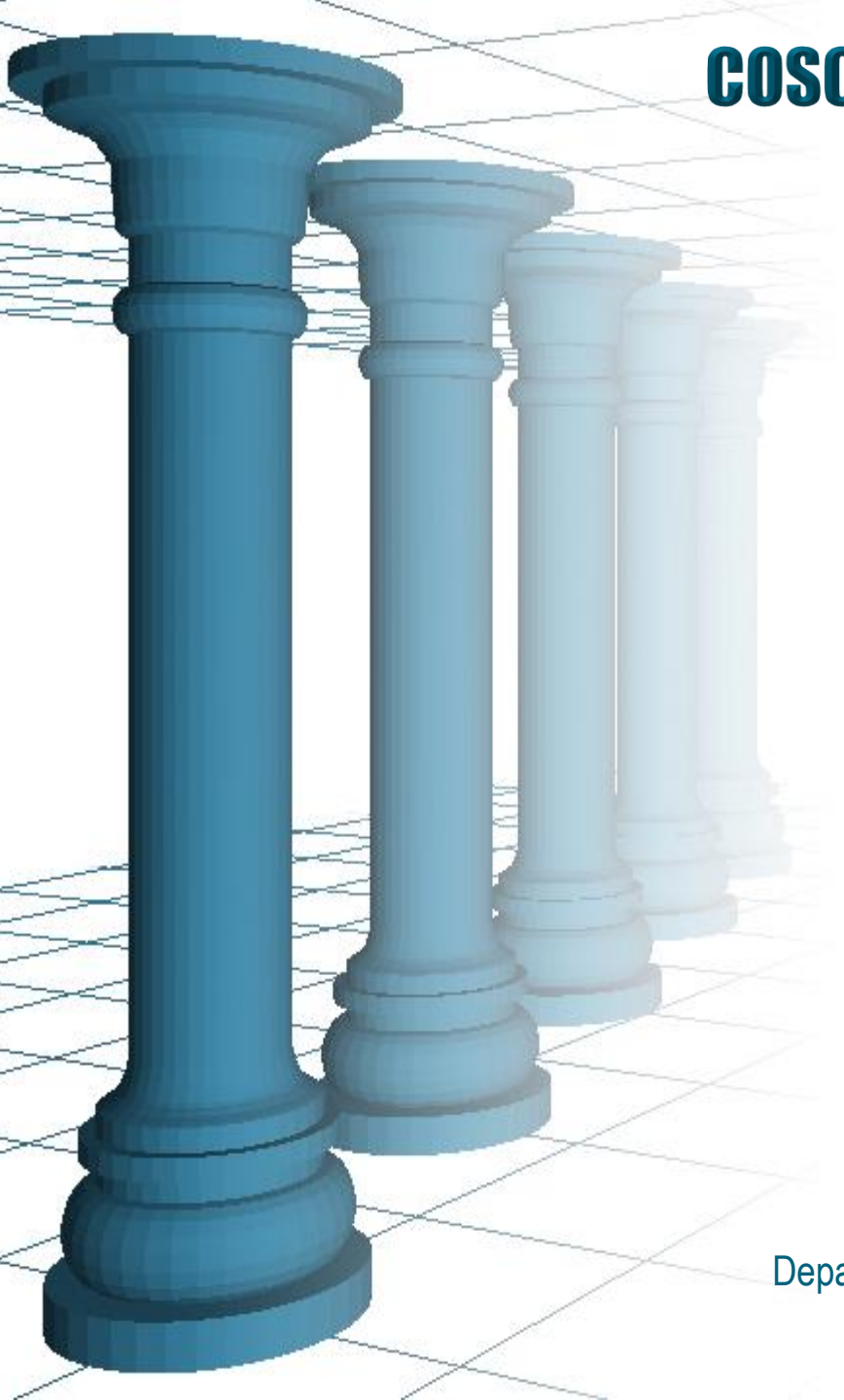


COSC363 Computer Graphics

Assignment-2

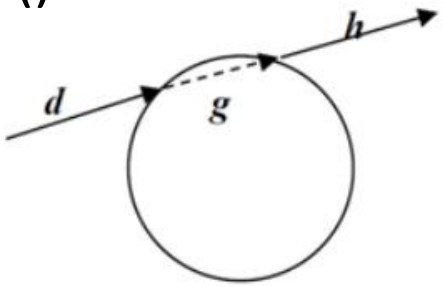
R. Mukundan (mukundan@canterbury.ac.nz)
Department of Computer Science and Software Engineering
University of Canterbury, New Zealand.

Assignment-2

- Due: 11:55pm, **2 June 2023**
- Drop-dead date: 9 June 2023.
- Your submission must be based on Lab 6,7 code. Implementations using shaders, path tracing, photon mapping etc., not allowed.
- Not a group project. Your submission must represent your own individual work
- Students are encouraged to discuss assignment related problems using course forum. However, code segments or any part of your assignment submission should not be posted on Learn.

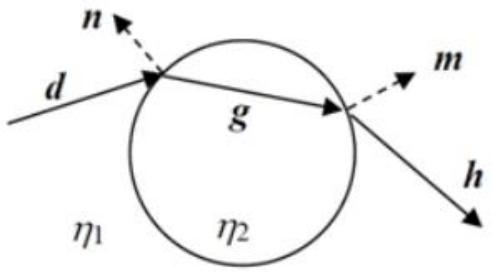
Assignment Specs

- Minimum Reqs (Max. 10 marks)
  - A good spatial arrangement of objects inside a box
  - Shadows
    - lighter shadows for transparent and refractive objects
  - One planar mirror-like object
  - Chequered pattern on a planar surface
  - A transparent object. Even though transparency may be treated as a special case of refraction where $\eta_1 = \eta_2$, the implementation of transparency effect does not require the refract() function.



Transparency

d = *g* = *h*



Refraction

g = *refract*(*d*, …)

