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

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

**IGME 309-01, 2015 Spring**

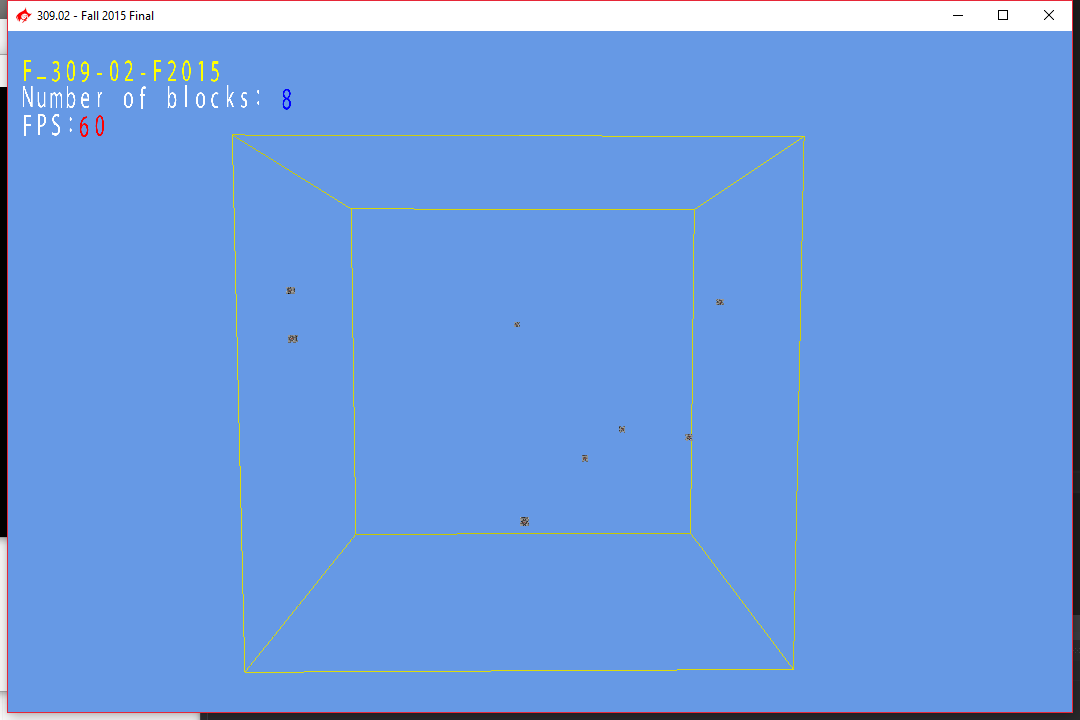
**Final – Practical**

Instructions:

A) Read this whole document before you start.

B) Using the provided code, implement a program that replicates the behavior described below and in the provided sample (FINAL\_DEMO.exe under \_Binary)

You are handed out code that will draw this out of the box:



Whenever you press “p” a new Companion Cube is added to the list of models and a new BoundingObject is assigned based on that model to the manager at a random position in the screen.

Your goal is to subdivide the octant and display the sub-octants necessary to divide the space in such a way that only an ideal count of 2 objects are inside of each octant leaf.

Everything in the main program is already in place so there is no need to modify ANYTHING other than two methods under MyOctant.cpp. Said methods are the Display and the Subdivide method, both of them should be recursive methods.

You are provided starting code and you will only need to focus on the MyOctant.cpp, no other files need any changes, assume that nothing in the provided code need any changes (unless you are aiming for extra credit).

Tips:

Each octant starts marked as it does not contain any children (m\_nChildren = 0) when you subdivide the octant you will always subdivide it into 8 children and thus you need to properly mark that.

You can check if a node needs to get subdivided by calling ContainsMoreThan(m\_nIdealBOCount); this will return a Boolean flag. Check the example at ConstructTree.

Requirements:

* Your code MUST compile AND execute. I will not take points out of the program if it doesn’t compile AND/OR run, I will simply not grade it. If your program does not run it will receive a 0/100. If you are having trouble with something in the code comment out the lines, say what you wanted to do and what you suspect the issue is. That will result in partial credit, which is better than not having a grade.
* Memory Leaks are acceptable, points will be taken off, but the code will be reviewed.
* You only need to modify MyOctant.cpp; there is no need to modify anything else.
* You get rid of the “trash files” (intermediary files).
* Zip your solution and upload it to the dropbox in my courses.
* There are provided variables to hold your values, threat them as hints so feel free to use them or create your own.

Grading:

(-???) Cheating:

Talking with anyone in person or online. You are only allowed to use MyCourses to download this file or upload your solution. Anything else is considered cheating.

(-100) Code not compiling or executing.

(-10 to -20) Memory leaks (You are not reserving new memory for this test so this shouldn’t be an issue)

(-10) For each uncommented method; I need to know what you are doing or trying to do.

(-10) You forgot to delete the \_Delete folder

(-10) You forgot to delete the .sdf file

(-50) You are not subdividing your octant properly displaying.

(-50) You are not displaying the Octants.

Extra points:

For the extra points you are allowed to modify anything in the program but you need two submissions one for the regular part and one that is clearly labeled extras:

(+50%) Collision Check and response

Surprise me (in a good way). As I don’t know how surprised I will be I can’t tell you how many extra points I will give you, just do your best, and as usual, in order to get the extra points you need to have a satisfactory degree in the required part.