|  |  |  |
| --- | --- | --- |
|  | **Rochester Institute of Technology**  **Golisano College of Computing and Information Sciences**  **School of Interactive Games and Media**  **2145 Golisano Hall – (585) 475-7680** |  |

**Data Structures & Algorithms for Games & Simulation II**

**IGME 309, 2014 Spring**

**A8 - Rough Axis Aligned Bounding Box**

Due: March/22/2015

This is homework assignment is meant to be completed in pairs. For this purpose a repository needs to be setup. The way you will deliver your assignment is by providing me and the grader access to your repository, this could be done freely in github or bitbucket, you will also need to provide your final solution to the labeled dropbox in MyCourses. The purpose of the repo is not only to facilitate your workflow but also to see who is doing what in the code. As such both partners are expected to have at least 1 meaningful commit to the repository. I do not tolerate a slacker and neither should you!

There should be a Readme file in your .zip file submission to my courses and in your repo, stating the name of the team members and what part of the assignment each one fulfilled.

Only one submission to MyCourses is required, if you have more than one submission (if you submitted something and improved on it after, for instance) this new submission needs to be done in BY THE SAME team member.

10% of your grade is a peer-evaluation, this evaluation is personal and needs to be submitted in the same dropbox in MyCourses as the homework assignment. In it you will grade from 0 to 10 how useful your partner was AND WHY. Failing to provide a reason for the assigned grade or failing to submit this file will result in losing this 10% from YOUR grade, “Because he or she showed up to the meeting” is not a valid reason for a grade.

The goal of this homework assignment is to get familiar with Axis Aligned bounding Boxes (implementation and collision detection, no collision response is necessary, Oriented Bounding Boxes will be covered in your next homework assignment); get familiar with teamwork through repositories and as practice before the final project.

For this homework assignment you will implement a Bounding Box class which will be able to create Axis Aligned Bounding Boxes (no Oriented Bounding Boxes just yet but if you feel like working ahead of it by all means do); calculate their position in the world and finally draw it on the screen surrounding different objects.

After the creation of the Bounding Box class you should create a box manager through a singleton which will let the user create new boxes for different objects (Releasing previously existing boxes is not necessary at this point but again if you want to go ahead please do); manage the different transformations of those boxes and calculate intersections among all the boxes in the class. If there is a collision involved demonstrate it using different colors.

Very much like the provided BoundingSphereClass and BoundingSphereManagerSingleton.

You can base your code on the Bounding Sphere Class and the Bounding Sphere Manager singleton class provided. If you are working on your own engine, I will be expecting the same amount of functionality and same names for your methods.

If you want to go ahead and implement other kind of collision response I can take that as extra.

There is a binary executable example under \_Binary that demonstrates what I’m expecting of this homework assignment.

It is not necessary to use “MyEngine” for this homework assignment, you can implement your own code for this, as usual “MyEngine” is meant to give you a head start on all the functionality that is not relevant to the homework assignment but it is completely optional. If you decide to implement your own code instead of working with it, at the very minimum I will need a framework able to load any number of .obj files and display them with textures on the screen; be able to provide a “model to world” matrix and that can display your bounding boxes. Everything in your code should be commented in such a way that the grader does not take much time reading and understanding the code. There are no extra points for implementing your own framework.

Your grade will be as follows:

60% for the Axis Alinged Bounding Box (rough box, no transformations other than translation)

At a bare minimum the Axis Aligned Bounding Box class it should let the user:

1. Add a box based on a model
2. Switch the visibility of the AABB
3. Get the Centroid of the AABB
4. Get the Minimum and Maximum of the AABB
5. Get and Set the “Model to World” matrix of the Box
6. Set the color of the AABB
7. Draw the AABB (if you are working with MyEngine, add it to the render list)

40% of your grade is related to the Bounding Box Manager singleton.

At a bare minimum it should:

1. Add a box based on a model
2. Switch the visibility of the AABB
3. Get the number of boxes in the manager
4. Set the color of a box
5. Set the visibility of a box
6. Render any specific box or all of them
7. Check the collision of all the boxes in the manager
8. Respond to those collisions (change the color of the box if there is a collision).

For either class you can implement as many extra methods as you need.

If you decide to go for any of the extra challenges you should write in the readme file what you did as extra and explain and why do you think you deserve the extra points.

Extra:

15% - Have a different collision response other than changing the color of the shapes, like moving the other objects as well or impeding the movement of the shapes in a spot that is already occupied.

15% - Implementing other collision detection like Box to Sphere or ray to Box, etc.

Submit to the dropbox labeled A8 – Rough Axis Aligned Bounding Box