Detail Information of Python project using Linear Algebra

Sketcher

Introduction

- A graphical application designed for digital sketching,
- Integrating mathematical transformations such as rotation and scaling.
- Developed using Python’s Tkinter library,
- To demonstrate the practical use of mathematics in computer graphics

Objectives

- ·To understand and apply transformation matrices such as scaling, rotation, and translation in computer graphics.
- ·To observe how linear algebra concepts are used to manipulate graphical objects in a digital environment.
- ·To develop an interactive drawing application that allows users to create and modify strokes, shapes, and text using mathematical transformations.
- To demonstrate the real-world significance of linear algebra in modern computer graphics and digital design.

Background Theory

Linear Algebra in Computer Graphics

For enabling

- Rotating
- Size changing and
- Moving

Transformation Matrices

Includes 3 Matrices

1. Rotation Matrix
2. Scaling Matrix
3. Translation Matrix

Background Theory

1. Rotation Matrix

A rotation transformation rotates an object around a fixed point.

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Used to rotate shapes in real-time while preserving their original structure.

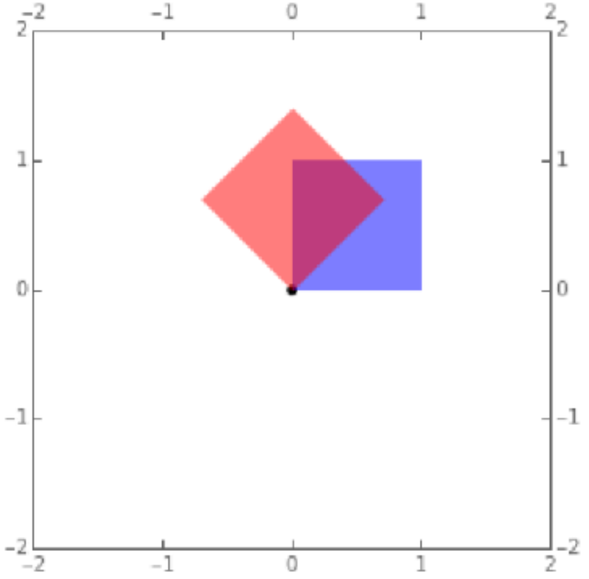


Figure – 2D Rotation

Background Theory

2. Scaling Matrix

Scaling transformations increase or decrease the size of an object.

$$S(s_x, s_y) = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

where and are the scaling factors along the x- and y-axes, respectively.

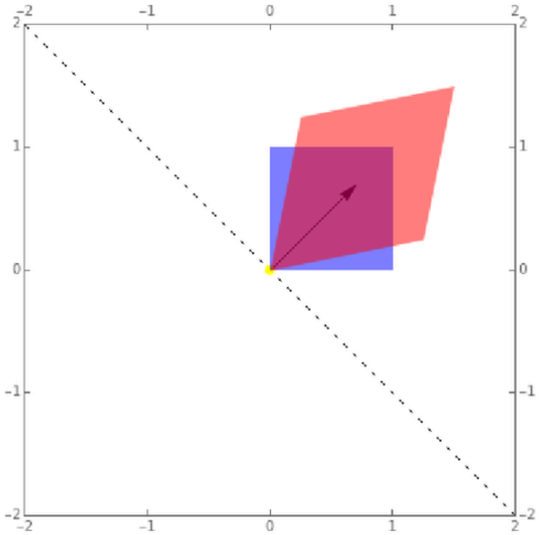


Figure – 2.2 2D Scaling

Background Theory

3. Translation Matrix

Translation moves an object from one position to another without altering its shape or size.

$$T(t_x, t_y) = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

where and represent movement along the x- and y-axes respectively.

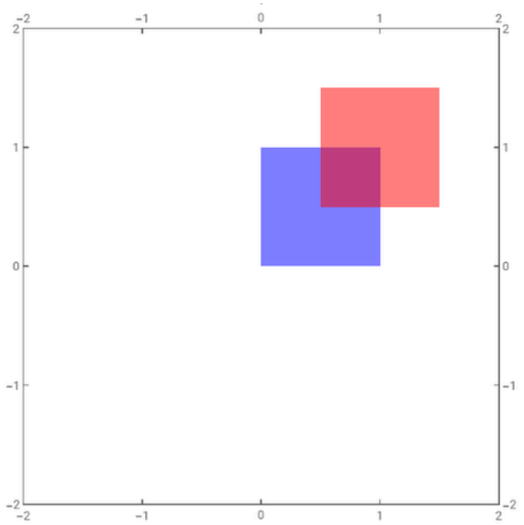


Figure – 2.3 2D Translation

Tools and techniques used

Programming Language : **Python** (Version 3.6 or later).

