Algorithm Process Documentation

General:

In order to implement the various movement behaviors, we made each behavior a state. Each of these states then takes a bunch of Vector3’s generated from a set of movement algorithms (henceforth called behaviors) in order to create a final Vector3 which is the final velocity that the object would be set to.

Collision Avoidance:

This is a major behavior that is used by almost all of the various movement states. This algorithm avoids obstacles by projecting a raycast from the object’s rigidbody. We then

**Etc. please detail stephen**

Wandering:

This movement state is generated from a combination of a wandering behavior and collision avoidance. If the collision avoidance behavior returns a zero vector (meaning there are no collisions to avoid), we simply take the vector returned by the wandering behavior. If collision avoidance is not a zero vector, the target velocity is 0.4\*avoidCollision + 0.6\*wander.

The wandering behavior works as follows. After a period of time (whose length varies based off of a range), generate a vector by slightly tweaking the object’s orientation by adding a random value. Return the newly generated vector adjusted by speed. Otherwise, if the period of time has not elapsed yet, just return the object’s current velocity vector.

Reach Goal:

This movement state is generated from a combination of a kinematic arrive behavior and collision avoidance. If the collision avoidance behavior returns a zero vector (meaning there are no collisions to avoid), we simply take the vector returned by the arrive behavior. If collision avoidance is not a zero vector, the target velocity is 0.4\*avoidCollision + 0.6\*kinematic arrive.

Flocking Wandering:

This movement state is generated by a combination of velocity matching, cohesion, and separation, along with avoid collisions.

Fleeing:

Formations:

This movement state is generated by having the game objects do the reach goal behavior towards specific positions in a formation.