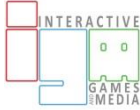
	<p style="text-align: center;"> Rochester Institute of Technology Golisano College of Computing and Information Sciences School of Interactive Games and Media 2145 Golisano Hall – (585) 475-7680 </p>	
---	--	---

Data Structures & Algorithms for Games & Simulation II
IGME 309, 2014 Spring
A10 - Bounding Object Manager

Due: Wednesday Nov/04/2015 at 11:59pm

This homework assignment is meant to be completed with your project team.

For this purpose a project in your repository needs to be setup. The way you will deliver your assignment is by providing me and the grader access to your repository (if you haven't done so already), 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 good doing what in code. As such, all 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 (in the team dropbox), if you have more than one submission (if you submitted something and improved on it after, for instance) this new submission will override the previous submission from ANY member, I will only grade the very last submission to MyCourses.

10% of your grade is a peer-evaluation, this evaluation is personal and needs to be submitted in the A10 **INDIVIDUAL** dropbox in MyCourses. 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(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 (Re)Aligned Bounding Boxes and the use of pre-tests for collision detection (implementation and collision detection, no collision response is necessary yet, Oriented Bounding Boxes through the Separation Axis Test 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 Object class which will be able to create Bounding Spheres and Axis (Re)Aligned Bounding Boxes (no Oriented Bounding Boxes just yet but if you feel like working ahead of time, by all means do); calculate their position in the world space and finally draw them on the screen surrounding different objects.

After the creation of the Bounding Object class you should create a BOManager through a singleton which will let the user create new BO for different objects from their list of vertices; 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 for the visual representation of the BO.

This is related to the MyBoundingSphereClass and MyBoundingBoxClass from ICE 11 and 12.

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.

It is not necessary to use “ReEngine” for this homework assignment, you can implement your own code for this, as usual “ReEngine” 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 the screen (if not with textures, at least with different colors); 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 IsColliding method needs to first check for a BoundingSphere vs BoundingSphere if that gives a positive then you will need to try the reoriented Box vs reoriented Box

Your grade will be as follows:

60% for the MyBoundingObjectClass

At a bare minimum the MyBoundingObjectClass should let the user:

- a) Add a BO based on a model (by its list of vertices)
- b) Switch the visibility of the BO
- c) Get the Centroid (in world coordinates) of the BO

- d) Get the Minimum and Maximum of the BO
- e) Get and Set the “Model to World” matrix of the BO
- f) Set the color of the BO
- g) Draw the BO (if you are working with ReEngine, add it to the render list)
- h) Check the collision with other BOs

40% of your grade is related to the BoundingBoxManager singleton.

At a bare minimum it should:

- a) Add a box based on a model
- b) Switch the visibility of the AABB
- c) Get the number of BO in the manager
- d) Set the color of a BO
- e) Set the visibility of a BO
- f) Render any specific BO or all of them
- g) Check the collision of all the BO in the manager
- h) Respond to those collisions (change the color of the BO 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 A10 – BoundingBoxManager and A10 – Peer Eval.