Programming Assignment 3 (100 points)
Texture Mapping & Particle Systems
Due on your Moodle drop-box by 12:00 Noon on Tuesday, October 31, 2017
In this extension of the “Jumping Brains” program, you’ll add some texture mapping and alter the animation so the characters will not plummet over the edge of the disk platform, instead turning around and hopping in the opposite direction. In addition, you’ll add a pattern of particle systems representing clumps of grass covering the disk. We return to the single-sphere brain in this assignment, so you may use a modified version of one of your previous assignments, but code illustrating how to texture map the disk is available on the course web site (http://www.cs.siue.edu/~wwhite/CS482/Syllabus.htm), and that may prove useful in this assignment.

Three changes will be needed to extend the “Jumping Brain” program appropriately. Specifically:

- Several bitmaps have been provided to you (in the zipped folder on the web site) for use on the brain spheres of the seven Jumping Brain objects. Using the disk texture mapping as an example (along with additional examples in your textbooks), apply these textures to those spheres to afford them a more brain-like appearance, as illustrated on the back of this page. Use the color of each Jumping Brain’s limbs to determine which texture to apply to which object.

- In the previous assignment, the Jumping Brain characters plummeted off the side of the disk when they had hopped far enough. In this assignment, alter that behavior so that each Jumping Brain will spin around 180-degrees when it approaches the disk’s border. After completing its spin, the Jumping Brain will resume hopping in the opposite direction. When it reaches the border on the other side of the disk, it should spin around and hop back in its original direction.

- The provided code includes a particle system class that is used to implement a basic raindrop system, which you will remove. Instead, you will enhance the scene by using this same approach to add blades of grass covering the “ground” disk on which the jumping brain characters reside. Specifically:
  1. The blades of grass should be arranged in small concentric circles to make a “clump” of grass (illustrated at right). These clumps should then be arranged in concentric circles to cover the ground disk (illustrated on the back of this page).
2. Each blade should be comprised of several particle segments, arranged to follow an appropriate path*. By randomly varying the parameters of the equations for these paths, the circular patterns of the grass clumps should be effectively disguised from the viewer’s vantage point.

3. When the application first begins, each blade of grass will "grow" from its source until a set number of particle segments have been generated.

4. No effort to deal with collisions between blades of grass, or between the jumping brain characters and blades of grass is required.

* - Graduate students are required to use parabolic paths.

Keep your code modular and readable, with an extensive explanation (including your name) at the top of each program file, explanatory sentences preceding each function, and in-line comments every place within the code where your logic is particularly complicated. Modify all comments in the provided code to reflect the changes that you make in your revised version. Avoid code redundancy by foregoing cut-and-paste in favor of placing any code that is needed repeatedly into its own module (function, class, structure, etc.).

Place all of your program files (just the .cpp and .h files, not your project file) into a single folder named with your last name. When the instructor creates a new Visual C++ project and properly attaches the OpenGL libraries and your code, it must compile and execute in order to be graded. Zip-compress this folder and copy it to your Moodle drop-box by Tuesday, October 31, 2017, at 12:00 Noon. Late assignments are not accepted without verifiable medical documentation. You must write your own code (with the exception of clearly annotated code that you receive from the instructor or as part of the material available from the textbook publishers), and no one but the instructor may see your code.