MyVector Class:

Explanation: Class containing three floats in order to act as a replacement for the vec3 class used previously.

Functions:

Direction : returns the vector normalized.

Magnitude: returns the length of the vector.

dotProduct and crossProduct:: returns the given product given two vectors.

Particle Class:

Explanation: Class containing attributes in order to simulate physics, such as position, size, mass, velocity, acceleration and forces acting on it.

Functions:

Constructor (Particle(Model\* model)): This is the one mainly used for the MP, being passed only the model so that it can internally generate all the values necessary to act as a spark, as designated by the requirements of Phase 1.

Update: Updates the particles position based on its velocity, acceleration and forces. Also deducts from its lifespan whenever a second (assuming 60FPS) has passed.

Destroy: returns a Boolean depending on the particle’s lifespan

Draw: Draws the particle if it isn’t destroyed yet

AddForce: Adds the force being passed to the accumulatedForce of the particle.

ResetForce: Resets acceleration and forces accumulated to zero.

getPos: Used to easily gain access to the whole position vector.

CleanUp: used for cleaning up the VAOs and VBOs of the model used.

ForceGenerator Classes:

Explanation: Used for simulating the forces being applied to the particles. The two used here are for gravity and drag specifically.

Functions:

UpdateForce: Used to apply the force calculated by the generator. To be overwritten by the class’s children.

UpdateForce (Gravity): Applies the force of gravity to the particle assuming it has a valid mass (above 0).

UpdateForce (Drag): Applies the drag to the particle via damping.

ForceRegistry Class:

Explanation: Contains a list of force generators and particles paired together so that is the forces can be applied in an organized manner.

Functions:

Add: Adds a particle and generator pair to the list.

Remove: Removes a particle and generator pair from the list given that one exists that matches the ones passed to this function.

Clear: Empties the list (but doesn’t destroy the contents).

UpdateForces: Applies the forces to their paired particle.

PhysicsWorld Class:

Explanation: Used to simplify the process of updating and rendering multiple particles at the same time.

Functions:

Constructor (PhysicsWorld(Model \* m, DragForceGenerator \* d)): This constructor saves references to the sphere model and drag force generator for generation of particles within the world class to replace the ones that reach the end of their lifespan.

AddParticle: Adds a particle to the list and pairs it with a GravityForceGenerator for its ForceRegistry

Draw: Calls the draw function of all particles inside the world.

Update: Calls the particles’ update funcion after removing and replacing the ones which reach the end of their lifespan and applying their paired forces in the PhysicsWorld’s ForceRegistry.

UpdateForces: Removes and Replaces particles which reach the end of their lifespan.