

Cairo University

Faculty of Engineering

Computer Engineering Department

First Year

**Data Structures and Algorithms**

**Supervised by:** Dr.Magda

**Done By:**

Ahmed Muhmmed AbdelHamid

Ahmed Mahmoud Ghareeb

Omer Muhammed Yacine

Ziyad Hassan Abdelmaksoud

**June 2020**

**Final Assessment Report**

|  |  |
| --- | --- |
| Team Name: Team\_9 | Number of Members: 4 |
| Email: ziyadhassan500@gmail.com | |

Section1: Ziyad Hassan AbdelMaqsoud,136

# **Assgin\_Cook\_Order**

**Member of class:** Restaurant

**Inputs:**

* Timestep: the current timestep of the system.
* 3 string: status bar strings.

**Returns:**

* Void.

**Called by:**

* Restaurant:: Interactive\_Mode().
* Restaurant:: StepByStep().
* Restaurant:: Silent().

**Calls:**

* Restaurant:: Assgin\_Normal\_Order().
* Restaurant:: Assgin\_Vegan\_Order().
* Restaurant:: Assgin\_VIP\_Order().
* Restaurant:: Assgin\_Urgent\_Order().

**Function Logic description:**

It is the main function that assigns all types of orders to the free cook by calling the other functions ( Assgin\_Normal\_Order(), …… ) and sends the remaining orders to the waiting list if it was not assigned to any cook.

# **Assgin\_Normal\_Order**

**Member of class:** Restaurant

**Inputs:**

* Ord: the order that will be assigned.
* Timestep: the current timestep of the system.
* NCooks\_Assgined: status bar strings if the order is assigned to a normal cook.
* VCooks\_Assgined: status bar strings if the order is assigned to a VIP cook.

**Returns:**

* Bool.

**Called by:**

* Restaurant:: Assgin\_Cook\_Order().

**Calls:**

None.

**Function Logic description:**

It assigns the orders of type Normal to the free cook and returns true if the order was assigned to a cook and returns false if not.

# **Assgin\_Vegan\_Order**

**Member of class:** Restaurant

**Inputs:**

* Ord: the order that will be assigned.
* Timestep: the current timestep of the system.
* GCooks\_Assgined: status bar strings if the order is assigned to a normal cook.

**Returns:**

* Bool.

**Called by:**

* Restaurant:: Assgin\_Cook\_Order().

**Calls:**

None.

**Function Logic description:**

It assigns the orders of type Vegan to the free cook and returns true if the order was assigned to a cook and returns false if not.

Section2: Ahmed Mahmoud Ghareeb,110

# **Assgin\_VIP\_Order**

**Member of class:** Restaurant

**Inputs:**

* Ord: the order that will be assigned.
* Timestep: the current timestep of the system.
* NCooks\_Assgined: status bar strings if the order is assigned to a normal cook.
* VCooks\_Assgined: status bar strings if the order is assigned to a VIP cook.
* GCooks\_Assgined: status bar strings if the order is assigned to a normal cook.

**Returns:**

* Bool.

**Called by:**

* Restaurant:: Assgin\_Cook\_Order().

**Calls:**

None.

**Function Logic description:**

It assigns the orders of type VIP to the free cook and returns true if the order was assigned to a cook and returns false if not.

# **Assgin\_Urgent\_Order**

**Member of class:** Restaurant

**Inputs:**

* Ord: the order that will be assigned.
* Timestep: the current timestep of the system.
* NCooks\_Assgined: status bar strings if the order is assigned to a normal cook.
* VCooks\_Assgined: status bar strings if the order is assigned to a VIP cook.
* GCooks\_Assgined: status bar strings if the order is assigned to a normal cook.

**Returns:**

* Bool.

**Called by:**

* Restaurant:: Assgin\_Cook\_Order().

**Calls:**

None.

**Function Logic description:**

It assigns the orders of type Urgent to any free cook, in break cook, in reset (injury) cook respectively, and returns true if the order was assigned to a cook and returns false if not.

