**Міністерство освіти і науки України**

**Національний технічний університет України «КПІ» імені Ігоря Сікорського**

**Кафедра обчислювальної техніки ФІОТ**

**ЗВІТ**

**з лабораторної роботи No1**

**з навчальної дисципліни «Технології Computer Vision»**

**Тема:**

**ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ПОБУДОВИ ТА ПЕРЕТВОРЕННЯ КООРДИНАТПЛОЩИННИХ (2D) ТА ПРОСТОРОВИХ (3D) ОБ’ЄКТІВ**

**Виконала:**

Студентка 3 курсу кафедри ОТ ФІОТ,

Навчальної групи ІМ-24

Іванова Дар'я

**Перевірив:**

Професор кафедри ОТ ФІОТ

Олексій Писарчук

**Київ 2024**

**І. Мета роботи:**

Виявити дослідити та узагальнити особливості формування та перетворення координат площинних (2d) та просторових (3d) об’єктів.

**ІІ. Завдання:**

Лабораторія провідної ІТ-компанії реалізує масштабний проект розробки

універсальної платформи з цифрової обробки зображень для задач Computer Vision.

Платформа передбачає розташування back-end компоненти на власному хмарному сервері

з наданням повноважень користувачам заздалегідь адаптованого front-end функціоналу

універсальної платформи. Цим формується унікальна для потреб замовника ERP система

з технологіями Computer Vision

Замовниками ресурсів платформи є: державні та комерційні компанії, що

розробляють медичне обладнання з діагностування захворювань за візуальною

інформацією; автоматизації аграрного бізнесу в аспекті обліку посівних територій за

даними з БПЛА; візуального контролю безпекових заходів на об’єктах критичної

інфраструктури: аеропорти, торгівельно-розважальні центри, житлові комплекси тощо.

Вам, як Computer Vision поставлено завдання.

|  |  |  |
| --- | --- | --- |
| День народження | Завдання І рівня складності – максимально 7 балів. | Фігура |
| 4 | Реалізувати операції: обертання – переміщення – масштабування.  3. операцію реалізувати циклічно, траєкторію зміни положення цієї операції відобразити. Обрати самостійно: бібліотеку, розмір графічного вікна, розмір фігури, параметри реалізації операцій, кольорову гамму усіх графічних об’єктів. Всі операції перетворень мають здійснюватись у межах графічного вікна. | ромб |

Завдання І рівня складності **– максимально 7 балів.**

Здійснити синтез математичних моделей та розробити програмний скрипт, що

реалізує базові операції 2D перетворень над геометричними примітивами. Для розробки

використовувати матричні операції та технології композиційних перетворень. Вхідна

матриця координат кутів геометричної фігури має бути розширеною.

Функціонал скрипта, що розробляється має реалізувати технічних вимог табл.1

Додатку 1.

### ІІІ. Результати виконання лабораторної роботи. **5.3. Результати виконання лабораторної роботи**

**5.3.1. Синтезована математична модель**

Для реалізації перетворень (обертання, переміщення, масштабування) ромба використано матричні операції. Математичні моделі для кожного з перетворень:

* **Обертання:**
* **Переміщення:**
* **Масштабування:**

**5.3.2. Результати архітектурного проектування та їх опис**

Архітектура проекту включає наступні основні модулі:

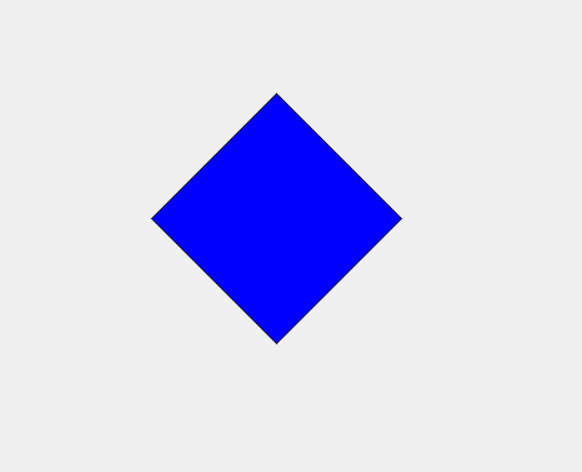
1. **Головний модуль** – ініціалізація графічного вікна, виклик функцій перетворень.
2. **Функція відображення ромба** – створення об'єкта ромба та його малювання.
3. **Функції трансформацій**:
   * rotate\_shape(win, obj): обертання об'єкта ромба.
   * move\_shape(win): переміщення об'єкта ромба.
   * scale\_shape(win): масштабування об'єкта ромба.

