# Minecraft 3D Castle Viewer

## INTRODUCTION

This project demonstrates rendering a textured 3D castle model inside a Python application using Pygame and OpenGL 3.3. It features a live video background, smooth camera controls, and audio playback to create an immersive exploration experience.

### How to Run and Use the Program:

1. Run the program by executing the main Python script:

**python main.py**

1. **Auto mode** will play for 25 seconds after opening/running the main.py and will then switch to manual mode **automatically**.
2. Explore the 3D scene using the following controls:

* Toggle Auto/Manual Camera Mode: **Press M**
* Auto mode cycles through preset camera views automatically.
* Manual mode lets you control the camera view.
* Navigate Views (Manual Mode):
* Press **Arrow keys** or **WASD** to cycle through available views.
* Press **number keys** **1–9 or 0** to jump directly to a specific view.
* Press **R** or **Spacebar** to jump to a random view.
* Camera Zoom: Use the mouse scroll wheel to zoom in and out within limits.
* Camera Rotation: Click and drag the left mouse button to rotate the camera smoothly.
* Adjust Camera Movement Speed: Press + to increase and - to decrease the smoothness of camera motion.

1. Exit the program by closing the window.

## OVERVIEW

This project renders a 3D castle model over a live video background inside a Pygame window using OpenGL. It includes smooth camera controls, automatic and manual view modes, zooming, rotation, and background music playback. The background video is played using OpenCV and displayed as an OpenGL texture mapped onto a full screen quad. The 3D castle model is loaded from custom vertex and index files, textured, and rendered using GLSL shaders.

## FEATURES

* 3D model rendering with textures using OpenGL 3.3
* Live video background rendered with OpenCV frames
* Multiple camera views with automatic cycling and manual control
* Smooth mouse-controlled camera rotation and zooming
* Background music playback with intro and loop tracks
* Zoom limit feedback with beep sound

## REQUIREMENTS

* Python 3.x
* Pygame
* PyOpenGL
* Pillow (PIL)
* NumPy
* OpenCV (cv2)
* pyglm

## PROJECT STRUCTURE

* `main.py` — Main application and rendering loop
* `model\_loader.py` — Loads 3D model vertex and index data into OpenGL buffers
* `texture\_loader.py` — Loads and prepares textures for OpenGL
* `textured\_shader.py` — Shader source code and compilation for 3D textured model
* `bg\_loader.py` — Video background shader, quad creation, and texture updating
* `source/` — Assets (models, textures, videos, sounds)

## CONTROLS SUMMARY

|  |  |
| --- | --- |
| **Control Key/Button** | **Action** |
| M | Toggle auto/manual camera mode |
| Arrow keys / WASD | Cycle through views (manual mode) |
| Number keys 1–9, 0 | Jump to specific view |
| R or Spacebar | Random view |
| Mouse Wheel | Zoom camera in/out |
| Left Mouse Drag | Rotate camera |
| + / - | Adjust camera smoothness (lerp speed) |
| Window Close | Exit program |