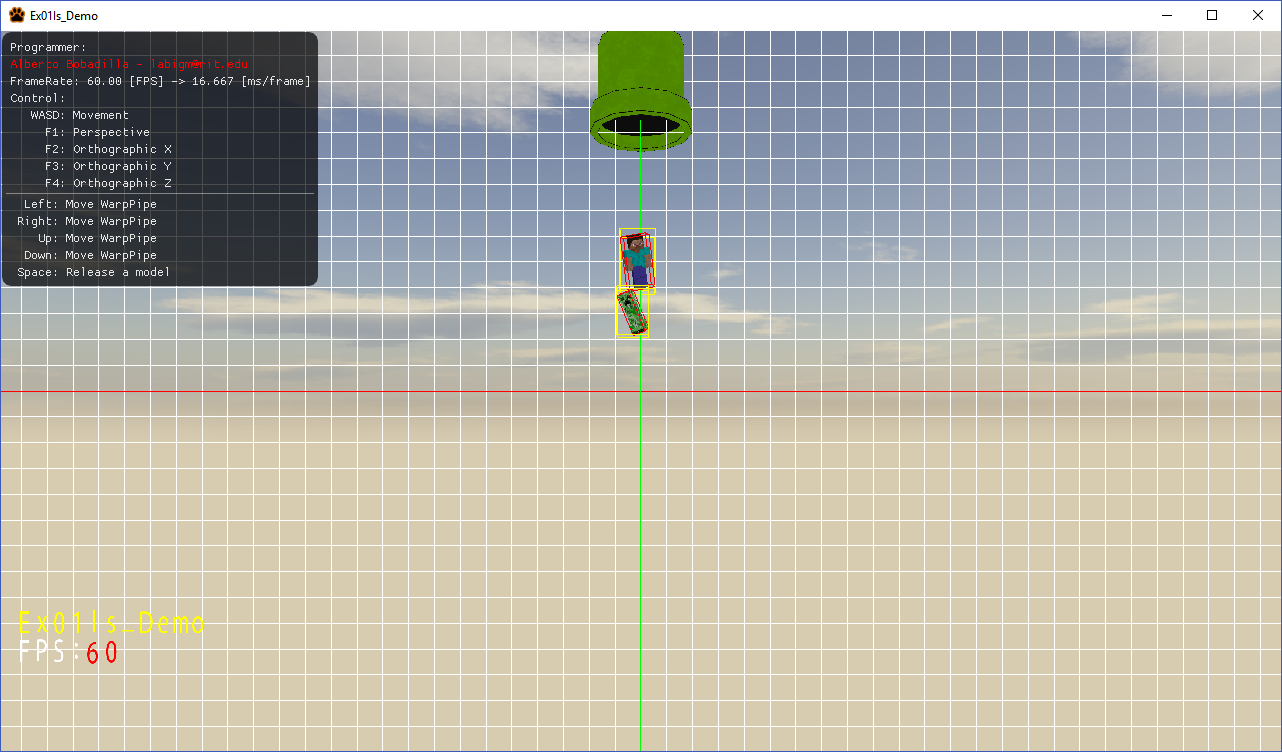
|  |  |  |
| --- | --- | --- |
|  | **Rochester Institute of Technology**  **Golisano College of Computing and Information Sciences**  **School of Interactive Games and Media**  **2145 Golisano Hall – (585) 475-7680** |  |

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

**IGME 309**

**Midterm – Practical (version i)**

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



You are working on a game app that will contain an element that will shoot models down a to the game world unfortunately the player has control over when the models are shoot down by pressing the space bar and this means that the object can collide among themselves and collisions need to be resolved. Thankfully for now your job consists in knowing if the objects are colliding only and not resolving the collisions. There is another part of this that the lead programmer forgot to add and that is controls over the position of the pipe, it should be located where the variable m\_v3PipePos is but it needs to be modified trough the directional keys.

Fortunately for you everyone in the programming team worked hard and have the vast majority of the code working; they made all the variables you may need for the main program and the basic drawing framework and made sure all memory was handled correctly.

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

1. Change the name of the programmer (you)
2. Make sure the warp pipe moves when the arrow keys are pressed
3. Make sure the models are drawn in the respective positions and orientations (m\_lPosition m\_lOrientations)
4. You will need to construct the Oriented Bounding Box (AKA the AABB when is viewed in its local space)
5. You will need to construct the Axis Realigned Bounding Box
6. You will need to check collisions between the objects using the ARBBs (the part on the AppMain is already calling the function for you but the IsCollidingMethod is not complete)

Tips

1. You are allowed to add more variables and methods as needed.
2. Memory is handled for you unless you make new variables

Your grade will happen as follows (up to 110%):

20% (up to) if you can move the WarpPipe

20% (up to) if you display the meshes in the right position and orientation (they will be falling down as m\_lPosition[i] += vector3(0.0f, -0.1f, 0.0f); will keep decrementing their height)

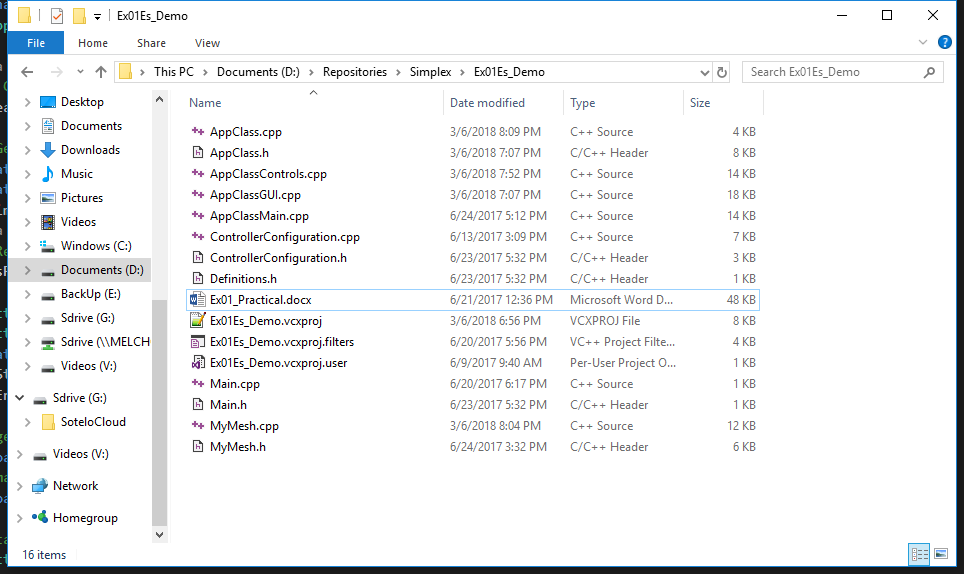
30% (up to) if you can draw the oriented bounding box (AKA the AABB)

30% (up to) if you can draw the ARBB

10% (up to, extra) if you calculate the collisions correctly

***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, if you are using extra media files (i.e. for music) please include them and let me know in a README file the right location relative to the executable. The content of your .zip file in your submission should look like this (names might be different):



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.**