RayTracer Project - Build & Usage Documentation

CSE 461 - Computer Graphics

Environment Requirements

This project is designed to be built and run on **Ubuntu Linux**. Ensure you are working in a Unix-based environment with the following tools installed:

- `g++` with C++17 support

- GNU Make

- A terminal or shell capable of running `make`, `bash`, and standard Linux utilities

- Makefile is provided

Tested on Ubuntu 22.04 LTS and compatible with most modern Debian-based distributions.

1 Compiling the Project

To compile the main raytracer and test runner binaries, use the following command:

```
make
```

This will:

- Compile the source files from `src/` and `lib/`

- Place object files in `build/obj/`

- Generate the final executable `raytracer` at:
  ```
  build/release/raytracer
  ```

2 Running the RayTracer

You can run the raytracer manually by providing:

- A scene XML file path

- An output PNG file path

- A render mode: `single` or `multi`

Example command:

```
./build/release/raytracer ./assets/scenes/scene_low_tree.xml outputs/
    ↪ output_tree.png multi
```

This will render `scene_low_tree.xml` using multithreaded mode and save the output to `outputs/output_tree.png`.

3   Output Directory

Rendered images are saved in the `outputs/` directory. This folder is automatically created if it does not exist.

4   Running System Tests

To build and run the test runner (which automatically renders all scenes in `assets/scenes/`):

```
make tests
```

This will:

- Build the test binary at `build/tests/test_runner`

- Run the binary

- Render all scenes with multithreading

- Save output images as:
  ```
  outputs/output_scene_<scene_name>_<timestamp>.png
  ```

Example generated output:

```
outputs/output_scene_scene_low_tree_20250412_154015.png
```

5   Cleaning Up

To remove all compiled files and outputs:

```
make clean
```

To remove only output images:

```
make clean_outputs
```

To remove only object files:

```
make clean_obj
```

Notes

- If `assets/scenes/` is missing, the test runner will exit with an error.

- If `assets/textures/` is missing, a warning will be shown, but the test will proceed.

- All timestamps are generated based on the local system time.