CSE382 Computer Graphics, Spring Semester 24/25

Computer Graphics Project

Student ID: 222100081

Full Name: Ahmed Sameh Basuny

Table of Contents

---

1. Introduction

This report presents the design and development of an interactive graphics application for CSE382. The application demonstrates analytical and practical solutions to ten transformation-based problems in computer graphics using Python, OpenGL, and Pygame. Each transformation is visualized in real-time, enhancing conceptual understanding through user interaction.

---

2. Application Overview

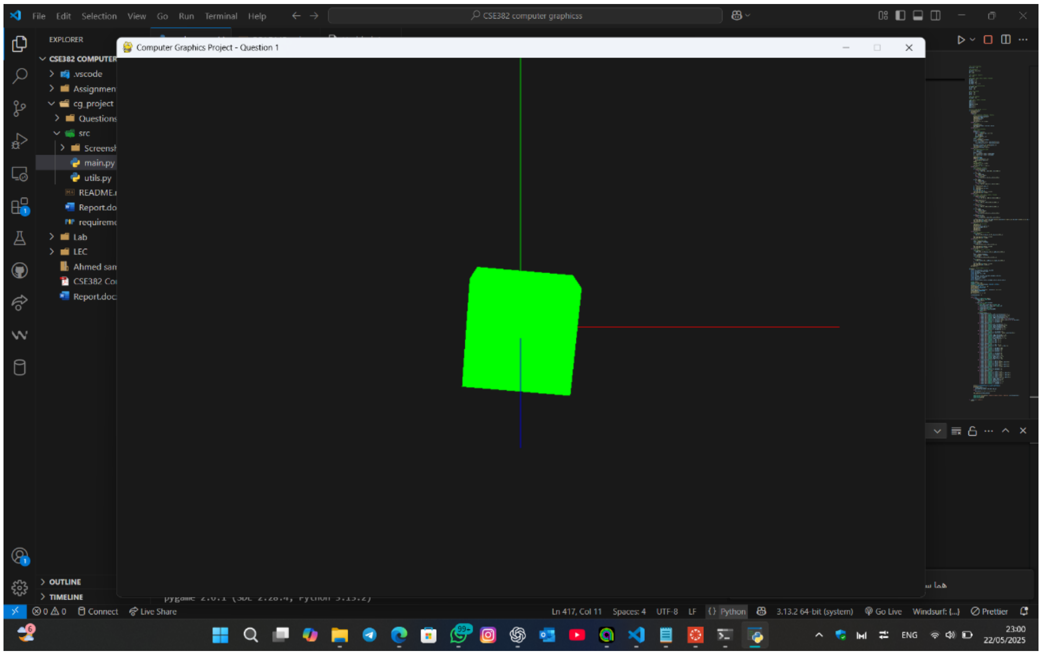
The application is implemented in Python (main.py) and uses OpenGL for rendering. It supports ten transformation scenarios, accessible by pressing number keys (1–0). The user can manipulate transformation parameters in real-time using keyboard inputs. The application's architecture allows modular updates, making it scalable for additional transformations.

