Computer Lab 7

Concepts

- "More complicated" detection algorithm
- Use of mediators
- `RandomNumber`
- Supporting multiple `RandomVariate` objects with the same RandomNumber object

Description

You are to create a model for a "constant-rate" sensor for which the time of detection is an exponentially distributed amount of time after a target has entered it range. The test scenario is basically the same as in Lab 6. You will need to write three classes: the `ConstantRateSensor` class, the `ConstantRateMediator` class, and an execution class.

ConstantRateSensor Class

The `ConstantRateSensor` class is a subclass of `BasicSensor`. It should have an instance variable of type `double` called `meanDetectionTime`, with an associated getter and setter methods. The constructor should have signature `(Mover, double, double)`. The first two arguments should be passed to the superclass constructor; the third is stored as an instance variable, the mean time to detection. Override `toString()` to add the mean detection time to the end of the String.

ConstatRateMediator Class

Write the `ConstantRateMediator` as a `SimEntityBase` that implements the `Mediator` interface. To fulfill the "contract" for Mediator your class must implement the following seven methods:

```
public Mover getTarget();
public Sensor getSensor();
public double getDetectionTime();
public double getUndetectionTime();
public void disconnect();
public void connect();
public boolean isTargetInRange();
```

For the `MediatorFactory` to correctly create instances of the `ConstantRateMediator`, it must have a constructor with signature `(Sensor, Mover)`. Define instance variables of type `Mover` and `ConstantRateSensor` and set them in the constructor.[1] Define instance variables of type `double` called `detectionTime` and `undetectionTime`. You can now implement the first four methods in the `Mediator` interface as getters for these four instance variables.

The `connect()` method simply adds the `ConstantRateMediator` as a `SimEventListener` to the sensor and the target and adds the sensor to the `ConstantRateMediator` as a `SimEventListener`. The `disconnect()` method reverses these three registrations.[2]

---

1. You will have to cast the Sensor to a ConstantRateSensor.
2. Use the removeSimEventListener() method.

The `isTargetInRange()` method returns a boolean indicating whether the target is within the sensor's range. I will leave it to you to figure out how to implement this.

When you have implemented all these methods you will be able to try for a "clean" compile, and you should do so. To get the `ConstantRateMediator` fully functional, however, requires a bit more work.

To generate the exponential detection times, declare a `RandomVariate` instance variable called `detectionTimeGenerator`. There will be a single static variable of type `RandomNumber` that will be used to generate the uniform variates for `detectionTimeGenerator`. Declare and instantiate the `RanndomNumber` variable as follows:

```
private static RandomNumber rng = RandomFactory.getRandomNumber();
```

Define a static method called `setSeed(long)` that invokes `setSeed(long)` on `rng`, passing it the parameter. Now, back in the constructor, get the `RandomVariate` instance as follows:

```
detectionTimeGenerator =
	RandomFactory.getRandomVariate(
		"Exponential",
		new Object[] {new Double(mySensor.getMeanDetectionTime())},
		rng);
```

The way this works is that each instance of `ConstantRateMediator` will have its own `RandomVariate` instance to generate the detection times, but just a single instance of `RandomNumber` will be generating the uniform random variates for all `RandomVariates`. This means that just one seed must be managed and that the detection times will be different across all `ConstantRateMediator`s.

You now need to write two event methods, `doEnterRange(RangeParameters)` and `doExitRange(RangeParameters)`. The `RangeParameters` class holds a reference to a sensor/target pair, which are accessed by the getters `getSensor()` and `getTarget()`, respectively.

```
public void doEnterRange(RangeParameters rp) {
	if (rp.getSensor() == mySensor && rp.getTarget() == myTarget) {
		this.connect();
		interrupt("Detection", myTarget);
		waitDelay("Detection", detectionTimeGenerator.generate(), myTarget);
	 }
}

public void doExitRange(RangeParameters rp) {
	if (rp.getSensor() == getSensor() && rp.getTarget() == getTarget()) {
		this.disconnect();
		interrupt("Detection", myTarget);
		waitDelay("Undetection", 0.0, myTarget);
	}
}
```

Mover Manager

You can use the `PathMoverManager` from the last lab without modification.

