CS 1181 Project 3: Data Science (Package Delivery Simulation)

PURPOSE: This assignment explores simulation of a real-world situation which leads to meaningful actions to solve a problem. A very common use of computers is to run simulations to answer practical questions with no easy closed-form mathematical solution. A simulation is a program designed to model a real system of interest. By setting parameters and selecting an appropriate model, results are produced that model the behavior of the real system. If the model is a valid representation of the real system, then the simulation can even predict the future performance of the real system.

PROBLEM: For this project, you will be simulating a package delivery system involving drones and trucks. A delivery company is interested in decreasing the time it takes them to deliver packages. Currently, the company only uses trucks to deliver packages, but a railroad crossing has been causing costly delays for their trucks. Thus, the company recently purchased a large number of drones capable of delivering packages without the delays of the crossing. Now, the company is looking to optimize their delivery system but is unsure how many packages should be delivered by the new drones. Your goal is to determine what percent of packages should be delivered via drones (vs trucks) to result in the shortest delivery time of all packages.

CONSTANTS: The company has provided you with the following information:

- Each day the company delivers 1,500 total packages.
- The distance to the train crossing is 3,000 units.
- The distance from the train crossing is 27,000 units. (The total distance is 30,000 units.)
- Drones and trucks will follow the exact same path (same distance).
- The daily train schedule (see below for more information)
- Each drone travels at a speed of 500 units/min.
- Each drone has a max load capacity of one package.
- One drone may take off every three minutes.
- Each truck travels at a speed of 30 units/min.
- Each truck has a max load capacity of ten packages.
- One truck may depart every 15 minutes.
- The percent of packages that will be delivered via drone (PERCENT_BY_DRONE)

Train: The train schedule will be provided to you in a text file called "train_schedule.txt". This file contains a list of start times and durations that the train will be blocking the track in minutes. For example, the first line of the file reads "94 77". This means that a train begins blocking the crossing at 94 minutes and continues to block the crossing for 77 minutes. To find the time that the train is no longer blocking the crossing, you can add the duration to the start time (i.e. 94 + 77 = 171).

Number of Drones and Trucks: One of the first questions that must be answered is how many drones and how many trucks are needed. You should have a variable called PERCENT_BY_DRONE that indicates what percent of the packages will be delivered via drone. Using this value, you need to determine how many drones and how many trucks are needed. Remember that each drone can carry one package and each truck can carry ten packages. Therefore, if you need to deliver 100 packages, 50% by drone and 50% by truck, then you would need 50 drones (100 * .5 = 50) and five trucks ((100 * .5) / 10 = 5). If the number of packages to be delivered by truck does not divide by ten evenly, you will need to round up as an additional truck will be necessary.

Next, it is important to note that there are two distinct problems that need to be addressed: the total time it takes for all the drones to deliver their packages and the total time it takes for all the trucks to finish delivering their packages. Since the drones do not have any obstacles, simple algebra can be used to determine the total time. However, because the trucks may be interrupted by the train, discrete event simulation must be used to simulate each of the trucks' journey.

Drones: You need to calculate the time it takes for one drone to complete its trip and the total time it takes for all drones to complete their trip. The first drone should take off at time 0.0. Keep in mind that multiple drones may be flying at the same time, so the total time is *not* the sum of each drone's trip time. Again, since the drones do not have any obstacles, event simulation is not necessary to determine the total time it takes for all drones to deliver their packages.

DISCRETE EVENT SIMULATION: To simulate the truck portion of this project, discrete event simulation will be used. This means time is managed by using a simulation clock that is advanced whenever an interesting event occurs. For example, if the simulation time is 0.0 minutes and a truck begins its route at a time 10.0 minutes, we mark the passage of time by advancing the simulation clock to 10.0 minutes (current time = 0.0 + time of truck starting event = 10.0). If a truck reaches the crossing 35.0 minutes later, we advance the simulation clock to 45.0 (10.0 + 35.0). Our assumption is no interesting activities occur between events (i.e. nothing significant happens in the intervals (0.0, 10.0) or (10.0, 35.0) so we can record the passage of time by moving the simulation clock forward to the time when the next important event occurs).

For this project there are several different events of interest. These consist of train events and truck events. Below is a brief description of the significant events a truck may experience. Keep in mind that all the trucks are traveling at the exact same speed on a one-lane road, so under no circumstances may one truck pass another.

- TRUCK_START: The first truck may start at time 0.0. Due to the required offset between truck departures, the second truck will start at 15.0. Since the truck speed and distance to the tracks have been provided to you, a simple calculation can be used to determine when the truck will arrive at the crossing.

