

Indian Institute of Information Technology, Allahabad
Object Oriented Methodology (OOM)
Mini Project-3

Instructors: Prof. O. P. Vyas

OOM Mini Project #3: Development of 3D Computer Animations Through Matrix Operations

Project Abstract:

In this project, you are tasked with developing a **3D animation system** that demonstrates basic 3D transformations such as rotation, scaling, and translation. The goal is to apply matrix operations to move a 3D object (like a cube) across the screen, and animate its rotations as users drag the mouse. A **Z-buffer** will handle hidden face removal, while **double buffering** will ensure smooth animation rendering.

The system will feature both wireframe and solid shapes for rendering the 3D models. You will focus on designing classes that are cohesive and loosely coupled, ensuring that the system is flexible and maintainable. Key classes include **Object3d**, **Polygon3d**, and **Point3d**.

Specific Technology:

- **Programming Language:** Java
- **User Interface:** Swing, JSF
- **Data Storage:** XML (optional)

Project Tasks:

1. **Use Case Analysis:**
 - Identify the key interactions between the user and the 3D animation system.
 - Create use cases for rendering, rotating, and scaling 3D objects.
2. **Design:**
 - Design a well-structured class architecture with clear responsibilities. Focus on **high cohesion** within classes and **low coupling** between them.
 - Develop a **Class Diagram** and **CRC Cards** for the main classes (Object3d, Polygon3d, Point3d).
 - Design the **User Interface** using Swing components, allowing users to interact with the 3D object through mouse events.
3. **Implementation:**
 - Implement the 3D transformations using matrix operations.
 - Ensure the system supports both **wireframe** and **solid shape** rendering modes.
 - Implement the **Z-buffer** algorithm for hidden surface removal.

- Use **double buffering** to enhance the smoothness of animations.

Functional Components:

Your 3D animation system should include the following features:

- **3D Object Manipulation:** Allow the user to rotate, scale, and move the object using the mouse.
- **Rendering Modes:**
 - **Wireframe Mode:** Display the object's wireframe model.
 - **Solid Shape Mode:** Display the object as a solid, shaded shape.
- **Z-Buffer Implementation:** Ensure proper hidden surface removal for solid shapes.
- **Double Buffering:** Implement double buffering to avoid flickering and ensure smooth animations.

Submission Requirements:

1. **Design Artifacts:**
 - **UML Diagrams:**
 - **Use Case Diagram:** Capture the main interactions between the user and the 3D animation system.
 - **Class Diagram:** Show the relationships between the classes (Object3d, Polygon3d, Point3d, etc.).
 - **CRC Cards:** Illustrate Class-Responsibility-Collaborator diagrams for key classes, focusing on functional design with cohesion and low coupling.
2. **Code Submission:**
 - Submit all **Java source code** necessary to compile and run the 3D animation system.
 - Include clear comments in your code explaining your design decisions, particularly with regard to coupling and cohesion.
3. **Project Presentation:**
 - Each team member must periodically present the project progress.
 - The final presentation should demonstrate your system's functionality and explain your design choices, focusing on how you ensured **low coupling** and **high cohesion**.