COMP4490 Graphics 2 Project Report: Ray Tracing Extension

***Abstract*— This report presents a detailed explanation of the implementation of various techniques for accelerating ray tracing, such as bounding volume hierarchies, along with camera modeling and non-photorealistic shading models. Additionally, the report covers the implementation of Schlick's approximation and super sampling anti-aliasing to improve the rendering quality of the scene. The methods used in the implementation are described in detail with their respective benefits and limitations. The report concludes with the performance analysis of the implemented techniques and their impact on the overall rendering speed and quality.**

Keywords—Bounding Volume Hierarchies, Camera, Non-photorealistic shading models, Schlick's Approximation, Super-Sampling Antialiasing.

I. Introduction

This project comprises 9 different scenes, including 2 newly designed scenes and 7 pre-existing ones, The rendered images of each scene are included at the end of the report. The aim of this report is to provide a detailed account of the rendering time required for each scene, as well as a comprehensive explanation of the implementation of various features. The report will delve into the specific techniques used to accelerate the rendering process, such as bounding volume hierarchies, camera modeling, and non-photorealistic shading models. It will also describe the implementation of Schlick's approximation and super sampling anti-aliasing to enhance the quality of the rendered scenes. Overall, this report aims to provide a thorough understanding of the implementation and performance of the features used in the project. For your convenience, we have included larger versions of all the images in this report at the bottom of the page.

II. Ray Tracing Performance and Feature

A. Ray Tracing Performance

Scene	Without BVH	With BVH	Without BVH & With Anti-Aliasing	With BVH & With Anti-Aliasing
A	361.703 s	39.257 s	1835.976 s	1365.890 s
B	398.393 s	78.022 s	4210.320 s	2766.724 s
C	27.104 s	36.732 s	105.384 s	97.698 s
D	19.546 s	28.501 s	146.675 s	73.976 s
E	64.788 s	72.597 s	312.532 s	276.668 s
F	19.433 s	26.604 s	90.223 s	67,961 s
G	79.772 s	101.049 s	593.413 s	312.366 s
H	56.607 s	100.621 s	512.623 s	223.621 s
I	253.527 s	45.520 s	1716.342 s	1066.886 s

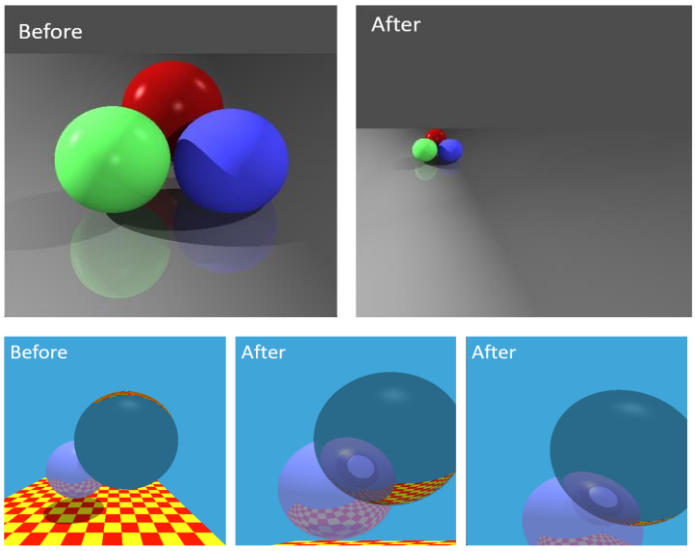
In analyzing the results presented above (a more clear image include at the bottom of report), we observe that the rendering of scenes without BVH trees is generally faster than those with BVH trees, particularly in scenes with fewer objects, such as scenes c, d, e, f, g, and h. However, for scenes with a larger number of objects, such as scenes a, b, and i, utilizing BVH trees for ray intersection checks will result in a much faster rendering speed. It is also important to note that including four samples for super-sampling antialiasing will inevitably slow down the rendering speed, albeit enhancing the overall quality of the rendered images. Therefore, in deciding which features to include in a given rendering task, it is crucial to consider the specific characteristics of the scene and prioritize accordingly.

B. Bounding Volume Hierarchies

In the implementation, a top-down version of the bounding volume hierarchy (BVH) tree was utilized, which was heavily inspired by Peter Shirley's book "Ray Tracing: The Next Week" [1]. To construct the BVH tree, the function buildBVH() in bvh.cpp was called, which takes a list of objects as input. The recursive algorithm subdivide() is then invoked, which subdivides the longest axis and splits the screen into two spaces, storing it in a binary tree format. Finally, the root node of the BVH tree is returned. As we can see in the previous performance table, this implementation allows for efficient ray-object intersection checks, greatly improving the rendering speed for scenes with a large number of objects.

C. Camera Position and Direction

In implementing the Camera feature, two new vectors were introduced, namely "lookFrom" and "lookAt", which is also inspired by the book by Peter Shirley[1]. The "lookFrom" vector replaces the previously used eye vector, while the "lookAt" vector is used as an offset for the screen vertex. By binding the "lookFrom" and "lookAt" vectors to keys, the camera position and direction can be easily changed. Several result images were generated to showcase the effect of changing the camera position and direction on the rendered scenes. These images are included below.

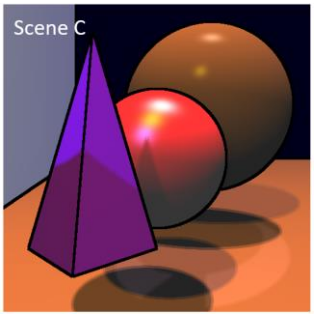


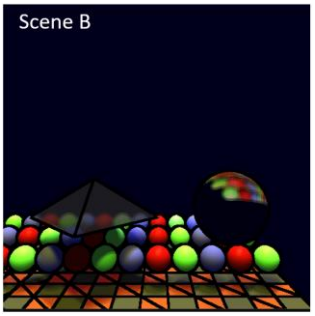
D. Non-Photorealistic Shading Models

Our implementation enables the use of non-photorealistic shading models, which can give objects a unique and artistic appearance. We have included three different types of shading models to achieve this effect.

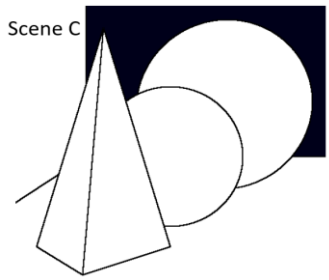
The first type involves outlining all objects to create a cartoon-like appearance. This approach has been successfully

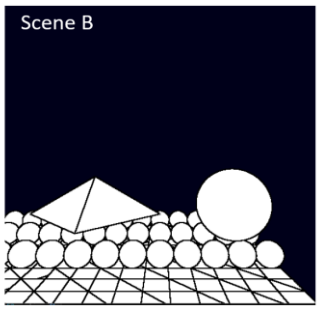
applied and resulted in images that resemble cartoons. You can see some examples of the output in the attached pictures.



