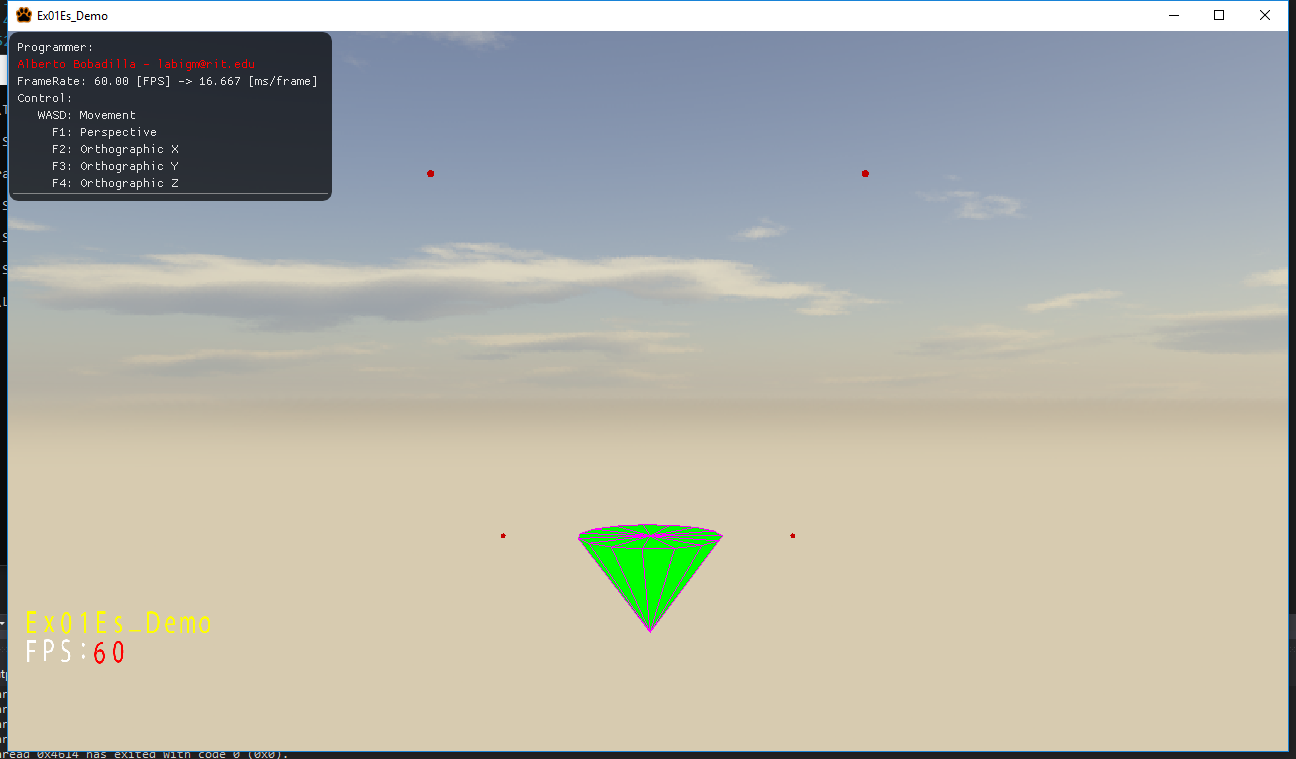
|  |  |  |
| --- | --- | --- |
|  | **Rochester Institute of Technology**  **Golisano College of Computing and Information Sciences**  **School of Interactive Games and Media**  **2145 Golisano Hall – (585) 475-7680** |  |

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

**IGME 309**

**Midterm – Practical (version e)**

In the class repository, I’ve included a solution under the \_Binary folder. Please take a look at that before continuing reading this document.



What you are expected to do is replicate the demo to the best of your abilities.

You will start with a cube at the center of the world, if you middle click you can rotate it around its center.

From the starter code there are two things you need to do:

1. Modify the MyMesh::GenerateCone method under MyMesh.cpp in order to generate a Cone. The center of the circular face needs to be located at (0,0,0) and the tip of the cone at (0,-a\_fHeight, 0) so the cone is upside down.
2. Modify the Application::Display method under AppClass.cpp modify the m\_sProgrammer variable accordingly. Then, based on the positions provided under m\_v3StopList you need to do perform linear interpolation between all stops and restart from the last one (no jumping from the last to the first, it needs to do a smooth transition). The list will include a larger or smaller number of stops for grading purposes so your transition between routes need to be dynamic, no hardcoded values)

The program will do a linear interpolation between an undetermined number of points in space. These points will be read from an .ini file with the same name as the program executing it, if there is no .ini file the program should be able to generate one and default values within it. The user should be able to modify, add, remove existing points in that file to create and design new stops. The reading from that file is already coded in the program so no need to focus on that, just the linear interpolation between stops repeating from the last one.

Your grade will be 100% if your solution behaves like the one provided and deductions will happen as follows:

-50% (up to) if you don’t generate a Cone based on the arguments

-50% (up to) failing to perform the linear interpolation

-20% If your shape doesn’t visit all the stops (you hardcoded the routes).

-20% If your shape does not perform a smooth transition between stops or flicks between them. This includes not transitioning from the last stop to the first one.

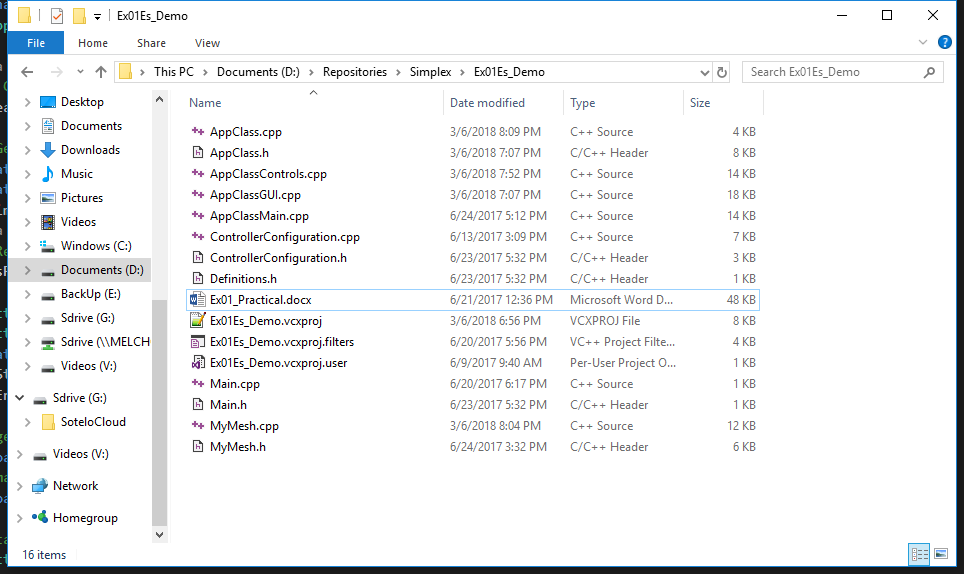
-10% If your linear interpolation does not use real time (the provided fTimer variable can be used and modified in any way you need)

-20% If you did not comment your code.

---In no case the grade could be less than 0%

***Submit to the dropbox labeled Ex01 –Practical***

As usual the required submission asks only for the project folder, not the whole solution, it should be no larger than 200kb if you are using the class framework. The content of your .zip file in your submission should look like this:



After you submit your file it is your responsibility to download your submission and make it is what you worked on and not the starter code, it has happened to other student before, do not let it happen to you.