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

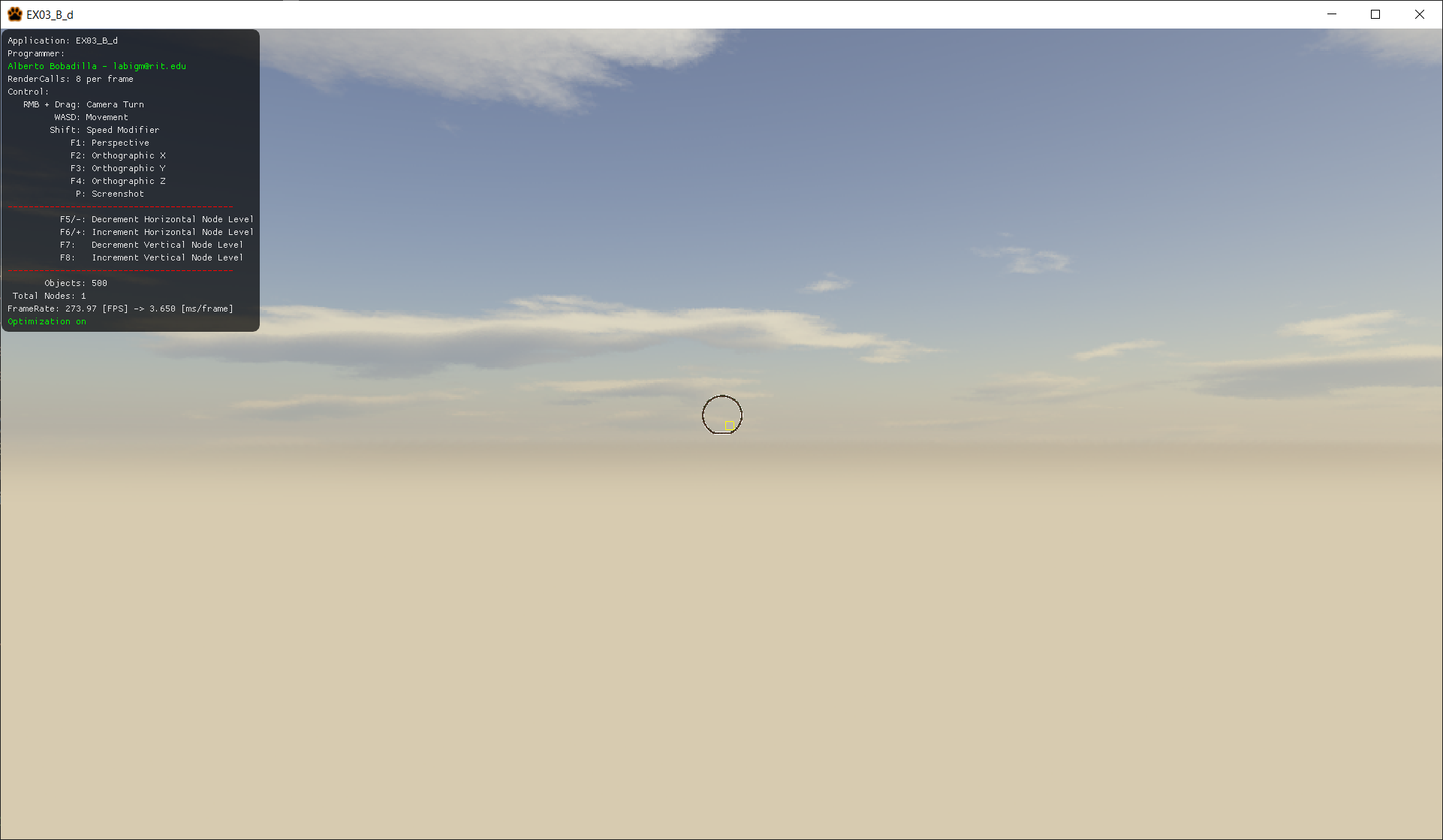
**Final – Practical**

Instructions:

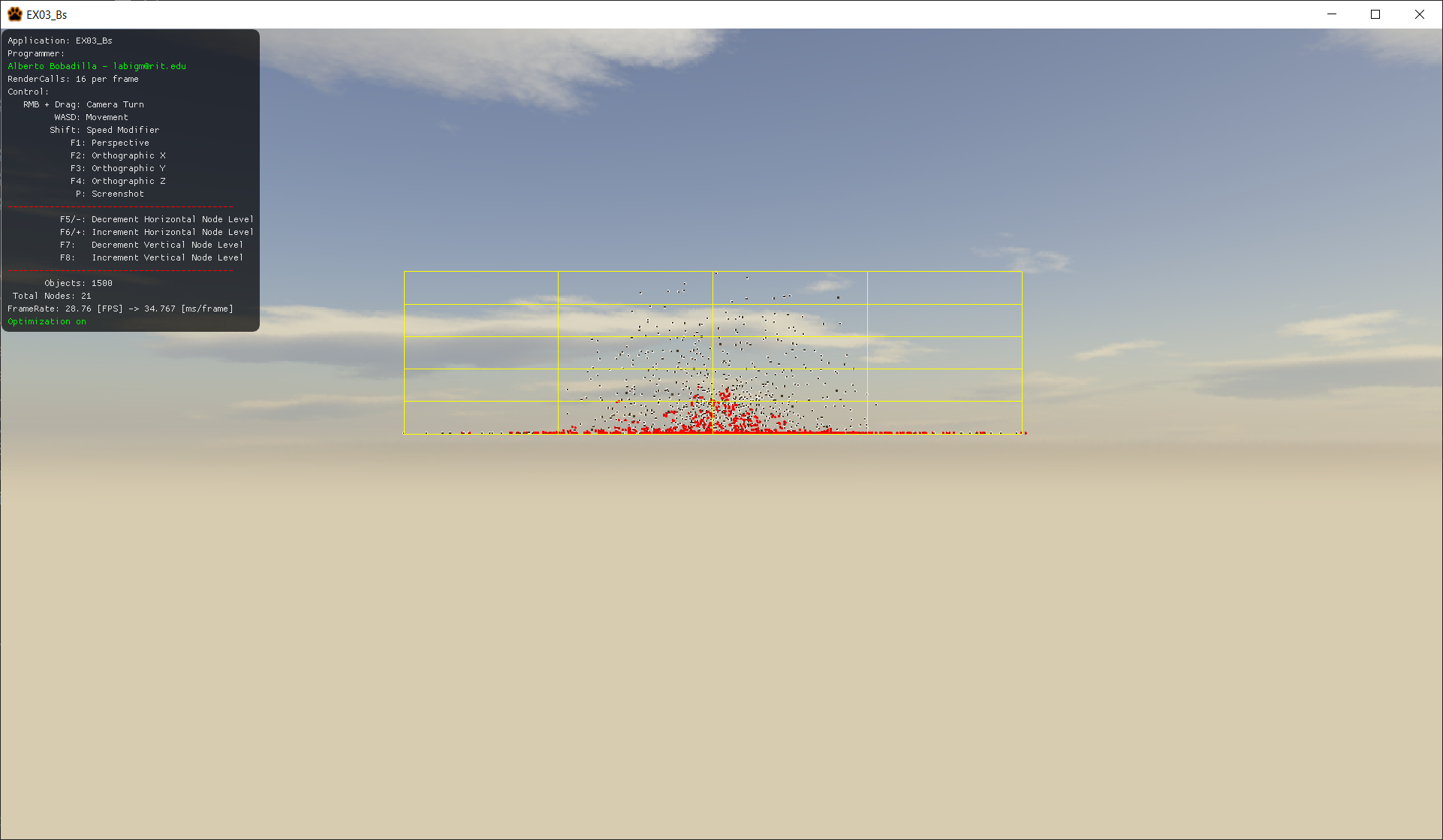
A) Read this whole document before you start.

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. In order to prevent killing the performance, this space resizing will only happen 10 times a second.

**(Example demo under Binary)**

There are only four methods to complement in order to fix this code all of them inside of Space.cpp:

Space::Space(uint a\_uWidthSubdivisions, uint a\_uHeightSubdivisions), In There you are expected to construct the variables for the Space class, some values are initialized to 5 just for the sake of having something, all of those values are wrong.

bool Space::IsColliding(uint a\_uRBIndex), In There you are supposed to check if the space is colliding with the specified entity (by index) in the entity manager.

void Space::Subdivide(uint a\_uWidthSubdivisions, uint a\_uHeightSubdivisions), In there you are supposed to subdivide your scene in a grid of uWidthSubdivisions by uHeightSubdivisions in size, F5 to F8 will let you add or decrease the subdivisions, but you do not have to worry about that functionality, it is already in place.

void Space::AssignIDtoEntity(void) ), In this one, after subdividing the space into smaller spaces you are supposed to assign the ID of this Space to any entity in the EntityManager that would be colliding with a leaf.

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.

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 one file I mentioned above, no other files need any changes. All of the 4 methods are worth the same number of points.

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; there is no need to modify anything else and this is the only file you need to submit UNZIPPED.

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.

(-20) You submit anything other than Space.cpp UZIPPED

Each of the four mentioned methods will receive 25 points each.