---

3. Analytical and Practical Solutions

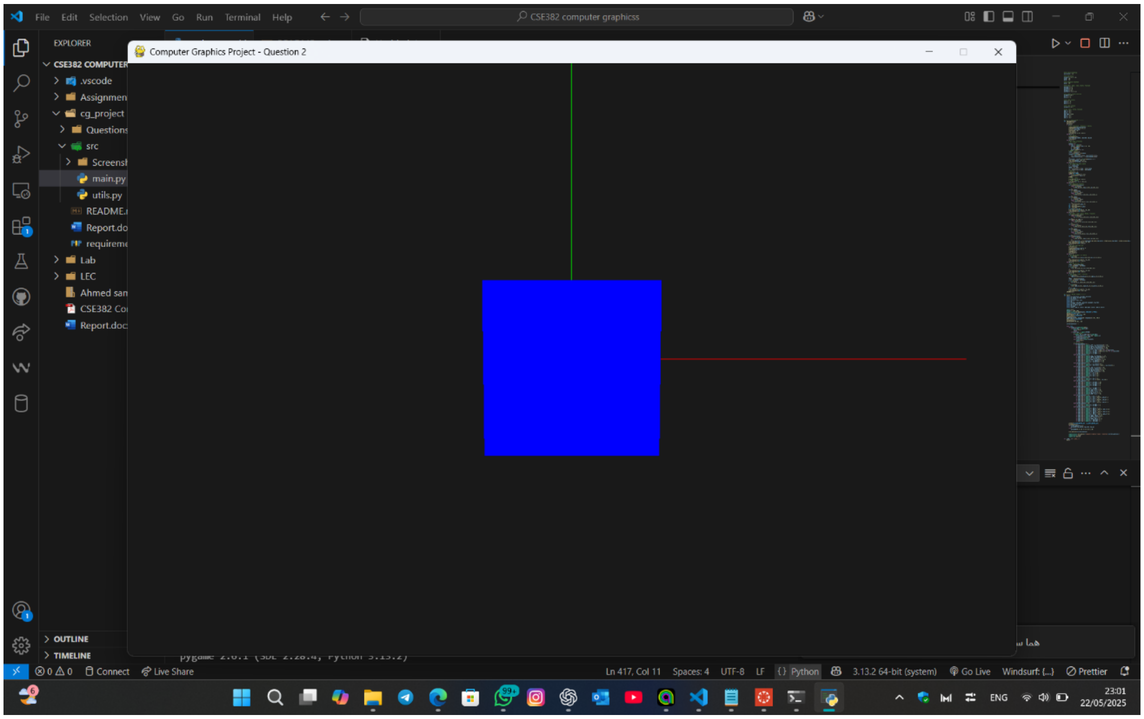
3.1 Question 1: Translation, Rotation, Scaling

Analytical Solution: A cube is transformed using matrix multiplication of translation, rotation (Z-axis), and scaling matrices.
OpenGL Implementation: Applied via glTranslatef, glRotatef, and glScalef.



3.2 Question 2: Camera Movement

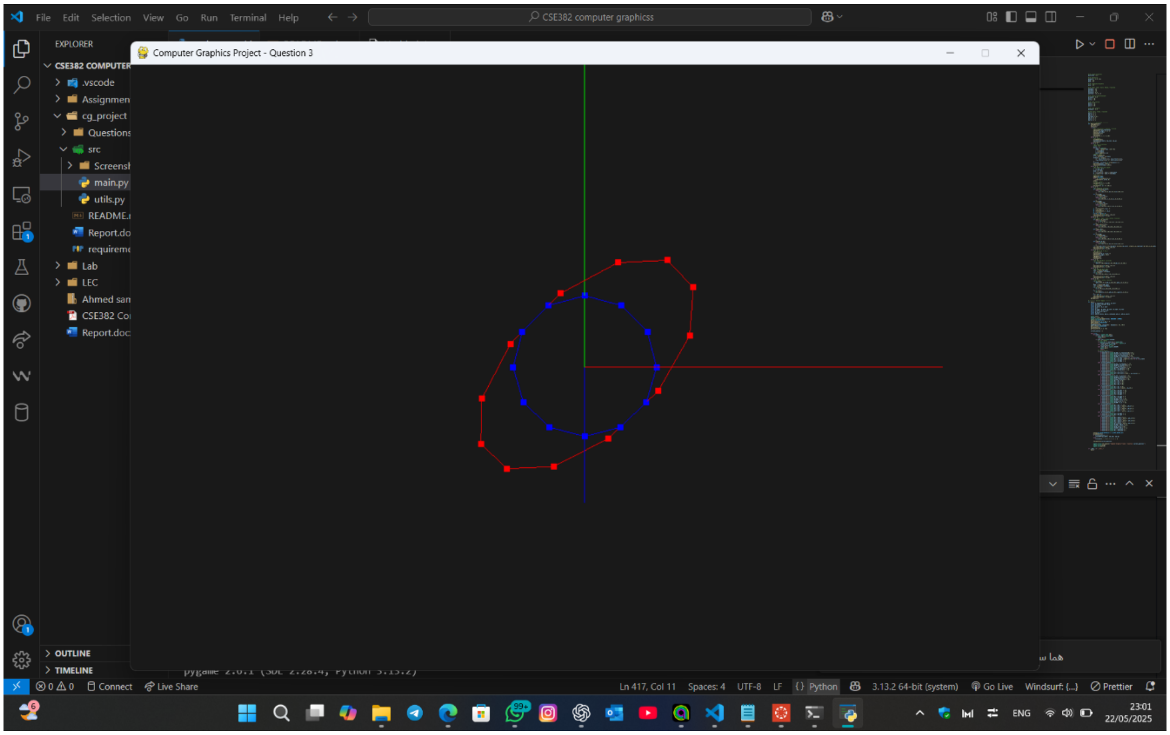
Analytical Solution: View matrix formed using camera position, target, and up vector.
OpenGL Implementation: gluLookAt used to define the camera.