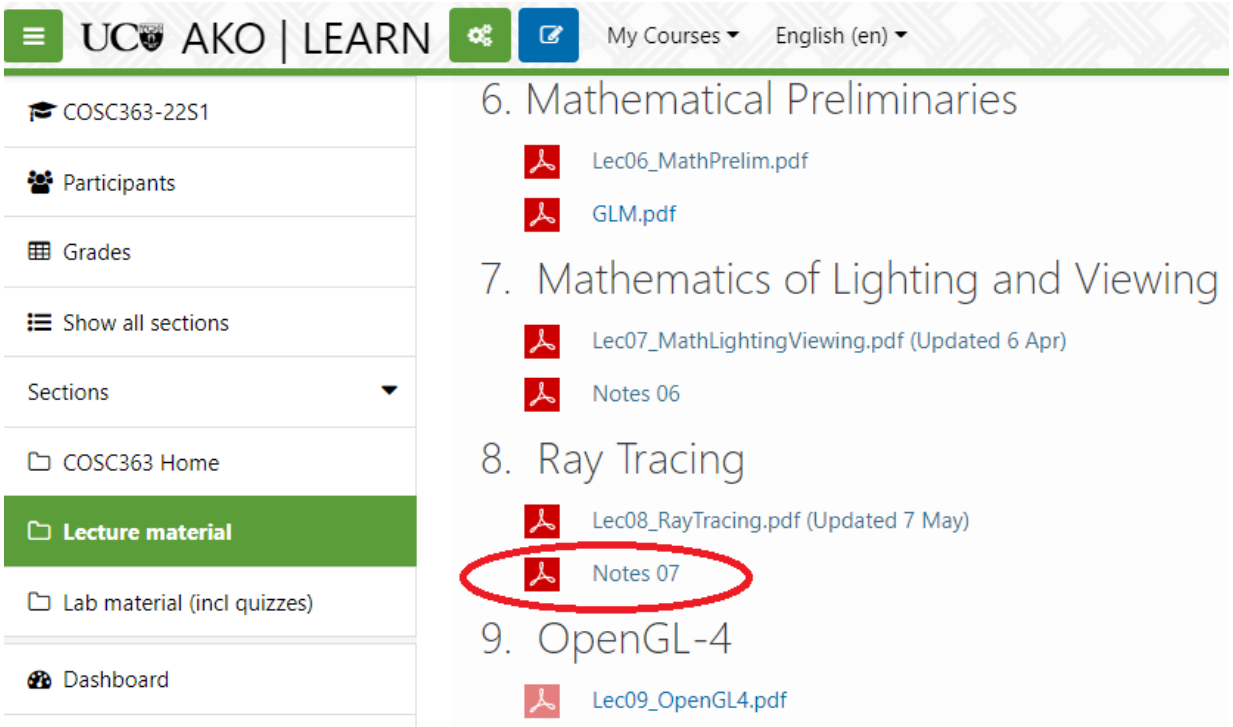
h = *refract*(*g*, …)

Assignment Specs

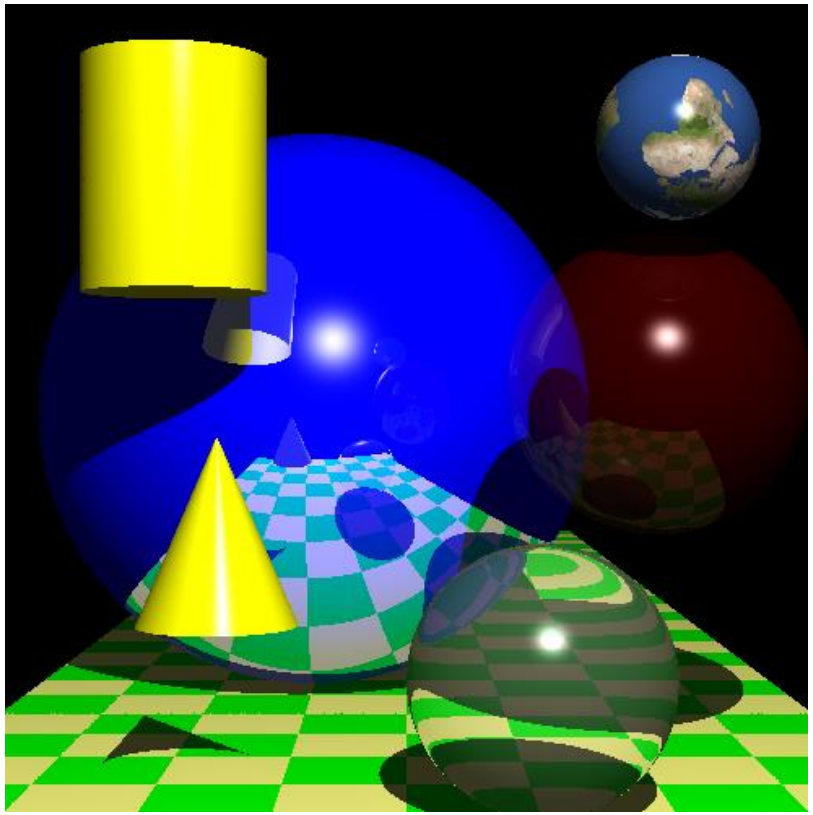
- Extensions (Max. 7 marks)
  - Cone/Double cone, Cylinder, Torus (?), + Cap
  - Refraction
  - Multiple light sources: multiple shadows, specular highlights
  - Multiple reflections on parallel surfaces
  - Spotlight
  - Anti-aliasing
  - Non-planar object textured using an image
    - E.g., textured sphere, textured cylinder.
  - Procedural patterns
  - Fog
  - Depth of field
  - Soft shadows

Supplementary Notes

- Information on modelling transparency, multiple light sources and shadows, spotlights, and fog can be found in "**Notes 07**" (Note07_RayTracing.pdf) in lecture material section.



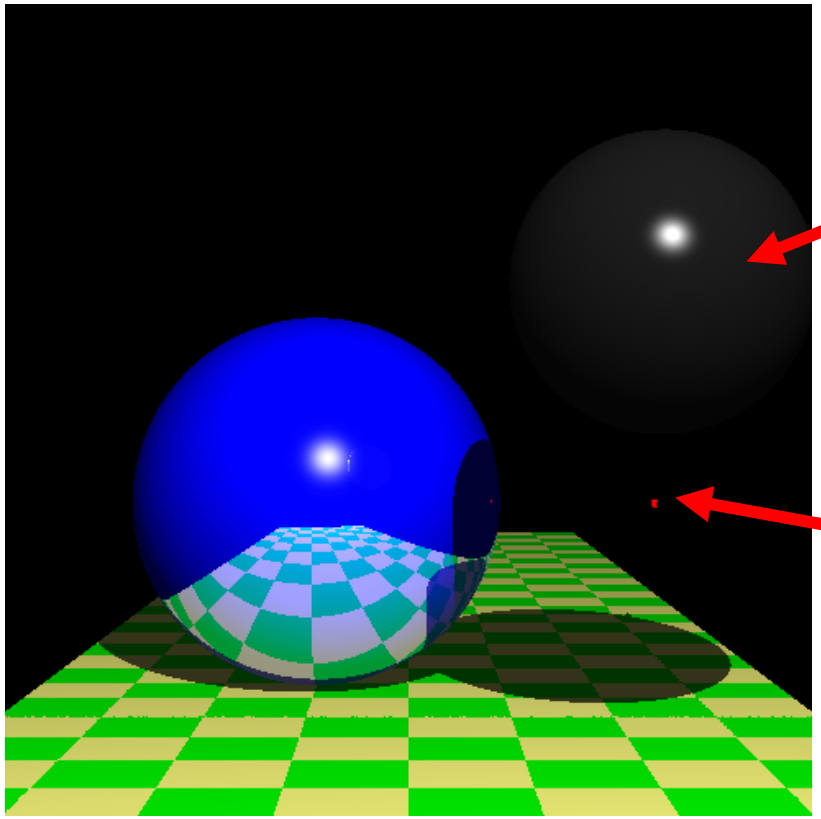
Bad Design



- Random placement of objects
- Objects and features not clearly visible
- Scene clutter
- Incorrect mapping of textures