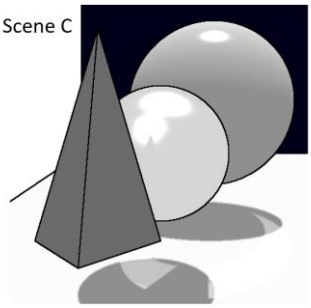


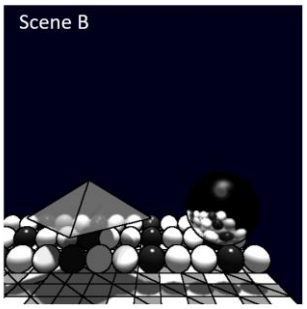
The second type focuses on adding outlines to the scenes, giving them a hand-drawn appearance. This technique produces stunning results, as you can see in the examples we've included.

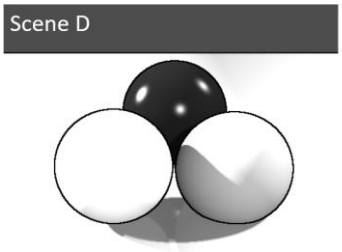




The third and final type of shading we've implemented creates a hand-sketched appearance. This approach gives scenes a more personal and artistic feel. We've also included some examples of this type of shading for you to see.







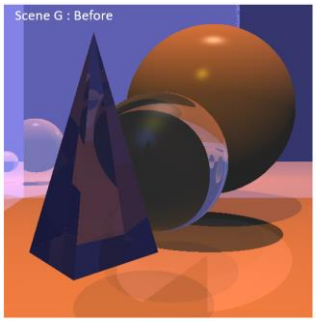
Overall, our non-photorealistic shading models allow for a range of unique and interesting visual effects that can be used to create captivating and creative scenes.

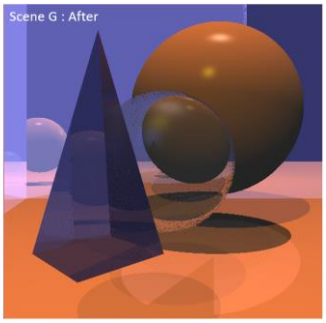
E. Schlick's Approximation

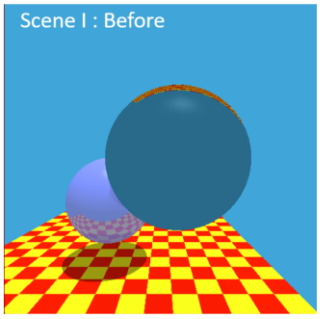
Our rendering engine now includes Schlick's approximation for refraction as its third feature. We drew inspiration for this implementation from Peter Shirley's book, specifically Chapter 10. By incorporating this feature, objects rendered using our engine can now appear more realistic, mimicking the appearance of glass.

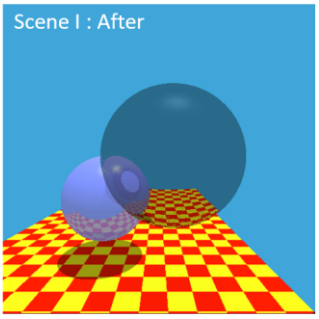
The implementation for this feature is straightforward. We simply added a reflectance calculation and compared it to a random float number. The ray is only allowed to pass through if the reflectance value is larger than the random float number generated.

We have included some stunning images that showcase the results of this implementation. You can see how the objects appear to have a transparent and reflective surface, creating a more immersive and believable scene. This feature adds to the realism of our rendering engine and enables the creation of visually-compelling scenes.







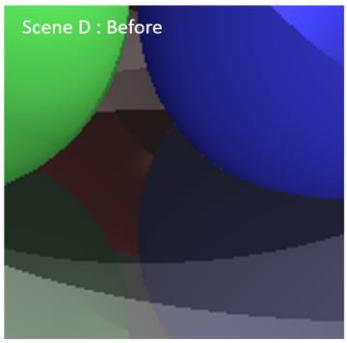


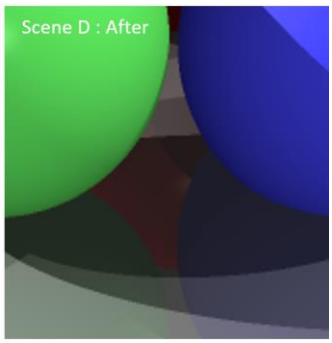
In summary, Schlick's approximation for refraction is a valuable addition to our rendering engine, elevating the quality and realism of our rendered scenes. We are excited to see the creative possibilities this feature will unlock for our users.

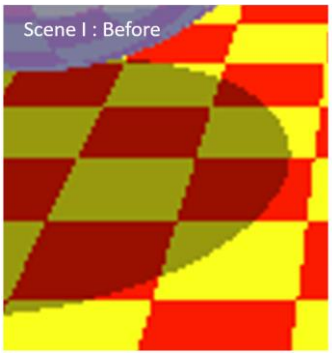
F. Super-Sampling Antialiasing

The final feature we've implemented in this project is super-sampling antialiasing. Due to the limitations of computer hardware, we've applied a 4-sample super-sampling technique in this implementation. This technique involves casting four rays for each pixel and then averaging the resulting colors to smooth out the edges of objects and eliminate blocky appearance.

We have included some images that showcase the results of this implementation. You can see how the objects appear smoother and more natural, without any distracting jaggies or aliasing artifacts.







Although we've implemented only 4 sample super-sampling in this version, we plan to expand this feature in future iterations to allow for even more precise and detailed renders.

Overall, this feature enhances the quality of our rendering engine and enables users to create smoother and more visually pleasing scenes. We look forward to seeing the creative possibilities that this feature will unlock for our users.

III. Conclusion

After an extensive exploration of ray tracing intricacies, this project has provided a fantastic opportunity for growth and learning. While significant progress has been made, there is still a vast potential for further improvement. One area that requires attention is the correction of Schlick's approximation calculation's flaws, which would significantly enhance the accuracy of the engine's output. Moreover, upgrading the acceleration structure to bottom-up BVH and tight bound, and moving the ray intersection calculation to the shaders file would enable us to leverage GPU acceleration better. Furthermore, expanding the variety of materials that we can render would be a worthy pursuit, such as implementing conductor materials. As we continue to work on this project, we are excited to explore new techniques and push the boundaries of what is possible with ray tracing. We would like to express our gratitude for this opportunity to work on this project and look forward to seeing the future developments of this rendering engine.

