Closed Task

1. Visitor pattern -> i) new Visitor via inner class (Class getDestination)

ii) anonymous Visitor (method updateLog())

2. Lambda expressions -> getPlayerTickets()

Open Task

\*\* Referenced **priority queue** in order to implement algorithm of Dijkstra

1. Helper inner class to store data to meet our needs (class Score, class Node)

2. implements another interface (Comparable in class Node) in order to use its attributes

3. Functional visitor to match two distinct functions depending on the data type (method evaluateTicket())

4. Dijksstra algorithm

-- OUR AI CAN –

1. Calculate the distance between mrX and the detectives via Dijkstra algorithm

2. Evaluate the nodes in the map (Places that have variety of selections of transport is transferred)

3. Evaluate the ticket to use even if the destination is same

4. Act differently by evaluating the current situation (Some detectives are so close, or mrX has just revealed his location)

5. Calculate the danger score in order to see how much he is on the cusp