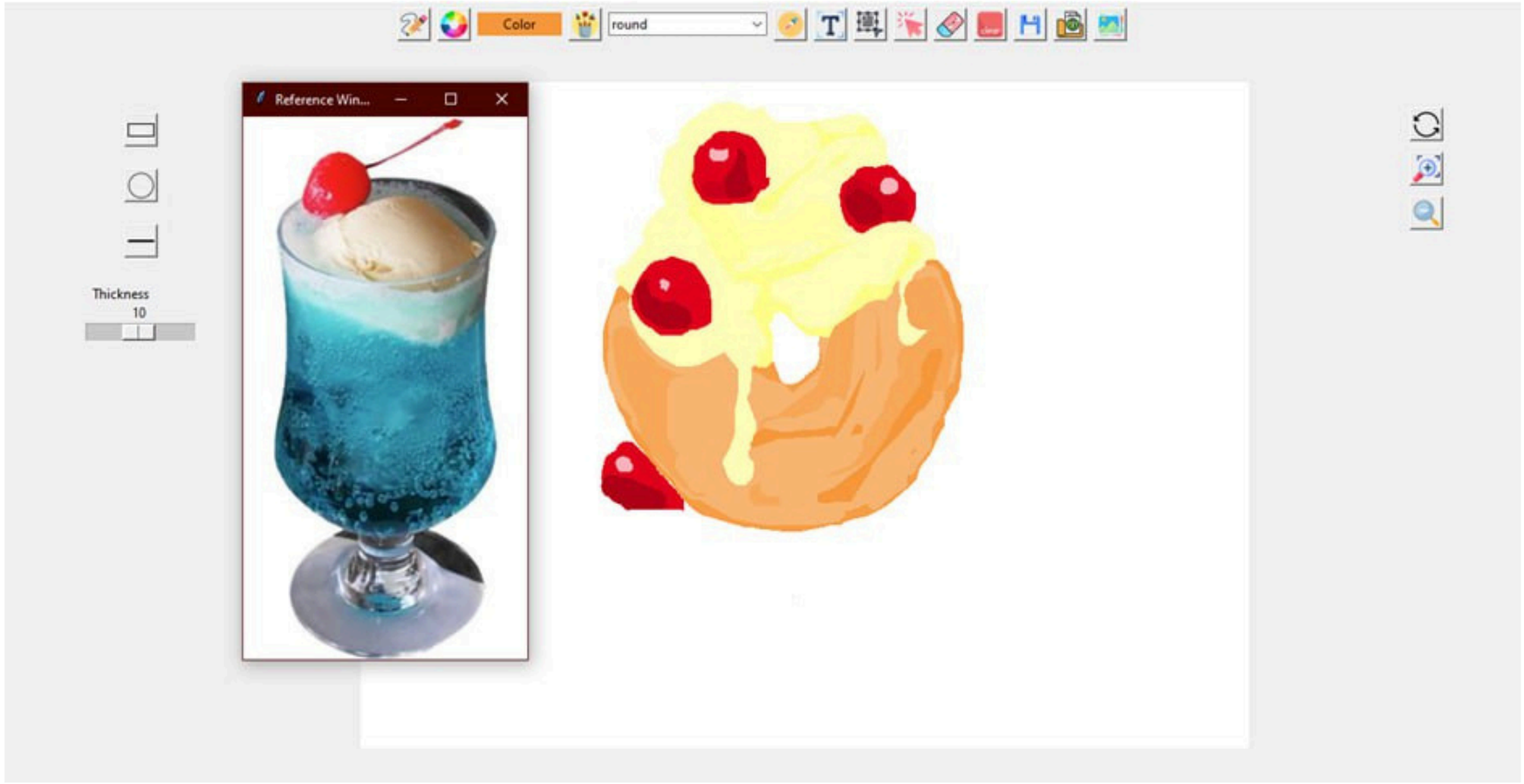
Libraries required / Software requirement

- tkinter – Provides **Graphical User Interface** (GUI).
- numpy – Used for M**athematical Operations, transformations** and **array.**
- Pillow (PIL) – Handles **Image Processing, Saving** and **Loading Images**.
- OpenCV (cv2) – Used for **Image Manipulation** and **Handling Drawing**
- math – Provides **Mathematical functions like Rotation and Scaling.**
- os – Used for **File Handling** and **Interacting with the Operating System**.
- random – Used for **Generating Random Numbers**, **Selections**, and **Variations**.

