A5 Project Proposal

Title: Ray traced Backgammon

Name: Mahammad Ismayilzada

Student ID: 20496519

User ID: mismayil

Final Project:

Purpose :

The main purpose of this project is to learn how to implement various ray tracing techniques and produce reasonably photo-realistic image at the end. The scene will be composed of rather simple shapes with more focus on making it look more realistic. If time permits, I will add more efficiency improvements as well.

Statement :

This project aims to extend A4 with more ray tracing effects. These effects will be implemented on a close up image of backgammon game. Couple of sample pictures will be attached to the project to show what kind of scene I am going to render at the end, but my final image may not look exactly like the image as I will try to add to the image as many ray tracing techniques as I can. At the end, my goal is not to render quite complex scene consisting of sophisticated meshes, but rather simple scene looking more realistic and elegant. As seen from the pictures, the scene will require at least the following features:

- Texture mapping
- Soft shadow
- Anti-aliasing
- Additional primitives
- Depth of field
- Constructive Solid Geometry
- Bump mapping

However, I will also make some objects reflective and refractive to add mirror reflection, refraction and possibly caustics as well. Hence, these features will constitute my objectives. In addition to them, if time permits, I will try to add motion blur to my scene which will make it more interesting and realistic.

The scene will demonstrate an instance of backgammon game with dice and pieces zoomed on a texture mapped backgammon board with blurred background.

To implement all the above features, my ray tracer from A4 will need to be extended with more data structures and algorithms and new Lua commands will also be added to support additional primitives and material features.

This project will be quite interesting and challenging for me for a couple of reasons. First of all, pure ray tracing techniques will be quite difficult to understand and implement on their own. Addition to this I will need to make the image more photo-realistic which will require keen attention to implementation details and choice of algorithms and methods. If all of these is done as planned, quite elegant image will be produced and if I have extra time left, I will apply motion blur to one of the dice that will add more realism to the scene.

I hope to learn various interesting ray tracing techniques, new mathematical and physical algorithms required to implement them and overall, skills for creating a complete ray tracing project from scratch.

Technical Outline :

Here I would like to cover the objectives in more detail.

Objective 1. Mirror reflection

For this objective, I can use the shininess property of the material or extend lua command gr.material to support it. If I use shininess property, then there will be some threshold that if shininess value exceeds it, then the object with that material will be reflective. Otherwise, gr.material can be extended to accept more parameters such as reflectiveness value.

Objective 2. Refraction

For this objective, secondary ray will need to be cast through refractive surface. Snell's law will be used to compute the angles. New refractive material will need to be added to Lua commands as well or gr.material will be extended.

Objective 3. Texture mapping

Texture mapping will be implemented for at least cubes and spheres. Either a new command (e.g. gr.texture_cube) will be added or existing commands will be extended to support this feature. Pixel will be mapped onto polygon and then into texture map. Weighted average of covered texture will be used to computer the final colour.

Objective 4. Bump mapping

Bump mapping will be similar to texture mapping except that normals will be perturbed rather than colours. New lua command will be added to support this feature.

Objective 5. Soft shadow

For this objective, in addition to point lights, area lights will need to be implemented. Area lights will have a center position and two opposite corners. Also instead of casting just one ray towards light, multiple rays will be casted to the area light. As a simple implementation, Light class will be extended to PointLight and AreaLight classes in C++.

Objective 6. Adaptive anti-aliasing

Instead of just casting N rays per every pixel, adaptive approach will be used. More rays will be applied to only those pixels that differ greatly from their neighbors. Also instead of supersampling, stochastic sampling will be used to cast random rays for each pixel. I plan to use Cook stochastic sampling for this purpose. This will also improve the efficiency as well. To show that adaptive anti-aliasing is working, pixels that this sampling is applied to can be highlighted in a different color.

Objective 7. Additional primitives

At least cylinder and cone primitives will be supported. Hence, new lua commands(gr.cone(), gr.cylinder()) will be implemented respectively. Also new classes with new intersection algorithms will be created for these primitives in C++.

Objective 8. Constructive Solid Geometry

To support more complex shapes, CSG will be implemented as described in section 18.6 of the course notes. At least, spheres and cubes with CSG will be supported. A new derived class from Primitive will need to be added that can allow union, intersection and difference of shapes.

Objective 9. Depth of Field

This objective will be a little bit more challenging. An imaginery focal plane will need to be defined behind image plane. A ray will be then casted from eye through the pixel and intersection with the focal plane will be focal point. There will also be defined an aperture of some size m. Then DOF rays will be cast from grid of size m on the image plane to the objects through the focal point and final color will be determined based on these rays.

Objective 10. Final scene
All of the above features will be demonstrated using possibly Cornell boxes and at the end, a unique scene will be created with as many above techniques as possible for the showcase.

Bibliography :

1. **Computer Graphics, Principles and Practice**, Third Edition, John F.Hughes, et al., 2014, pp. 547-551, 557-559, 702-706, 727-734, 1060-1062
2. **Laine S., Aila T., Assarsson U., Lehtinen J., and Akenine-Moller T. Soft shadow volumes for ray tracing.** *ACM Trans. Graph. 24, 3* (July 2005), pp. 1156-1165
3. **Robert L.Cook, Thomas Porter, Loren Carpenter. Distributed Ray Tracing** *Acm 1984.* pp. 137-145
4. **Mark A.Z.Dippe, Erling Henry Wold, Antialiasing through stochastic sampling** *Acm 1985.* pp. 69-78

Objectives:

Full UserID:____________________ **Student ID:**____________________

___ 1: **Mirror Reflection**
Mirror-like surfaces are supported and reflection is implemented correctly.

___ 2: **Refraction**
Transparent objects support refraction of light.

___ 3: **Texture mapping**
Texture mapping from a picture file is implemented correctly.

___ 4: **Bump mapping**
Bump mapping is correctly implemented for hard surfaces.

___ 5: **Soft shadow**
Area lights and more shadow rays are used to produce soft shadows.

___ 6: **Adaptive anti-aliasing**
Adaptive anti-aliasing is implemented along with stochastic sampling.

___ 7: **Additional primitives**
Support for cones and cylinders have been implemented correctly.

___ 8: **Constructive Solid Geometry**
Union, intersection and difference operations can be applied to spheres and cubes.

___ 9: **Depth of field**
Depth of field is implemented correctly to give focus to objects.

___ 10: **Final Scene**
A unique final scene is created to demonstrate most features from above list.

A4 extra objective: Anti-aliasing using supersampling with 9 rays.