

Path Tracer Project

Learning Outcomes:

- Understand the basics of path tracing and Monte Carlo methods.
- Implement ray generation for perspective cameras.
- Implement basic materials like lambertian materials and perfectly-smooth metals.
- Implement ray intersection with simple shapes such as spheres and triangles.
- Implement reinhard tonemapping and gamma correction.
- Implement a simple path tracer.

Assignment Requirements

First, modify `run-all.ps1` such that the last scene is `Your First Name` in lowercase followed by an underscore followed by your student ID. For example, if your name is `Ahmed Mohamed Mahmoud` and your student ID is `123456789`, then the last scene should be `ahmed_123456789`.

Then, there are a set of TODOs in the code that you need to complete. These are marked with `TODO:` comments. You should fill these out according to the instructions provided.

The TODOs can be found in the following files:

- `color.hpp`
- `material.cpp`
- `shapes.cpp`
- `camera.cpp`
- `pathtracer.cpp`

After correctly implementing, run the `run-all.ps1` script.

Finally, submit the following files zipped together with the archive name being your student ID:

- `color.hpp`
- `material.cpp`
- `shapes.cpp`
- `camera.cpp`
- `pathtracer.cpp`
- The folder `output` containing all the output images.

Hints




- In the folder `expect_output`, you will find outputs for all the scenes except the last one.
 - You can visually compare your results with these images to ensure that your implementation is correct.
 - The noise in the images won't exactly match with your output since the path tracing process is inherently stochastic.
- If the project is built in debug mode, 1000 samples will probably be slow.

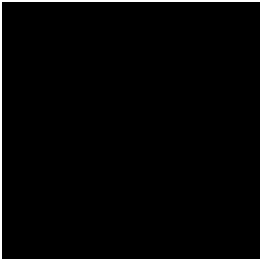
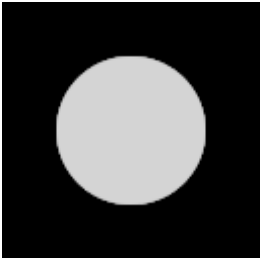
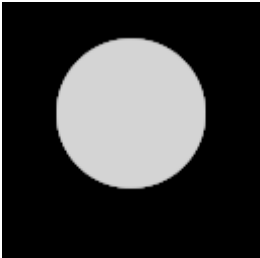
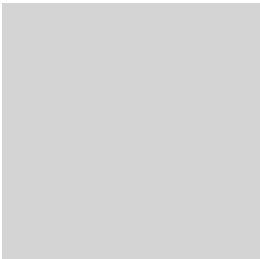
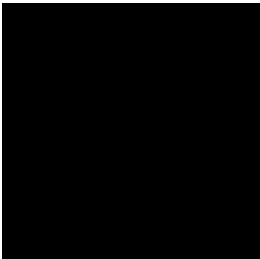
- Decrease the number of samples while debugging.
- After you are done, build the project in release mode and increase the number of samples to 1000 again.
- The project implements a BVH to accelerate ray tracing. The BVH could make debugging harder.
 - You can disable it using the `--nobvh` or `-b` flag.
 - Also, you can change the default BVH config in the top of the `main` function.
- The project implements some debug modes that you may find helpful while debugging.
 - You can enable them using `--debug MODE` or `-d MODE` flag, where MODE can be `distance` or `normal`.
 - Also, you can change the default debug config in the top of the `main` function.
- Feel free to change the default config at the top of the `main` function during development, then return them back when you are done.

Scenes

The project includes a set of scenes that you can use for testing. These include:

Single Shape Scenes

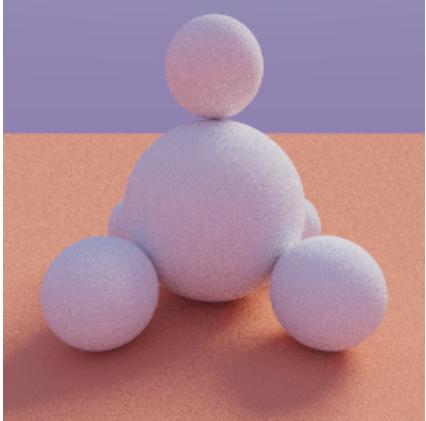
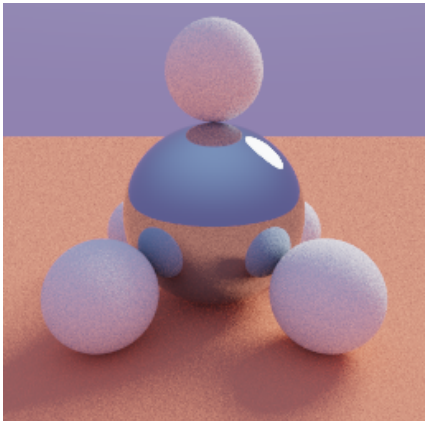
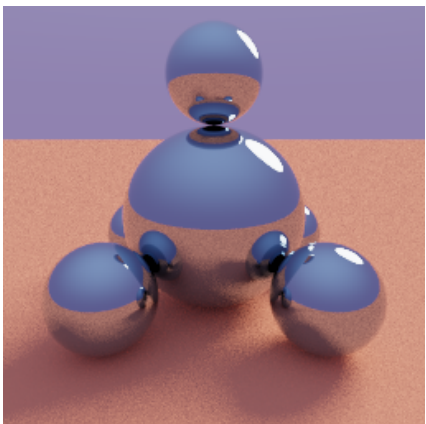
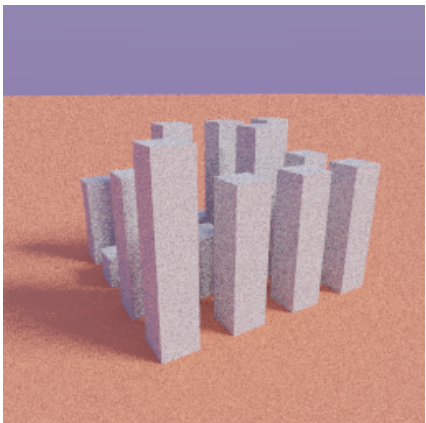
Scene Name	Description	Expected Output
<code>tri_test0.0</code>	A triangle orthogonal to the camera direction.	
<code>tri_test1.0</code>	A triangle tilted up slightly.	
<code>tri_test2.0</code>	A triangle paralalled to the camera direction.	

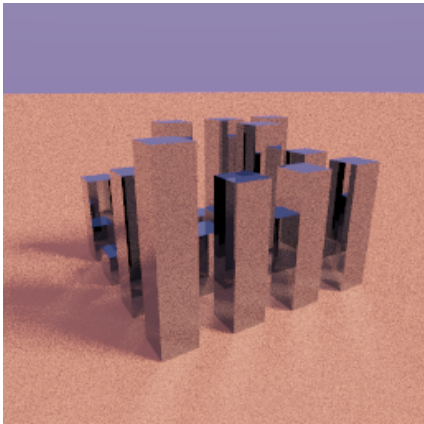
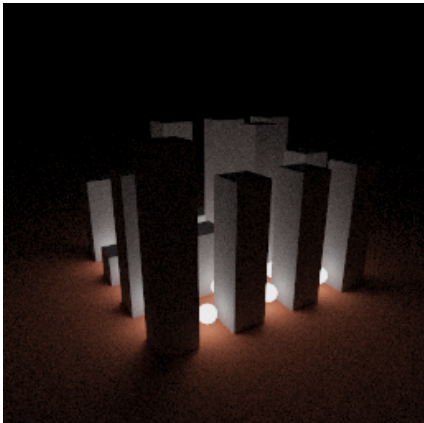
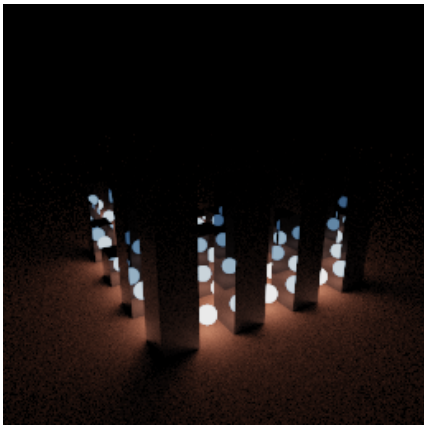
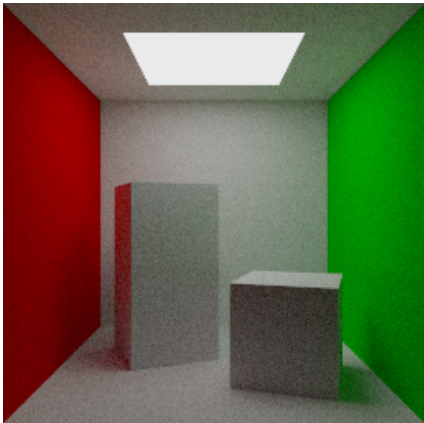
Scene Name	Description	Expected Output
<code>tri_test3.0</code>	A triangle behind the camera.	
<code>sph_test0.0</code>	A triangle orthogonal to the camera direction.	
<code>sph_test1.0</code>	A triangle tilted up slightly.	
<code>sph_test2.0</code>	A triangle paralled to the camera direction.	
<code>sph_test3.0</code>	A triangle behind the camera.	