Testing and Validation of Sketcher Application

UNIVERSITY of INFORMATION TECHNOLOGY

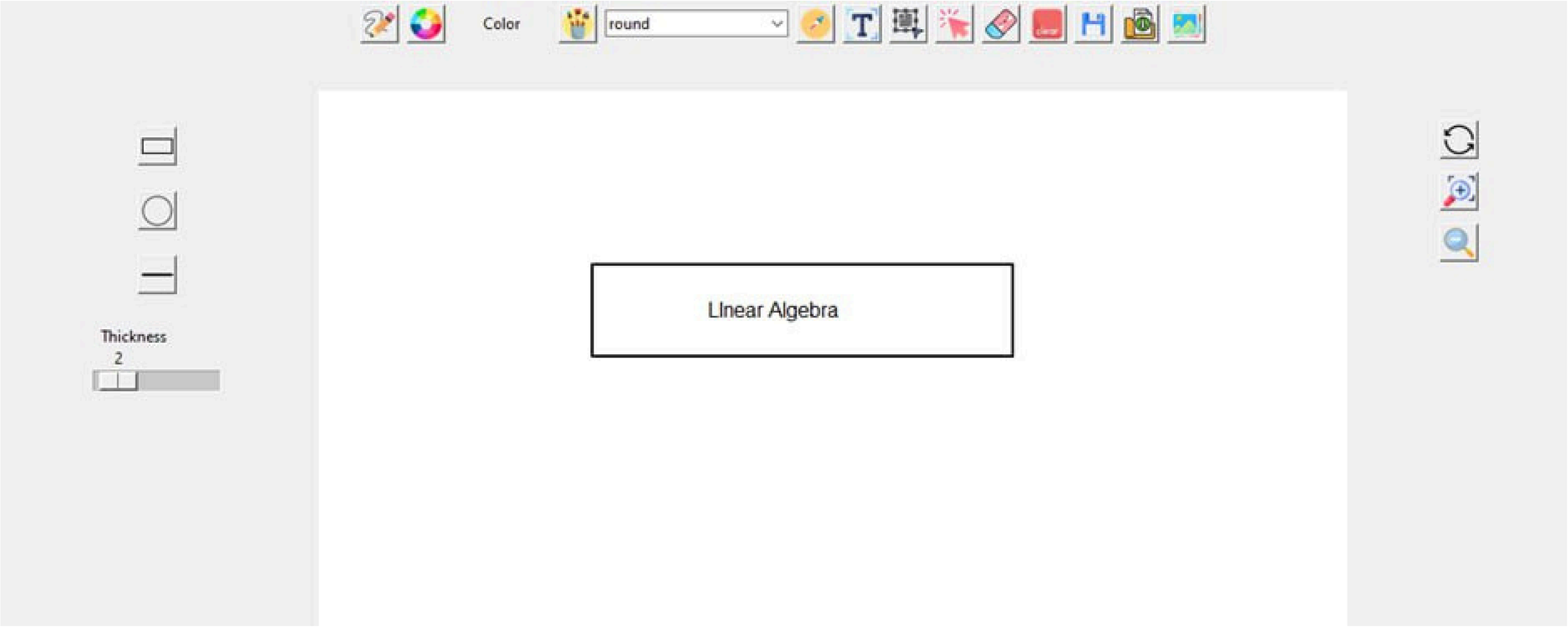
Reference-Based Digital Sketching Test in Sketcher