- TRUCK_AT_CROSSING: Here two outcomes may occur. First, if there is no train blocking the crossing, the truck may cross immediately and continue its journey. The other possibility is that the truck must wait. There are two possible conditions that cause a truck to wait. If a train is blocking the crossing the truck must wait. Additionally, if there is no train, but there is a line of trucks waiting to cross, then the truck must wait. Therefore, it is possible that a truck may have to wait even if the train is already gone. Each truck must also take one minute to start up from a stop and cross. For example, if a train leaves the crossing at time 35.0, then the first truck waiting will cross at 36.0, followed by the next truck at 37.0, and so on until all the trucks have crossed.

- TRUCK_CROSS: The truck crosses the tracks. Again, since the truck speed and distance from the tracks have been provided to you, simple algebra can be used to determine when the truck will complete its delivery.

- TRUCK_END: Lastly, once the truck has crossed the crossing, the truck can end its journey.

In addition to keeping track of truck events, it will be useful to keep track of the significant train events. These include when a train begins to block the crossing and when it is no longer blocking the crossing.

Event Sorting: To process the events, they should be stored in a data structure that can allow them to be sorted by event time. Additionally, some sorting rules must be implemented in case multiple events occur at the same time. First, a train event should always take priority over a truck event. Second, a truck arriving at the crossing should be handled before a truck that is crossing at the same time. This will ensure that two trucks do not cross at the same time.

Real-Time Tracking: For each significant event, a description should be printed to the console for tracking. Event tracking should be printed in correct time order. Reminder: only the trucks and trains use discrete event simulation, and therefore, only truck and train events need to be tracked. Below is a list of every event type that should be printed to the console:

- Truck beginning its journey
- Truck crossing the crossing
- Truck waiting at the crossing (if necessary)
- Truck completing its delivery
- Train beginning to block the crossing
- Train no longer blocking the crossing.

Each event print statement must include the time that the event is occurring, the type of vehicle (truck or train), the id (if a truck), and a brief description of the event. For example, if the second truck is waiting at the crossing, "76.0: TRUCK #2 waits at crossing" could be printed to the console.

Special Considerations: Via proper implementation of the discrete event simulation and real-time event tracking, your solution should demonstrate that no truck is able to cross the crossing when a train is present. Additionally, it should be clear that your solution properly queues trucks that cannot cross and dequeues them when the train is gone. This includes ensuring that a new truck arriving to the crossing does not cut to the front of a pre-existing line of trucks.

Statistics: After every event has been processed for the truck simulation and the drone calculations have been performed, the following should be printed to the console.

- The number of drones and trucks for the simulation
- The full train schedule
- The total trip time for each truck ID (time from each truck departure to time it reaches the destination)
- The average trip time for all trucks
- The total time for all the trucks to deliver their packages
- The trip time for one drone (the same for every drone)
- The total time for the drones to deliver all their packages
- The total time for all packages to be delivered

Remember that the total times for both the drones and the trucks are not the sum of all drone trip times and truck times respectively. This is because multiple drones and trucks are traveling at the same time.

End Goal: Once your simulation is running properly and your statistics are being calculated correctly, you should run your simulation several times with different values for PERCENT_BY_DRONE. *In a comment at the top of your main file*, indicate what PERCENT_BY_DRONE you found produces the lowest total time for all packages to be delivered. Please round your value to the nearest percent.

<u>GRADING</u>

(75 pts) **Base Functionality**

[10] "train_schedule.txt" is correctly read and parsed
[5] The number of drones and trucks is calculated correctly given a certain PERCENT_BY_DRONE
[10] Drone calculations are correct (individual drone trip, total drone trip time)
[5] Data structure holding all events sorts them properly
[10] Real-time tracking information is printed to the console
[10] Truck calculations are correct (total trip time per truck, total and average trip time for all trucks)
[10] Trucks properly queue, do not cross the tracks when a train is present, and dequeue when the train is gone
[10] Truck and drone statistics are printed to the console
[5] Final best value for PERCENT_BY_DRONE is correct

(25 pts) **Style**

[10] Object-oriented principles such as encapsulation, inheritance, and polymorphism, are used appropriately
[10] The coding is ***meaningfully*** commented, and there is not commented out junk code
[5] Standard coding conventions are followed, including variable names and indentation

NOTE: There is no valid excuse for turning in a program which does not compile. A program will receive no credit if it does not compile and execute.