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

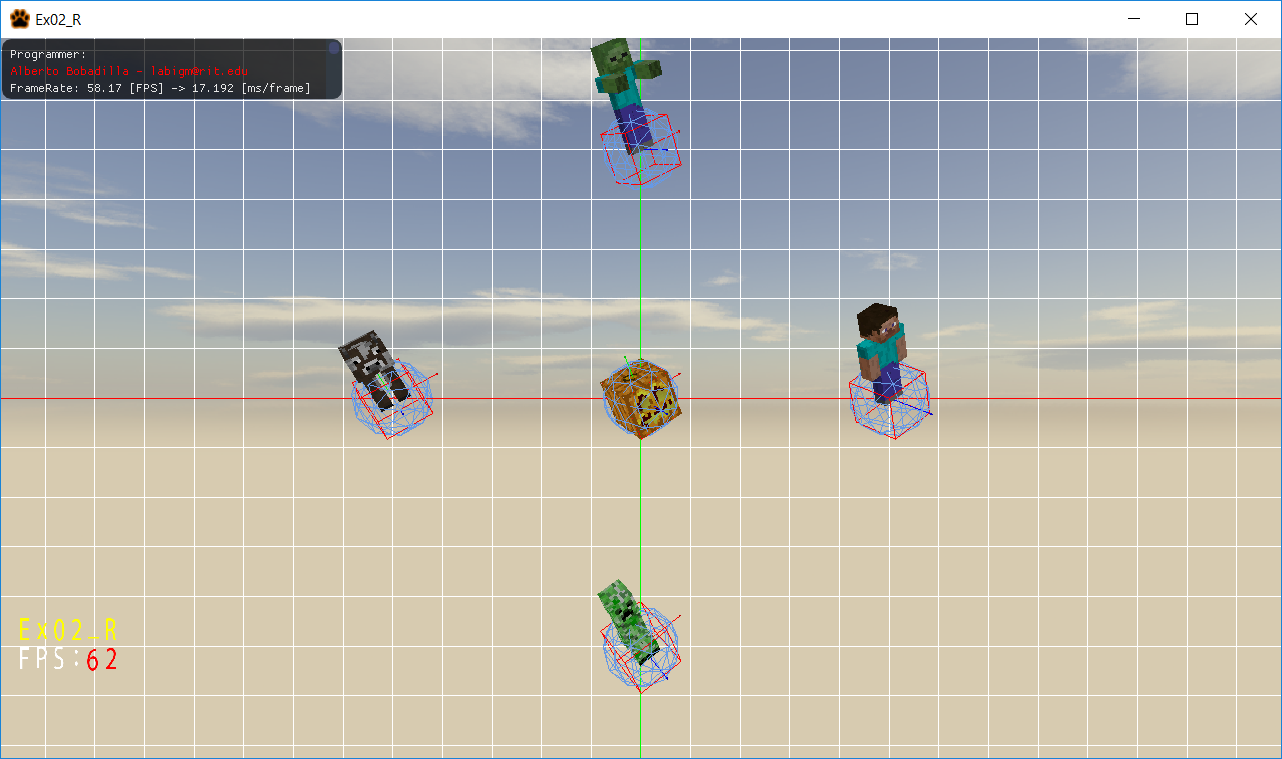
**IGME 309**

**Second exam – Practical (version r)**

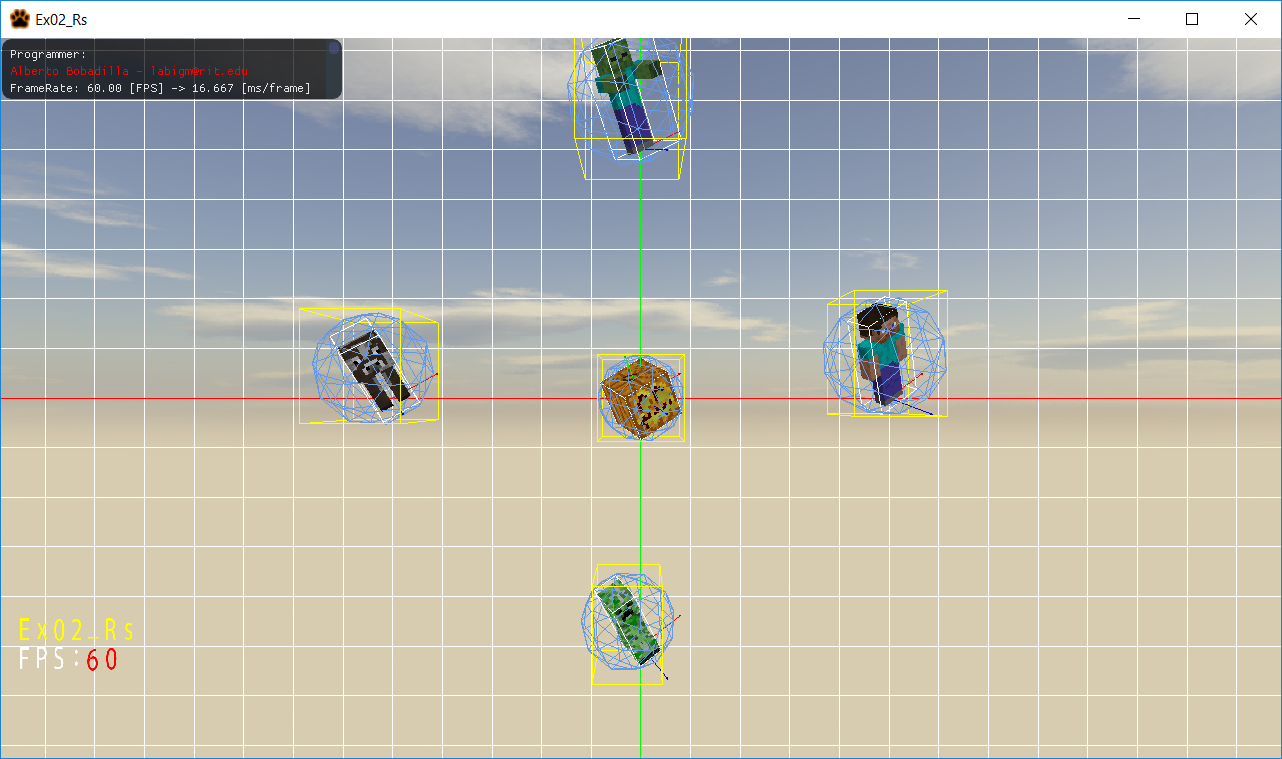
Video instructions can be found here: <https://youtu.be/I5K6LDO3tZE>

