Noah Frost, Adegbesan Dolapo

**Summary:** We will be simulating half a day or a day at a restaurant called In-N-Out, which is a fast food restaurant. We will be using an event-driven simulation in which people will come and go from the restaurant, and these events will create other events, such as a person entering a queue and then ordering food. We can implement some tracking variables that can give us statistics of the simulation after it runs. The challenges that we expect to run into in this project will be building an effective event class that can cover and expand to meet all of the possible events, and how to keep track of the desired information.

**Definition:** The project that we are doing will require us to build a vector of a class called Event, and we will randomly fill this up with some random events that have to do with people coming into the restaurant at a certain time. Then a loop will then find when the next event should happen, which is the event with the lowest time, and when this event happens, a new event will be created and put into the vector with some time associated with it. These new events include someone getting in line, ordering food, and leaving.

First we will fill up the vector with random times that people come into the restaurant. Then we have to assume that people come in at a relatively constant rate. Given enough time, we could implement some statistical adjustment to the rate at which people come in. This would likely look like a normal curve that gives the likelihood of someone coming in at a certain mealtime. This might make results more interesting and accurate, but also would be more complicated and require extra time.

Every time that someone walks into the restaurant – the first event – they will enter the line. After waiting some time they will order food – event 2. Then they will leave the restaurant after some time.

While the events take place, certain stats will be tracked. This includes the time people wait in the queue, how much money people spend, how long people wait for food, and how many people are served. These will be kept track of by adding to running totals, then dividing by the total number of people served to find averages. It is also possible that other stats could be thought of as we build the project and these can be added later.

**Class:**

|  |
| --- |
| Event |
| * int time //the time that the event will occur * int type:  //the type of event where 1=entering, 2=getting in line, 3=ordering food, 4=getting food, 5=sitting down, 6=leaving * Group group; //the group associated with the event happening |
| * Event(); * Event(int time,int type) * Void setEvent(int time,int type,Group group) * Void setTime(int time) * Void setType(int type) * Void setGroup(Group group) * Void setName(int name) * Int getTime() * Int getType() * Group getGroup() * void doEvent(vector<Group> queue, vector<Event> &list,int &queueTime,int &money,int &foodTime,int &customers) * void doQueue(vector<Group> &queue,vector<Event> &list,int &customers) * void doOrder(vector<Group> &queue,vector<Event> &list,int &money,int &queueTime) * void doFood(vector<Group> &foodQueue,vector<Event> &list,int &customers) * Void doLeave(vector<Event> &list,int &customers) * Int orderTime() |