Testing and Validation of Sketcher Application

UNIVERSITY of INFORMATION TECHNOLOGY

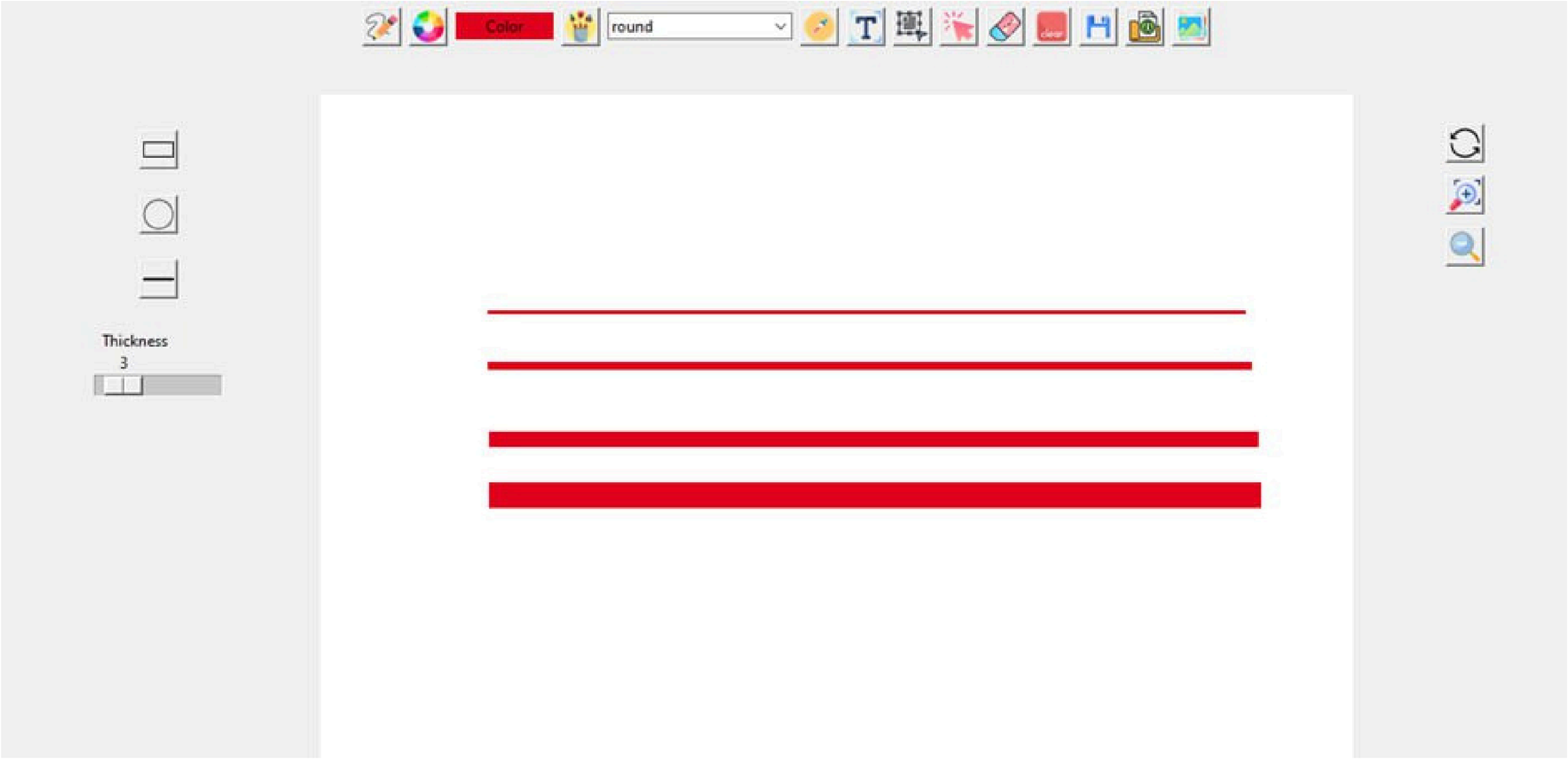
Text and Shape Testing: Linear Algebra Box in Sketcher



