|  |  |  |
| --- | --- | --- |
|  | 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 309-02, 2015 Spring**

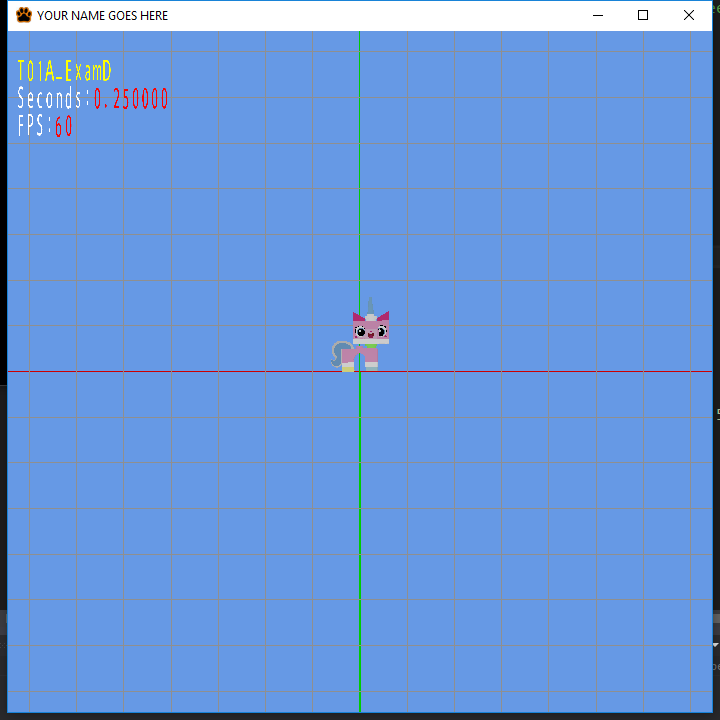
**Midterm – Practical**

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 (MIDTERM DEMO.exe under \_Binary)

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



Your goal is to make the model perform a triangular move around the origin in a time based manner. (Example demo under Binary)

You are provided starting code and you will only need to focus on the AppClass.cpp, no other files need any changes, assume that nothing in the provided code need any changes (other than setting up the values for the variables).

**Your animation need to last 1.5 seconds per stop (4.5 seconds total). It needs to move to 3 different locations, given by the control points:**

(-2.25f,-2.25f, 0.0f)

( 0.0f, 2.25f, 0.0f)

( 2.25f,-2.25f, 0.0f)

**After that the animation restarts. Again, please refer to the example demo.**

Requirements:

* You need to scale down the model as well. When loaded it is 4 times larger than what it should be when rendered. Although that transformation is already stored in the matrix m4Scale
* 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 AppClass.cpp; there is no need to modify anything else.
* You get rid of the “trash files” (intermediary files).
* Zip your solution and upload it to the dropbox in my courses. The solution will be around 20 megabytes in size.
* It is not required to show the control points but if you want to make your life easier it is not a bad idea to do so. The function call looks like this:
  + m\_pMeshMngr->AddSphereToQueue(matrix, color, WIRE);

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

(-50) Your application is not time based

(-50) Your model translates, but does not reset to the original position and start over.

Extra points:

There are no other extra points specified for this exam, except for one thing: 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. If you are going for the extra credit make a new project within the same solution and name it something else.