**What needs to be programmed**

**Pedestrian Modeling:**

* Has a speed (either constant or static)
* only moves at right angles
* is a circle with .5m in diameter, so it has an area
* has x,y coordinates, initial position

**Vehicle Modeling:**

* has a steady state speed
* has a width (collision zone)
* has x,y position & initial position
* has an acceleration

**Check for Collisions Function:**

* run every 100 ms because that’s when we receive input from sensor
* inputs: pedestrian location (x&y) \*there’s an accuracy on this, pedestrian velocity, vehicle position, vehicle velocity
* returns 0 if no collision possibility, otherwise the minimum deceleration value needed to avoid collision (need to do some number crunching or research on this)

**3 Different kind of Scenario Functions:**

\*Note the scenario functions are going to have to take the responsibility of the Brake by actuator system

1. Moving then stopped
   * Takes in an initial position, end position, and initial speed for the pedestrian
   * Creates pedestrian object based on parameters
   * Creates vehicle object
   * Will update the position of the vehicle and pedestrian based on their velocity and a sensor update time of 100ms, until both the pedestrian and vehicle are stopped and collision is avoided
   * Runs check collision function every 100ms.
   * Return the time lost due to avoidance maneuver
2. Static then moving
   * Takes an initial position, delay time, and final speed for pedestrian
   * Creates pedestrian object based on parameters
   * Creates vehicle object
   * Updates position of vehicle and pedestrian based on their velocity and a sensor update time of 100ms, starting the pedestrians movement after the time delay; not quite sure how to determine stop point of the scenario
   * Runs check collision function every 100ms
   * Return the lost time due to avoidance maneuver
3. Static
   * Takes an initial position
   * Creates a pedestrian object based on parameters
   * Creates vehicle object
   * Updates position of vehicle and pedestrian based on the their velocity and a sensor update time of 100ms, until the vehicle successfully stops
   * Runs check collision function every 100ms
   * Return lost time due to avoidance maneuver

**Time lost calculation function**

* Probably give some time system was activated and at what speed, the distance traveled, the stopping time at which the system was turned off
* Return time lost due to avoidance maneuver