

Restaurant Delivery Jam Level Generation Automation

Author: Abhijeet Munje

Overview

Objective:

To develop an algorithm to automate level progression or improve its efficiency.

Document Audience:

Product Manager, Game Developer, Management, Art Department, QA and Publisher.

Algorithm Design

Components:

- Delivery Agents Wave
- Wave Data
- Tray System
- Packaged Order Inventory (Grid)
- Levels
- Features
- Difficulty Scale: Early Stage, Mid Stage, Late Stage, Tycoon Stage

Basic Algorithm Steps

Let's consider we are designing pseudo code by considering first 10 levels only.

```
1)  for(level= 1, level<10,level++)
2)  {
3)    Delivery agent wave=1;
4)    Packaged order inventory=1
5)    GenerateLevel();
6)  }
7)
8)  Void GenerateLevel()
9)  {
10)    must(delivery agents=order inventory)
11)    Delivery agent wave ++;
12)    Packaged order inventory ++;
13)    if(Difficulty Scale == Early Stage && level==5)
14)    {
15)      Add (Feature 1 to the level)
16)    }
17)  }
```

How it works:

- We procedurally generate levels based on the defined component variables.
- So we have a function named GenerateLevel() that accommodates logic which is as below:
 - 1) It mandates that no. of delivery agents are equal to order inventory for game balance.
 - 2) It increments wave and Inventory. Example- 1 wave has 3 agents, 2 waves will have 6 agents
 - 3) It checks if a level is at Early Stage progression and is Level 5, then a Feature 1 is added to the level, let it be the timer feature for an agent.
- Now we are executing this function from Level 1 to 10 and the levels will be generated as per the function's logic till Level 10.
- Note: We can create various such functions and can iterate them for different level numbers and generate new levels each time.

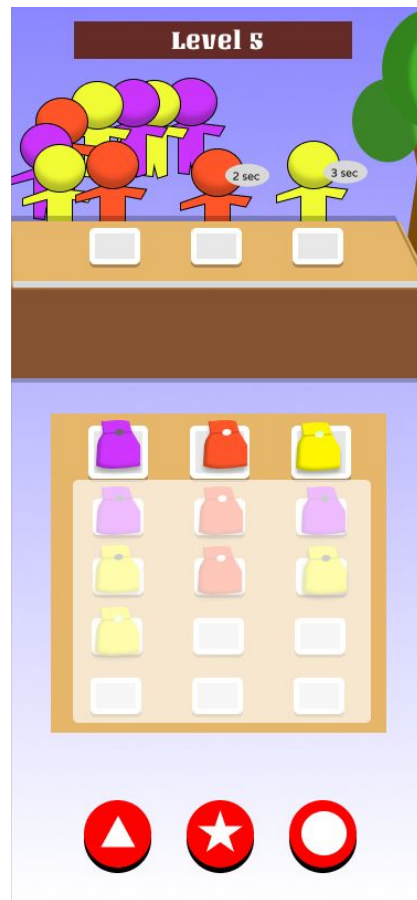
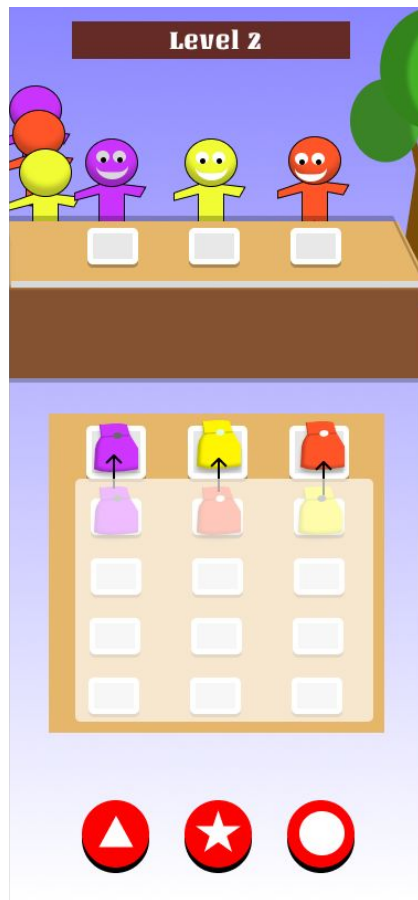
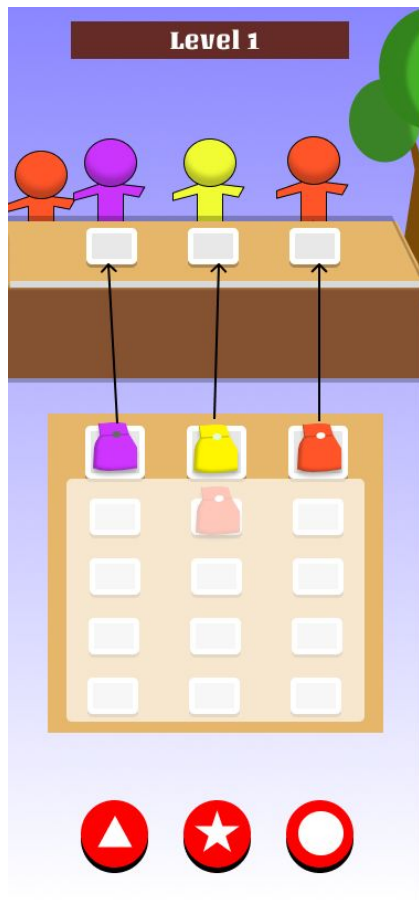
Agent Wave - Inventory Balance Algorithm

- 1) **if(Wave 2 agents = Orange, Yellow, Purple, Green)**
- 2) **Then**
- 3) **(Order Inventory 2 Packages = Orange, Yellow, Purple, Green && Position = 1st || 2nd Row)**

How it works:

- The main thing is to balance the matching sequence of delivery agents in the queue and the orders that are in the inventory.
- So for example, if we have Wave 2- Orange, Yellow, Purple & Green Agents in the queue after the wave 1, then the Order Inventory 2 must have same colored packages slightly shuffled in rows 1 and 2 for beginning normal level.
- This shuffling between rows can be made difficult based on difficulty stage.

Visual Demo



The agents have increased in Level 2, so as the order packages have also increased making a match and in Level 5 as per the algo, we have added timer feature to the agents.