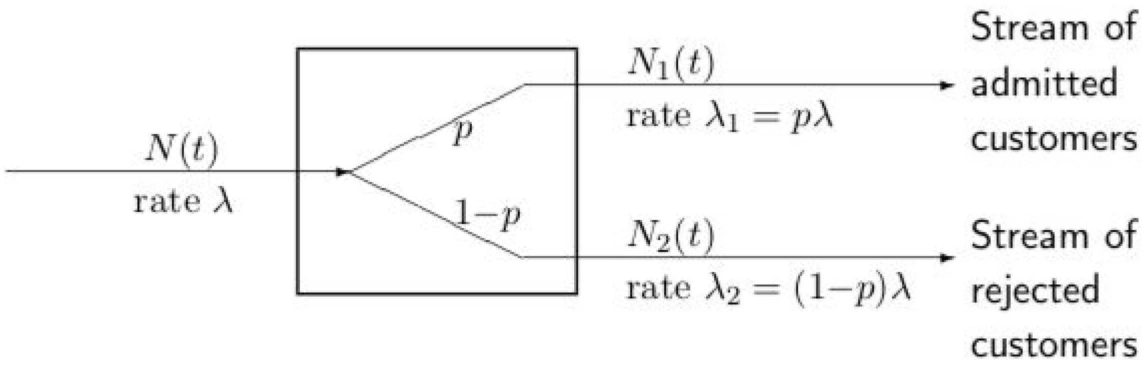
Mathematical Model

1) Thinning Poisson Process

Thinning of a Poisson Process




```
Step 1. Let U = Math.random()
Step 2. If P(accept) < U:
                return True
        Else:
                return False
```

-Firstly, we let the variable U to be a uniform random generator between 0 to 1.

-Secondly, if the acceptance probability that we assign is less than variable U, boolean value True will be returned. Otherwise, boolean value False will be returned.

2) Inverse Transform for customer arrival

-Let X be a random variable of customers arrival with is distributed exponentially with rate 0.5.

if X has the probability density function:

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0. \\ 0 & x < 0. \end{cases}$$

Note: $E[X] = 1/\lambda$.

-Assuming rate lambda = 0.5, $Inverse\ CDF\ = X = -2ln(1-U)$.

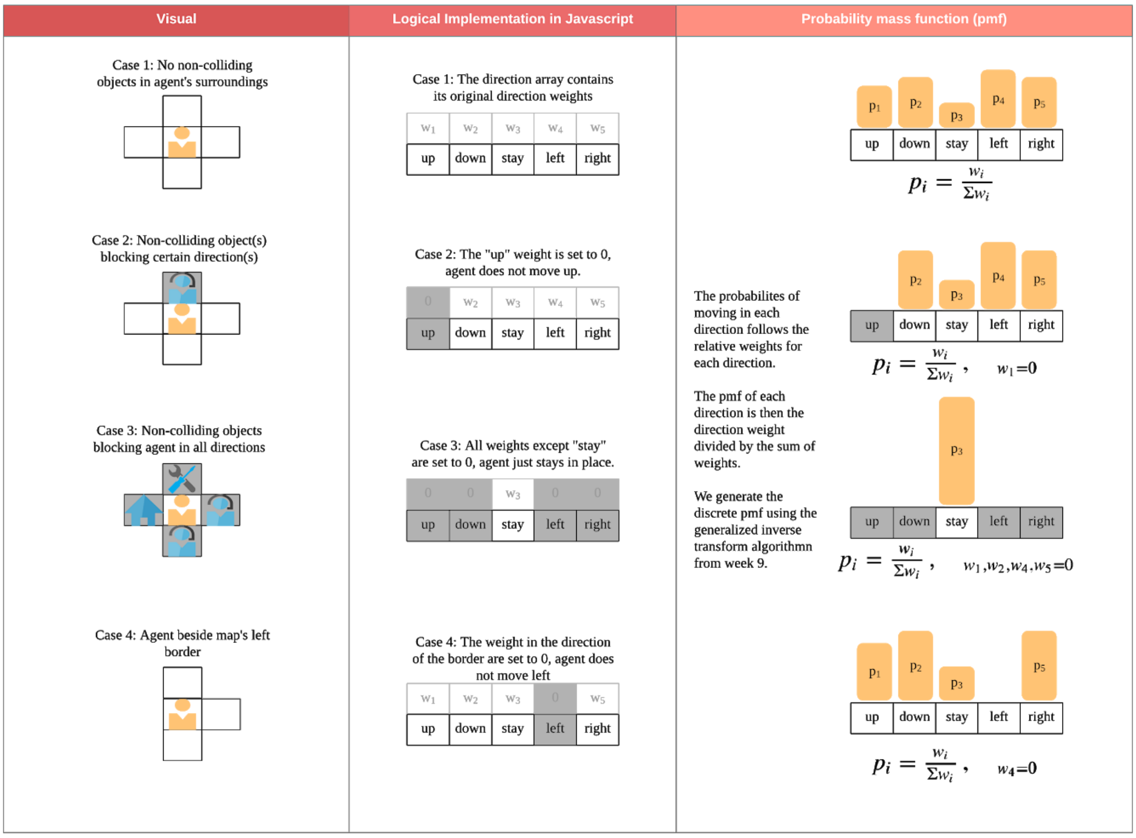
Step 1. Let U = Math.random()
Step 2. Let time_delta = -2 * Math.log(1 - U)

-Firstly, we let the variable U to be a uniform random generator between 0 to 1.

-Secondly, we let the variable time_delta =
$Inverse\ CDF\ of\ Exponential\ =\ X =\ -2ln(1-U)$

3) <u>Movement of the agents</u>

Non-Colliding Agent Movement Logic



The diagram above shows how the non-colliding agent moves using a probability mass function model. There are a total of 5 directions : up, down, stay, left and right. Each direction is assigned to some weight at the user's discretion. For example, agent direction's = [1, 1, 1, 1, 1]. The agent has a probability of $\frac{1}{5}$ of moving in each direction. We can also adjust the agent's behaviour such that they will generally move up or down at a specific area as well or such that they will avoid moving in that specific direction at all.

For Case 1, we simulate a situation such that there are no objects near the agent. Thus, the agent can move in the intended direction based on the given weight in the specific area.

For Case 2, we simulate a situation such that there is a non-colliding object at the top on the agent. Therefore, we assigned the ‘up’ weight to be 0. If the agent direction’s = [0, 1, 1, 1, 1], the agent has a probability of $\frac{1}{4}$ of moving in each direction except for the up direction.

For Case 3, we simulate a situation such that there are non-colliding objects at every side of the agent. Therefore, we assigned the weights to be 0 in those direction. If the agent direction’s = [0, 0, 1, 0, 0], the agent is just going to stay at the same spot.

For Case 4, we simulate a situation where the agent have reached a left border of the map of the simulation. Therefore, we assigned the ‘left’ weight to be 0. If the agent direction’s = [1, 1, 1, 0, 1], the agent has a probability of $\frac{1}{4}$ of moving in each direction except for the left direction.