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

Right out of the box you will begin with this:

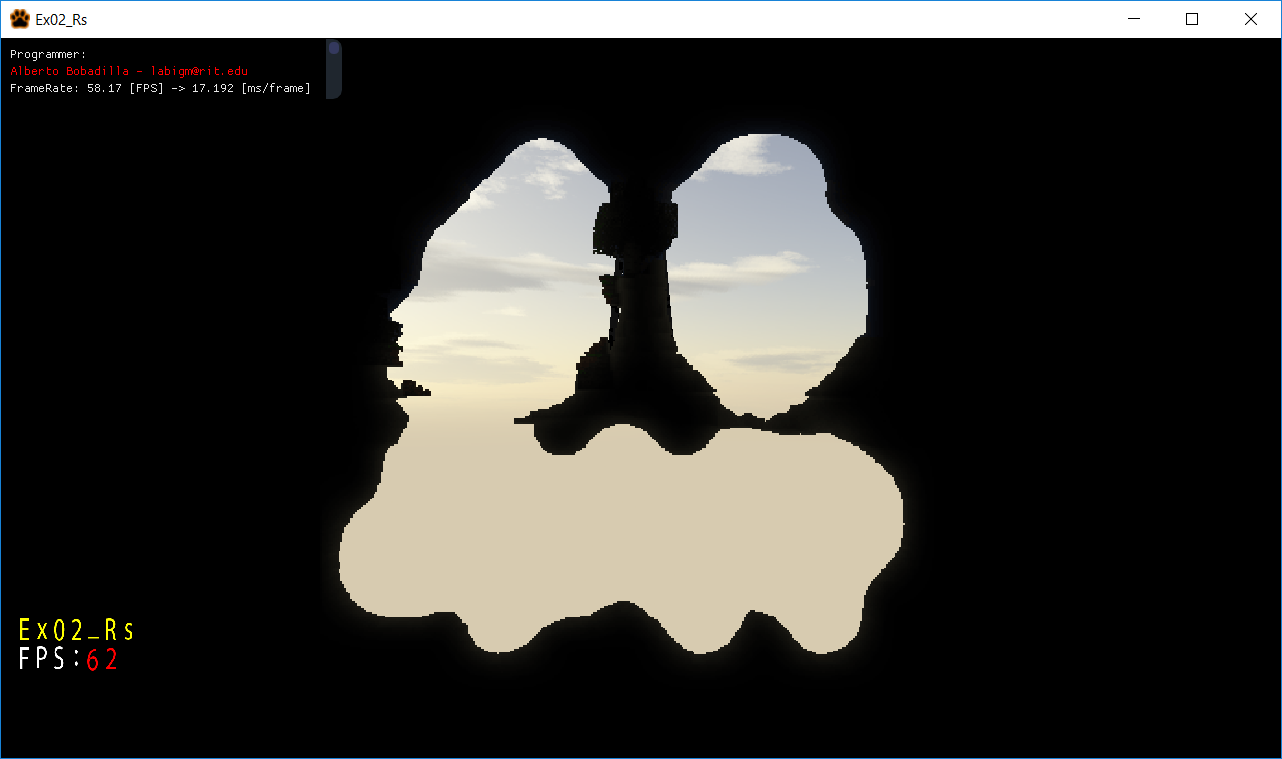


You are working on an app that will display the bounding sphere, axis realigned bounding box and oriented bounding box of every model in the scene. The end result should look like this:



Aside from calculating the Rigid Bodies you are also expected to calculate collisions among the objects, either using ARBB or BS.