3.3 Question 3: Clock Stretching

Analytical Solution: Twelve 2D points scaled non-uniformly with a stretching matrix.

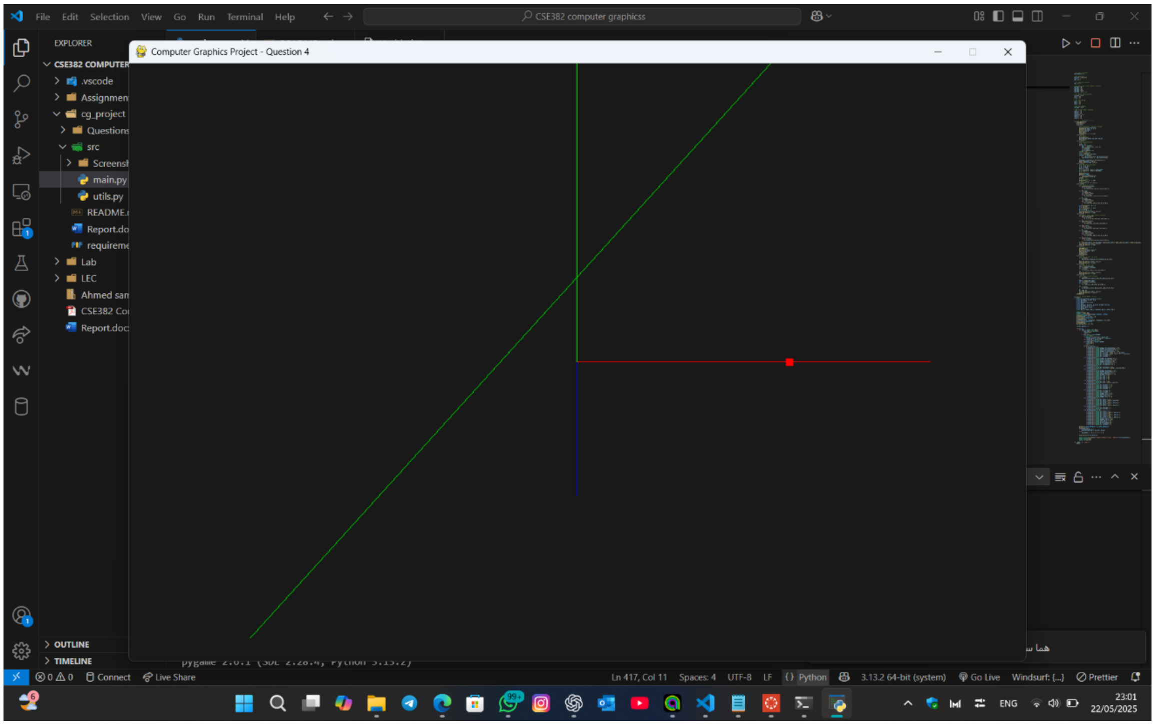
OpenGL Implementation: Drawn with GL_LINE_LOOP for original and stretched clocks.



3.4 Question 4: Reflection Over a Line

Analytical Solution: Points reflected over y = mx + b using reflection formulas.