**5.3.3. Опис структури проекту програми**

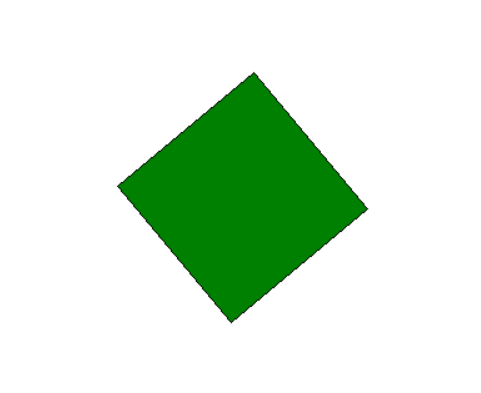
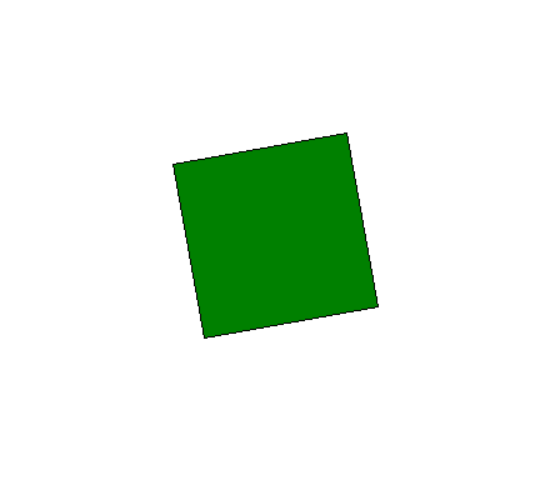
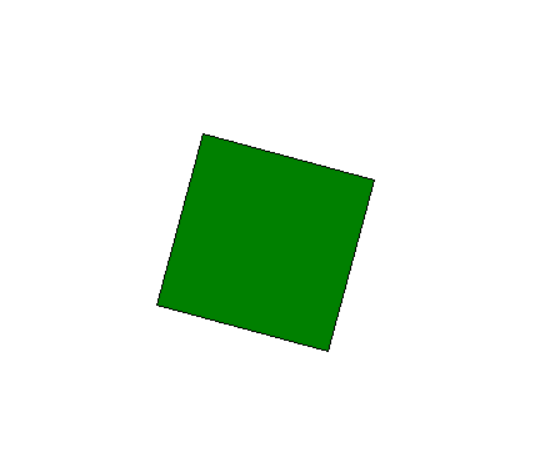
Структура проекту складається з таких компонентів:

* **Основний файл transformation.py**:
  + Імпортування необхідних бібліотек (graphics, numpy, math)
  + Оголошення констант розміру вікна та параметрів ромба
  + Функції:
    - draw\_initial\_shape()
    - rotate\_shape()
    - move\_shape()
    - scale\_shape()
  + Основний цикл виконання програми