Execution Class

Your execution class (`TestConstantRateSensor`) is nearly identical to that of the last lab. The data for the Movers, Sensor ranges, and waypoints is the same. The `Sensor` objects, however, will be instances of `ConstantRateSensor`, and the `MediatorFactory` must have `oa3302.ConstantRangeMediator` added as a mediator type for `simkit.BasicMover` and `oa3302.Con-`

`stantRateSensor`.[3] Use the parameters in Table 1, which identical to those in the previous lab for the entities. In this case, the two Movers simply will switch places. You still can, however, use the PathMover-

Table 1. PathMoverManager Properties

Manager	*Mover Type*	*Initial Location*	*Speed*	*Sensor Range*	*Mean time to Detection*	*Path*
PathMover-Manager 0	Basic-Mover	(0.0, 0.0)	10	10	1.0	(100, 100)
PathMover-Manager 1	Basic-Mover	(100, 100)	15	20	1.5	(0, 0)

Manager from Lab 6.

Output

```
simkit.smd.Referee@b2cec993
Sensors:
oa3302.ConstantRateSensor.3 10.0 locationDelegate: simkit.smd.BasicMover.1 (0.00, 0.00) 0.0 [1.0]
oa3302.ConstantRateSensor.4 20.0 locationDelegate: simkit.smd.BasicMover.2 (100.00, 100.00) 0.0
[1.5]
Targets:
simkit.smd.BasicMover.1 (0.00, 0.00) 0.0
simkit.smd.BasicMover.2 (100.00, 100.00) 0.0
 ** Event List -- Starting Simulation **
0.000  StartMove {simkit.smd.BasicMover.1 (0.00, 0.00) 10.0}
0.000  StartMove {simkit.smd.BasicMover.2 (100.00, 100.00) 15.0}
 ** End  of Event List -- Starting Simulation **
Time: 0.000 Current Event: StartMove {simkit.smd.BasicMover.1 (0.00, 0.00) 10.0}[1]
 ** Event List --  **
0.000  StartMove {simkit.smd.BasicMover.2 (100.00, 100.00) 15.0}
0.000  StartMove {oa3302.ConstantRateSensor.3 10.0 locationDelegate: simkit.smd.BasicMover.1
(0.00, 0.00) 10.0 [1.0]}
14.142 EndMove {simkit.smd.BasicMover.1 (0.00, 0.00) 10.0}
 ** End  of Event List --  **
Time: 0.000 Current Event: StartMove {simkit.smd.BasicMover.2 (100.00, 100.00) 15.0}[2]
 ** Event List --  **
0.000  StartMove {oa3302.ConstantRateSensor.3 10.0 locationDelegate: simkit.smd.BasicMover.1
(0.00, 0.00) 10.0 [1.0]}
0.000  StartMove {oa3302.ConstantRateSensor.4 20.0 locationDelegate: simkit.smd.BasicMover.2
(100.00, 100.00) 15.0 [1.5]}
5.257  EnterRange {oa3302.ConstantRateSensor.3, simkit.smd.BasicMover.2, exitRangeTime=0.800}
9.428  EndMove {simkit.smd.BasicMover.2 (100.00, 100.00) 15.0}
14.142 EndMove {simkit.smd.BasicMover.1 (0.00, 0.00) 10.0}
 ** End  of Event List --  **
Time: 0.000 Current Event: StartMove {oa3302.ConstantRateSensor.3 10.0 locationDelegate: sim-
kit.smd.BasicMover.1 (0.00, 0.00) 10.0 [1.0]}[1]
 ** Event List --  **
0.000  StartMove {oa3302.ConstantRateSensor.4 20.0 locationDelegate: simkit.smd.BasicMover.2
(100.00, 100.00) 15.0 [1.5]}
5.257  EnterRange {oa3302.ConstantRateSensor.3, simkit.smd.BasicMover.2, exitRangeTime=0.800}
9.428  EndMove {simkit.smd.BasicMover.2 (100.00, 100.00) 15.0}
14.142 EndMove {simkit.smd.BasicMover.1 (0.00, 0.00) 10.0}
 ** End  of Event List --  **
Time: 0.000 Current Event: StartMove {oa3302.ConstantRateSensor.4 20.0 locationDelegate: sim-
```

