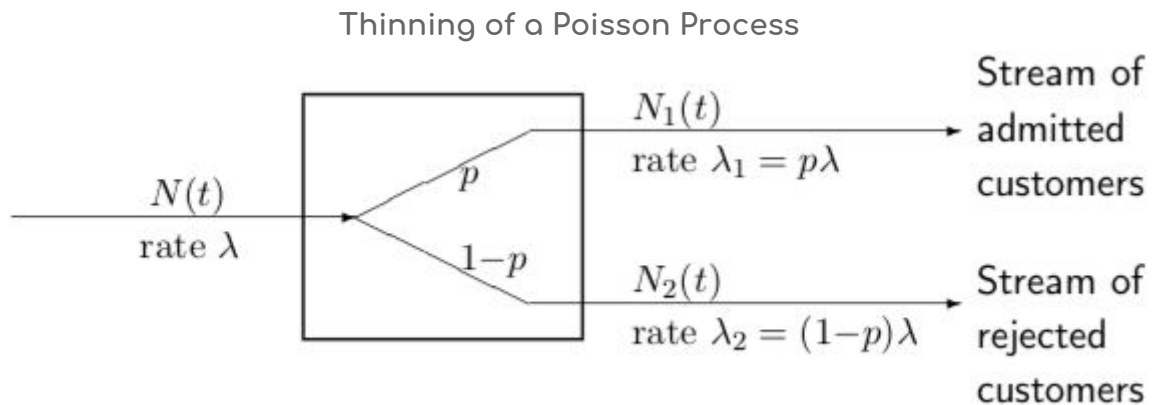


# Mathematical Model

## 1) Thinning Poisson Process



Step 1. Let  $U = \text{Math.random}()$

Step 2. If  $P(\text{accept}) < U$ :

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  $\lambda = 0.5$ ,  $\text{Inverse CDF} = X = -2\ln(1 - U)$ .

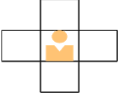

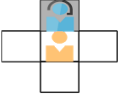
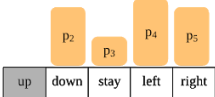



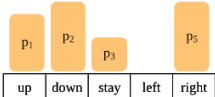
Step 1. Let  $U = \text{Math.random}()$   
 Step 2. Let  $\text{time\_delta} = -2 * \text{Math.log}(1 - U)$

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

-Secondly, we let the variable  $\text{time\_delta} =$   
*Inverse CDF of Exponential*  $= X = -2\ln(1 - U)$

### 3) Movement of the agents

#### Non-Colliding Agent Movement Logic

Visual	Logical Implementation in Javascript	Probability mass function (pmf)										
<p>Case 1: No non-colliding objects in agent's surroundings</p> 	<p>Case 1: The direction array contains its original direction weights</p> <table border="1" data-bbox="710 992 922 1050"> <tr> <td><math>w_1</math></td> <td><math>w_2</math></td> <td><math>w_3</math></td> <td><math>w_4</math></td> <td><math>w_5</math></td> </tr> <tr> <td>up</td> <td>down</td> <td>stay</td> <td>left</td> <td>right</td> </tr> </table>	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	up	down	stay	left	right	 $p_i = \frac{w_i}{\sum w_i}$
$w_1$	$w_2$	$w_3$	$w_4$	$w_5$								
up	down	stay	left	right								
<p>Case 2: Non-colliding object(s) blocking certain direction(s)</p> 	<p>Case 2: The "up" weight is set to 0, agent does not move up.</p> <table border="1" data-bbox="710 1167 922 1225"> <tr> <td>0</td> <td><math>w_2</math></td> <td><math>w_3</math></td> <td><math>w_4</math></td> <td><math>w_5</math></td> </tr> <tr> <td>up</td> <td>down</td> <td>stay</td> <td>left</td> <td>right</td> </tr> </table>	0	$w_2$	$w_3$	$w_4$	$w_5$	up	down	stay	left	right	<p>The probabilities of moving in each direction follows the relative weights for each direction.</p>  $p_i = \frac{w_i}{\sum w_i}, \quad w_1 = 0$
0	$w_2$	$w_3$	$w_4$	$w_5$								
up	down	stay	left	right								
<p>Case 3: Non-colliding objects blocking agent in all directions</p> 	<p>Case 3: All weights except "stay" are set to 0, agent just stays in place.</p> <table border="1" data-bbox="710 1357 922 1415"> <tr> <td>0</td> <td>0</td> <td><math>w_3</math></td> <td>0</td> <td>0</td> </tr> <tr> <td>up</td> <td>down</td> <td>stay</td> <td>left</td> <td>right</td> </tr> </table>	0	0	$w_3$	0	0	up	down	stay	left	right	<p>The pmf of each direction is then the direction weight divided by the sum of weights.</p>  <p>We generate the discrete pmf using the generalized inverse transform algorithm from week 9.</p> $p_i = \frac{w_i}{\sum w_i}, \quad w_1, w_2, w_4, w_5 = 0$
0	0	$w_3$	0	0								
up	down	stay	left	right								
<p>Case 4: Agent beside map's left border</p> 	<p>Case 4: The weight in the direction of the border are set to 0, agent does not move left</p> <table border="1" data-bbox="710 1559 922 1617"> <tr> <td><math>w_1</math></td> <td><math>w_2</math></td> <td><math>w_3</math></td> <td>0</td> <td><math>w_5</math></td> </tr> <tr> <td>up</td> <td>down</td> <td>stay</td> <td>left</td> <td>right</td> </tr> </table>	$w_1$	$w_2$	$w_3$	0	$w_5$	up	down	stay	left	right	 $p_i = \frac{w_i}{\sum w_i}, \quad w_4 = 0$
$w_1$	$w_2$	$w_3$	0	$w_5$								
up	down	stay	left	right								

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.