Testing and Validation of Sketcher Application

UNIVERSITY of INFORMATION TECHNOLOGY

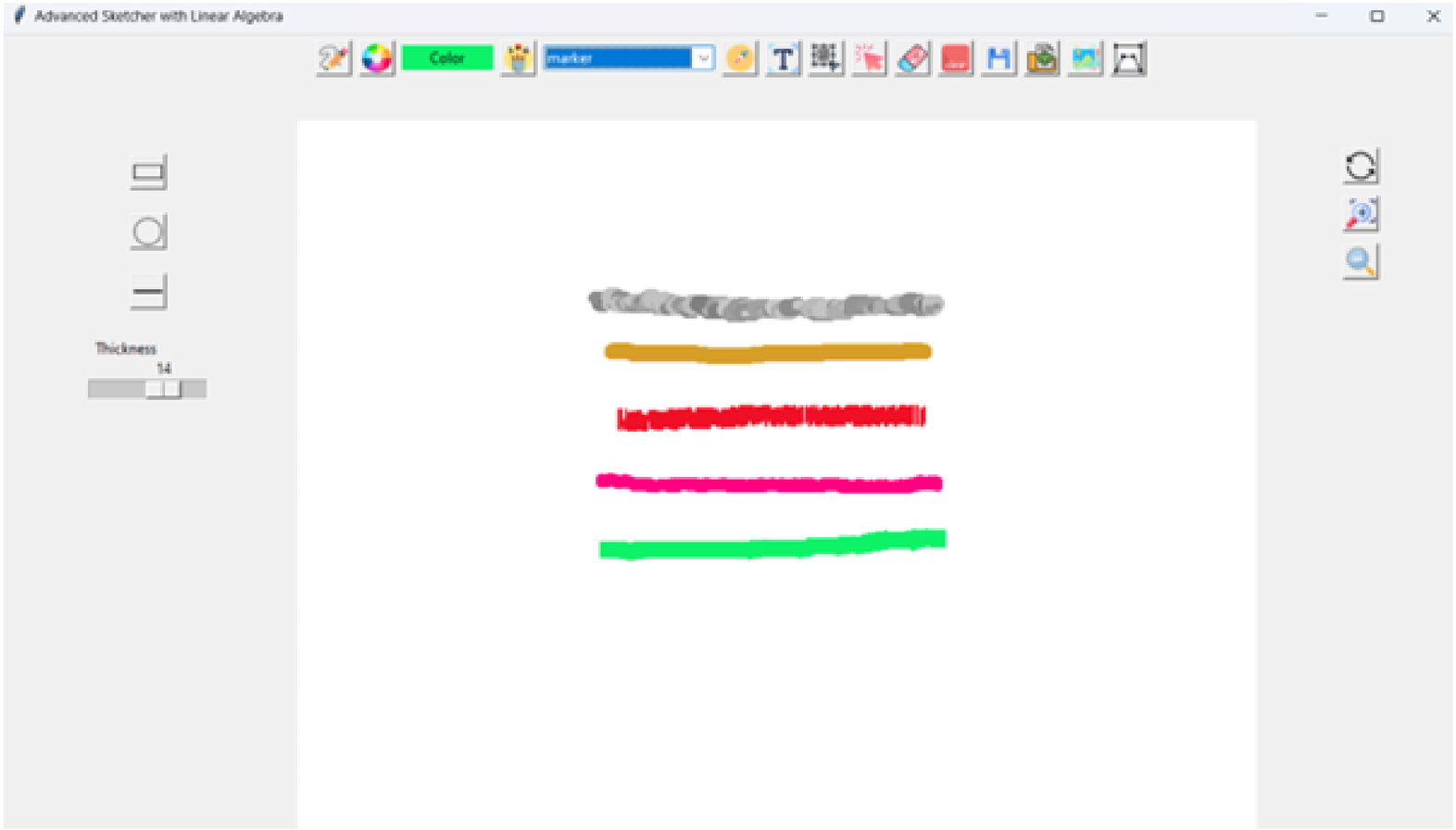
Input Stroke: Red Horizontal Line Thickness Test in Sketcher