3. Simply substitute the class names here (oa3302.ConstantRateSensor, oa3302.ConstantRateMediator)for the ones you used in Lab 6 in MediatorFactory.addMediatorType().

```
kit.smd.BasicMover.2 (100.00, 100.00) 15.0 [1.5]}[2]
 ** Event List --  **
4.857  EnterRange {oa3302.ConstantRateSensor.4, simkit.smd.BasicMover.1, exitRangeTime=1.600}
5.257  EnterRange {oa3302.ConstantRateSensor.3, simkit.smd.BasicMover.2, exitRangeTime=0.800}
9.428  EndMove {simkit.smd.BasicMover.2 (100.00, 100.00) 15.0}
14.142 EndMove {simkit.smd.BasicMover.1 (0.00, 0.00) 10.0}
 ** End  of Event List --  **
Time: 4.857 Current Event: EnterRange {oa3302.ConstantRateSensor.4, simkit.smd.BasicMover.1, exitRangeTime=1.600}
[1]
 ** Event List --  **
5.061  Detection {simkit.smd.BasicMover.1 (34.34, 34.34) 10.0}
5.257  EnterRange {oa3302.ConstantRateSensor.3, simkit.smd.BasicMover.2, exitRangeTime=0.800}
6.457  ExitRange {oa3302.ConstantRateSensor.4, simkit.smd.BasicMover.1, exitRangeTime=1.600}
9.428  EndMove {simkit.smd.BasicMover.2 (48.49, 48.49) 15.0}
14.142 EndMove {simkit.smd.BasicMover.1 (34.34, 34.34) 10.0}
 ** End  of Event List --  **
Time: 5.061 Current Event: Detection {simkit.smd.BasicMover.1 (35.78, 35.78) 10.0}[1]
 ** Event List --  **
5.257  EnterRange {oa3302.ConstantRateSensor.3, simkit.smd.BasicMover.2, exitRangeTime=0.800}
6.457  ExitRange {oa3302.ConstantRateSensor.4, simkit.smd.BasicMover.1, exitRangeTime=1.600}
9.428  EndMove {simkit.smd.BasicMover.2 (46.32, 46.32) 15.0}
14.142 EndMove {simkit.smd.BasicMover.1 (35.78, 35.78) 10.0}
 ** End  of Event List --  **
Time: 5.257 Current Event: EnterRange {oa3302.ConstantRateSensor.3, simkit.smd.BasicMover.2, exitRangeTime=0.800}
[2]
 ** Event List --  **
5.615  Detection {simkit.smd.BasicMover.2 (44.24, 44.24) 15.0}
6.057  ExitRange {oa3302.ConstantRateSensor.3, simkit.smd.BasicMover.2, exitRangeTime=0.800}
6.457  ExitRange {oa3302.ConstantRateSensor.4, simkit.smd.BasicMover.1, exitRangeTime=1.600}
9.428  EndMove {simkit.smd.BasicMover.2 (44.24, 44.24) 15.0}
14.142 EndMove {simkit.smd.BasicMover.1 (37.17, 37.17) 10.0}
 ** End  of Event List --  **
Time: 5.615 Current Event: Detection {simkit.smd.BasicMover.2 (40.44, 40.44) 15.0}[2]
 ** Event List --  **
6.057  ExitRange {oa3302.ConstantRateSensor.3, simkit.smd.BasicMover.2, exitRangeTime=0.800}
6.457  ExitRange {oa3302.ConstantRateSensor.4, simkit.smd.BasicMover.1, exitRangeTime=1.600}
9.428  EndMove {simkit.smd.BasicMover.2 (40.44, 40.44) 15.0}
14.142 EndMove {simkit.smd.BasicMover.1 (39.71, 39.71) 10.0}
 ** End  of Event List --  **
Time: 6.057 Current Event: ExitRange {oa3302.ConstantRateSensor.3, simkit.smd.BasicMover.2, exitRangeTime=0.800}
