For this assignment we simulated a 2D scene at a park. We have birds flying in the sky using flocking behavior implemented through cellular automata, various sports balls bouncing on the ground using simulated rules of physics, and a water fountain made using particle simulation.

Jack created the balls bouncing around, and created a Ball class that uses projectile motion equations with gravity and Euler’s Method to accurately represent the balls as they bounce off the ground and walls of the sketch. The balls also incorporate air resistance (drag) in their calculations, and there is a coefficient of restitution implemented that slows down each ball’s motion slightly when it hits the floor or walls. This coefficient and the air resistance slow the balls down over time as they bounce around, as if you were actually to bounce a ball. Jack created basketballs, soccer balls, and baseballs. The balls use different radii, masses, and coefficients of restitution to more accurately depict the type of ball and affect the drag and bouncing calculations. For example, the basketball has a higher coefficient of restitution than the baseball, so it loses less energy when it hits the floor than the baseball does, causing it to bounce higher and for a longer time. The Ball objects do not interact with each other. The only methods created for this class are the initializer, which sets the variables and randomly generates the position and velocity of the ball, and the update method, which calculates the new position of the object using Euler’s method and draws it every frame. The update method checks to see if the ball has hit a wall, and reverses the direction and implements the coefficient of restitution if it has.

Braxton created the fountain spouting from the lake using a particle simulation class. The structure will be set up as follows: Particle System class followed by a generic Particle class which contains methods to add the forces to the individual particle as well as adding a new particle itself. The particle class itself is complete with motion; random movement of the individual particles. The fountain features two separate particle systems which in turn show a main stream in the middle and smaller streams on the side, like one you would find in a public park. The separate streams in the fountain each have their own velocity. To achieve this, I utilized two nested ArrayLists, one for the particles, and one for the particle system. The water appears to fall back into the lake, which was completed by editing the particle’s opacity based on their position.

Jade used a rule-based system (cellular automata) to simulate flocking behavior in birds in the sky. A single .png file represents many identical birds. There are two classes: Bird and Flock. Class Bird imports the .png file and sets PVectors for the three rules: separation, alignment, and cohesion and applies them to all birds. Separation is weighted more heavily than alignment and cohesion since it is most important for the objects to not collide. It has a function for wrapping around the canvas so birds are always visible in the sky and don’t get too close to the ground. Class Flock defines an ArrayList that holds all Bird objects. The class has a function “addBird” that is called in the main program in a “for” loop to add an easily modifiable amount of birds to the flock. addBird is also called in the mousePressed() function so the user can click on the canvas and add a new bird where they clicked.

One background image of a park scene was shared among the members for everyone to appropriately scale and locate their objects with relation to one another. As each member finished their part, they added their objects to the scene.