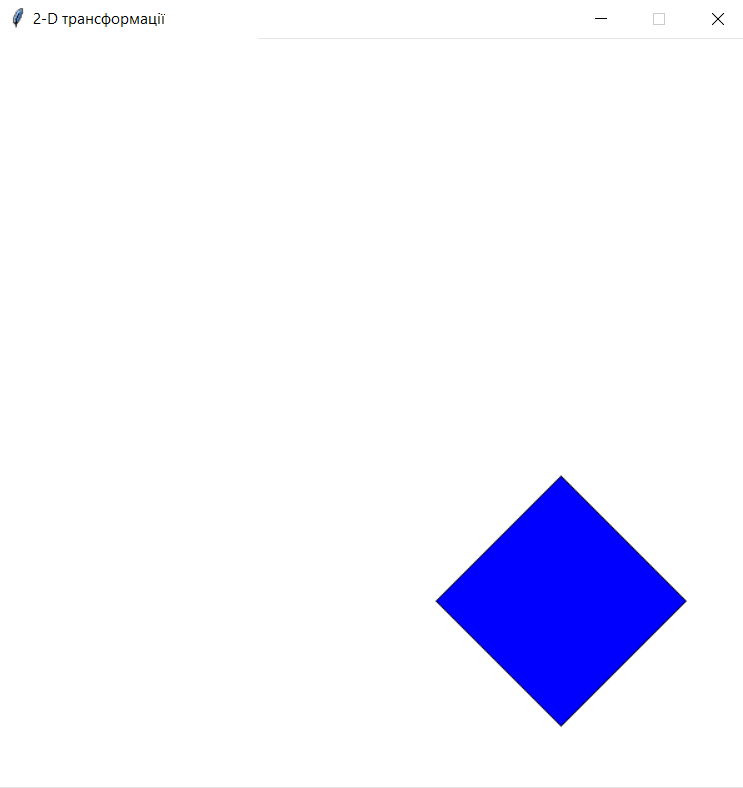
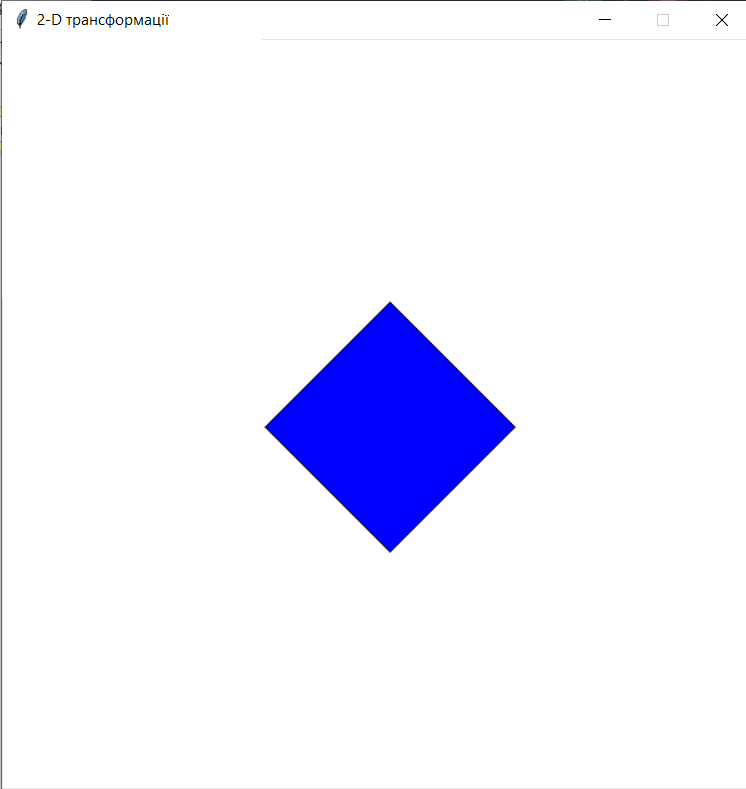
**5.3.4. Результати роботи програми відповідно до завдання**