References

[1] Shirley, P. (n.d.). Ray Tracing in One Weekend Series.” Ray Tracing in one weekend series. Retrieved April 24, 2023, from https://raytracing.github.io/books/RayTracingTheNextWeek.html

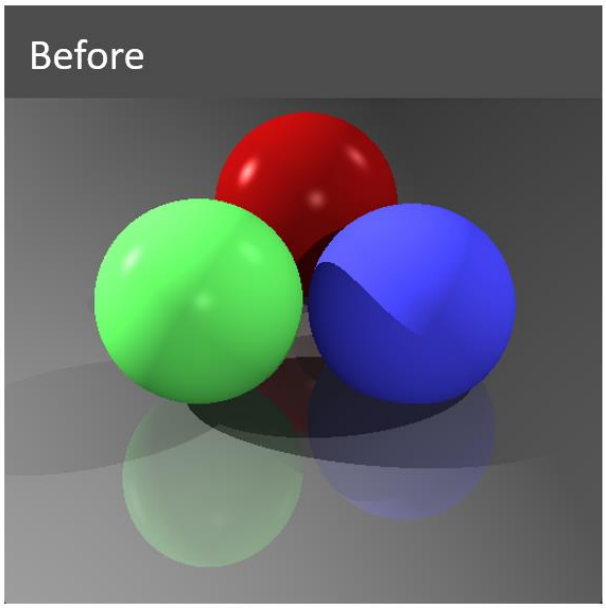
[2] Fussell, D. (n.d.). Anti-aliased and accelerated ray tracing - University of Texas at Austin. Retrieved April 24, 2023, from https://www.cs.utexas.edu/users/fussell/courses/cs384g-fall2011/lectures/lecture10-Aa_and_accel_raytracing.pd

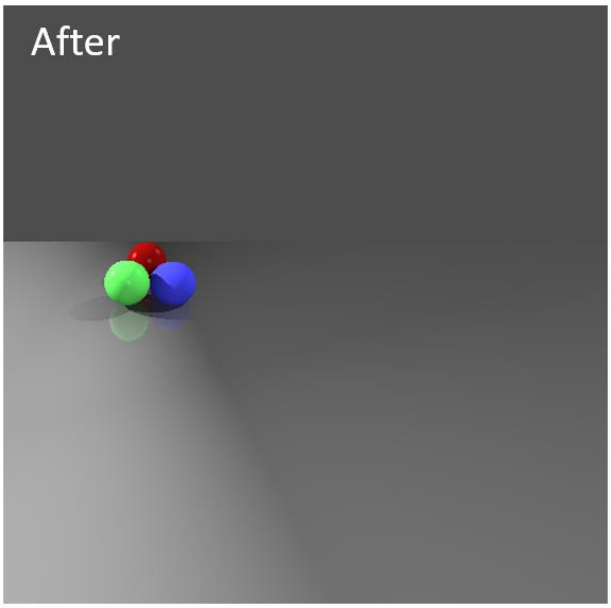
Larger version of all Images Below.

Ray Tracing Performance Table

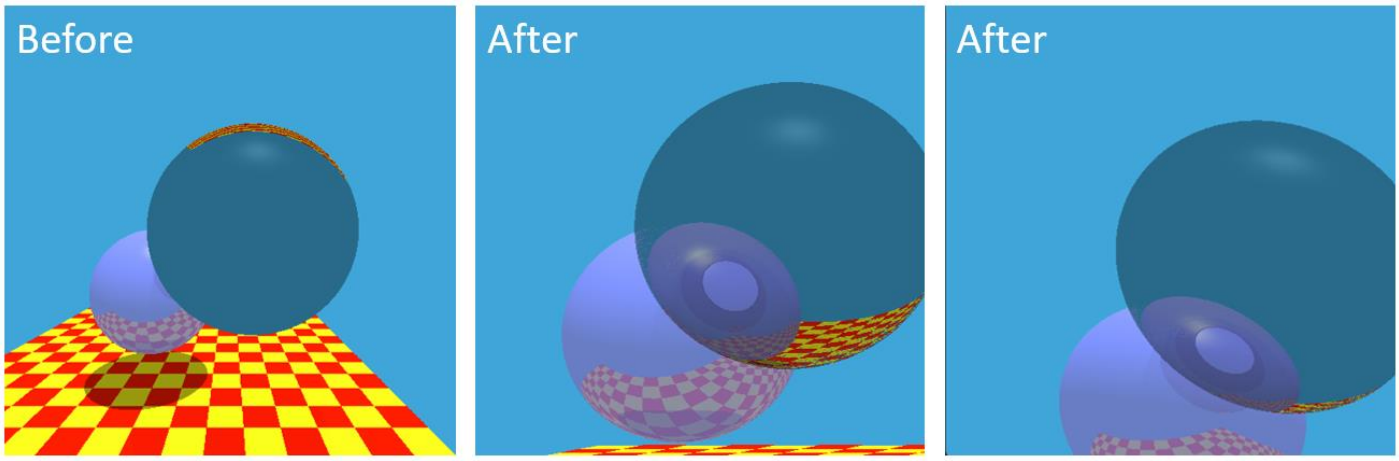
Scene	Without BVH	With BVH	Without BVH & With Anti-Aliasing	With BVH & With Anti-Aliasing
A	361.703 s	39.257 s	1835.976 s	1365.890 s
B	398.393 s	78.022 s	4210.320 s	2766.724 s
C	27.104 s	36.732 s	105.384 s	97.698 s
D	19.546 s	28.501 s	146.675 s	73.976 s
E	64.788 s	72.597 s	312.532 s	276.668 s
F	19.433 s	26.604 s	90.223 s	67,961 s
G	79.772 s	101.049 s	593.413 s	312.366 s
H	56.607 s	100.621 s	512.623 s	223.621 s
I	253.527 s	45.520 s	1716.342 s	1066.886 s

