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

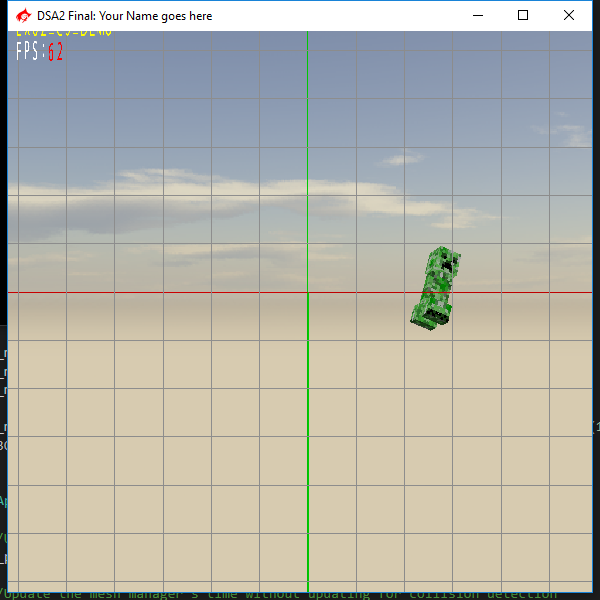
**Final – Practical**

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 (Ex02\_Cs\_DEMO.exe under \_Binary)

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



Will draw a creeper model rotating in the in 1 degree in all of the 3 X Y and Z axis. And moving using a lerping method (feel free to remove the lerping of the matrix while working, it is there to help you visualize your BO’s are actually moving with your creeper)

The rotations of said creeper are affected by the dreadful Gimbal Lock.

Your goal is to eliminate the Gimbal Lock by any means and have the object rotate the right way (again there is an example under the \_Binary folder)

Everything in the Update method is already in place for this effect. Each render call the Update Method will call the UpdateOrientation method which in turn will update the **m\_m4Orientation** matrix that will be used to update the model and the Bounding Object.

For the first part of the final you are expected to eliminate the Gimbal Lock of the orientation matrix, all your code should be executed on the UpdateOrientation method.

For the second part of your final you are expected to complete the Constructor of the BoundingObject class, it will take the list of vertices from the model and generate an Oriented Box out of them.

For the third and last part of your final you are expected to modify the SetModelMatrix method of the Bounding Object class in such a way that when executed it will calculate the Axis ReAligned Bounding Box.

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 the following methods:
  + void AppClass::UpdateOrientation(void)
  + MyBOClass::MyBOClass(std::vector<vector3> a\_lVectorList)
  + void MyBOClass::SetModelMatrix(matrix4 a\_m4ToWorld)
* You get rid of the “trash files” (intermediary files).
* Zip your project (just your project not the whole solution, it should be less than 200kb) and upload it to the dropbox in my courses.

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

(30) Getting rid of the Gimbal Lock

(30) Generating the Oriented Bounding Box

(30) Generating the Axis ReAligned Bounding Box

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:

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.