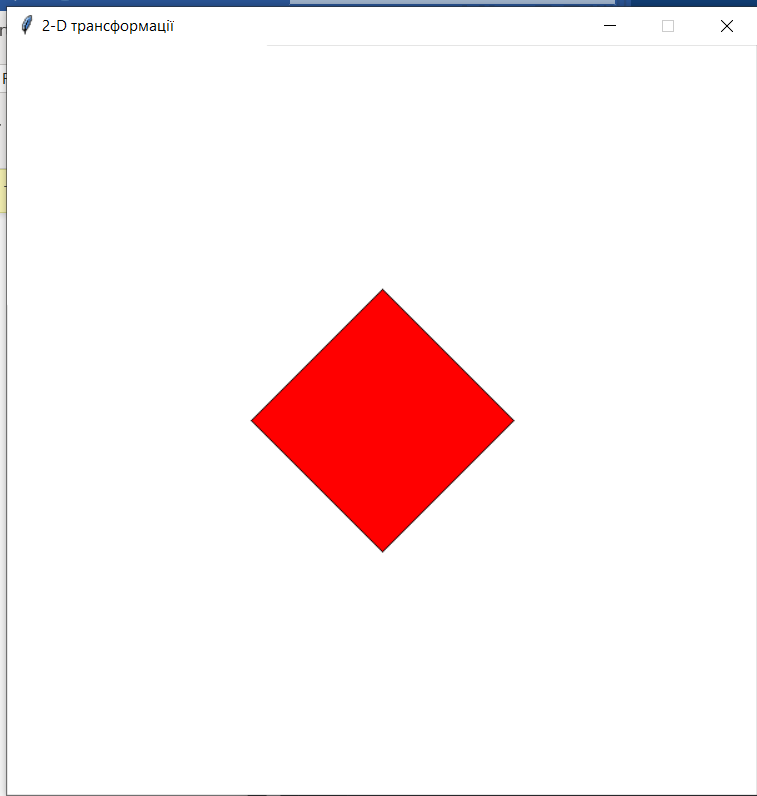
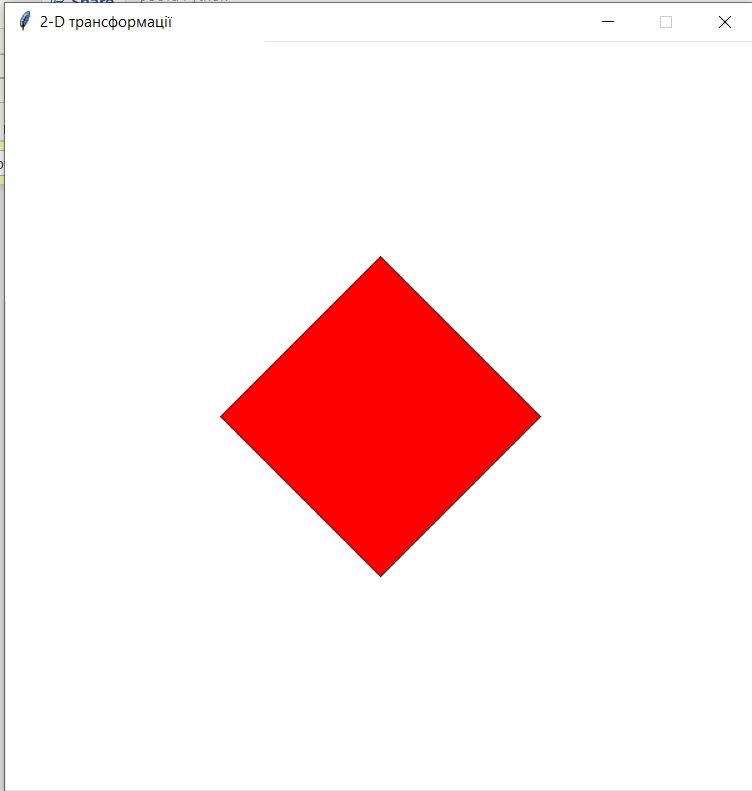
* rotate\_shape(win, obj): обертання об'єкта ромба.



* move\_shape(win): переміщення об'єкта ромба.



* scale\_shape(win): масштабування об'єкта ромба.



**5.3.5. Програмний код, що забезпечує отримання результату**

*from* graphics *import* \*  
*import* time  
*import* numpy *as* np  
*import* math *as* mt  
  
*#---------------- Формування та відображення статичного ромба ------------------------*xw = 600; yw = 600; st = 100 *# Розміри графічного вікна та параметри перетворень  
  
# Розміри ромба*x1 = xw//2; y1 = yw//2 - st  
x2 = xw//2 + st; y2 = yw//2  
x3 = xw//2; y3 = yw//2 + st  
x4 = xw//2 - st; y4 = yw//2  
  
*def* draw\_initial\_shape(win):  
 obj = Polygon(Point(x1, y1), Point(x2, y2), Point(x3, y3), Point(x4, y4))  
 obj.setFill('blue')  
 obj.draw(win)  
 *return* obj  
  
