1 Final Project Plan

1. Feature descriptions:
   - Collision Detection: we will be using bounding boxes to detect the collision between balls. When two balls collide, we will be recalculating the new position and velocities of both balls accordingly, following laws of physics. The bounding boxes will let us know if the collision between given two objects may be happening. For objects for which collisions may be happening, we will use the position of the balls to then calculate more details about position of collision and resultant change in position of the balls. All collisions would be assuming perfect elastic collision, which ignores loss of kinetic energy through external forces like friction.
   - Shadow Mapping: we have not formally learned this topic. However, we scrolled through the shadow mapping resources in the Final Topics lecture, which are fairly well documented and will give us a good outline for how to proceed.
   - Texture Mapping: we will be adding textures to the pool table. The table itself will be made of wood, and the flat part will be made of felt. We may also make textures for the balls themselves. This will involve creating new shaders, and using the lab code for texture mapping. The general process would be finding a image of wood, and mapping each one of our primitives to a point in the texture map image.
   - Bump Mapping: for bump mapping, we plan on generating a simple noise function, and following the instructions in the lecture slides. Generally, the process seems to be straightforward: mapping each point on the object (which will be the pool table) to the noise modification.

2. Resources (click for link):
   - 3D collision detection algorithm
   - Shadow mapping
   - Texture mapping
   - Bump mapping
   - Elastic collision equation

3. High Level Overview:
   The scene will contain a table, along with the cue ball and the 15 other balls. We will have a camera that allows the user to view the scene from any angle, and a preset camera that allows them to look down the line of sight of the cue, and from bird-eye view. There will also be an option for the user to reset all of the balls into the starting position.
   There will be a direction vector on the cue ball which the user will be able to control to determine the direction that the ball is hit. We will also have a slider to control the force of the ball being hit.
   When the user says "hit", the ball will be hit by the cue according to the specifications, and the physics of all the balls being hit will be simulated. The bump, texture, and shadow mapping will all affect the visual details

but not affect the overall logic of the program.

4. Division of labor:
   One person will handle constructing the scene graph, along with the necessary camera, integration, and user interface code code.
   One person will handle collision detection, and the physics of hitting a ball.
   One person will handle the fine grained details - texture mapping, bump mapping, and shadow mapping.

   We will start ASAP, with the scene graph person getting all the skeleton code set up and the basic pool table set up.