**Nilanjan B. Mitra (2023BCS-501)**
**19th September, 2025**

**CS 302: Computer Graphics**
**Minor Lab Exam Assignment: 3D Graphics Viewer**

## Project Overview

Link to the Project: https://github.com/DarkPhoenix645/cg-lab-minor

1. This assignment involved the design and implementation of a 3D Graphics Viewer which has 4 built-in objects which can be viewed viz. Cube, Pyramid, Tetrahedron and Octahedron in both Orthographic and Perspective Projections.
2. The line drawing is done using the DDA (Differential Drawing Analyser) Algorithm.
3. It also includes support for moving / rotating the object, scaling the object and resetting all transformations back to the initial state.
4. It is built using Python and the pygame library.

## Learnings & Observations

1. **Understanding Projections:** A key learning was the distinct mathematical and conceptual differences between orthographic and perspective projections:
   (a) **Orthographic Projection:** Achieved by discarding the Z-coordinate and scaling, preserving sizes and parallel lines. This is crucial for technical drawings and CAD applications where accurate measurements are paramount. The mathematical formulation is relatively simple, involving scaling and translation.
   (b) **Perspective Projection:** Achieved by dividing coordinates by the depth (Z-value), simulating how humans perceive depth. This results in a more realistic visual representation where distant objects appear smaller. The implementation required careful handling of camera positioning and depth calculations.
2. **Geometric Transformations:** Implementation of translation, rotation, and scaling using 4x4 matrices was also done. This uses standard matrix transforms for the associated geometric operations.
3. **Line Drawing Algorithms:** The DDA (Digital Differential Analyzer) algorithm was implemented for drawing wireframe lines. This reinforced the understanding of rasterization principles, where continuous lines are approximated by discrete pixels. While simpler than Bresenham's, DDA's reliance on floating-point arithmetic reduces performance.
4. **pygame Integration:** Leveraging pygame provided a framework for window creation and event handling (keyboard inputs for transformations and projections).

## Code Structure

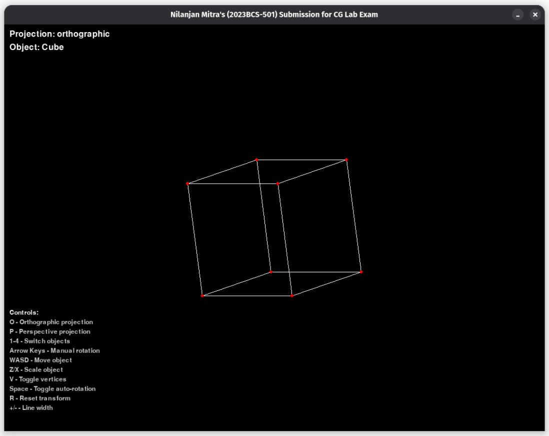
The project is organized as follows:

- `pyproject.toml`: Project configuration and dependency management.
- `src/`: Main Python package directory.
  - `main.py`: Entry point of the application.
  - `gui.py`: Handles user interface, event loop, and user input.
  - `renderer.py`: Manages drawing objects to the screen, including line rendering and projection application.

- `projections.py`: Implements orthographic and perspective projection logic.
- `transformations.py`: Handles geometric transformations (translate, rotate, scale).
- `objects.py`: Defines 3D object structures and creation functions (e.g., cube, pyramid).
- `math_utils.py`: Provides fundamental vector and matrix operations.

## Output Images

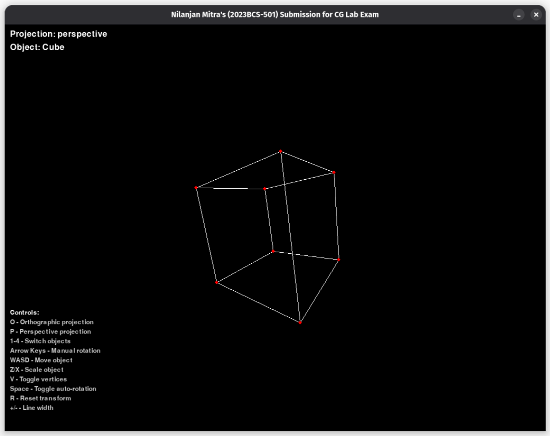
The following grid displays outputs of the lab assignment:



```
Nilanjan Mitra's (2023BCS-501) Submission for CG Lab Exam
Projection: orthographic
Object: Cube

Controls:
O - Orthographic projection
P - Perspective projection
1-4 - Switch objects
Arrow Keys - Manual rotation
WASD - Move object
Z/X - Scale object
V - Toggle vertices
Space - Toggle auto-rotation
R - Reset transform
+/- - Line width
```