Camera Position Sample : Scene D

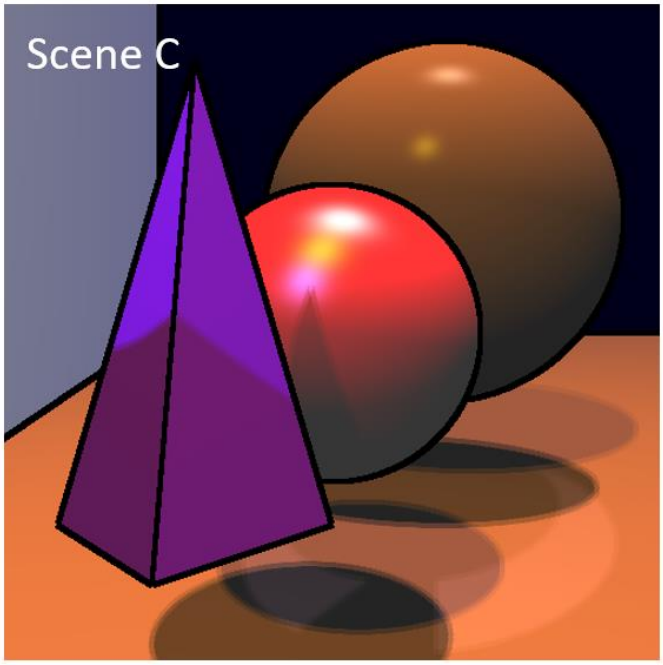


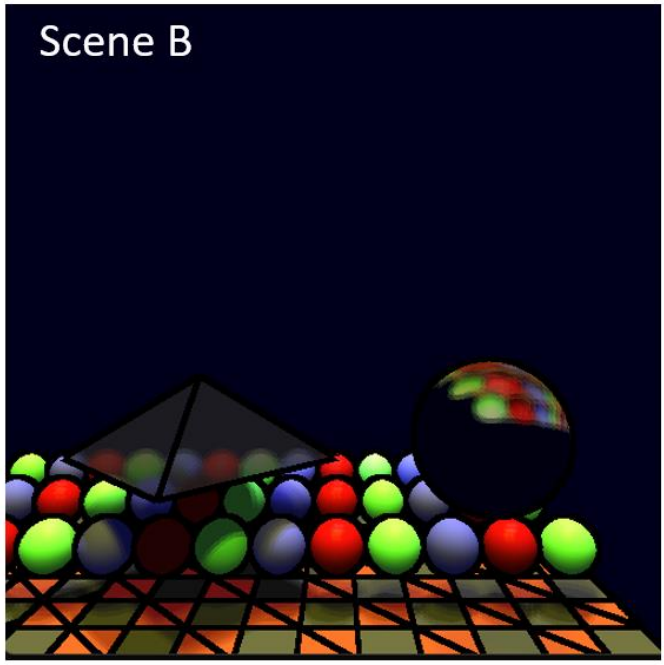


Camera Position Sample : Scene I

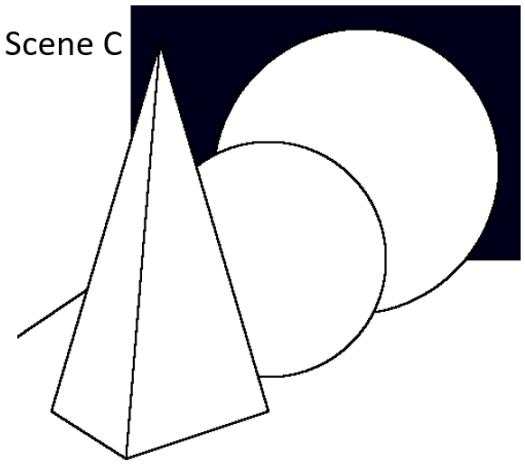


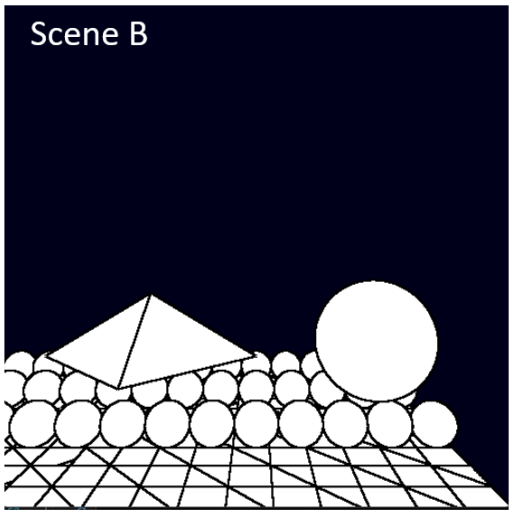
Non-Photorealistic Shading Type 1 Sample : Scene B and C



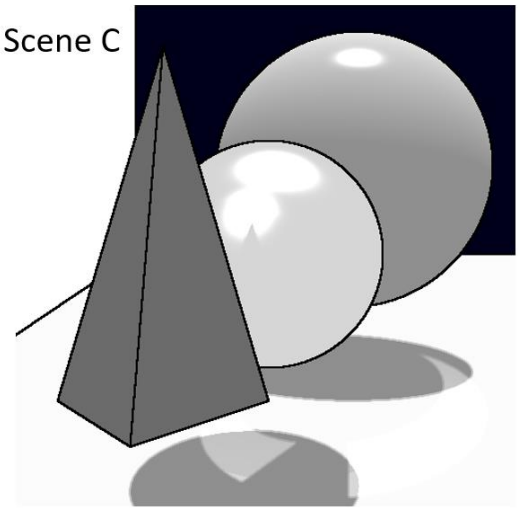


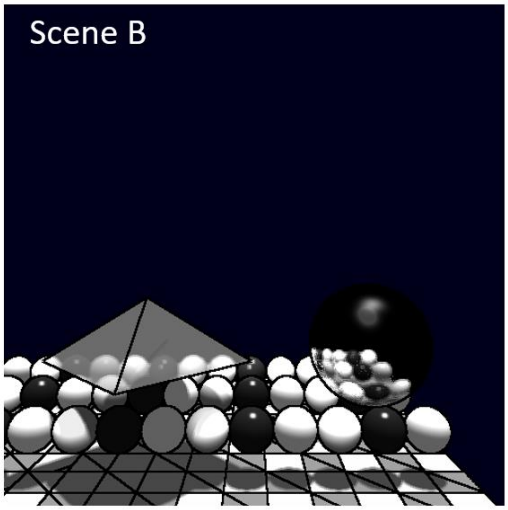
Non-Photorealistic Shading Type 2 Sample : Scene B and C

