|  |  |  |
| --- | --- | --- |
|  | 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**

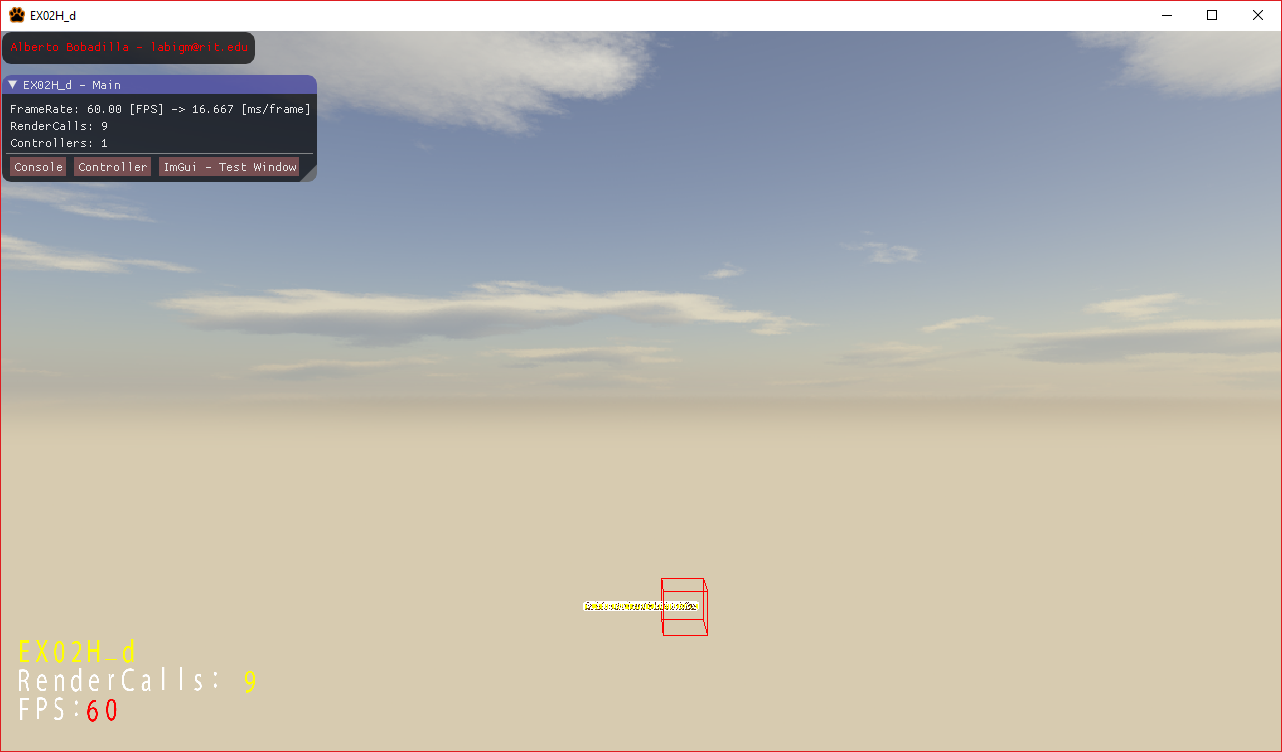
**Final – Practical**

Instructions:

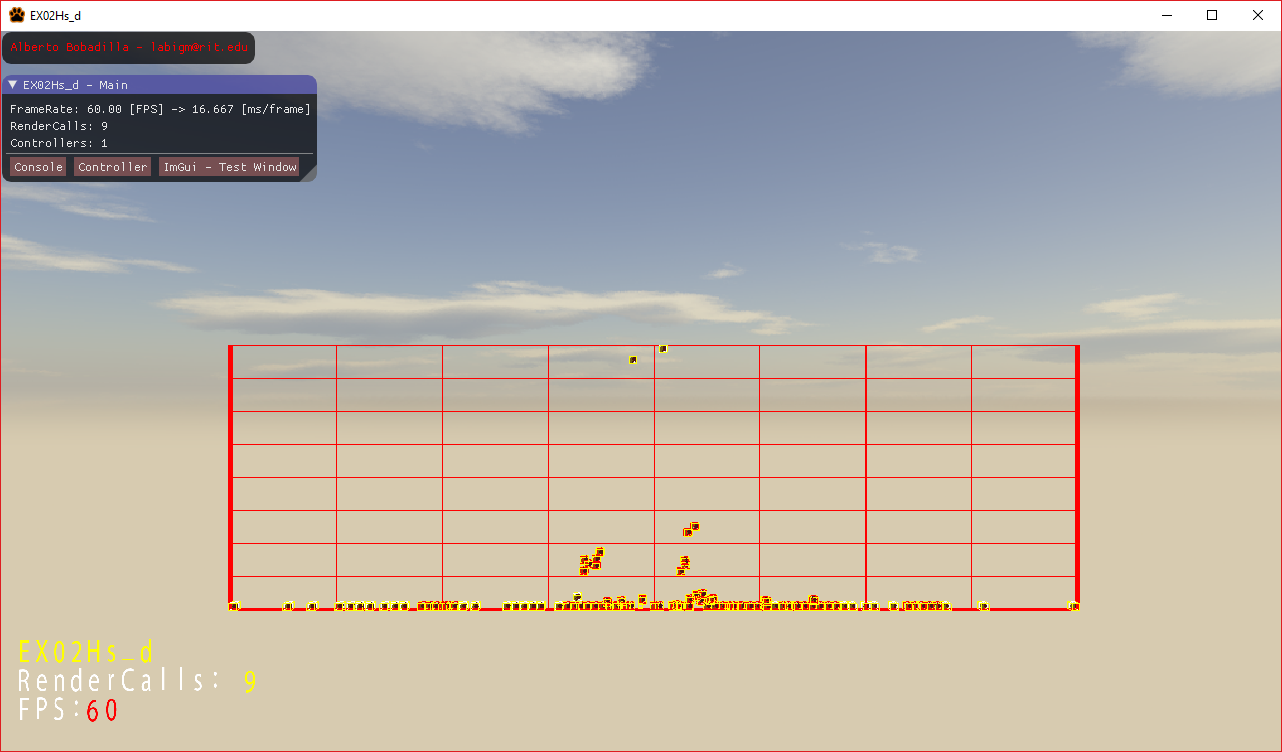
A) Read this whole document before you start.