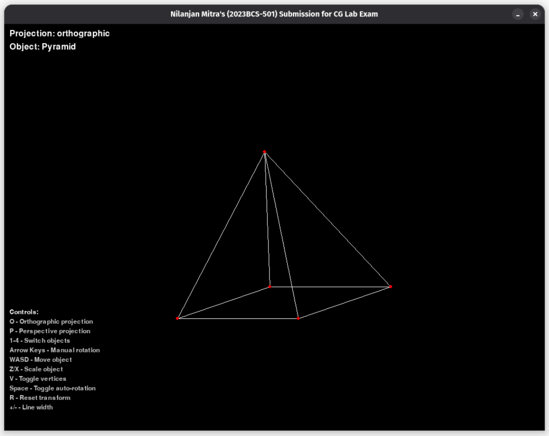
Orthographic Projection (Cube)



```
Nilanjan Mitra's (2023BCS-501) Submission for CG Lab Exam
Projection: perspective
Object: Cube

Controls:
O - Orthographic projection
P - Perspective projection
1-4 - Switch objects
Arrow Keys - Manual rotation
WASD - Move object
Z/X - Scale object
V - Toggle vertices
Space - Toggle auto-rotation
R - Reset transform
+/- - Line width
```

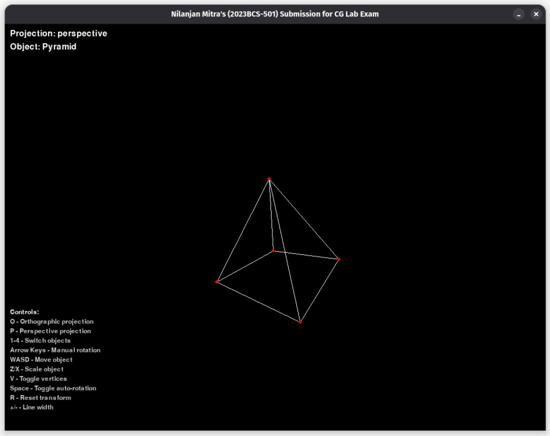
Perspective Projection (Cube)



```
Nilanjan Mitra's (2023BCS-501) Submission for CG Lab Exam
Projection: orthographic
Object: Pyramid

Controls:
O - Orthographic projection
P - Perspective projection
1-4 - Switch objects
Arrow Keys - Manual rotation
WASD - Move object
Z/X - Scale object
V - Toggle vertices
Space - Toggle auto-rotation
R - Reset transform
+/- - Line width
```

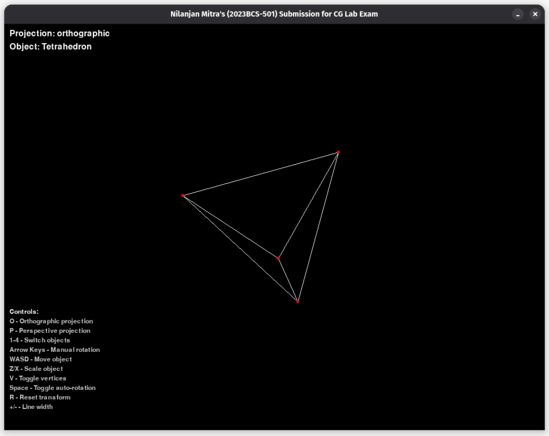
Orthographic Projection (Pyramid)



```
Nilanjan Mitra's (2023BCS-501) Submission for CG Lab Exam
Projection: perspective
Object: Pyramid

Controls:
O - Orthographic projection
P - Perspective projection
1-4 - Switch objects
Arrow Keys - Manual rotation
WASD - Move object
Z/X - Scale object
V - Toggle vertices
Space - Toggle auto-rotation
R - Reset transform
+/- - Line width
```

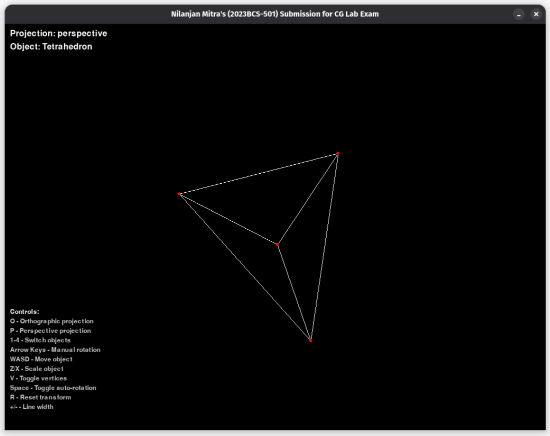
Perspective Projection (Pyramid)



```
Nilanjan Mitra's (2023BCS-501) Submission for CG Lab Exam
Projection: orthographic
Object: Tetrahedron

Controls:
O - Orthographic projection
P - Perspective projection
1-4 - Switch objects
Arrow Keys - Manual rotation
WASD - Move object
Z/X - Scale object
V - Toggle vertices
Space - Toggle auto-rotation
R - Reset transform
+/- - Line width
```

Orthographic Projection (Tetrahedron)



```
Nilanjan Mitra's (2023BCS-501) Submission for CG Lab Exam
Projection: perspective
Object: Tetrahedron

Controls:
O - Orthographic projection
P - Perspective projection
1-4 - Switch objects
Arrow Keys - Manual rotation
WASD - Move object
Z/X - Scale object
V - Toggle vertices
Space - Toggle auto-rotation
R - Reset transform
+/- - Line width
```

Perspective Projection (Tetrahedron)