Testing and Validation of Sketcher Application

UNIVERSITY of INFORMATION TECHNOLOGY

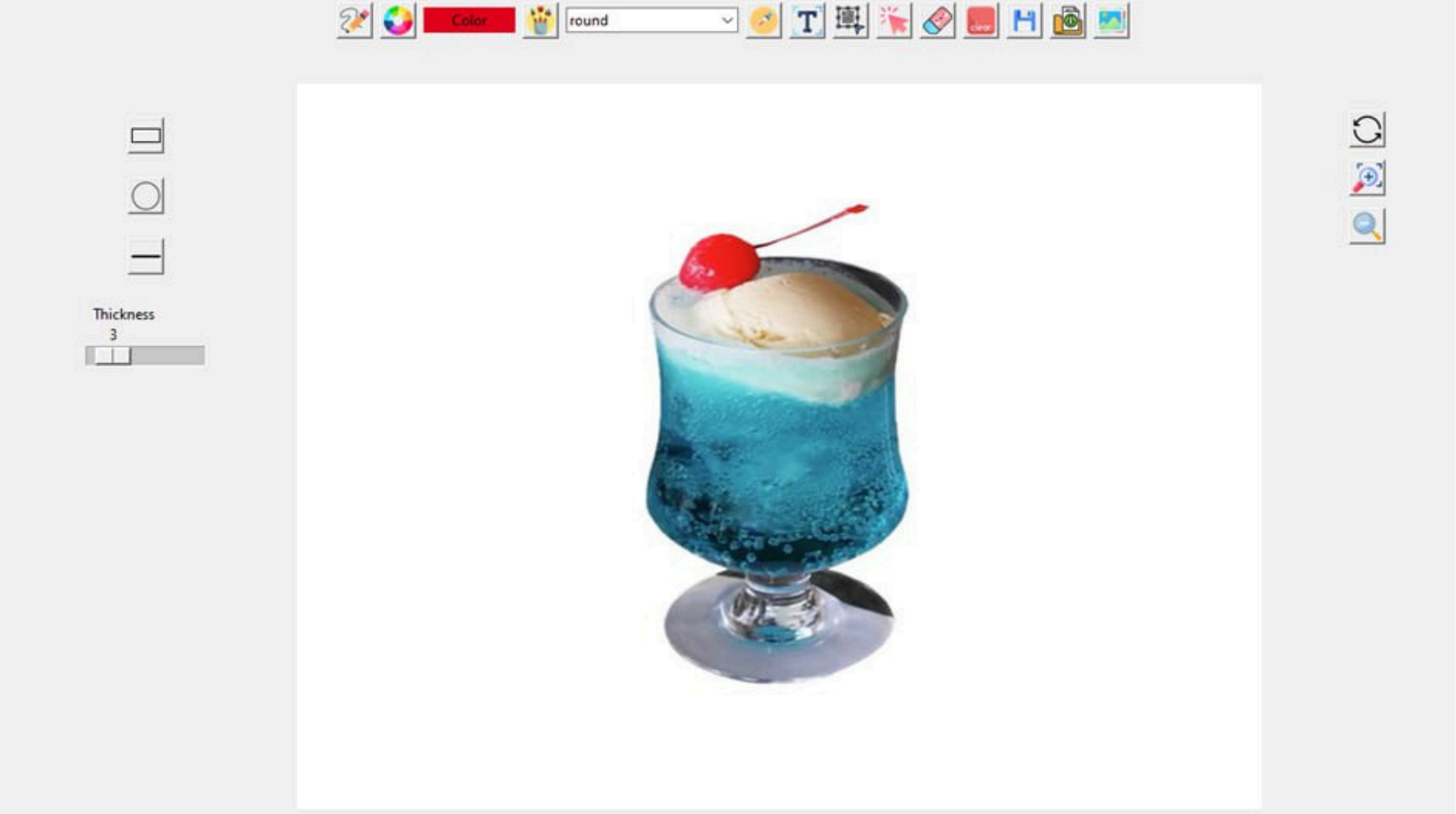
Stroke and Color Testing: a variety of lines in Sketcher



Testing and Validation of Sketcher Application

UNIVERSITY of INFORMATION TECHNOLOGY

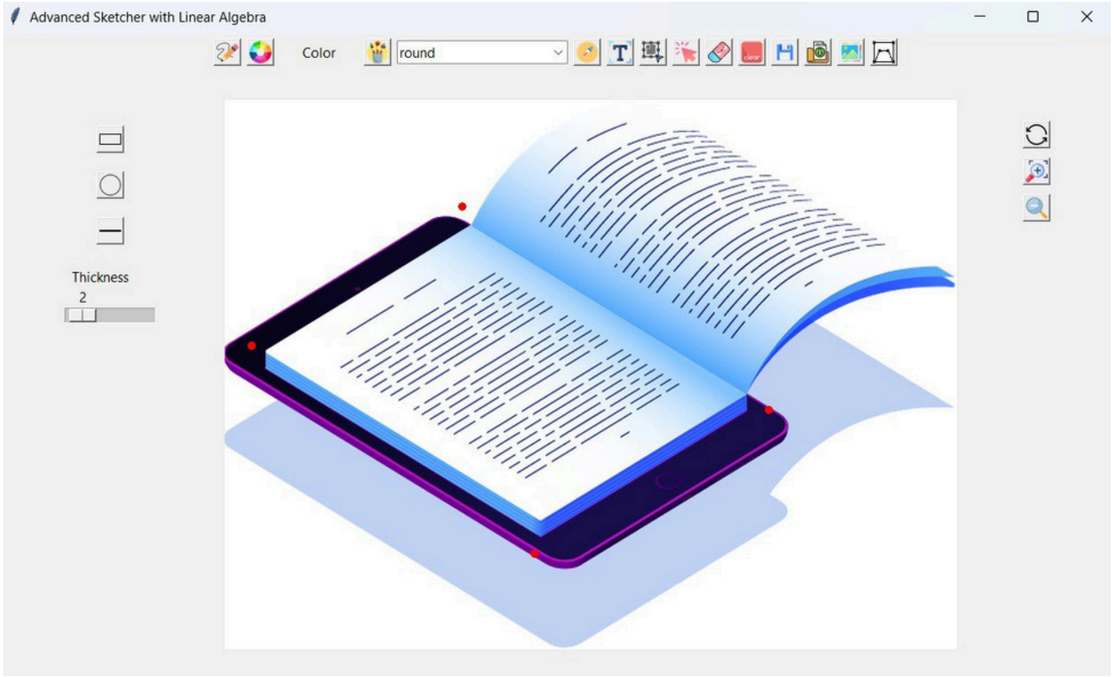
Image Import Testing: Blue Soda Float in Sketcher



