

## OVERALL SUBMISSION

We have completed the full assignment with Part A and Part B completed.

In Part A, we implemented

- (1) a scene signature where each pixel shows a unique color identifier for the first hit
- (2) a render scene with only the diffuse and ambient components of the Phone model
- (3) a rendered scene with all three terms of the Phone model

In Part B, we implemented a partitioning structure using AABB tree and completed four features including:

- (1) Compound object - cylinder
- (2) Anti-Aliasing
- (3) Soft shadows
- (4) Texture-mapping

## CODE & FILE STRUCTURE

We didn't add any external files to this assignments, most implementations are done within the function that provided in the assignment.

We modified and add several structs:

- Material Struct:
  - Add 1 double reflect-ratio number to set ratio of each object reflection
- Intersection Struct:
  - Point3D uvCoord to handle the texture mapping coordinates
  - 1 boolean texture flag to control texture mapping for both and for each individual object
- Ray3D Struct:
  - Modified the constructors of Ray3D to adapt texture mapping
- Add new Struct(AABBNode):
  - Copy most data from SceneDraNode, coving obj, mat etc.
  - Record the values of corresponding AABB data and surface area

We added a few helpers functions:

- In Scene\_object.cpp:
  - UnitCylinder::UnitCylinder
  - UnitSphere::textureCoord
  - UnitSquare::textureCoord
  - UnitCylinder::textureCoord
- In raytracer.cpp:
  - Raytracer::get\_texture\_color

```
Raytracer::read_texture
Raytracer::findBounding
Raytracer::CreatingAABB
Raytracer::calculatesurfacearea
Raytracer::intersectBB
Raytracer::AABBMerge
Raytracer::fixUpwardsTree
Raytracer::traverseSceneBB
```

Part A was done within two files:

scene\_object.cpp: where ray intersection calculation is implemented

light\_source.cpp: where light and phone shading is implemented

Part B was done in different files for different features:

(1) **Cylinder**: Done in scene\_object.cpp, Unitcylinder() function

(2) **Anti-Aliasing**: Done in raytracer.cpp, render() function

(3) **Soft Shadow**: Done in raytracer.cpp, computeShading() function

(4) **Texture Mapping**:

- Texture Coordinates are calculated in scene\_object.cpp
- Texture Colour is calculated in get\_texture\_color in raytracer.cpp
- Read texture file is done with a helper function read\_texture() in raytracer.cpp
- Shade Texture color is done in function shade\_ray() in raytracer.cpp

(5) **BSP tree**: Done in raytracer.cpp, traverseSceneBB() function

## IMPLEMENTATION & REFERENCES

(1) BSP tree: (Traversal function does not work very well, while AABB tree was completely implemented)

-AABB tree is based on the surface area and depth of node as search heuristics.

-Reference:

<https://github.com/JamesRandall/SimpleVoxelEngine/blob/master/voxelEngine/src/AABB>

Tree.cpp

(2) Cylinder:

-Reference:

<http://woo4.me/wootracer/cylinder-intersection/>

(3) Anti-Aliasing:

- Anti-Aliasing is implemented by using the supersampling method.

- To see the Anti-Aliasing effect, change the boolean ANTI\_ALIASING to true (in raytracer.cpp on the top).

- Reference:

<https://www.ics.uci.edu/~gopi/CS211B/RayTracing%20tutorial.pdf> (Page 12)

(4) Soft Shadow

- Soft Shadow is implemented by sampling the light direction with random jitter

(5) Texture Mapping

- A square and sphere texture mappings are implemented, and each is controlled by a flag in Intersection struct, hence, you can see both texture mapping by setting until.n file, texture\_flag to true, and square\_texture or sphere\_texture to true depends on what you want to map. Setting texture\_flag to false will disable the texture mapping for all objects.
- The overall method to produce texture mapping is to map every point on an object to a pixel on the image, then get the colour of the pixel and shade.
- Reference: <https://www.cs.unc.edu/~rademach/xroads-RT/RTarticle.html#texturemap>

## GROUP ROLES

Huan Wei, 1000446485 weihuan1

Part A: UnitSquare: Intersect

Part B: Anti-Aliasing, Soft Shadow, Texture mapping

Zhirun Qian, 100335078 qianzhir

Part A: UnitSphere: Intersect, shade in light\_source.cpp

Part B: reflection, Simple Shadow, BSP tree with AABB, Cylinder