*#------------------------- Циклічне обертання ромба ------------------------  
  
def* rotate\_shape(win, obj):  
 win.setBackground('white')  
 coords = np.array([[x1, y1, 1], [x2, y2, 1], [x3, y3, 1], [x4, y4, 1]])  
 center\_x, center\_y = xw // 2, yw // 2  
  
 *def* rotate(coords, angle, center):  
 angle = mt.radians(angle)  
 cx, cy = center  
 transformation\_matrix = np.array([  
 [mt.cos(angle), -mt.sin(angle), 0],  
 [mt.sin(angle), mt.cos(angle), 0],  
 [0, 0, 1]  
 ])  
 translated\_coords = coords - np.array([cx, cy, 0])  
 rotated\_coords = translated\_coords.dot(transformation\_matrix.T)  
 final\_coords = rotated\_coords + np.array([cx, cy, 0])  
 *return* final\_coords  
  
 *for* \_ *in* range(72):  
 *if* win.checkMouse():  
 obj.undraw()  
 *return* time.sleep(0.1)  
 obj.undraw()  
 coords = rotate(coords, 5, (center\_x, center\_y))  
 obj = Polygon(Point(coords[0, 0], coords[0, 1]),  
 Point(coords[1, 0], coords[1, 1]),  
 Point(coords[2, 0], coords[2, 1]),  
 Point(coords[3, 0], coords[3, 1]))  
 obj.setFill('green')  
 obj.draw(win)  
 obj.undraw()  
  
*#------------------------- Переміщення ромба ------------------------  
  
def* move\_shape(win):  
 win.setBackground('white')  
 obj = draw\_initial\_shape(win)  
 dx, dy = 5, 5  
  
 *for* \_ *in* range(50):  
 *if* win.checkMouse():  
 obj.undraw()  
 *return* time.sleep(0.1)  
 obj.move(dx, dy)  
 obj.undraw()  
  
*#------------------------- Масштабування ромба ------------------------  
  
def* scale\_shape(win):  
 win.setBackground('white')  
 obj = draw\_initial\_shape(win)  
 coords = np.array([[x1, y1, 1], [x2, y2, 1], [x3, y3, 1], [x4, y4, 1]])  
 center\_x, center\_y = xw // 2, yw // 2  
 scale\_factor = 1.05  
 scale\_limit = 1.5  
 scale\_step = 0.05  
 scale\_cycles = 3  
  
 *for* \_ *in* range(scale\_cycles):  
 increasing = *True  
 for* step *in* range(12):  
 *if* win.checkMouse():  
 obj.undraw()  
 *return* time.sleep(0.2)  
 obj.undraw()  
 factor = scale\_factor *if* increasing *else* 1 / scale\_factor  
 scaling\_matrix = np.array([  
 [factor, 0, center\_x \* (1 - factor)],  
 [0, factor, center\_y \* (1 - factor)],  
 [0, 0, 1]  
 ])  
 coords = coords.dot(scaling\_matrix.T)  
 obj = Polygon(Point(coords[0, 0], coords[0, 1]),  
 Point(coords[1, 0], coords[1, 1]),  
 Point(coords[2, 0], coords[2, 1]),  
 Point(coords[3, 0], coords[3, 1]))  
 obj.setFill('red')  
 obj.draw(win)  
  
 *if* step == 5:  
 increasing = *False* obj.undraw()  
  
win = GraphWin("2-D трансформації", xw, yw)  
obj = draw\_initial\_shape(win)  
win.getMouse()  
obj.undraw()  
rotate\_shape(win, obj)  
win.getMouse()  
move\_shape(win)  
win.getMouse()  
scale\_shape(win)  
win.getMouse()  
win.close()

**5.4. Висновки**

В результаті виконання лабораторної роботи було реалізовано алгоритми 2D-перетворень геометричних об'єктів у графічному вікні. Було досліджено та застосовано математичні моделі для обертання, переміщення та масштабування ромба. Розроблена програма коректно виконує всі необхідні перетворення, анімація працює плавно, зміна стану об'єкта відбувається за допомогою натискання миші. Всі завдання лабораторної роботи виконані у повному обсязі.