Bad Design

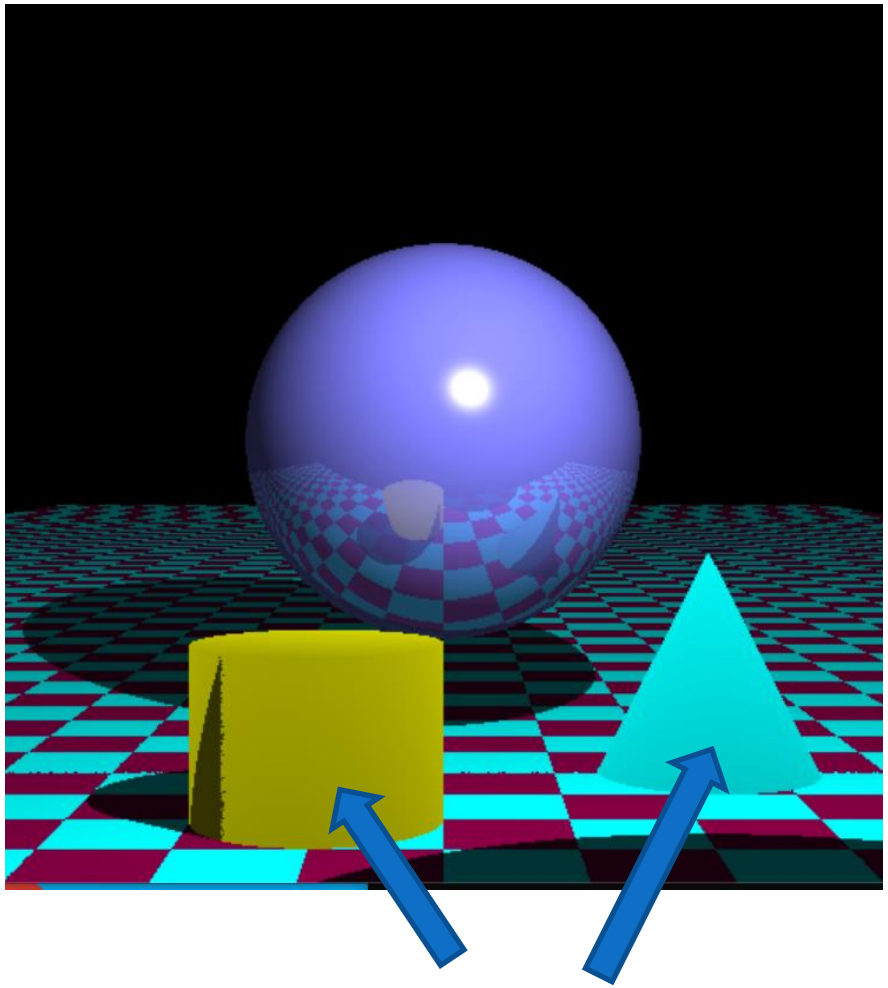
Marks will not be given to features not clearly visible in the output.



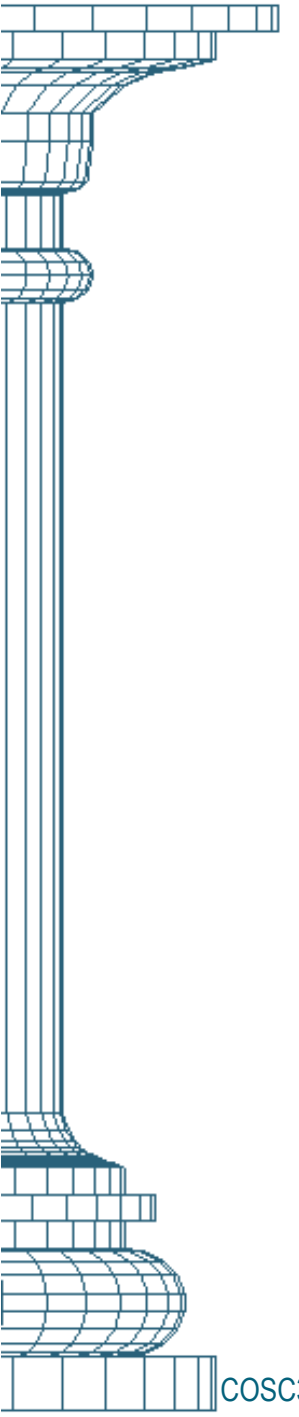
Refractive sphere ?

Cylinder ? Cone ?

Bad Design



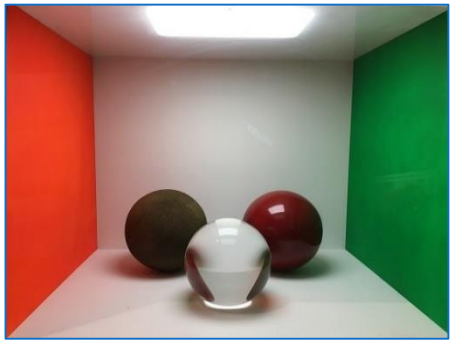
- Improper lighting



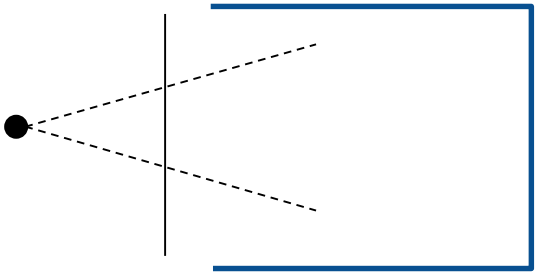
Examples of some of the Minimum Requirements

Box

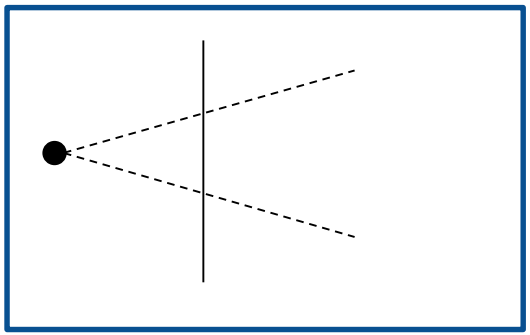
- A box environment is commonly used for testing global illumination algorithms
  - E.g. Cornell Box (Wikipedia)



- 5 or 6 axis-aligned planes, each having a different colour or pattern.

