



Development of 3D computer animations through Matrix Operations.



GROUP 3



Group Members

AASHITA CHOUHAN

IIT2020124

ADITYA VERMA

IIT2020125

MITUL VARSHNEY

IIT2020145

SHASHWAT MITTAL

IIT2020157

Github Link: <https://github.com/homewardgamer/3D-Simulation>



Abstract

The following program demonstrates how the basic 3D transformations can be programmed in Java to show the movements of a 3D object. When the applet starts, a 2D cube appears on the screen. When we press the drag mouse then cube performs rotations in the direction in which the mouse was dragged and thus, we can see the 2D rendered image of the cube. A z buffer is used to delete hidden faces. "Double buffering" is used for smooth animation. The present program has the feature of developing the required 3D model in either wireframe or as solid shape. Also, the present program uses classes namely Object3d, Polygon3D, Point3D.

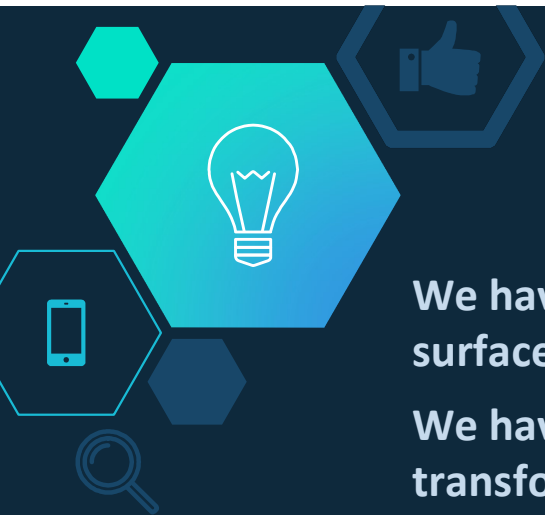


Concept

In this project, we have used the concept of matrix transformations in 3D space to transform and rotate our 3D Cube along the x,y, and z-axis.

Technologies Used: Java, Swing, XML





We have rendered the 3D Cube by using 12 triangle-shaped surfaces/planes where each Cube's face is divided into two triangles.

We have used mouse event listeners and triggered the 3D transformation using a matrix. In this way, we can render a 3D cube in the 3D space, but in reality, it is just a projection of the 3D Cube on a 2D screen.

We have also added darker gradients to refine the edges and depth to the Cube to make it look good. At last, we have implemented a field of view, where we increase the Cube's size inversely to the users' field of view.





