

Om Deore

+91 7038551995 | omdeore890@gmail.com | [linkedin.com/in/om-deore](https://www.linkedin.com/in/om-deore) | github.com/om101deore

EDUCATION

Vishwakarma institute of information technology

Pune, MH

Bachelor of Technology in Computer Engineering

Dec. 2021 – May 2025 (expected)

- Cumulative Grade Point Average (CGPA): 8.8/10

Sharad Pawar College

Nashik, MH

12th Maharashtra State Board

Jun. 2019 – May 2021

- Percentage: 86.5%

LPDP Maratha English School

Satana, MH

10th Maharashtra State Board

Jun. 2019

- Percentage: 95.6%

SKILLS

Technical Skills: C/C++, OpenGL, Bash, Java

Developer Tools: Git, Linux, CMake, Vim

PROJECTS

Multiplayer Game using Socket Programming | C++, OpenGL, GLFW, Linux Sockets

[Github](#)

- Created Two player game using **OpenGL** for graphics and Linux sockets for communication between server and clients
- Connection between server and client is established by using **TCP** protocol between server and client. Server has to accept connection request from client and do three-way handshake before sending data. Both server and client send positions of their respective players
- These positions are sent every frame of game. As there is very less data to be transmitted, TCP works as good and fast as UDP
- OOP is used to solve design problems with help of runtime polymorphism and virtual fun

3D Illuminated Scene | C++, OpenGL, GLFW, GLM

[Github](#)

- Developed a visually immersive voxel scene of a room using **OpenGL**. Implemented voxel rendering and scene setup leveraging OpenGL for efficient graphics rendering and manipulation.
- Utilized **Phong lighting technique** to create realistic light effects. Employed OpenGL shaders to implement the Phong lighting model, accurately simulating ambient, diffuse, and specular lighting.
- Leveraged **GLFW** for window and input handling, ensuring smooth user interaction. Implemented event handling and input management using **GLFW**, enabling intuitive navigation and control within the voxel scene.

Maze Solver | Java, Swing, AWT

[Github](#)

- Developed a maze solver application in Java using **AWT**. Implemented a grid-based maze structure and designed a tracker to navigate the maze using Depth-First Search (DFS) and Breadth-First Search (BFS) algorithms.
- Utilized Depth-First Search (DFS) algorithm to explore possible paths in the maze. Implemented DFS traversal logic to systematically explore paths within the maze, marking cells and backtracking when necessary to find the optimal solution.
- Implemented Breadth-First Search (BFS) algorithm for path finding efficiency. Utilized BFS to explore alternative paths in a breadth-first manner, ensuring optimal path finding and efficient maze-solving capabilities.