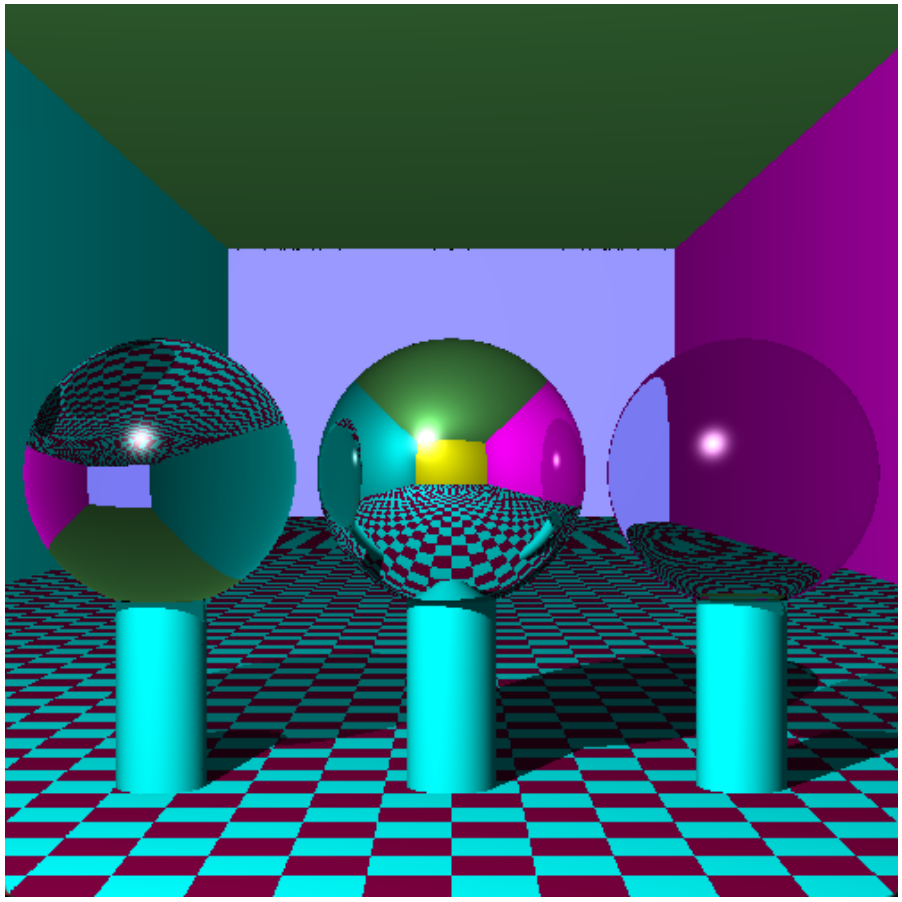


5 Planes



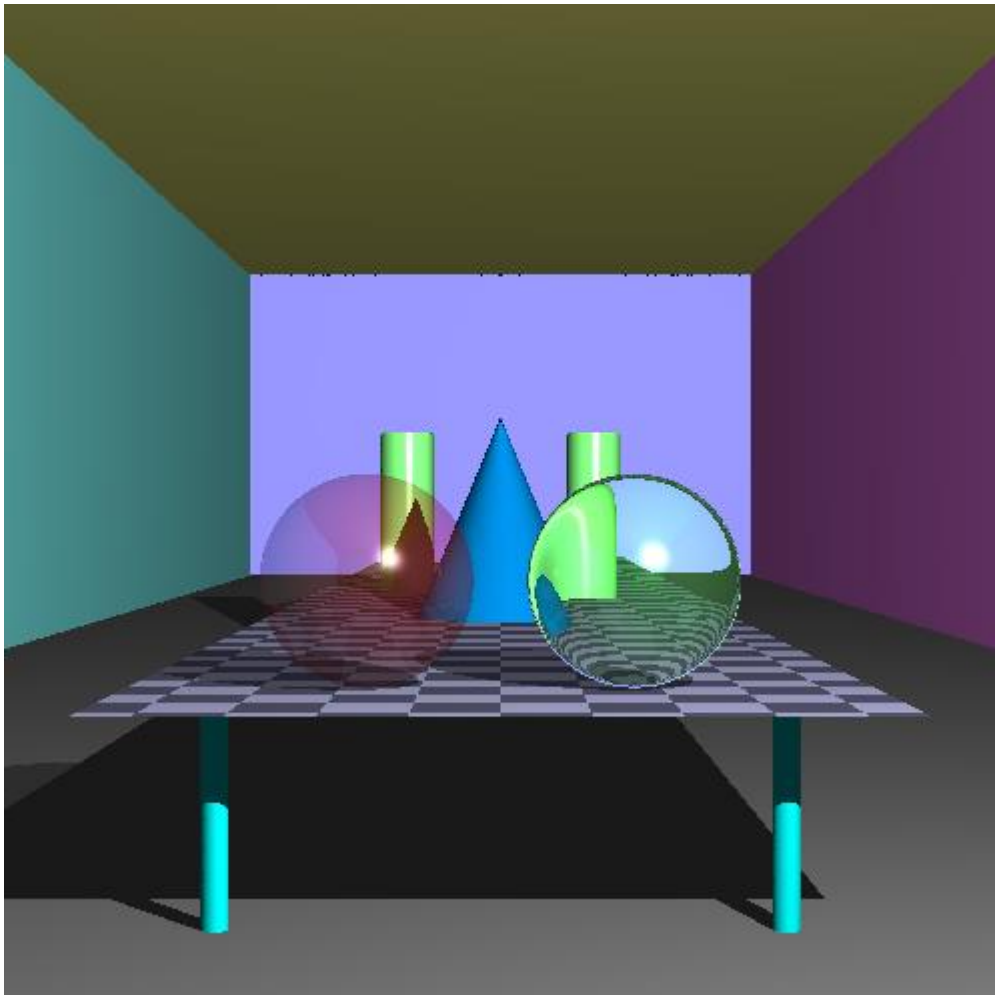
6 Planes

Box: Example



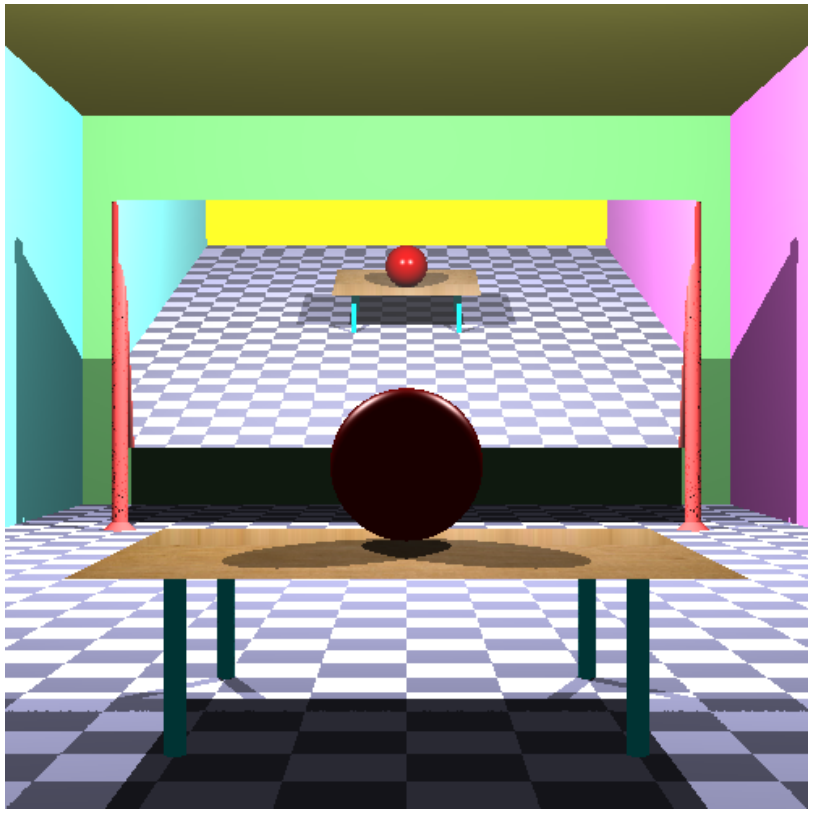
- The above example uses 6 planes for the box
- Spheres: Refractive (η =1.5), Reflective, Refractive (η = 1.005)
- Refractive spheres cast lighter shadows