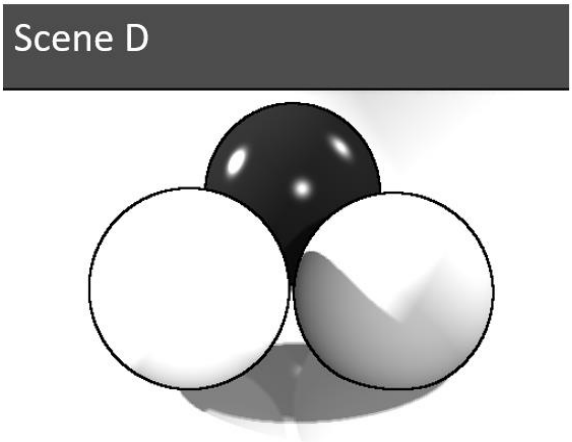




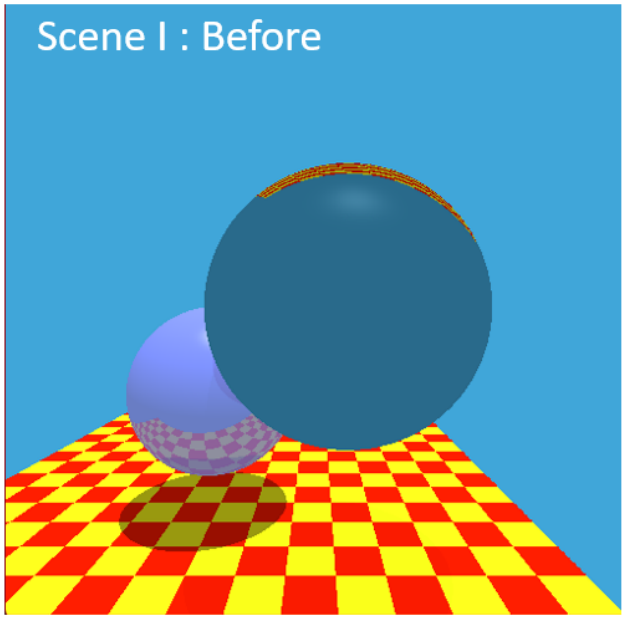
Non-Photorealistic Shading Type 3 Sample : Scene B, C and D

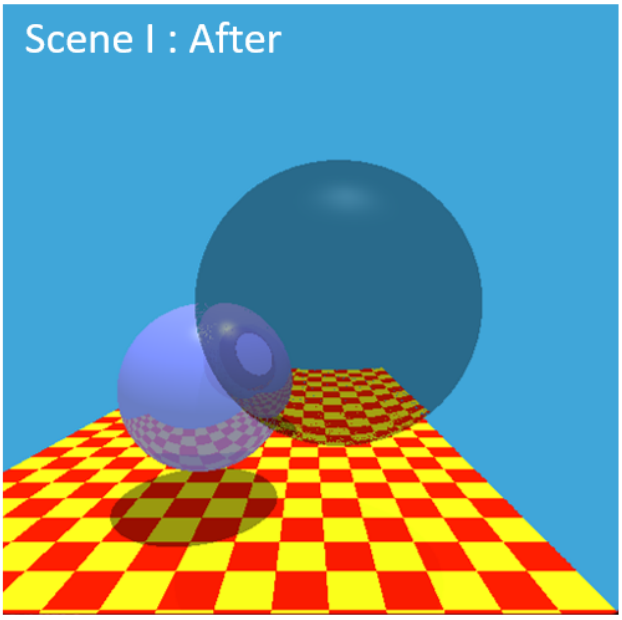






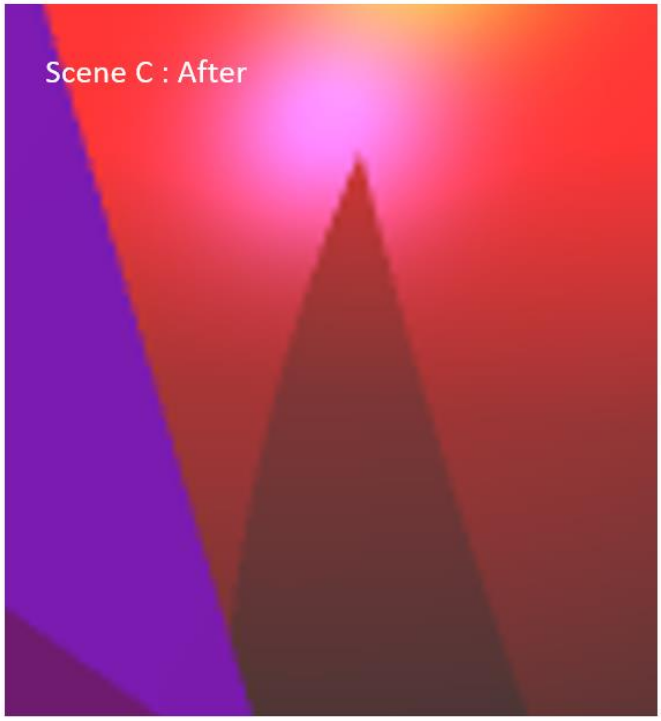
Schlick's Approximation Sample : Scene I



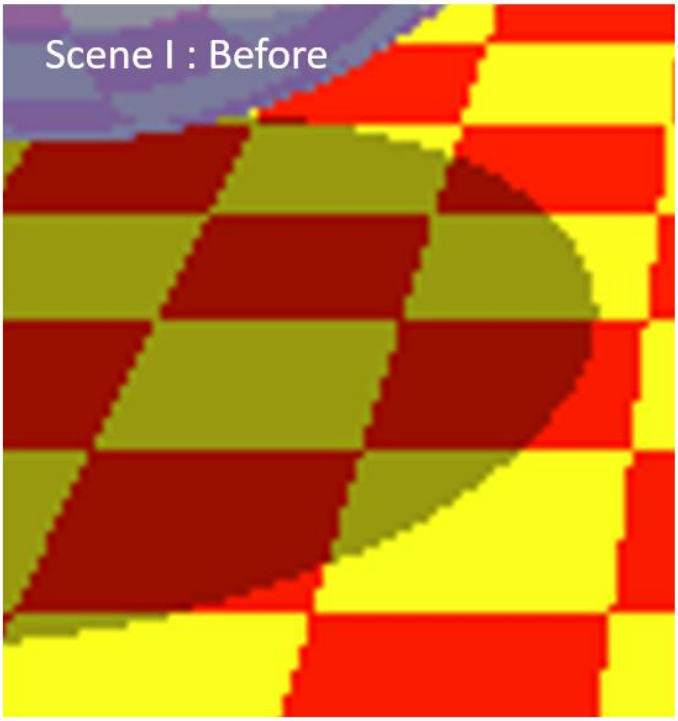


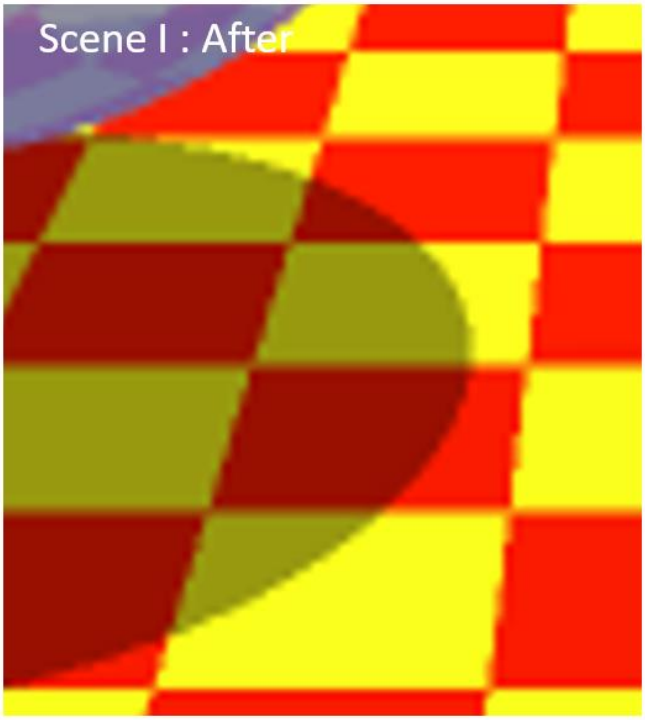
Schlick's Approximation Sample : Scene G