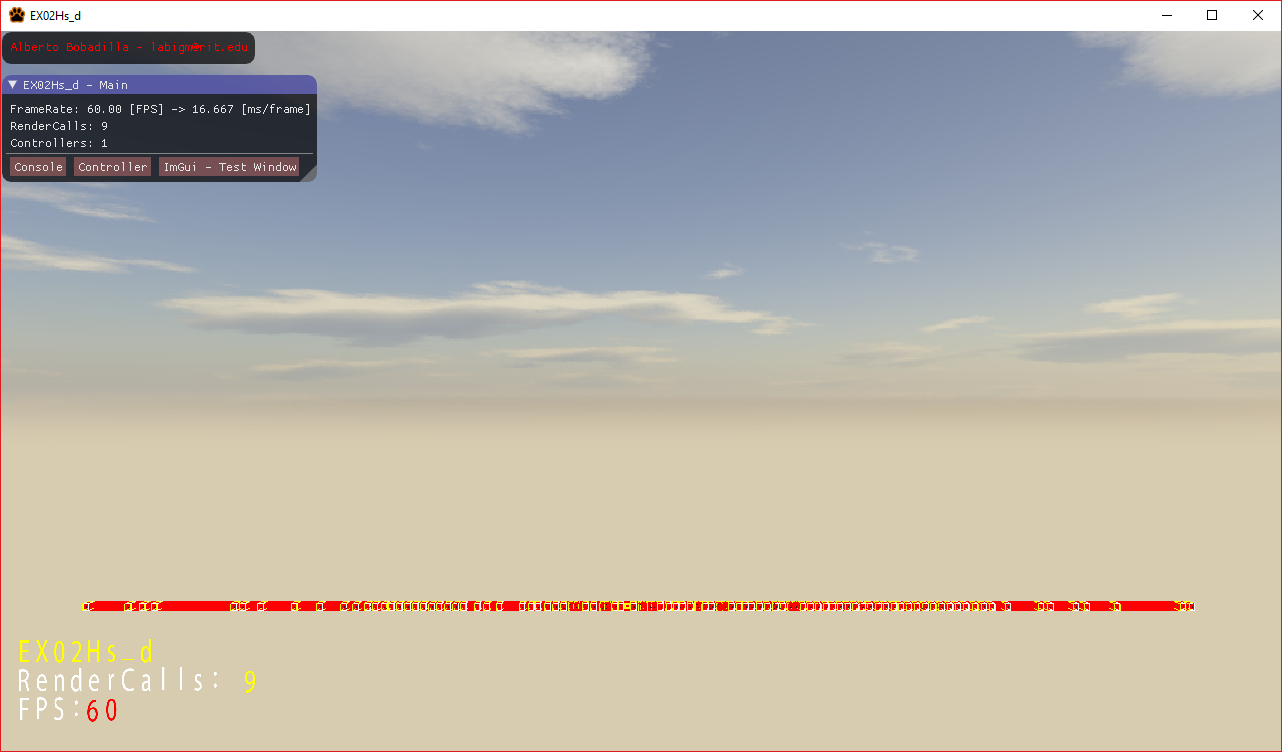
B) Using the provided code, implement a program that replicates the behavior described below and in the provided sample (FINAL\_DEMO.exe under \_Binary)

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



The full version should look like this:





As the objects fall the surrounding space will resize itself and the objects will separate from each other.

**(Example demo under Binary)**

There are only three methods to complement in order to fix this code, one is the void MyRigidBody::SetModelMatrix(matrix4 a\_m4ModelMatrix) under the MyRigidBody.cpp, in there you need to initialize the values for the globalized variables to construct the ARBB from the AABB (you can assume the AABB is calculated right). The second method to implement is the void Simplex::Space::Resize(void) under the Space.cpp file, this method is meant to calculate the right size and center of the space based on each of the ARBBs in the system, for this method to work properly you need to have completed the first one. The third method is void Simplex::Space::Subdivide() which will subdivide the space into smaller spaces each recursion, will be half the size of the parent and all will be evenly distributed.

Using the arrow keys you will be able to move one of the objects that might be able to push other around, that part of the functionality is done already but it will resize the scene so you can try the code. If you have the ARBBs are not right the objects will not push each other correctly.

Everything in the main program is already in place so there is no need to modify ANYTHING

You are provided starting code and you will only need to focus on the two files I mentioned above, no other files need any changes. You should leave alone the regions marked as “does not need changes” and all of the 3 methods are worth the same amount of points.

**There is no need to mark what objects are in what space, nor the other way around. This is just creating the space, I’m not going for the full homework assignment here.**

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 Space.cpp and MyRigidBody.cpp; there is no need to modify anything else.
* Your submission will only be the two mentioned files in a zipped folder.

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.

(33) SetModelMatrix

(33) Resize

(34) Subdivide