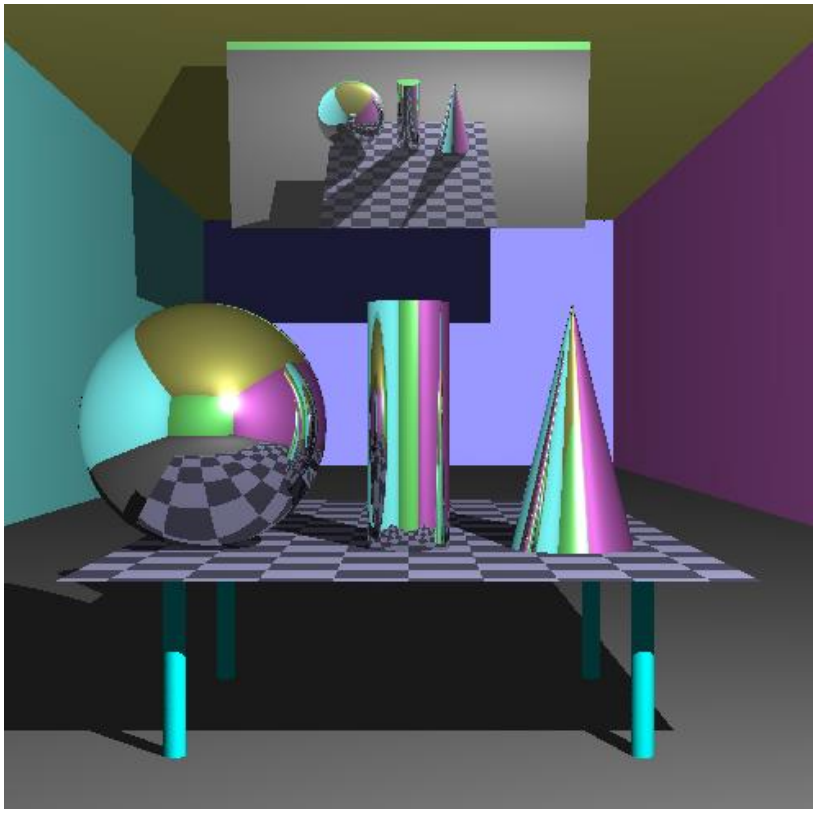
Transparent Object



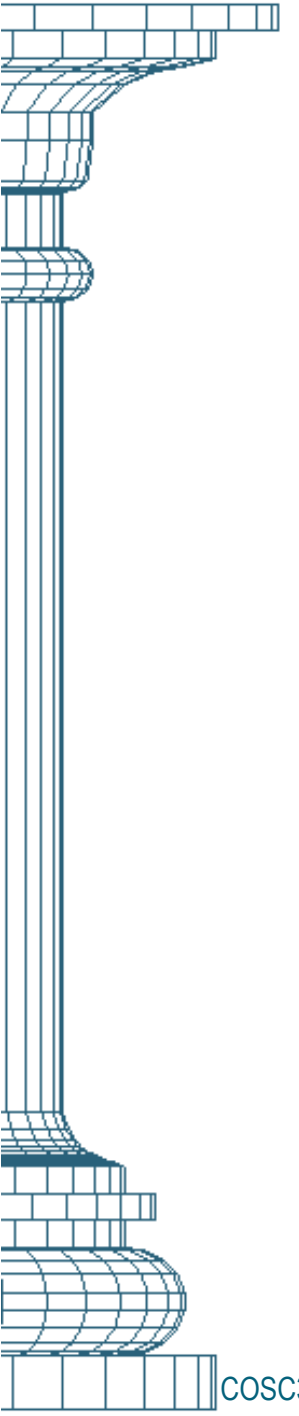
- Spheres: Transparent, Refractive ($\eta = 1.01$)
- Transparent and refractive spheres cast lighter shadows

Mirror: Examples



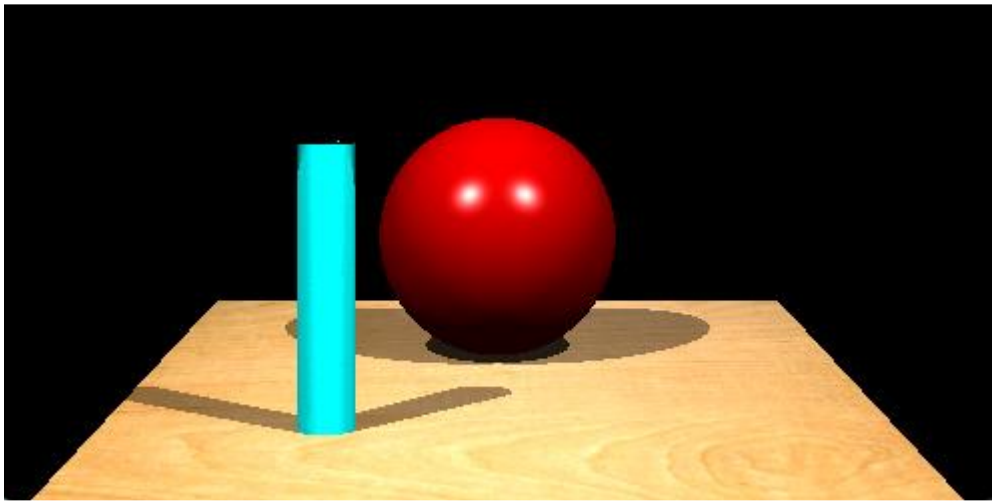


- The first example above contains two light sources



Extra Features

Multiple Lights



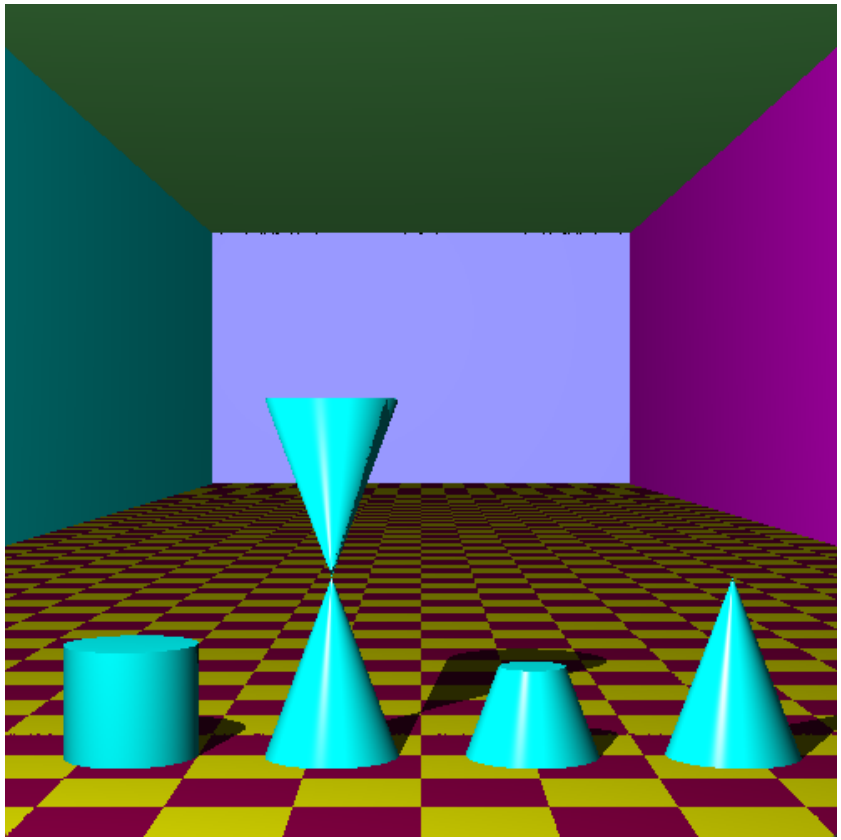
- Please trace shadow rays to each of the light sources to generate multiple shadows of objects in the scene.
- Multiple specular highlights must be visible on at least one object.

