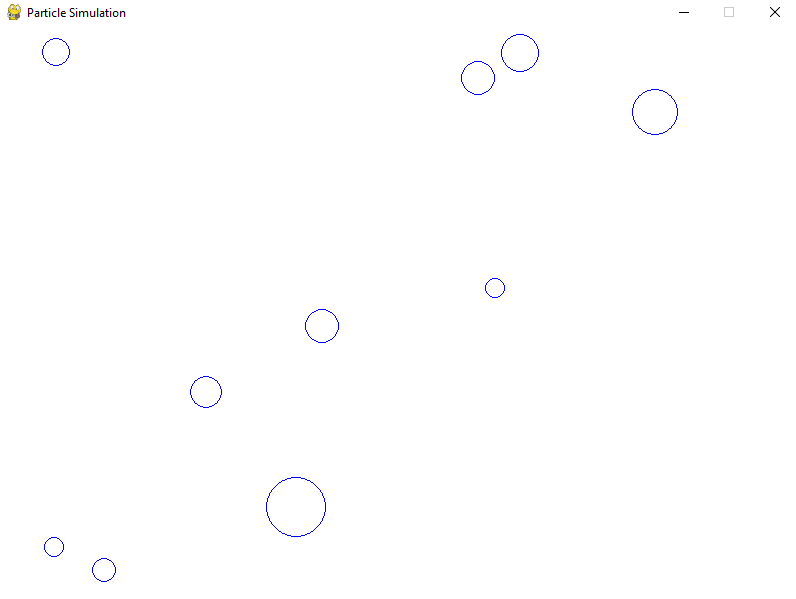
Particle Physics Lab

The purpose of this lab is to help you learn more about vectors and motion in pygame.

Part 1 – Creating our Basic Physics:

A screen shot of our finished part 1:



Create a directory called Particle-Simulation. This will be our working directory.

Create three new python modules: main.py, settings.py, and sprites.py.

Part 1A: The settings.py module.

This module will house all our constants for our world and its physical properties.

Use good commenting technique to label and describe each section of your settings.

At the top of your settings module, import pygame so we can use the pygame.math.Vector2 class.

Screen Settings:

Make a setting for width and height of the screen and the frame rate. Be sure to use ALL CAPS because that indicates what is a setting constant.

Particle Settings:

Make a constant for the number of particles in the simulation. Setting this to one during the debugging process is helpful for focusing on a single particle's behavior. Make constants for the minimum and maximum radius size of each particle. (The specific particle radius will be randomly chosen from within these bounds)

Make a constant for minimum and maximum initial speed of the particle. (The specific speed for each particle will also be chosen randomly.)

Part 1B: The sprites.py module.

At the top of this module, import pygame and import all settings using the from \_\_\_\_ import \* technique.

Create the Particle class which subclasses pygame.sprite.Sprite.

The signature for the init method is as follows:

def \_\_init\_\_(self, app, x, y, radius, angle, speed):

app and radius are used to set self.app and self.radius.

[We need to keep a reference self.app so we can access properties from out Application class we will create in the main.py module in Part 1C.]

The x, y values will be used to create self.pos a vector created with the pygame.math.Vector2 class.

To create the particles velocity vector self.vel, first make a vector in the direction . Using pygame.math.Vector2. Then, multiply this vector by speed. Finally rotate the vector to the proper angle using the rotate\_ip(angle) method.

The last bit of code in our init is to add our particle to the sprite group we will create in our Application class in main.py (in part 1C). We will name this group particles. So add the particle to this group through the self.app reference.