# **Free\_Cooks**

**Member of class:** Restaurant

**Inputs:**

* Timestep: the current timestep of the system.

**Returns:**

* Void.

**Called by:**

* Restaurant:: Interactive\_Mode().
* Restaurant:: StepByStep().
* Restaurant:: Silent().

**Calls:**

None.

**Function Logic description:**

It frees all the cooks that should be free at the current timestep either if they were doing orders and finish it or returning from a break.

Section3: Ahmed Muhmmed AbdelHamid,109

# **Injuered**

**Member of class:** Restaurant

**Inputs:**

* Timestep: the current timestep of the system.
* Probability: The probability of the cook to be injured (random number to be compared to the one read from the input file).

**Returns:**

* Bool.

**Called by:**

* Restaurant:: Interactive\_Mode().
* Restaurant:: StepByStep().
* Restaurant:: Silent().

**Calls:**

None.

**Function Logic description:**

If the probability (sent as a parameter) satisfies the Injury probability (read from the input file) it makes the first busy cook injured by making his speed decrease to half and take reset after finishing the order.

If the first busy cook was injured, it injures the second one. it was a try to reduce the probability of injuring the first busy cook server times as it might make the cook never finishes his order (very little speed and large finishing timestep).

# **Auto\_Promotion\_Event**

**Member of class:** Restaurant

**Inputs:**

* Timestep: the current timestep of the system.

**Returns:**

* Void.

**Called by:**

* Restaurant:: Interactive\_Mode().
* Restaurant:: StepByStep().
* Restaurant:: Silent().

**Calls:**

None.

**Function Logic description:**

If the promotes any Normal order or VIP order if it reached the time limit.

# **Read\_Input**

**Member of class:** Restaurant

**Inputs:**

None.

**Returns:**

* String: the name of the input file.

**Called by:**

* Restaurant:: Interactive\_Mode().
* Restaurant:: StepByStep().
* Restaurant:: Silent().

**Calls:**

* ArrivalEvent().
* CancellationEvent().
* Promotion().

**Function Logic description:**

It opens the input file and initializes the Normal\_Cook, Vegan\_Cook, and VIP\_Cook lists and it initializes the Event queue.

Section4: Omer Muhammed Yacine,205

# **OutputFile**

**Member of class:** Restaurant

**Inputs:**

* Filename: the name of the output file.
* Injeredcooks: the numbers of the cooks that got injured.

**Returns:**

None.

**Called by:**

* Restaurant:: Interactive\_Mode().
* Restaurant:: StepByStep().
* Restaurant:: Silent().

**Calls:**

None.

**Function Logic description:**

It creates an output file with the name of “OUTPUT\_” + the same name as the input file. And then writes all that is needed in it.

# **Draw\_Status\_Bar**

**Member of class:** Restaurant

**Inputs:**

None.

**Returns:**

* Void.

**Called by:**

* Restaurant:: Interactive\_Mode().
* Restaurant:: StepByStep().
* Restaurant:: Silent().

**Calls:**

None.

**Function Logic description:**

It draws the left part of the status bar ( the number of each normal type and how many of them is free at the moment).

# **Draw\_Assignation**

**Member of class:** Restaurant

**Inputs:**

* Normal\_Cooks\_Assgined: the cooks of Normal type which is assigned in the previous timestep.
* Vegan\_Cooks\_Assgined: the cooks of Vegan type which is assigned in the previous timestep.
* VIP\_Cooks\_Assgined: the cooks of the VIP type which is assigned in the previous timestep.

**Returns:**

* Void.

**Called by:**

* Restaurant:: Interactive\_Mode().
* Restaurant:: StepByStep().
* Restaurant:: Silent().

**Calls:**

* GUI:: PrintMessageN().

**Function Logic description:**

It draws the right part of the status bar ( the cooks of each type which were assigned in the previous timestep).