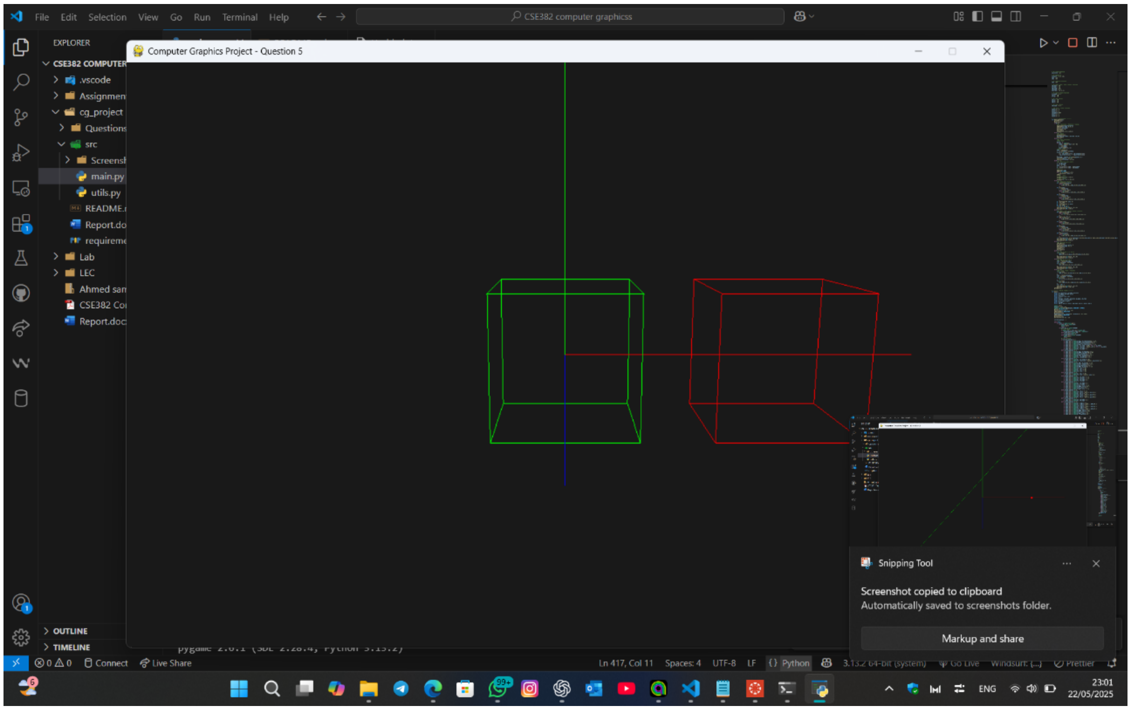
OpenGL Implementation: Line, original, and reflected points are displayed.


File Edit Selection View Go Run Terminal Help
CSE382 computer graphicss
EXPLORER
CSE382 COMPUTER
.vscode
Assignmen
cg_project
Questions
src
Screensh
main.py
utils.py
README.
Report.do
requireme
Lab
LEC
Ahmed san
CSE382 Co
Report.doc
OUTLINE
TIMELINE
Computer Graphics Project - Question 4
Ln 417, Col 11 Spaces: 4 UTF-8 LF Python 3.13.2 64-bit (system) Go Live Windsurf: {...} Prettier
Connect Live Share
ENG
23:01
22/05/2025

3.5 Question 5: Composite Transformation

Analytical Solution: Combined transformations via matrix multiplication.

OpenGL Implementation: Rendered cube shows both original and transformed states.