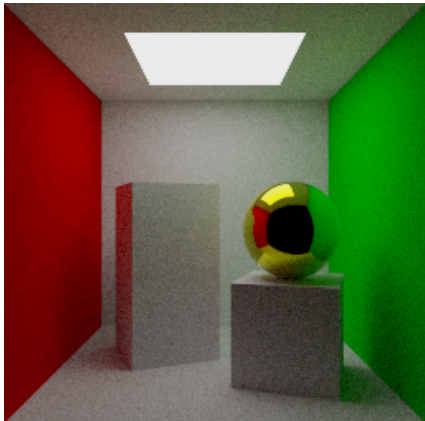
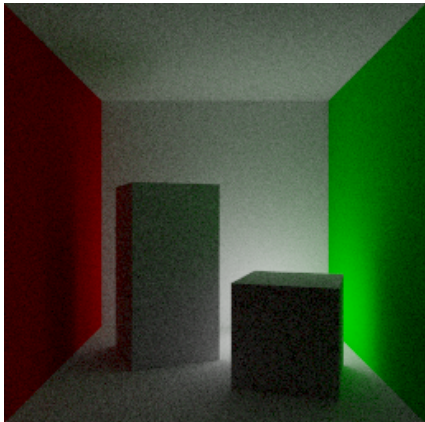
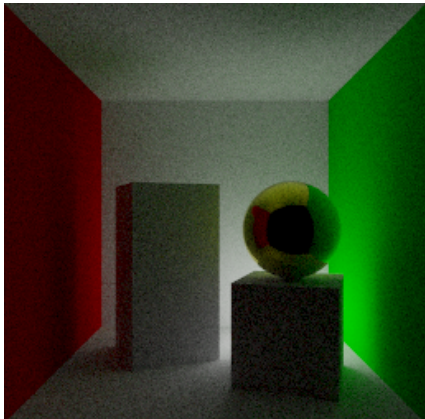
Note: Scenes `tri_test0.0`, `tri_test1.0`, `tri_test1.0`, `tri_test1.0`, `sph_test0.0`, `sph_test1.0`, `sph_test2.0`, and `sph_test3.0` are similar to the ones above but with a resolution of 4x4.

Complex Scenes

Scene Name	Description	Expected Output
------------	-------------	-----------------

Scene Name	Description	Expected Output
balls0	A group of lambertian balls under a day sky.	
balls1	A mix of lambertian and smooth metallic balls under a day sky.	
balls2	A group of smooth metallic balls under a day sky.	
city0	A group of lambertian blocks under a day sky.	

Scene Name	Description	Expected Output
city1	A group smooth metallic blocks under a day sky.	 A 3D rendering of several rectangular metallic blocks of varying heights and widths, arranged in a cluster. They are placed on a reddish-brown ground plane under a clear, light blue sky. The blocks have a highly reflective, metallic surface, showing clear highlights and reflections on the ground.
city2	A group of lambertian blocks lit by emissive balls.	 A 3D rendering of several rectangular blocks of varying heights and widths, arranged in a cluster. They are placed on a dark ground plane. The blocks are lit from below by several small, glowing white spheres (emissive balls). The blocks have a matte, lambertian surface, and the scene is dark, with the primary light source being the emissive balls.
city3	A group of smooth metallic blocks lit by emissive balls.	 A 3D rendering of several rectangular metallic blocks of varying heights and widths, arranged in a cluster. They are placed on a dark ground plane. The blocks are lit from below by several small, glowing white spheres (emissive balls). The blocks have a highly reflective, metallic surface, showing clear highlights and reflections on the ground. The scene is dark, with the primary light source being the emissive balls.
cornell_box0	A simple Cornell box.	 A 3D rendering of a simple Cornell box. The box is a rectangular prism with a white floor and ceiling. The left wall is red, the right wall is green, and the back wall is white. A small white rectangular light source is mounted on the ceiling. Two gray rectangular blocks are placed inside the box: one tall and thin, and one shorter and wider.

Scene Name	Description	Expected Output
cornell_box1	A Cornell box with a golden ball.	 A Cornell box with a red left wall, a green right wall, and a white back wall and floor. A white rectangular block is on the left, and a smaller white cube is on the right. A golden ball with a black center is on top of the cube. A bright light source is on the ceiling.
cornell_box2	A simple Cornell box lit by an emissive ball.	 A Cornell box with a red left wall, a green right wall, and a white back wall and floor. A white rectangular block is on the left, and a smaller white cube is on the right. A golden ball with a black center is on top of the cube. The scene is dimly lit, with the walls and floor showing some color bleeding from the walls.
cornell_box3	A Cornell box with a golden ball and lit by an emissive ball.	 A Cornell box with a red left wall, a green right wall, and a white back wall and floor. A white rectangular block is on the left, and a smaller white cube is on the right. A golden ball with a black center is on top of the cube. The scene is dimly lit, with the walls and floor showing some color bleeding from the walls. The golden ball is also lit by an emissive light source.