Multiple Light Sources

$$\text{colour} = (\text{ambient term}) + (\text{diffuse term})_{L1} + (\text{specular term})_{L1} + (\text{diffuse term})_{L2} + (\text{specular term})_{L2}$$

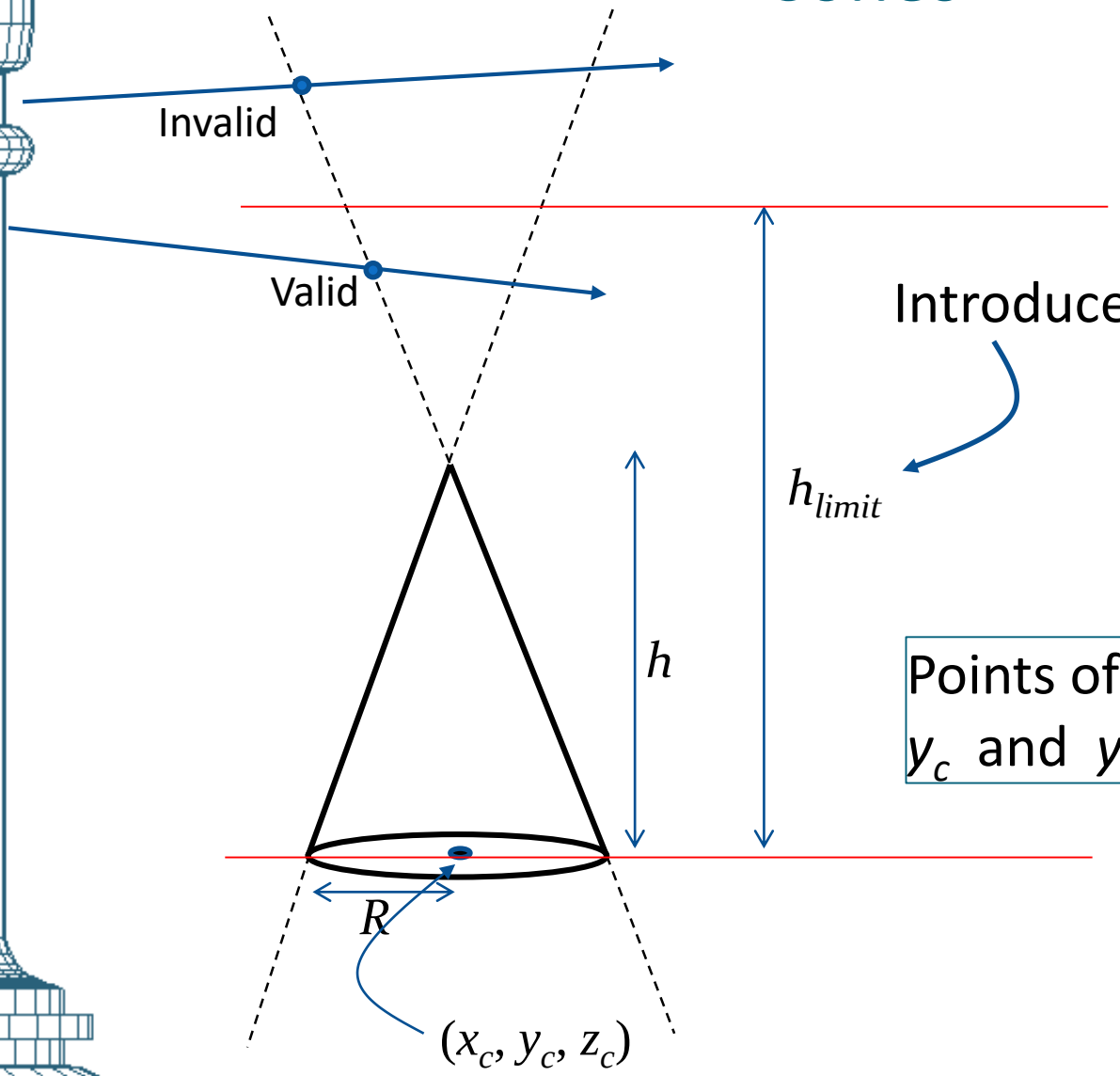
- Use only one ambient term.
  - You may have to modify the function "lighting()" in the SceneObject class.
- Reduce intensity of light sources if required.

Cones



- Doubles and truncated cones can be easily generated.
- In the above example, the cylinder and truncated cone have caps

