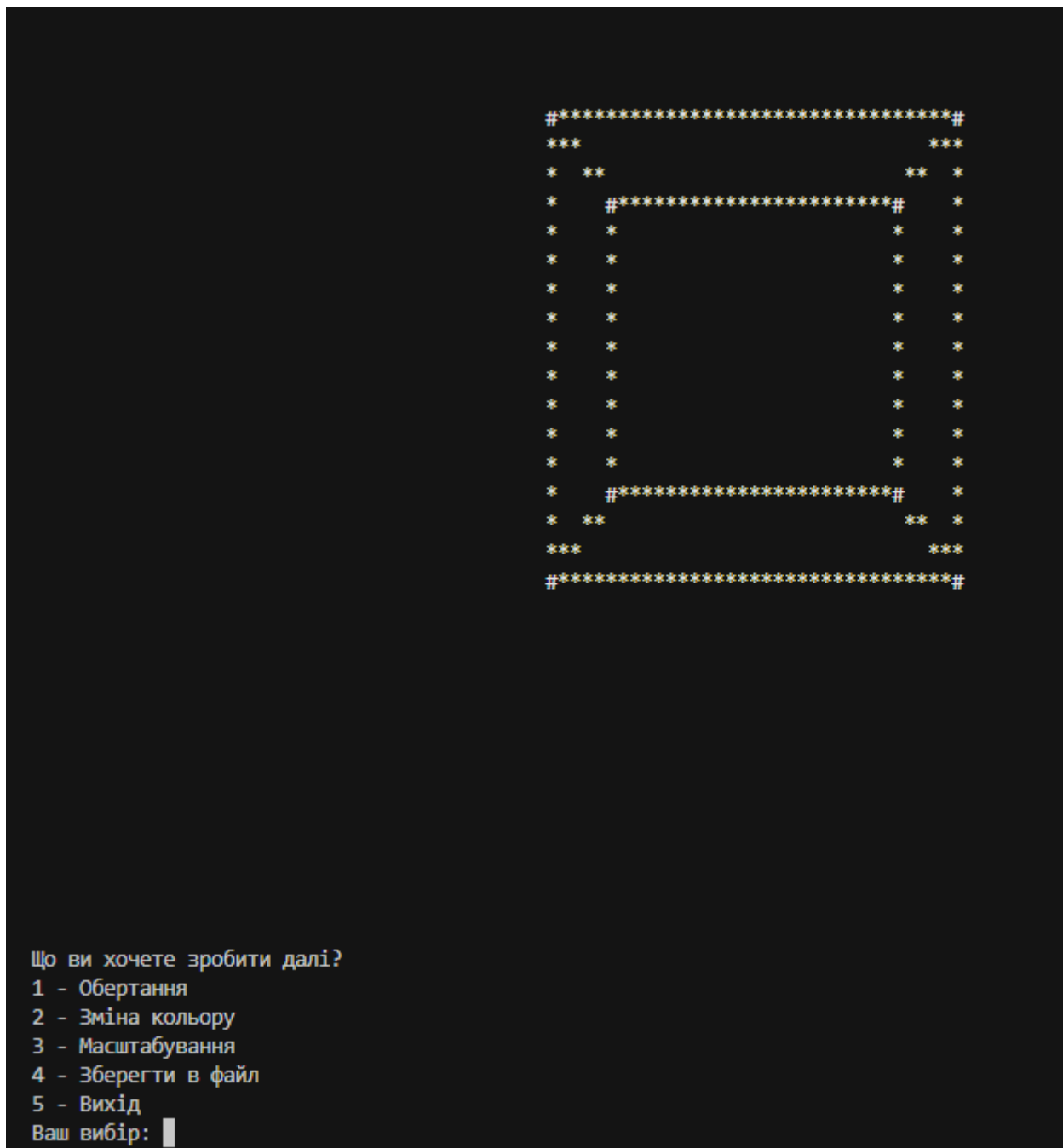


Рис. 2 – Згенерований куб

Висновок: У ході виконання лабораторної роботи я створив високорівневий об'єктно-орієнтований генератор 3D ASCII-арту, який дозволяє користувачам проектувати, відображати та маніпулювати 3D-фігурами в ASCII-арті, а також зберігати їх у файлах.