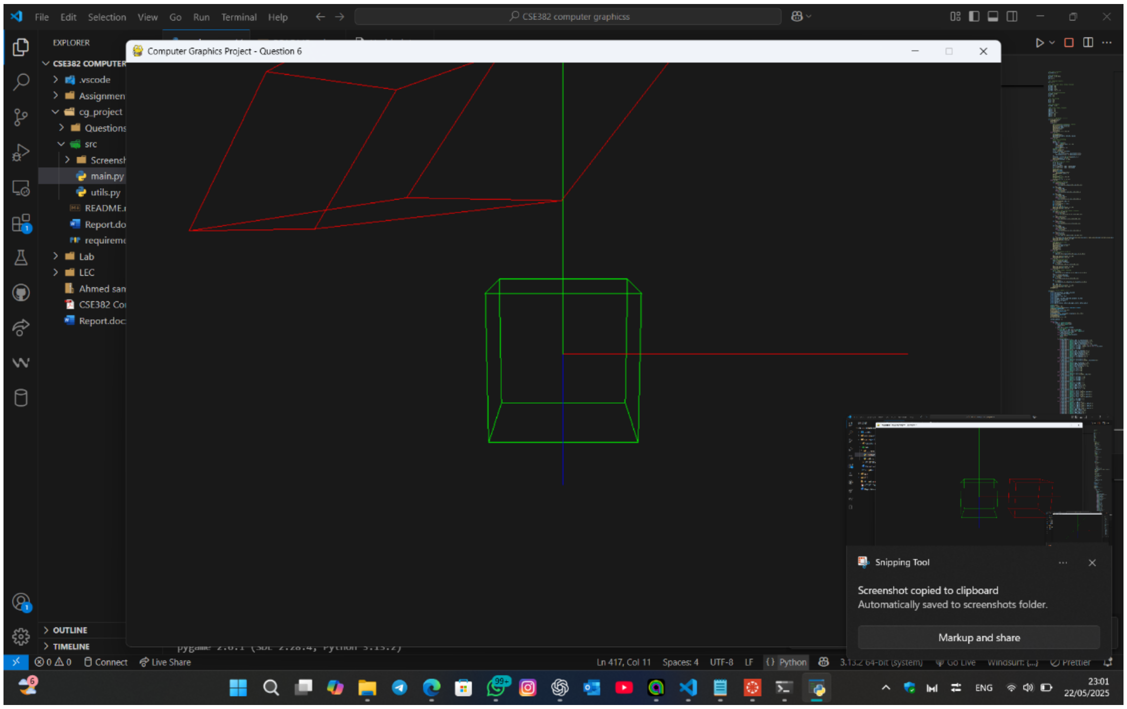


Screenshot copied to clipboard
Automatically saved to screenshots folder.

3.6 Question 6: Shear, Taper, Scale, Rotate, Translate

Analytical Solution: Transformation matrices applied sequentially.

OpenGL Implementation: glMultMatrixf used after building the matrix.

