The Beer Game

Overview

This game is a depiction of a supply chain and, hence, simulates the four stages - retailer, wholesaler, distributor and factory. Each supply chain produces and delivers certain units of beer. The factory produces and the other three stages of the supply chain deliver it to the customer. So, we'd be dealing with orders (flowing upstream) and deliveries (flowing downstream).

The goal of this game is to minimise the total cost for everyone in the supply chain by maintaining low stocks yet managing to deliver all the orders.

The game is simple - each stage/group has to fulfil the incoming orders of beer. The important aspect to be noted here is the delay or time lag between the various stages of the supply chain owing to the logistics and production time. Each delivery requires two rounds until they are finally delivered to the next stage. This is represented by two shipping delay fields located in between the supply chain stages as well as at the production end.

Customer -> Retailer -> Wholesaler -> (1 week delay) -> Distributor -> (1 week delay) -> Factory

Only 4 players can be play a game at a time.

How to play

- A user can select to create a new game or join an existing game.
    - Creating a game
        - Click on 'Create new game '. You will get a 5 digit unique key for the game session, which is needed for other players to join in.
        - The creator of the game is the leader.
    - Joining an existing game
        - Enter the game key to join an existing game.
- Upon entering a game, the player will be assigned a role from either of retailer, wholesaler, distributor or factory.
    - If the number of human players is less than 4, the leader can appoint ghost players to fill in the vacancies.
- In the beginning, each player has a pre-initialised inventory level of 5.
- To begin an order, each player must enter a finite number in their order basket.
    - The retailers would know the actual customer demand, however, they may choose to order any units of beer. The customer demand would be visible to the retailer on his screen.
    - Note that the factory's production is capped at 20 units of beer per week.
    - For ghost players the outgoing order is same as their incoming order.

- At the end of the week, the order is passed on to next stage in the supply chain. Deliveries, if any, come in at the end of a week.
- The players have to manage their orders such that they keep their inventories at the lowest level and yet are able to meet the demands of the customer(See scoring for more details).
- A game will have 20 rounds.

Admin Mode

- The admin can spawn x number of games for others to join. (required?)
- Since the admin is also a legitimate user, he may join a game like others.
- The admin will, however, have a default role of a customer for each game he has spawned.
    - Admin has to simply enter his demand in the order basket which would be relayed to the retailers at the starting of every week.

Scoring

- Each player starts with a balance of $0.
- Each unused unit of beer at the end of the week will incur a cost of $1 as handling charges.
- Each unit in backlog (backorders) will cost $2 per week.
- Each unit of beer sold will bring $4 in profits.

All the moves in the game would be recorded. At the end of the game, the players would be shown graphs to depict the bullwhip effect, with an option to view a history graph of all the games played so far.

- When a game is played in non admin mode, the computer will act as a customer and order certain number of units from the retailer each week.
    - The customer demand would be simulated as - consistent demand for first 5 weeks, huge increase in demand for next 5 weeks, huge drop in demand for next 5 weeks and then consistent demand for the next 5 weeks.
- Only the game which in which all the rounds were completed will be considered to be shown on the graph. All other games are ignored.
    - Graph depicts growth in customer demand, retailers and distributors orders

Architecture of the app

- Backend services would be built on Ruby on Rails with MongoDB as the database store.
- The front end would be purely on React Native (ReactJS)
    - ReactJS would help us make the app equally interactive and fast for mobile phones as well.

Schedule

- Week 1 - Discussions
- Week 2 - Confirm app specifications
- Week 3 - Submit wireframes and get the specifications revalidated
- Week 4 - Development time
- Week 5 - Submit the first prototype
- Week 6 - Development time
- Week 7 - Wrap up development and testing
- Week 8 - Submit the final app
- Week 9 - Testing and work on feature additions
- Week 10 - Testing and work on feature additions
- Week 11 - Final deployment on server and testing
- Week 12 - Buffer
- Week 13 - Buffer

Questions:

- Should the factory's max capacity be made public?
- Should scoring mechanism be extended? If yes, how can it be used to make the game more fun?
- Any more features? Addons?
- How does the university's server look like? Will it provide us just the site space and the app has to be managed by us? (PAAS?)
    - Would university's server allow us to provision MongoDB and support Ruby on Rails?