

=====

Setup

Player Options

- Set Amounts of each Ingredients
- Set Cost of Soup

System Options (Constants)

- GAME_LENGTH //length in min. of the game.
- NEXT_WAVE_IN //length in min. of how often a new wave of customers comes
- IMPULSE_CHANCE_RANGE //what the random number range will be
- IMPULSE_CHANCE //which number out of impulse range gives them a impulse buy
- IMPULSE_AMOUNT //\$\$\$ how much more the customer is going to spend with an impulse
- PRICE_DEVIATION //determines the deviation of what a customer is willing to pay
- PREFERENCE_DEVIATION //deviation of what cust. will like according to their set preference

=====

Start Simulation

- Runs for two minutes
- <<No pause or menu options available>>
- Create a new store and add a soup to it.
- Create a new customer driver.

=====

Create Customers

- Every ten seconds a new wave of customers is created.
 - Every customer is given a set of preferences, a possible impulse purchase, an “intuition,” and also a “life”.
 - The preferences will be a random int for each ingredient and the amount of money they’re willing to spend.
 - The impulse purchase will be selected at random at customer initialization.
 - Their intuition is a timer that determines when they’ll go up and order. If it’s set after their life has expired they’ll never order (i.e. they’ll just become a loiterer). We should run a test before the intuition timer is initialized to determine if they’re going to be a loiterer and if so don’t start the timer so we don’t waste resources.
 - The life is basically a timer that describes how long they stay in the restaurant. Once the timer runs out, they gracefully leave the restaurant by unsetting the intuition timer (if applicable), (exiting GUI [animation]), and removing themselves from the list of customers.
- The number of new customers is based on a Random number between 1 & 10 and is weighted by popularity. So initially (at popularity 0), at max only 10 customers would show up.
 - Popularity starts at 1 and then goes up indefinitely, it can’t be set at less than 0 so that it’s not impossible to recover from no one liking you (i.e. if you got negative -10

popularity then you would have zero customers coming in and have no one to increase your popularity).

- This will be set very primitively: every customer that likes their meal will add 1pt to popularity, and customers that don't enjoy it will decrement it by 1. If by some chance everyone of the ingredients matches the customers preference we'll add 10.

=====

Interaction

- Interaction will be what the customer does in the restaurant.
 - + interact()
 - this will control the process of interaction. it'll first call wantsToOrder(), then order(), then finally consume().
 - - wantsToOrder()
 - This will decide if the customer wants to place an order.
 - It will consult the price. If the price is within their deviation it will decide its a good idea to buy it.
 - - order()
 - Create a new instance of the soup which will increase the total number of soups purchased and the revenue.
 - - consume()
 - Determine if the ingredient amount is within the deviation of customers preference for each ingredient (iterate through the list of ingredients and check each one.) If the majority of ingredients fall within the preference range then add 1 to the popularity, else subtract 1.

=====

Score

- Total sales
- Total Revenues
- Customers vs Customers who Purchased
- Customer Satisfaction
 - "x" Customers liked your soup
 - "y" Customers disliked your soup
 - Your approval rating was $(\text{approved_customers}/\text{total_customers}) * 100$

=====