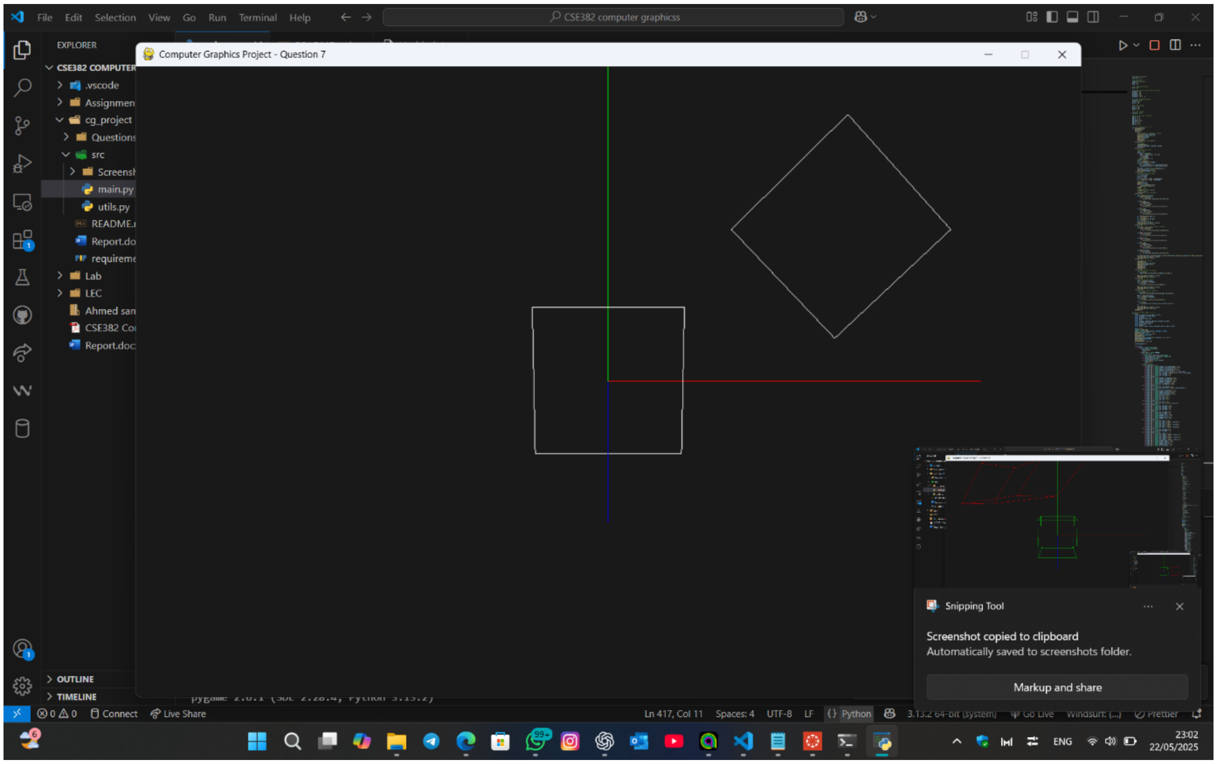


Screenshot copied to clipboard
Automatically saved to screenshots folder.

3.7 Question 7: Square Rotation and Translation

Analytical Solution: Rotate then translate a square using homogeneous coordinates.

OpenGL Implementation: Square drawn twice to illustrate change.



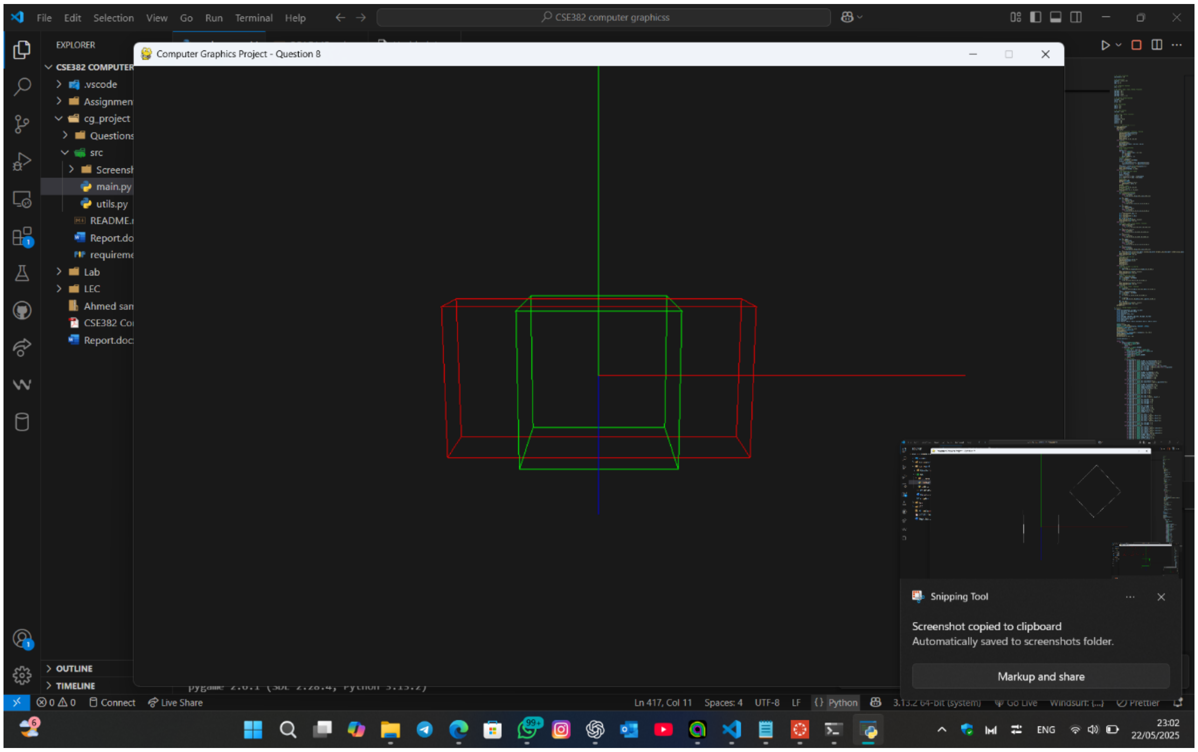
Screenshot copied to clipboard
Automatically saved to screenshots folder.

```
pygame 2.6.1 (SDL 2.28.4, Python 3.13.2)
```

3.8 Question 8: Cube Non-Uniform Scaling

Analytical Solution: Scale X, Y, Z with different values.