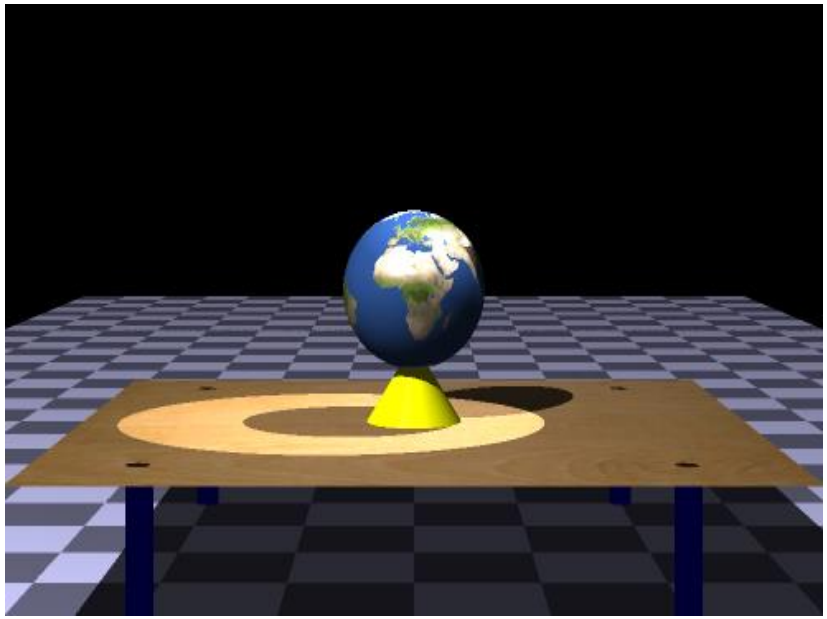
Cones

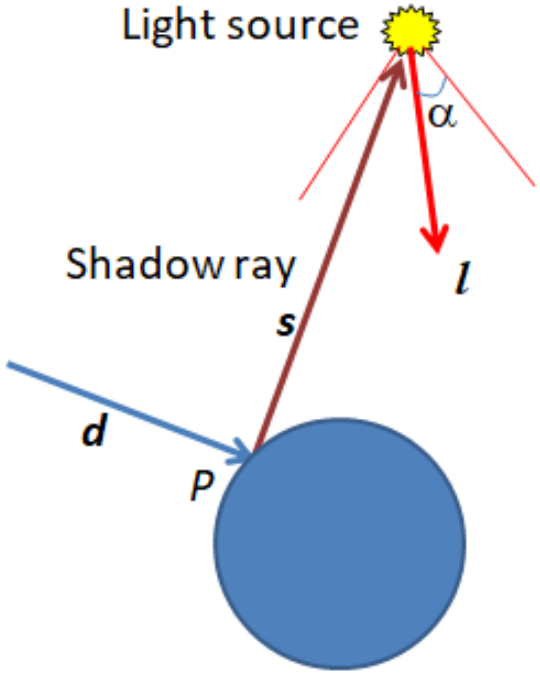


Introduce a new parameter h_{limit}

Points of intersection between y_c and y_c+h_{limit} are valid

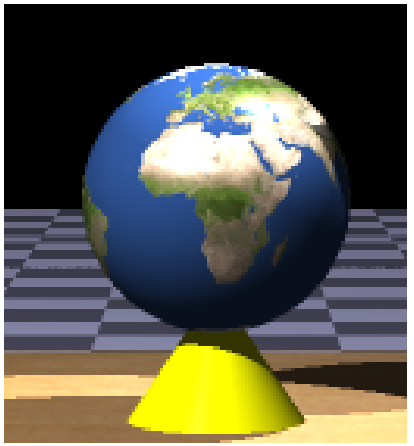
Spotlight



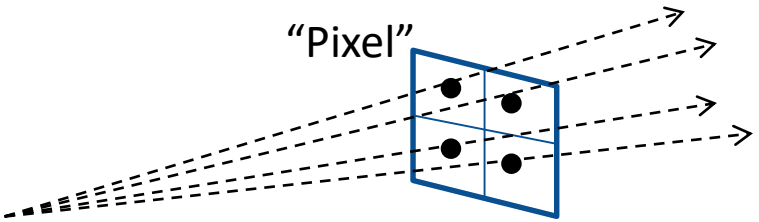


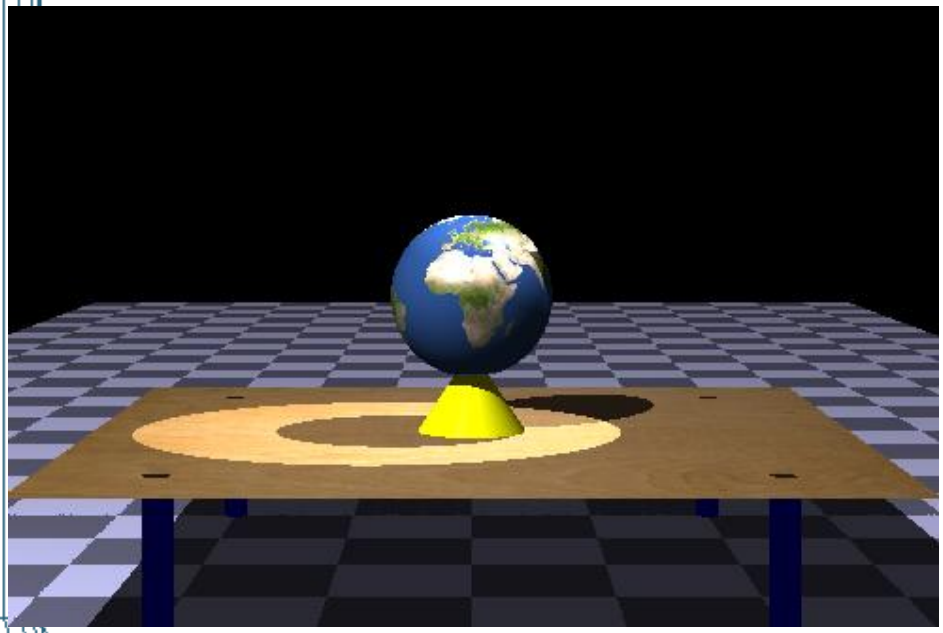
Texturing a Non-Planar Object

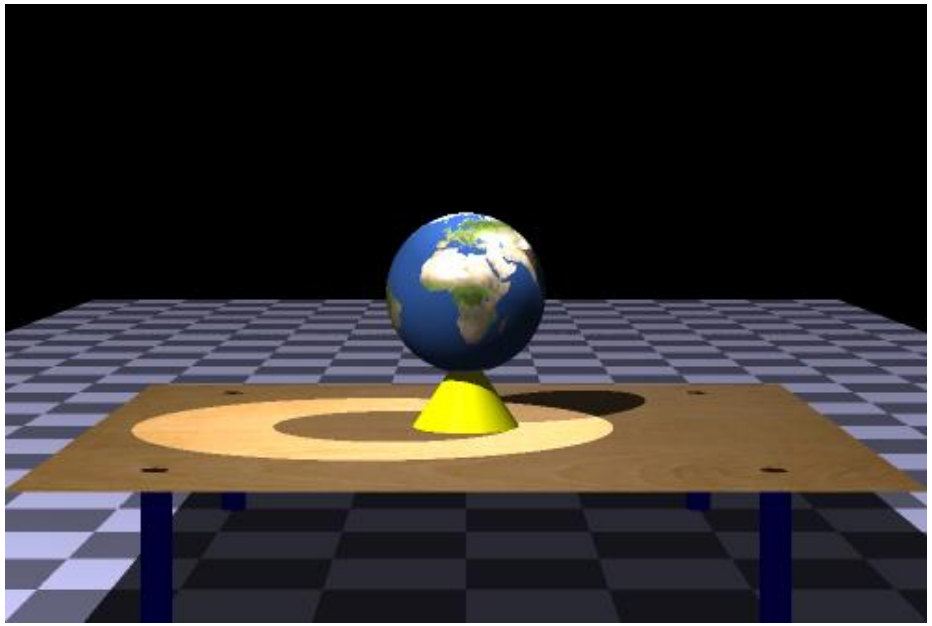
- Sphere: Compute spherical angles α, δ
  - Convert α to texture coordinate *s*
  - Convert δ to texture coordinate *t*
  - Ref: Wikipedia: UV Mapping
- Cylinder: Computer cylindrical angle α
  - Convert α to texture coordinate *s*
  - Convert *y* to texture coordinate *t*
- *BMP files*
  - 24 bits per pixel (not indexed color)
  - Uncompressed