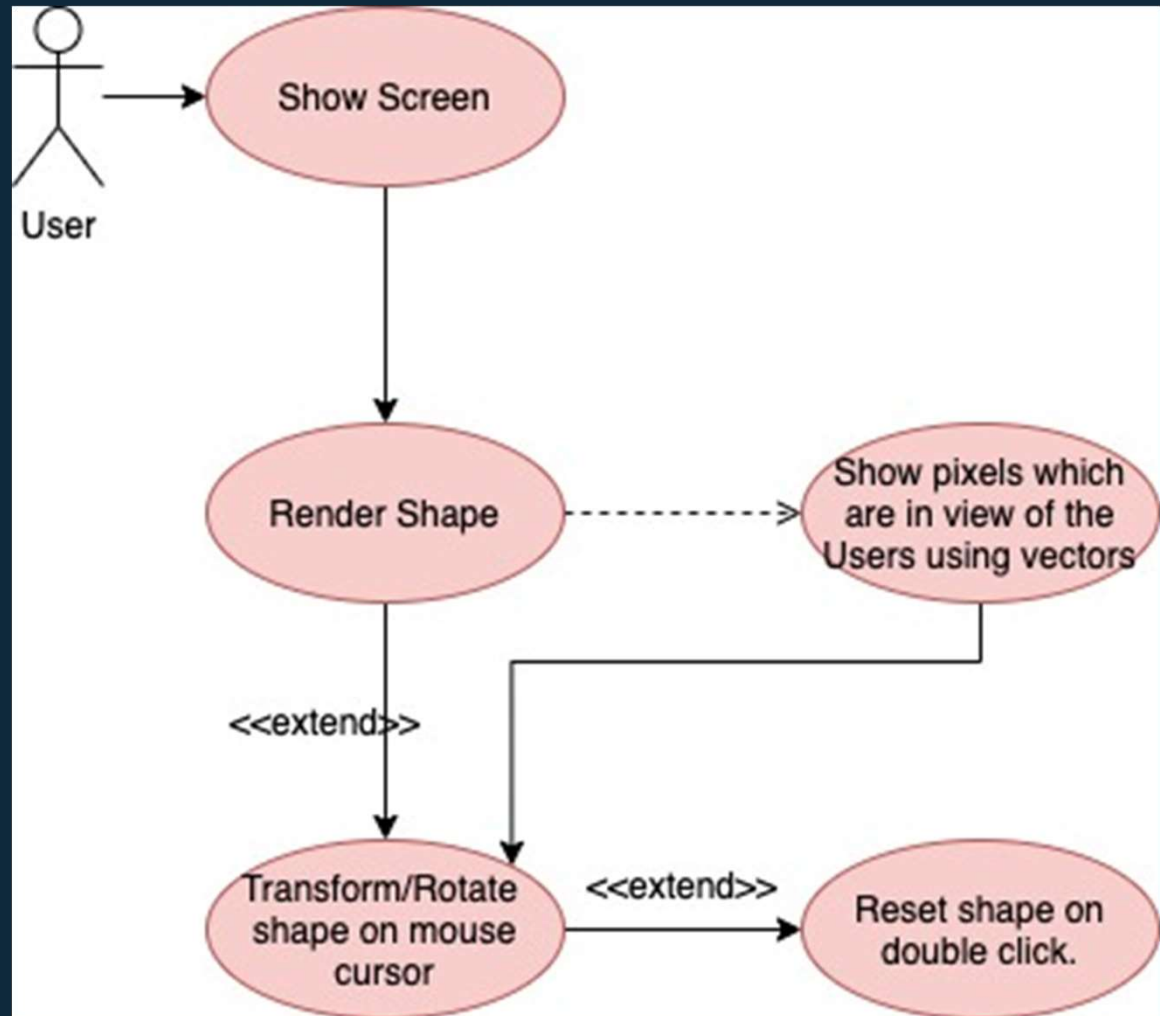
Features

- ◇ When we press and drag the mouse then cube performs rotations in the direction in which the mouse was dragged.
- ◇ When we double click on the mouse, the cube resets to its original state.
- ◇ On pressing space one can hide and show boundaries of the cube.
- ◇ We can zoom in and out of the screen to adjust the size of the cube.