OpenGL Implementation: glScalef(x, y, z) used.

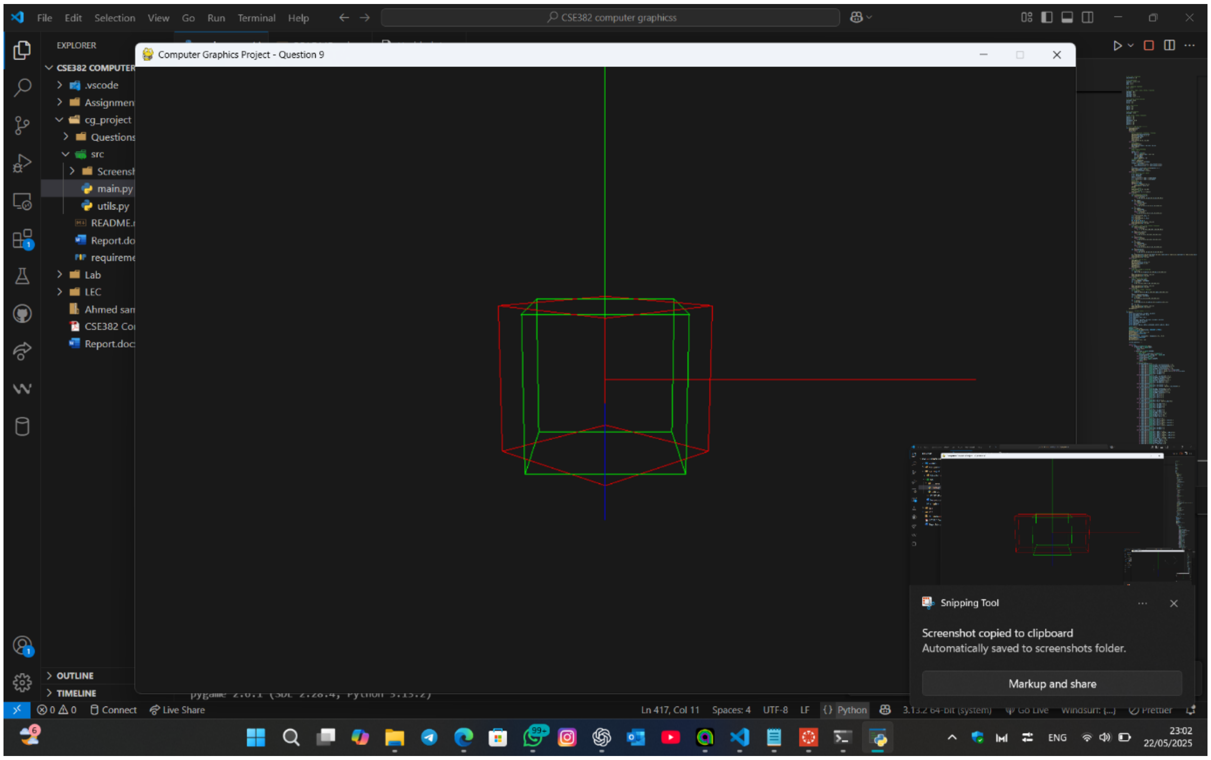


Screenshot copied to clipboard
Automatically saved to screenshots folder.

3.9 Question 9: Cube Rotation Around Y Axis

Analytical Solution: Rotation matrix about Y axis applied.

OpenGL Implementation: glRotatef(angle, 0, 1, 0) used.

