**Physics and Config Final Project Proposal**

I was thinking of building off of the midterm project and make a sort of shooting gallery where the user can navigate the scene with a first-person fly-camera, and launch projectiles from the cannon or from themselves (like a gun of some kind) at various targets. All of the targets will have collision detection, but they will behave differently. For example, one kind of target could disappear and re-appear (similar to the harder marks from the midterm), while others could be tied by rod or rope constraints to a point and swing around when they are shot. The cannon will shoot projectiles similar to the bullet or cannon ball from the midterm, while the player will shoot projectiles similar to the laser (almost straight, not affected or affected very little by gravity). Projectiles will have a timed lifespan while targets do not.

Technical Details:

As per the requirements, the targets and projectiles will inherit from cParticle so they may also inherit sphere-sphere collision handling. Projectiles fired from the cannon or player will also have sphere-plane collision, where there will be a bounding area, or a “play area”, represented by simple meshes like the previous assignment, but larger. The targets will not have sphere-plane collision, so if a target is placed near a plane, or even in the plane, it should not collide and react. The projectiles and targets will be represented by simple sphere meshes, and have a radius. As mentioned above, some of the targets will have different types of constraints, so we may observe their behaviour when hit by projectiles.

**Targets:**

Basic Target: Simple sphere mesh that represents a particle and has the basic sphere-sphere collision detection. Moves to a new position when hit, is static otherwise. Shots bounce off of them.

Basic Target (moveable): Basic target but it just bounces around the scene, basically the spheres from Project 3, with no lifespan.

Hanging Chain: Small pivot particle at the top, with several linked “rope” constraints before a larger target sphere particle at the bottom. The pivots in the links would need constraints, but maybe no collision detection, so that only when the target is hit, it causes the entire “chain” to move. Alternatively, the pivots and the “chain” itself could have collision detection. The pivot at the top would be static. The number of links and distance between them could/should be configurable. (Or, the number of links is configurable and within the program, calculates the maximum distance between them to reach the “target” position at the end of the chain).

Connected Targets: Targets connected by a rod/multiple rod constraints, so the whole entity is affected when hit. 2 versions I’m thinking of, 1 is as above, the whole entity can just be pushed around the scene. The second, when I was thinking about how rod constraints could be used in games, I thought of old-school dungeon traps, like swinging axes and the like. So, the second version of connected targets would have some kind of pivot point, and the target would be constrained to only move around one axis. So one might only be able to move around the xz-plane with the y-value static, combined with the rod constraint to make a “spinning blade”.

Spring Target: As suggested for the revision, we haven’t done this in class but I was reading a bit of the section about it in the textbook. Essentially, the target is locked or mounted to a point in space, and it can be bounced around but it always returns to the same position.

Barrels (for the buoyancy): if/when I get to the buoyancy part, the barrels will be what I use to show it. For the target part of the game, the barrels will explode when hit, and both it and the shot will disappear. After a short period of time, the barrel will re-spawn. They will likely be single sphere meshes and particles, or a small group of particles made with constraints (as we did with the coolTetraThing).

For the configuration requirements, the different projectiles/shots will be externally configurable. For the projectiles, similar to the midterm, they will have their particle attributes configurable, as well as their radius, lifespan, and any other attributes I may need to add. I will also have the different targets be configurable for their radius/size, restitution for the collision detection, and for the rod/rope constraints, the maximum distance will also be configurable. Also, per the requirements, in release mode, I will have a post-build script to create a deployment package and make an installer script to create a single installation exe for the program.

For harder/bonus marks, I would like to try to implement a buoyancy system, and to show it working, attempt to remake the cannon game from Zelda: Wind Waker, and have some “barrels” in the “water” that the cannon can shoot at.