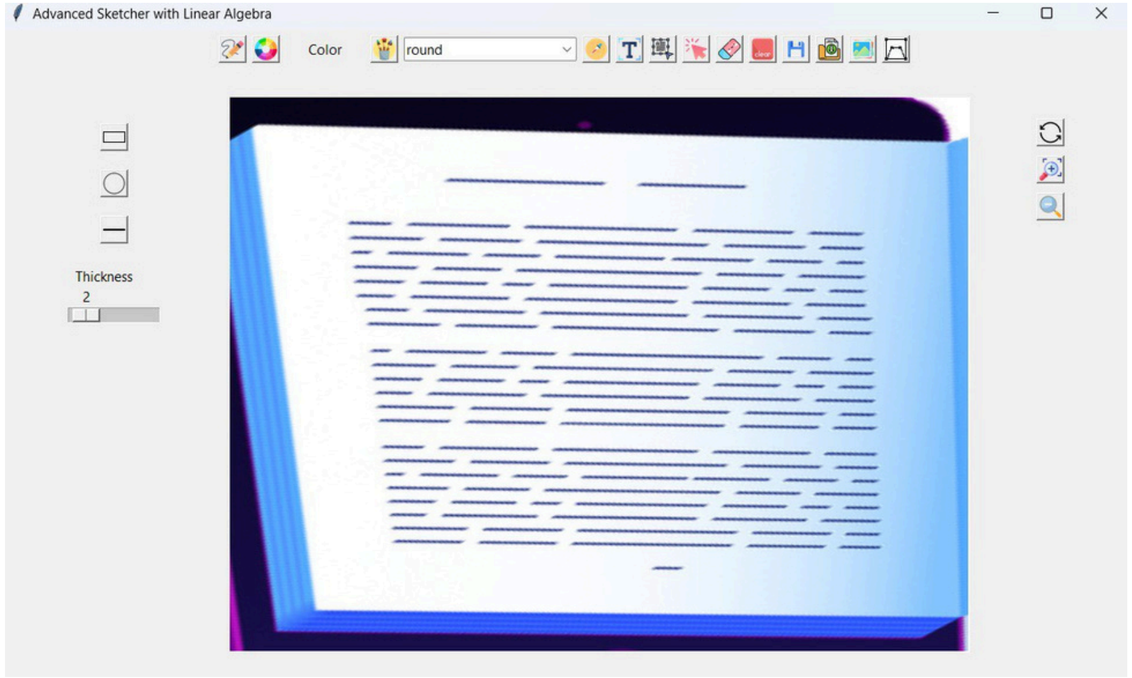
Testing and Validation of Sketcher Application