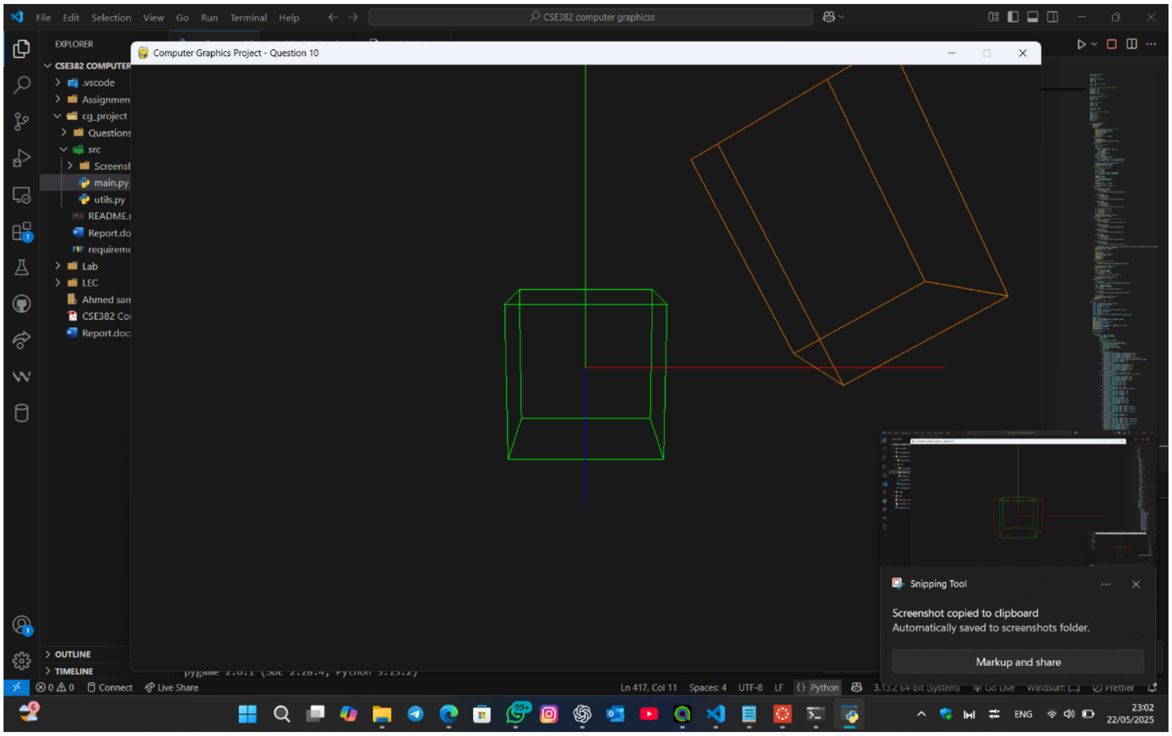


Screenshot copied to clipboard
Automatically saved to screenshots folder.

3.10 Question 10: Scale, Rotate, Translate (All Together)

Analytical Solution: Composite transformation matrix built from three matrices.

OpenGL Implementation: All transformations applied in one sequence.



Computer Graphics Project - Question 10

Screenshot copied to clipboard
Automatically saved to screenshots folder.

pygame 2.6.1 (SDL 2.28.4, Python 3.13.2)

4. User Controls

Question Controls

1	Arrows: Move, W/S: Scale, A/D: Rotate
2	Arrows/W/S: Move camera
3	W/S: Stretch clock
4	Arrows: Move point, W/S: Slope, A/D: Intercept
5	W/S: Change parameter
6	W/S: Shear, A/D: Rotate
7	Arrows: Move, A/D: Rotate
8	W/S: Scale X, A/D: Scale Y, Q/E: Scale Z

Question Controls

9	A/D: Rotate
10	W/S: Scale X, A/D: Scale Y, Q/E: Scale Z, Arrows: Move, Z/X: Rotate

Use number keys (1–9) to select questions; press 0 for question 10.
Use ESC to exit.

---

5. Conclusion

This project integrates analytical transformation concepts with practical OpenGL implementations. Through user interactivity and visual feedback, the application facilitates deeper understanding of 2D and 3D transformations in computer graphics. It serves as a valuable educational and development tool.

---

6. References

[1] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes, *Computer Graphics: Principles and Practice*, 3rd ed., Addison-Wesley, 2013.

[2] E. Angel, D. Shreiner, *Interactive Computer Graphics: A Top-Down Approach with WebGL*, 7th ed., Pearson, 2015.

[3] *Pygame Documentation*, [Online]. Available: https://www.pygame.org/docs/

[4] *PyOpenGL Documentation*, [Online]. Available: https://pyopengl.sourceforge.net/documentation/

[5] *Numpy Documentation*, [Online]. Available: https://numpy.org/doc/