Team ChickenNuggets APCS2 pd1
Bing Li, Kenny Li, Maggie Zhao 2018-05-02

Game Overview / UX: You opened up a new restaurant in Broadway and a lot of customers have come to try out your food. Every customer has a different order and you are responsible for serving your customers before they leave! You can do this by clicking on the customer to display their order and time limit, then click on the desired dish in the kitchen to serve. If you have successfully served the customers before time is up, you will get a generous tip that you can use to upgrade your store with new cooking equipment. Otherwise, you will receive a punishment. Once you have finished serving 25 customers, you win and may proceed to the next level!

Rules:

- Customers are served with a priority queue- you have to serve the customers that come in first, but any "special" customers that come in will shove to the front of the line.
- Must serve within time limit
    - If completed the goal, you will get a tip from the customer
    - If you fail to provide the service, you will lose money, and several other customers will leave the store as well.
    - If any food is burnt (or contains the wrong ingredient), you get a smaller amount of tips
- If you have enough tips, you can buy new equipments in the store that will make the cooking process more efficient.
- Win condition: successfully served everything to certain number of customers (25 maybe?), game ends. Otherwise, keep going until the win condition is reached

Possible Classes

+ Customer: creates a regular customer object and implements methods that a regular customer can do (ex. Ordering food, giving tips, leave, stay)
    + Gordon Ramsay: a subclass of regular customers, but this object has higher temper value and less time limit (more likely to leave)
    + Special customers: TBD
+ Food: should have prices, flavor, size, ingredients? and time for making. If a customer ordered food, it will be stored in a linked list (aka. inventory) and will be removed / served once the customer from the priority queue is called
    + Cooked: overwrites the instance variables in Food.
    + Topping: overwrites the instance variables in Food
    + Drinks: overwrites the instance variables in Food

+ Store: has a list of kitchen upgrades that can be bought. Each one will help reduce the amount of time it takes to cook food, or increase the time customers will wait for their food.

Rough Timeline:

1. 6 / 6 / 2018: first version of game done (should be the MVP, done in Java)
2. 6 / 9 / 2018: second version of game done (should contain GUI, done in Processing)

Prioritize TO-DO List

1. Customer class (java)
2. Foods class (java) and the subclasses
3. Subclasses of Customer (java)
4. Store class (java)
5. Turn everything into Processing (GUI optional)
6. Add different levels

How are we making use of at least two other second term concepts?

- We are using priority queue to store the customers
- We are using linked list to store the food orders, since removing the food from a linked list cost the least run time

How are we making meaningful use of OOP?

- The customers and foods are both objects. Each object has a state and behavior. For example, the customers have order and time limit as instance variables and they can make orders and raise tips