

Report for CGR Coursework S2810619

1. Implemented Features

Module 1: Camera, Exporter, and Image I/O

- Implemented a **Python Blender exporter (Export.py)** that extracts camera, light, and object data from Blender and writes them to an ASCII JSON file (`scene.json`).
- Implemented **Camera class (camera.h / camera.cpp)**:
- Loads all camera parameters (location, gaze, up, focal length, sensor size, resolution) from JSON.
- Computes orthonormal basis vectors (`gaze`, `right`, `up`) using cross products and normalization.
- Implements `generateRay(px, py)` to map pixel coordinates to 3D world-space directions via a pinhole camera model.
- Implemented **Image class (image.h / image.cpp)**:
- Supports reading and writing **ASCII PPM (P3)** images.
- Includes `setPixel()` for per-pixel color editing.
- Verified correctness by writing a color-coded ray direction map (`dirs.ppm`).

Module 2: Scene Parser (partial)

- Implemented **Scene class (scene.h / scene.cpp)** capable of reading lights, spheres, cubes, and planes from the exported JSON.
- Verified parser output via console printouts in `main.cpp`.
- Implemented a **test renderer** in `main.cpp` that:
- Loads camera and scene data.
- Iterates over all pixels, generates rays, and visualizes ray directions as RGB colors.
- Outputs a test image (`Output/dirs.ppm`).

2. Testing and Debugging

- **Integration testing:** Combined modules through `main.cpp` and compared console logs to expected object counts from Blender.
- **Visual debugging:** The generated image (`dirs.ppm`) was visually verified—color gradients correspond correctly to ray directions (red = x-axis, green = y-axis, blue = z-axis).
- **Error handling:** All file operations use error checks (e.g., cannot open file).