CS 350 |

Assignment 2| Geometry Toolbox

Files (submit archive) due

- Week 7
- By Sunday, 23:59

Remember: The major point of this assignment is for you to learn how to submit your homework correctly. You should strive to follow all of the directions exactly as specified in the handouts. The syllabus that was handed out during the first day of class also contains important information on how to submit your homework. If you are still unsure of how to do something, you should ask for help, either from myself or from a tutor in the lab or from another student.

Topics

The purpose of this assignment is to be able to construct and use core bounding volumes (Spheres and Aabbs) for simple optimizations. This includes using them for bounding volume spatial partitions to optimize ray-casting, frustum-casting (future assignment), and pair-queries.

Implementation

You should implement the primitive classes as discussed in class:

- Point3D
- Plane
- Triangle
- Bounding Sphere
- AABB
- Ray

Provide the following functionality for operations/tests on these primitives:

- Basic Intersection
  - Sphere Vs Sphere
  - AABB Vs Sphere
  - Sphere Vs AABB
  - AABB Vs AABB

- Point-based
  - Point Vs Sphere
  - Point Vs AABB
  - Point Vs Triangle
  - Point Vs Plane

- Ray-based
  - Ray Vs Plane
  - Ray Vs AABB
  - Ray Vs Sphere
  - Ray Vs Triangle

- Plane-based
  - Plane Vs AABB
  - Plane Vs Sphere

Note

- You can use OpenGL or DirectX
- You can develop on Windows or Mac OS or Linux

- You should have the ability to explain and derive every single line of code used in your project
- Provide a README.txt file that explains your implementation and all the applicable key mappings that you have used (see below.)

Your programming assignment submission must contain a README text file for **this and future assignments.** The README file MUST contain the following information:

a. How to use parts of your user interface that is NOT specified in the assignment description.
b. Any assumption that you make on how to use the application that, if violated, might cause the application to fail.
c. Which part of the assignment has been completed?
d. Which part of the assignment has NOT been completed (not done, not working, etc.) and explanation on why those parts are not completed?
e. Where the relevant source codes (both C++ and shaders, as applicable) for the assignment are located. Specify the file path (folder name), file name, function name (or line number).
f. Which machine did you test your application on.
    i. If on campus – put the machine name e.g. DIT1234
    ii. If remote – specify OS, GPU, and OpenGL Driver version. **Tip:** You can use the OpenGL Extensions Viewer to get this information readily.
g. The number of hours you spent on the assignment, on a weekly basis
h. Any other useful information pertaining to the application

Please refer to your course syllabus for submission guidelines.

Please refer to the references on Moodle for details about implementation.

Please refer to the Grading Sheet for grade breakdown and extra-credit opportunities.

GRADING SHEET

Implementation Point	Grade	Comments
Scene generation	**20%**	
Objects loaded correctly from the files and displayed in proper position	10	
Correct Rendering using CS 300 shaders	10	
Tests	**70%**	
Sphere Vs Sphere	5	
AABB Vs Sphere	5	
Sphere Vs AABB	5	
AABB Vs AABB	5	
Point Vs Sphere	5	
Point Vs AABB	5	
Point Vs Triangle	5	
Point Vs Plane	5	
Ray Vs Plane	5	
Ray Vs AABB	5	
Ray Vs Sphere	5	
Ray Vs Triangle	5	
Plane Vs AABB	5	
Plane Vs Triangle	5	
Interactivity	**5%**	
Camera Implementation	5	
Miscellaneous issues	**5%**	
Missing information in README	2	**You will receive zero grade for the rest of the assignment if this fails.**
Application does not compile	1	
Application cannot be executed	1	
Scene setup incorrect	1	
Total	**100%**	
Extra Credit	**30%**	
Display the primitives that are being tested	20	
Dynamic Picture-in-Picture (Map view) for the entire scene displayed on top right corner of the screen. The view should look top-down into the scene.	10	