Super-Sampling Antialiasing Sample : Scene C

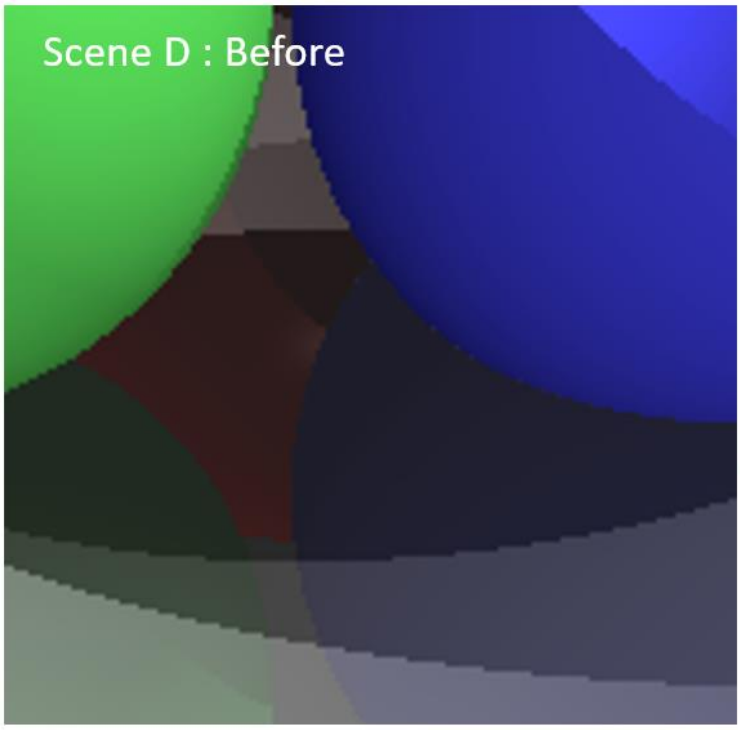


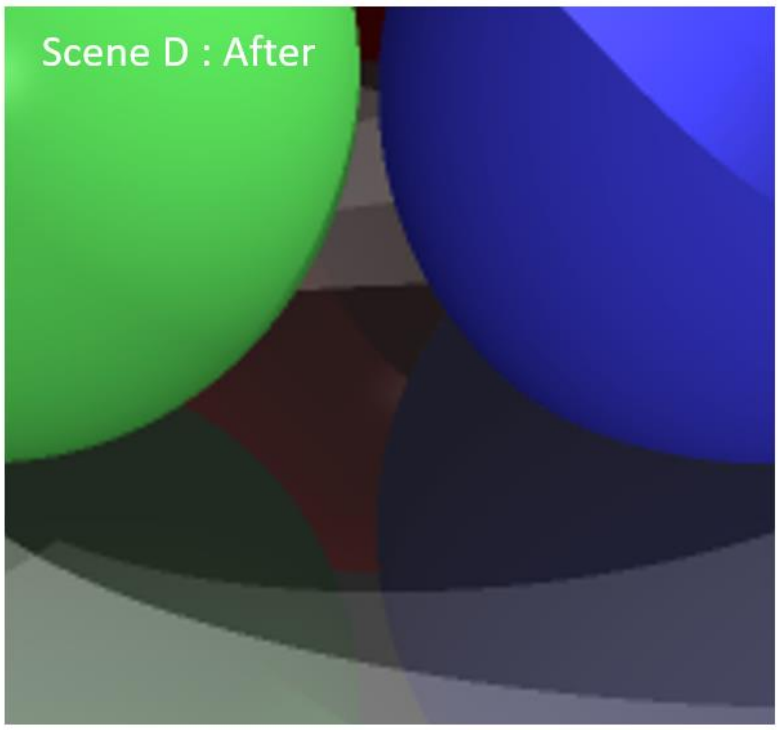
Super-Sampling Antialiasing Sample : Scene I



