CS350
Geometry framework

1 Objective

The objective of this assignment is to implement a basic geometry framework.

Details

- Many tests have been provided in `/test/geometry.cpp` which are meant to be a starting point for a geometry library
- Test data exist in `/test/in_geometry`, the first value is always the number of tests, the rest of the data depends on the geometry to load.
- There are different types of tests:
  - `classify_*`: Tell if a shape is inside another, overlapping or outside
  - `overlap_*`: Tells is two shapes share space (spheres and aabbs are considered to be solid, not hollow)
  - `intersection_time_*`: Mostly for rays, returns the time of intersection
  - Others: Helper functions that you see necessary
- Each test requires different functions, these functions should be declared in the files: `geometry.hpp/geometry.cpp`
- **Note that this assignment is using a functional approach**. There are no shape classes involved, you are free to add classes, if you do so, add them in a file called `shapes.hpp/shapes.cpp` and let it invoke functionality from: `geometry.hpp`
- You will be required to implement the **`operator`**<< for the math primitives.

2 Provided

The student is provided a basic project framework as starting point. This framework has several projects. The frame is divided in the following sections:

- `/demo`: Demo executable: **This project will not be graded**
  - It is recommended to have a playable demo to visualize the shapes that are loaded from tests
- `/src`: Continued from previous assignment
  - `geometry.cpp/geometry.hpp`: **Most of your work will be done here**, will hold the geometrical functionalities, note that this file should not depend on anything but `math.hpp`.
  - `shapes.cpp/shapes.hpp`: Optional, if you decide to encapsulate your geometry in classes (which you should, at some point) they should belong in these files.
- `/test`: Testing framework. **THIS FOLDER WILL BE REPLACED BY THE INSTRUCTOR**
  - `common.hpp`: Common for all tests. Note the variable `WORKDIR`:

```
// Modify this variable if working directory is giving issues
#ifndef WORKDIR
#define WORKDIR "../../"
#endif
```

    Modify this variable so if your IDE workdir does not match the expected
  - `geometry.cpp`: Holds some tests for this assignment

3 Notes

- You may include extra files if you see fit. You may not include other libraries.
- Comment your code: Files + Functions headers too!
- **Use GIT as a control version to track your progress**

4 Delivery

- Files to submit:
  - All source code!
  - CMakeLists.txt (as you may have added new files)
  - `.git` folder
- DO NOT SUBMIT BINARY FILES NOR INTERMEDIATE FILES