UNIVERSITY of INFORMATION TECHNOLOGY

Perspective Transform Testing: Transformed Book Image in Sketcher



Before

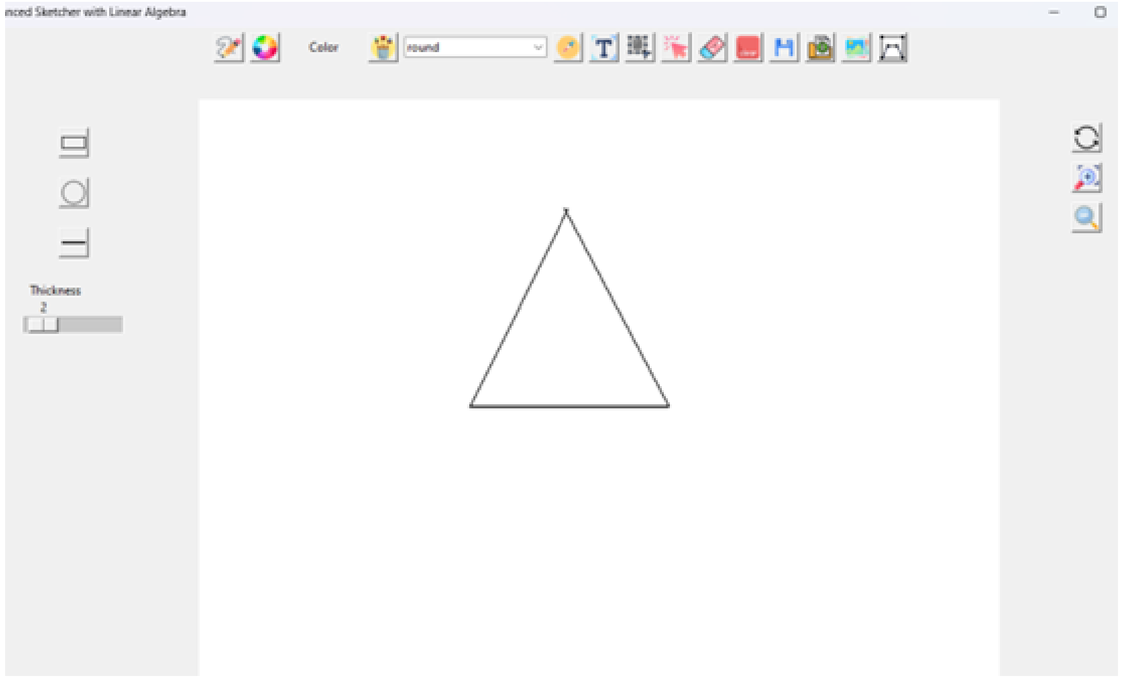


After

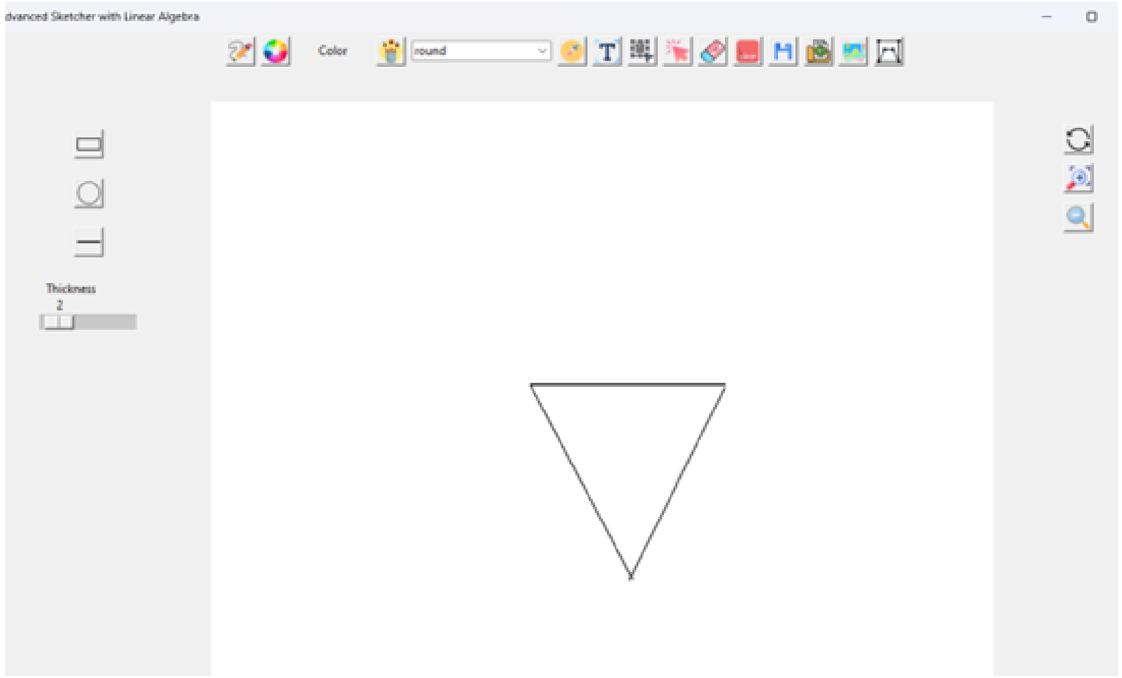
Testing and Validation of Sketcher Application

UNIVERSITY of INFORMATION TECHNOLOGY

Rotation Testing: Rotated drawing in Sketcher



Before



After