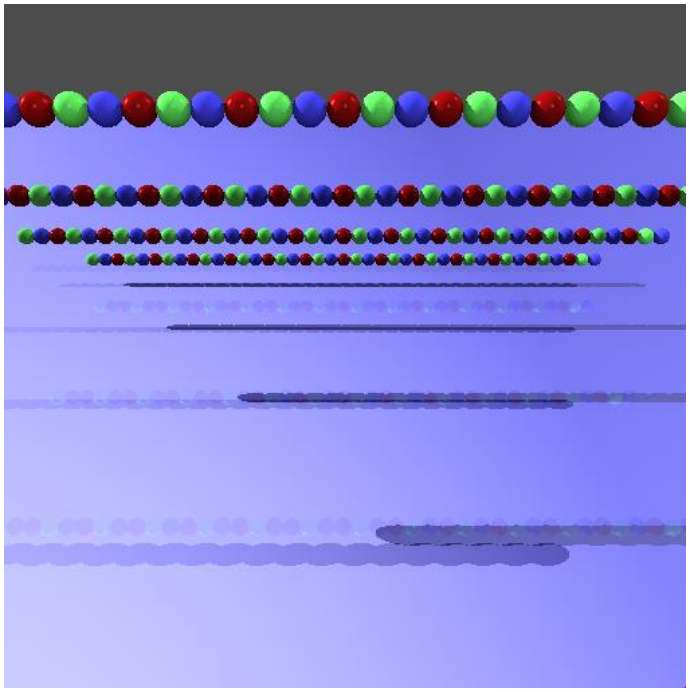


Super-Sampling Antialiasing Sample : Scene D





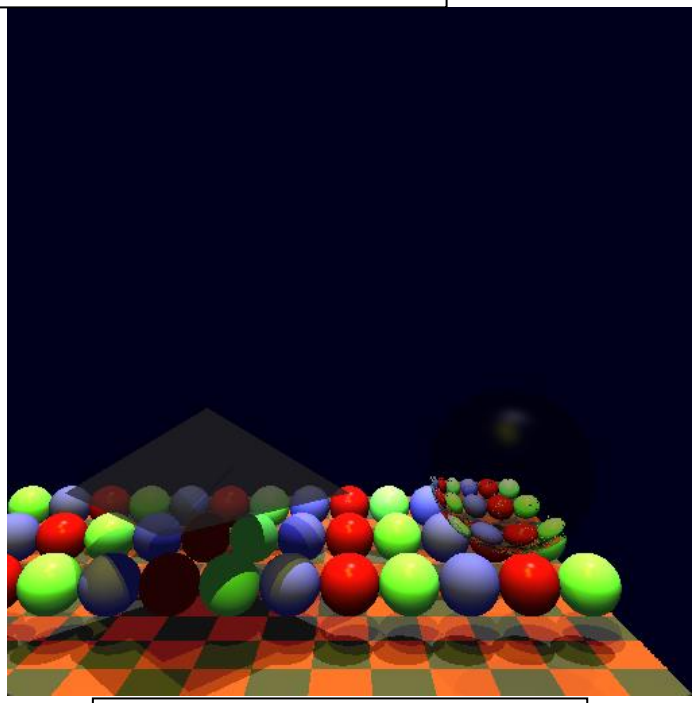
All Normal Sample Scenes details (without any features applied)



Scene a

Objects :

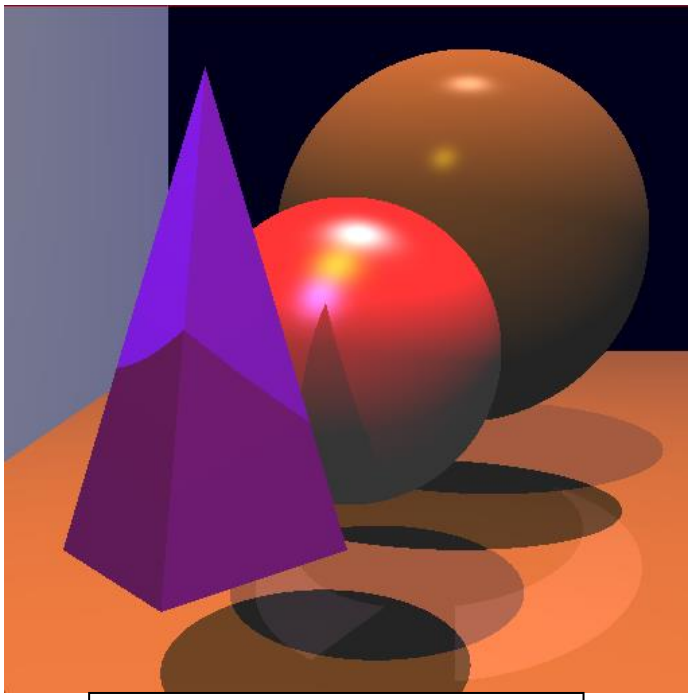
1. 164 spheres
2. 1 plane



Scene b

Objects :

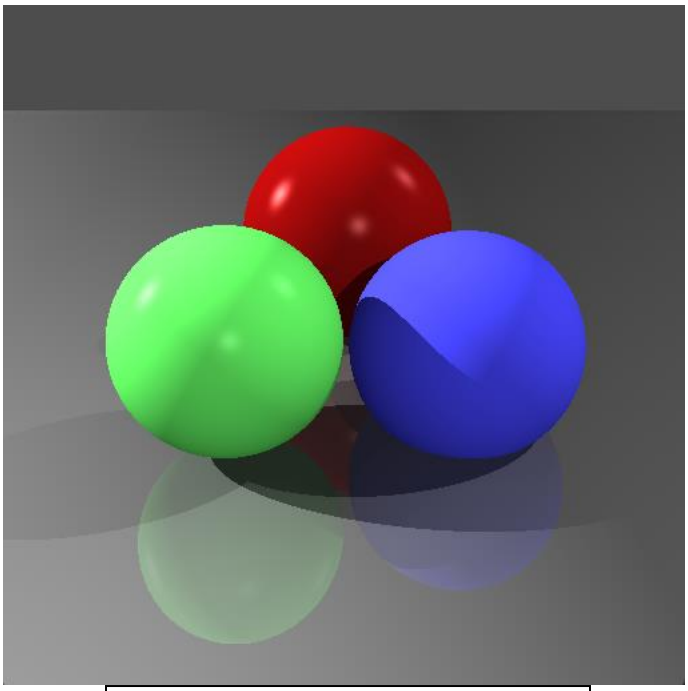
1. 39 non-reflective sphere
2. 1 transmissive triangle mesh
3. 1 transmissive sphere
4. 1 checker mesh plane



Scene c

Objects :

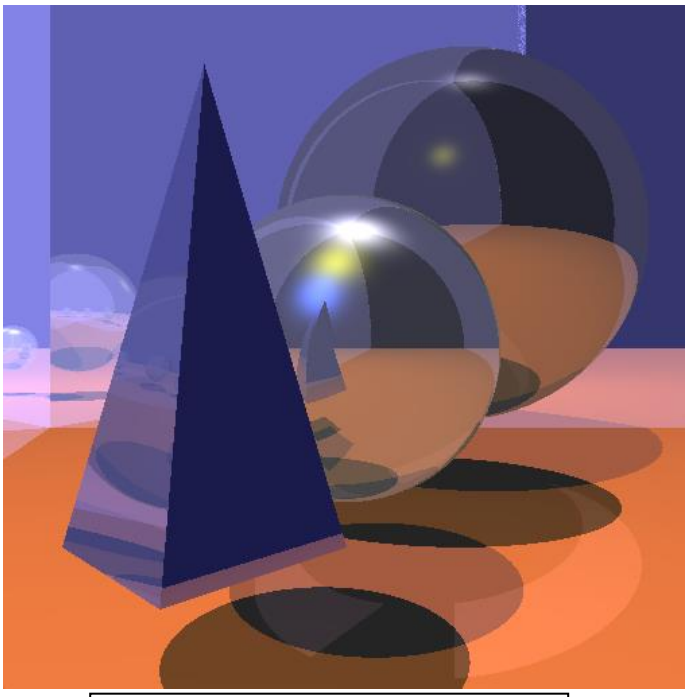
1. 1 triangle mesh
2. 2 spheres
3. 2 planes



Scene d

Objects :