Finally, by pressing ‘C’ on the keyboard you are expected to change the view matrix of the camera from within the pumpkin.



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

1. Modify the MyRigidBody.cpp Constructor to generate the initial values for the class.
2. Modify the MyRigidBody.cpp SetModelMatrix method to update the necessary values of the ARBB when the model matrix is modified.
3. Modify the MyRigidBody.cpp IsColliding method to check if a collision really exists (right now all objects falsely mark that they are colliding)
4. Modify the AppClass.cpp RenderFromInsidePumpkin method to not only see though the pumpkin (as in the starting code) but to also update the view as the object rotates.

Tips:

1. There is nothing to modify in any other method or file aside of what is mentioned above. You may add any helper functions as needed, but if you do, you will need to upload the modified files as well.
2. WASDQE will rotate the objects via a quaternion, the functionality is already in place and there is no need to modify or add anything. (when in pumpkin view only the pumpkin will rotate, not the other objects)
3. While in pumpkin view, there is no translation to keep things simple.
4. Arrow keys will help you move the pumpkin to check collision detection.
5. Your files need to compile, as if the code does not compile it gets an automatic 0. It’s better for you to comment the faulty lines and receive partial credit than no credit at all.
6. You are allowed to add more variables and methods as needed. But you should **only submit two files MyRigidBody.cpp and AppClass.cpp inside of a single zip file with the name Ex02.zip.** If you are adding more variables make sure you are giving me all the files I need to grade.
7. Memory is already handled for you unless you make new variables.

You will be graded as follows (up to 100%):

25% (up to) if you can generate the AABB and BS

25% (up to) if you can generate the ARBB

25% (up to) if you can check collisions

25% (up to) if you can modify the view matrix from inside the pumpkin in such a way that it follows the rotations

Deductions will happen as follows (no more than 100% deductions can happen):

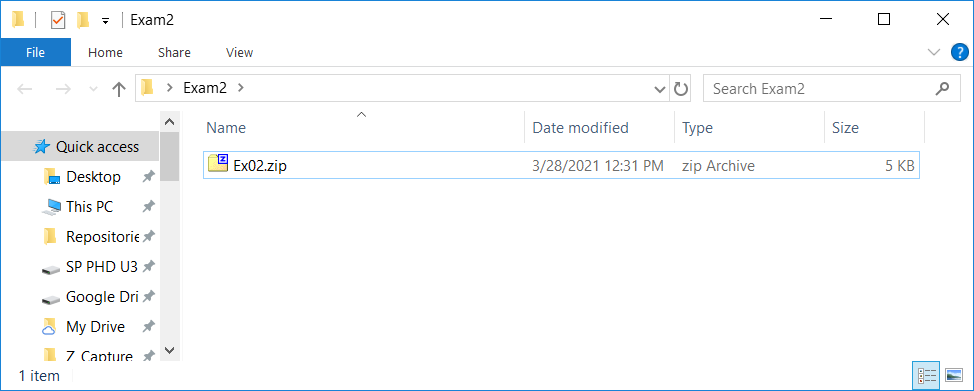
-100% your code does not compile

-100% (up to) your code is hardcoded some how

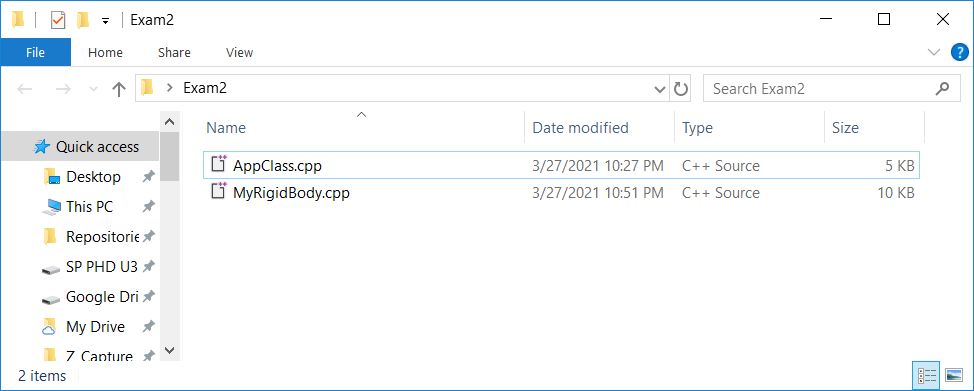
**-20% you submit anything other than a single zipped file (named Ex02.zip) containing MyRigidBody.cpp and AppClass.cpp files** (unless you really needed to include more files for some reason)

***Submit to the dropbox labeled Exam 2 –Practical***

The required submission asks only for a single zipped file (named Ex02.zip) containing MyRigidBody.cpp and AppClass.cpp, not the whole solution, it should be no larger than 10kb, if you are using extra files for creating variables and such, please include those files as well as a zipped file. The content of your submission should look like this (routes might be different):



The content of that zip file should be this:



You are also required to push to your repository and give me the link to it in the comment of your submission. But there is no penalty if you forget.

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 students before, do not let it happen to you.**