Anti-Aliasing

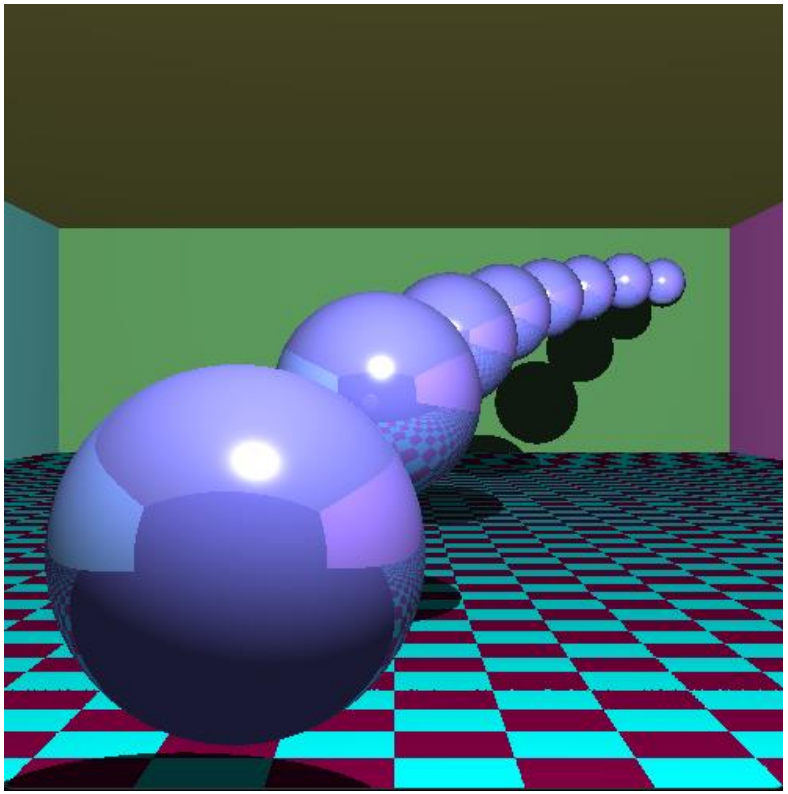


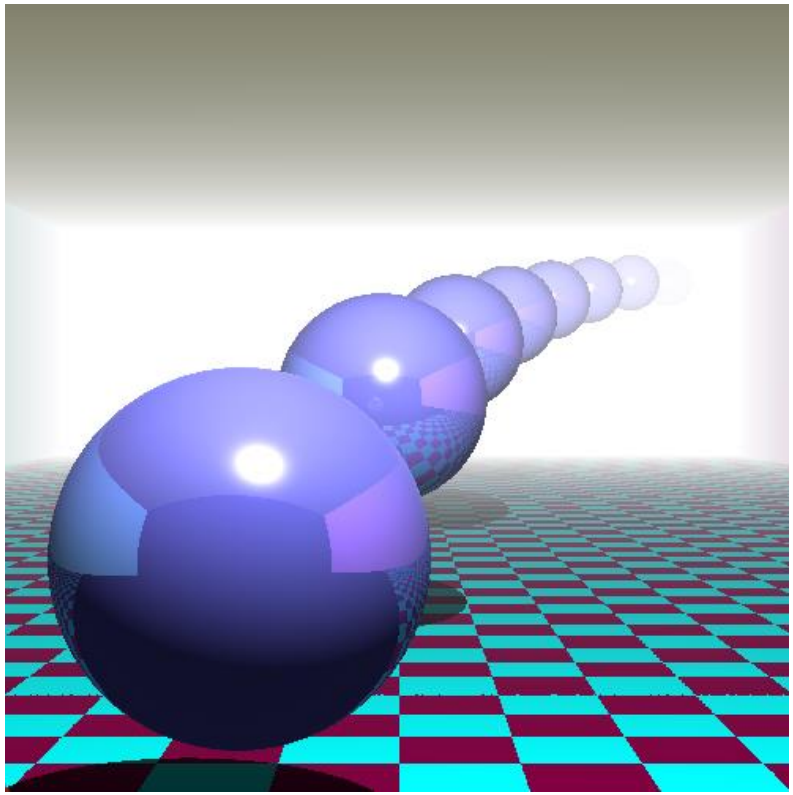




Please include screenshots of outputs with and without anti-aliasing.

Fog



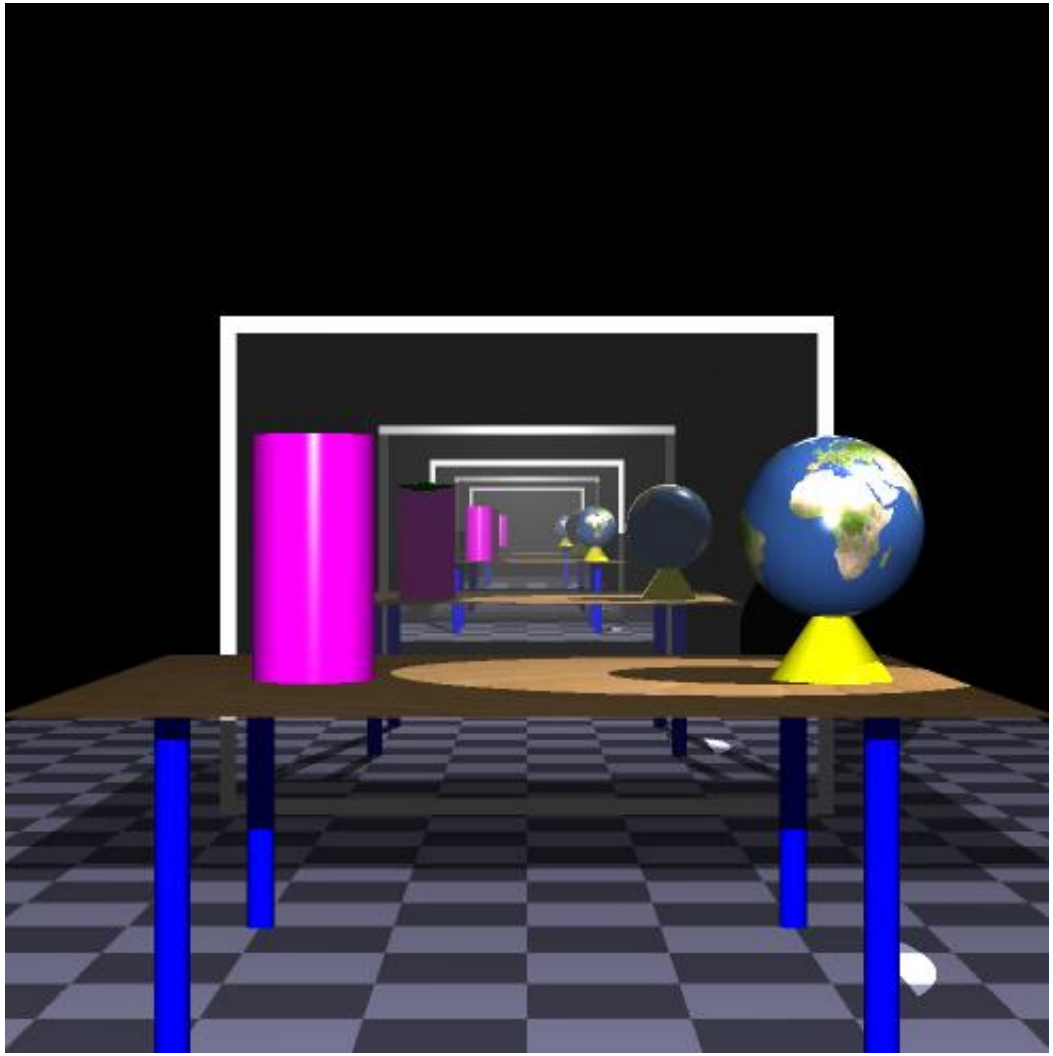


$$\lambda = \frac{(ray.hit.z) - z_1}{z_2 - z_1}$$

color = (1- λ) color + λ white

Please include screenshots of outputs with and without fog.

Multiple Reflections



The camera must be placed between the two reflecting surfaces

Assignment Submission

- Provide build details/command in the report
- Please submit report in PDF format only
- Please package the files as a zip file.