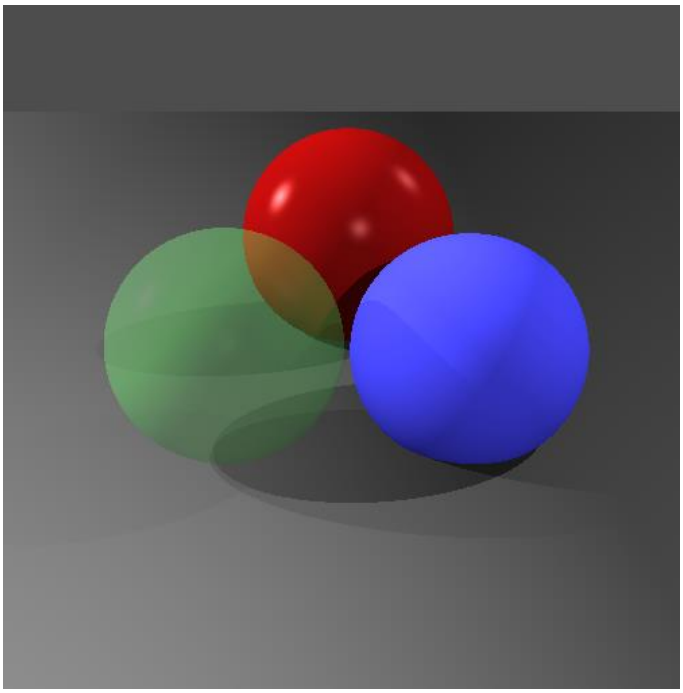
1. 3 spheres
2. 1 reflective plane



Scene e

Objects :

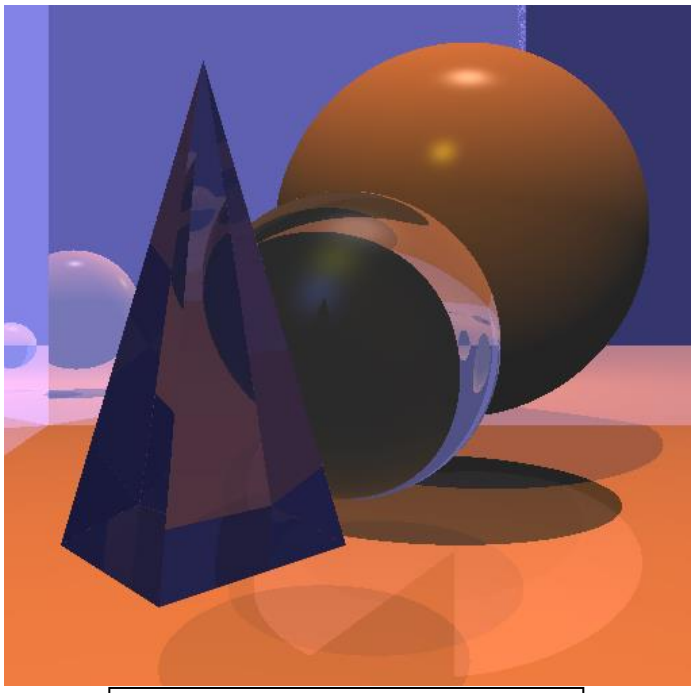
1. 1 reflective triangle mesh
2. 2 reflective spheres
3. 2 reflective planes



Scene f

Objects :

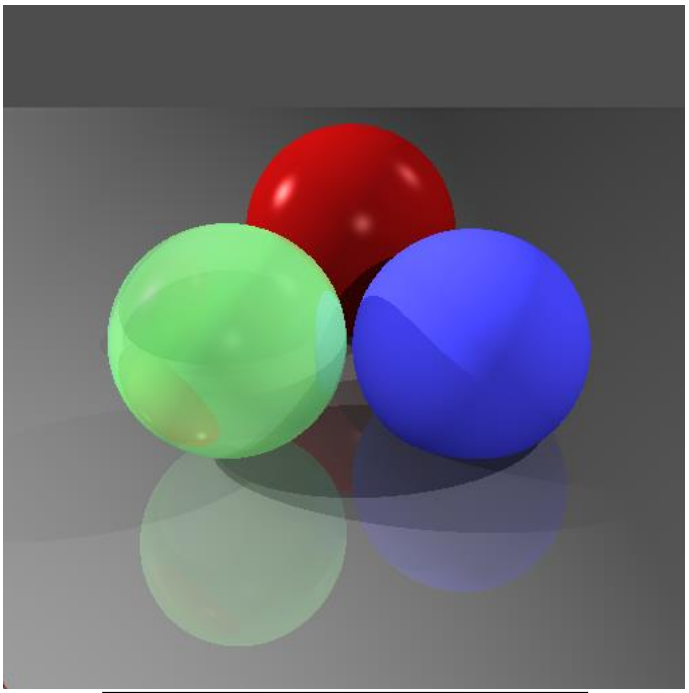
1. 3 spheres (1 transmission)
2. 1 plane



Scene g

Objects :

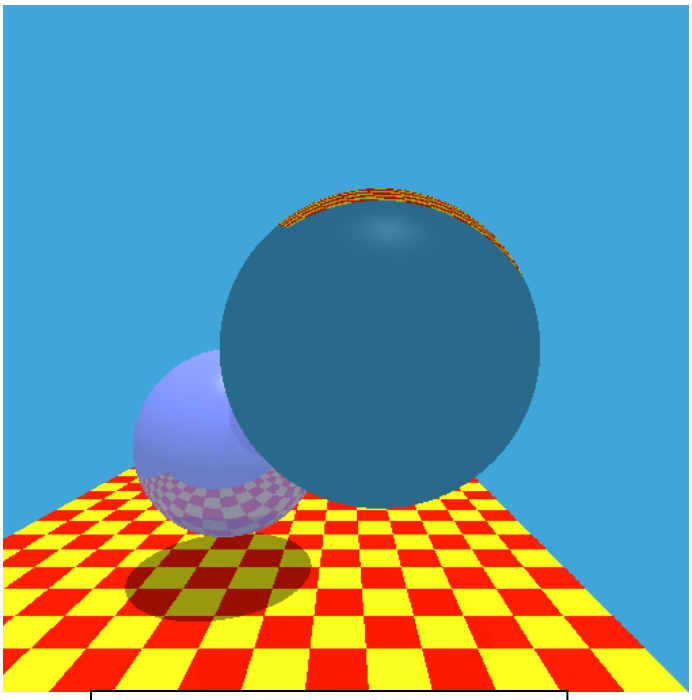
1. 1 transmissive triangle mesh
2. 2 spheres (1 transmissive)
3. 2 reflective planes



Scene h

Objects :

1. 3 spheres (1 transmissive)
2. 1 reflective planes



Scene i

Objects :

1. 1 transmissive sphere
2. 1 reflective sphere
3. 1 checker mesh plane