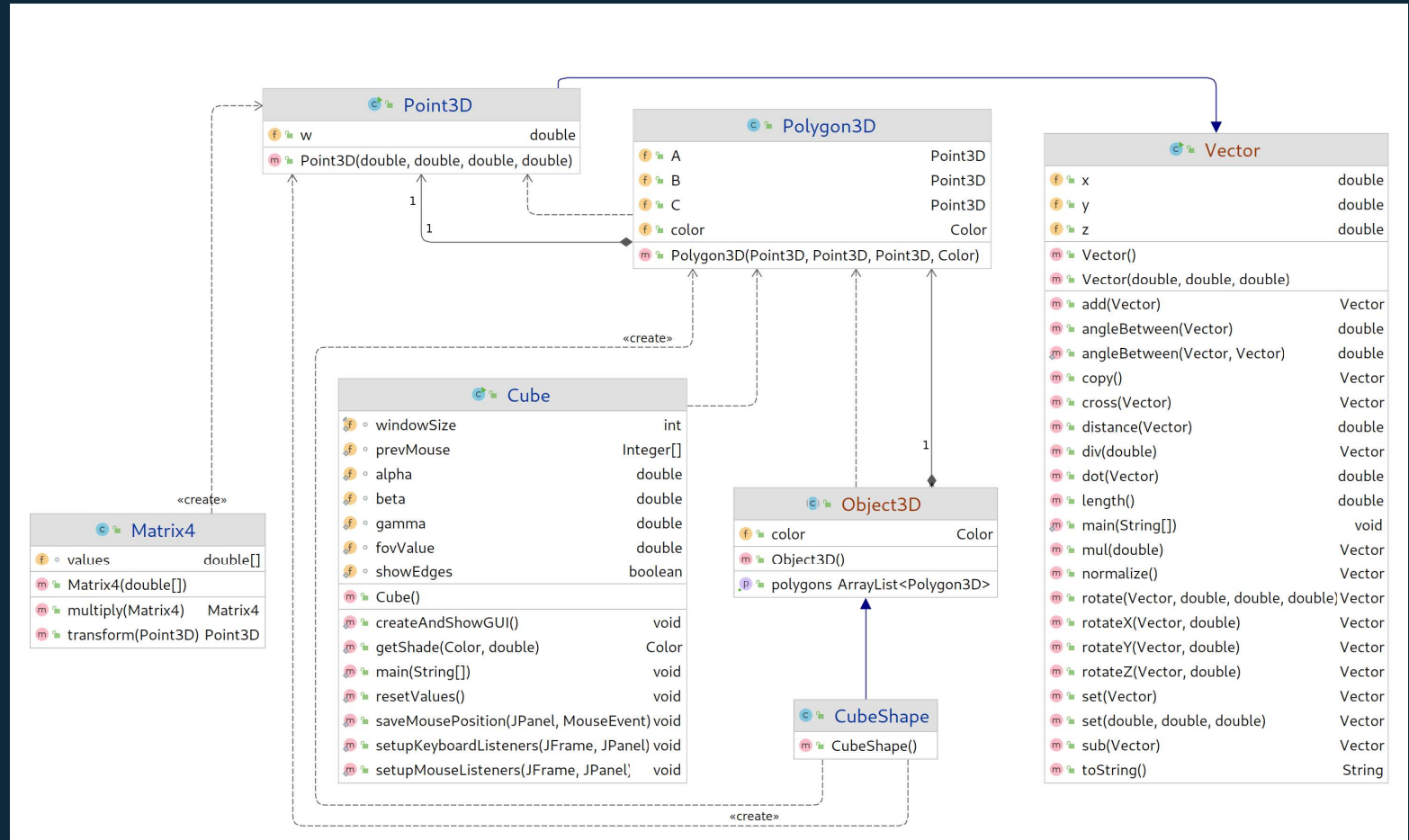




Use Case Diagram



Use ClassDiagram



CRC Diagram

Matrix4	
Responsibilities	Collaborators
Matrix multiplication Matrix Transformation	Point3D

Cube	
Responsibilities	Collaborators
Rendering the Cube Shading the Surface Detect and use Mouse Movements Detect and use Keyboard Input	Object3D Matrix4

Object3D	
Responsibilities	Collaborators
Collection of Triangles	Polygon3D

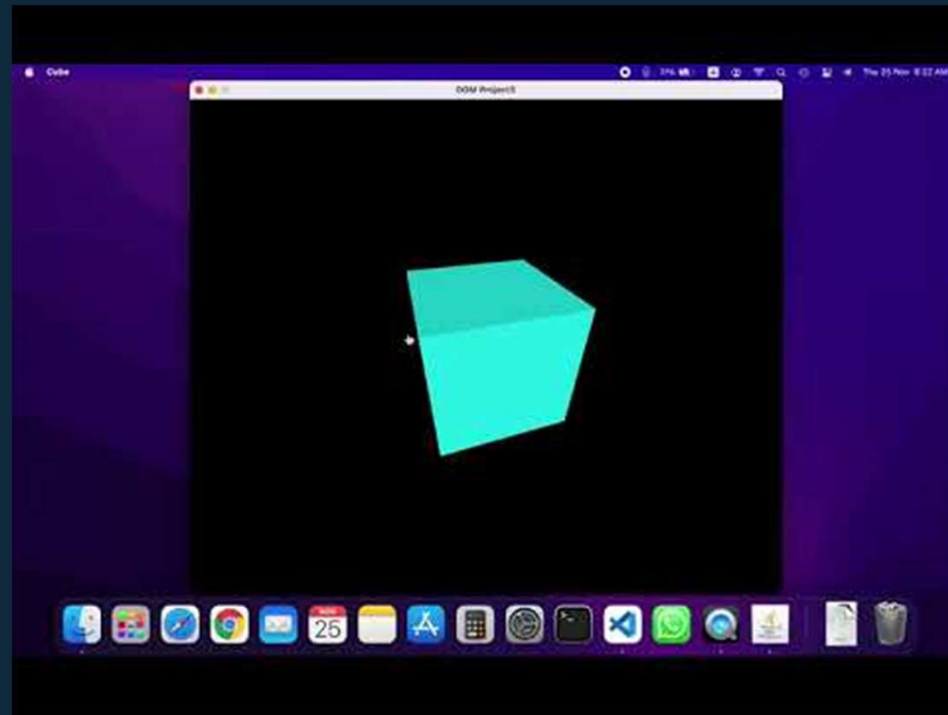
Vector	
Responsibilities	Collaborators
To Compute Vector Maths	

Point3D	
Responsibilities	Collaborators
Storing Coordinates in 3D space	Vector

Polygon3D	
Responsibilities	Collaborators
Storing Vertices of Triangle Storing the Color	Point3D Color



Demo Video





Thanks!

