Math 155A - Introduction to Computer Graphics – Winter 2017

Instructor: Sam Buss, Univ. of California, San Diego

Project #2 – Create and animate a rolling disk.

Due date: Friday, January 27 at 9:00pm.

Goals: Learn some simple uses of translation and rotation. Learn some animation techniques.

What to hand in: When you are done, place your C++ files, executable, and Visual Studio solution together in a separate folder in your PC computer account in the APM basement labs. As usual, there should be nothing in this folder except your files for this homework assignment, and the creation/modification dates should match the turn in date. The program must compile and run on these computers.

Grading will be personalized and one-on-one with one of the TAs Srivastava or Jingwen or with Sam Buss. Your program must run on the PC lab, you must come into the PC lab and meet one of us. You will have to show your source code, run the program, make changes on the spot to your program and recompile as requested by the grader, and be able to explain how your program works and why it renders what it does. The grading should be completed promptly, but no later than the due date for the next programming assignment.

FOR PROJECT #2, PLEASE DO THE FOLLOWING STEPS #1 - #5.

1. **Download** the RollDisk program from the zip file RollDisk_Winter2017.zip. Extract these into a directory named RollDisk. Be sure to save this in your networked, permanent folder. Files left elsewhere on the computer lab computers are likely to be to erased automatically! (The full URL for the zip file is: http://www.math.ucsd.edu/~sbuss/CourseWeb/Math155A_2017Winter/Project2/RollDisk_Winter2017.zip.)
2. There is an executable "**RollDiskDemo.exe**" that shows (more-or-less) **how your program should end up**. This version and should run on the computer lab. If you are using another system, you may also try the alternate version, RollDiskDemo_freeglut.exe.
   Experiment with this program. Notice the following items and keyboard commands.
   a. The disk is rolling back-and-forth on a circular track without slipping.
   b. If the rate of movement is too slow, press "R" to make it faster. If it is too fast, press "r" to make the rate slower.
   c. Press “s” to single-step the animation. Press “a” to restart the animation after single-stepping.
   d. Press “D” to make the dimension (the size) of the disk larger. Press “d” to make it smaller.
   e. Press “p” to toggle between two modes of animation: In the first animation mode, the disk moves at a constant velocity at all times. In the second animation mode, the speed varies with the height of the disk, roughly matching the physics of potential energy versus kinetic energy (but without taking the moment of inertia into account). For the second mode, note how the disk slows down at the extreme range of motion.

3. **Find the source files “RollDisk.cpp” and “RollDisk.h” in the zip fie and use these to create a Visual C++ project. Examine the source code and run this program.** This program acts somewhat like the **RollDiskDemo.exe**. However, it draws a sliding rectangle instead of the rolling disk. When you examine the source, do the following:
    a. Examine the code: You should be able to understand **nearly everything** except for the portions of the code dealing with the modelview matrix and the perspective matrix.
    b. The other part that is new is the use of **glTranslatef and glRotatef.** These will be discussed in class on Monday. You will need to use these when you rewrite the code (for step 4 below).
    c. You should try to understand how the animation works, at the very least the constant velocity animation mode.
4. **Re-write the code for RollDisk.** Match as, closely as you can, all the functionality of the demo program. The main task is to draw the rolling disk. Most of the rest of the code, including most of the animation logic can be left intact. Be sure to:
    a. Have the disk look essentially the same as in the demo program, with filled in and wireframe portions. For this, you might want to use **glPolygonMode** (see page 14 of the text), so that you can form the disk from a triangle fan plus additional triangles. Please use a loop to build the vertices of the disk instead of typing a large set of explicit vertices!
    b. Have the disk follow along the circular path of the “trellis”.
    c. Have the disk roll without slipping.
    d. Support all of the keyboard controls of the demo program.
5. **Turn in the project as described above.**

Program grading: Scale of 0 to 10. Personal grading session with a TA or the professor.