[1]
 ** Event List --  **
6.057  Undetection {simkit.smd.BasicMover.2 (35.76, 35.76) 15.0}
6.457  ExitRange {oa3302.ConstantRateSensor.4, simkit.smd.BasicMover.1, exitRangeTime=1.600}
9.428  EndMove {simkit.smd.BasicMover.2 (35.76, 35.76) 15
.0}
14.142 EndMove {simkit.smd.BasicMover.1 (42.83, 42.83) 10.0}
 ** End  of Event List --  **
Time: 6.057 Current Event: Undetection {simkit.smd.BasicMover.2 (35.76, 35.76) 15.0}[1]
 ** Event List --  **
6.457  ExitRange {oa3302.ConstantRateSensor.4, simkit.smd.BasicMover.1, exitRangeTime=1.600}
9.428  EndMove {simkit.smd.BasicMover.2 (35.76, 35.76) 15.0}
14.142 EndMove {simkit.smd.BasicMover.1 (42.83, 42.83) 10.0}
 ** End  of Event List --  **
Time: 6.457 Current Event: ExitRange {oa3302.ConstantRateSensor.4, simkit.smd.BasicMover.1, exitRangeTime=1.600}
[2]
 ** Event List --  **
6.457  Undetection {simkit.smd.BasicMover.1 (45.66, 45.66) 10.0}
9.428  EndMove {simkit.smd.BasicMover.2 (31.51, 31.51) 15.0}
14.142 EndMove {simkit.smd.BasicMover.1 (45.66, 45.66) 10.0}
 ** End  of Event List --  **
Time: 6.457 Current Event: Undetection {simkit.smd.BasicMover.1 (45.66, 45.66) 10.0}[2]
 ** Event List --  **
9.428  EndMove {simkit.smd.BasicMover.2 (31.51, 31.51) 15.0}
14.142 EndMove {simkit.smd.BasicMover.1 (45.66, 45.66) 10.0}
```

```
 ** End  of Event List --  **
Time: 9.428 Current Event: EndMove {simkit.smd.BasicMover.2 (0.00, 0.00) 15.0}[1]
 ** Event List --  **
9.428  EndMove {oa3302.ConstantRateSensor.4 20.0 locationDelegate: simkit.smd.BasicMover.2 (0.00,
0.00) 15.0 [1.5]}
14.142  EndMove {simkit.smd.BasicMover.1 (66.67, 66.67) 10.0}
 ** End  of Event List --  **
Time: 9.428 Current Event: EndMove {oa3302.ConstantRateSensor.4 20.0 locationDelegate: sim-
kit.smd.BasicMover.2 (0.00, 0.00) 15.0 [1.5]}[1]
 ** Event List --  **
14.142  EndMove {simkit.smd.BasicMover.1 (66.67, 66.67) 10.0}
 ** End  of Event List --  **
Time: 14.142 Current Event: EndMove {simkit.smd.BasicMover.1 (100.00, 100.00) 10.0}[2]
 ** Event List --  **
14.142  EndMove {oa3302.ConstantRateSensor.3 10.0 locationDelegate: simkit.smd.BasicMover.1
(100.00, 100.00) 10.0 [1.0]}
 ** End  of Event List --  **
Time: 14.142 Current Event: EndMove {oa3302.ConstantRateSensor.3 10.0 locationDelegate: sim-
kit.smd.BasicMover.1 (100.00, 100.00) 10.0 [1.0]}[2]
 ** Event List --  **
              << empty >>
 ** End  of Event List --  **
```

Deliverables

Turn in hard copies of your source code for just the classes you wrote for this lab. Turn in only the portion of the output showing the first few Detection events.