Sprint No.5

## Bestellungen

Die Bestellung wurden nur fertig implementiert.

Anstatt Pizza sofort zu verkaufen, werden Bestellungen erzeugt. Aufgewärmte Pizza wird nur verkauft, wenn es eine Bestellung dafür gibt.

Jede Bestellung soll 1 Pizza verlangen. Neue Bestellungen werden durch die Nachfrage und durch den Preis beinflusst. Falls Pizzen im Lager vorhanden sind, dann können die Bestellungen abgearbeitet werden.

gemacht:

* eine Variable und ein Element für die Pizza-Bestellungen erstellt
* mathematische Logik für die Anzahl an Bestellungen
* alle 5 Sekunden (1 Stunde im Spiel) kommen Bestellungen ein
* die Pizza-Button-Click-Funktion überschreiben, damit nur Pizzen gemacht werden können, falls Bestellungen da sind

## Random-Events

Jede Minute (also zwei Mal pro Tag) gibt es eine Chance, dass ein zufälliges Event stattfindet. Jedes Event hat eine bestimmte Dauer (muss noch gebalanced werden). Die Standardchance ist 10%.

* Pizza Festival: Nachfrage ist um 150% erhöht, Command: TriggerFestival()
* Massive Sale: Pizzen kosten nur halb so viel, Command: TriggerPizzaSale()
* Sudden Inflation: Pizzen sind um einiges teurer, Command: TriggerInflation()
* Power Outage: Alle Automaten sind deaktiviert, Command: TriggerPO()

Command “StopEvent()” beendet das momentan aktive Event, falls es eines gibt.

Command “SetEventInterval(num)” bewirkt, dass es alle num Sekunden eine Chance auf ein Event gibt.

## Pizza-Machine-Bugfix

Pizza Machine/Buy Machine wird nun sofort angezeigt, wenn das Level 2 aufgerufen wird.

Code-example:

if (l2active==false) {

addAutomats();

showModalLevel2();

console.log("You reached level 2!");

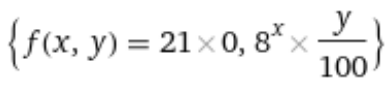
}

## setLevel()-Bugfix

Die setLevel()-Funktion ist nun funktionsfähig - falls zum Beispiel Level 3 geladen wurde, dann kann Level 2 nicht geladen werden → Neuladen der Website als einfache Lösung.

## Nachfrage

Die Nachfrage wurde nun vollständig implementiert; gefehlt hat eine Ergänzung der Nachfrage-Funktion - nun sieht sie so aus:



zur Formel: das sind die neuen Bestellungen pro Zeiteinheit

- x… Preis in Euro

- y… Nachfrage in Prozent (deswegen /100)

## Github-branches

Github-branches wurden zusammengefügt und es wurde auch eine neue erstellt.

## Werbung

Die Werbung wurde vollständig implementiert und ist funktionsfähig.

## Code testen bis inkl. Level 3

Bugfixes vorgenommen und Code des gesamten Projekts wurde durchgegangen und überarbeitet.

## Resümee Sprint 5

Sprint No. 5 wurde besprochen; Sprint-Review und Sprint-Review-Protocol wurden erstellt.

Vorbereitung für Level 4

## Leaderboard

* Implementierung eines Leaderbords von den Top-10 Restaurants
* Ziel:
  + Andere Restaurants zu überholen erster am Leaderboard zu sein
  + Evtl. andere Restaurants aufkaufen und dadurch noch mehr Geld zu generieren
* Schritte:
  + die anderen 9 Restaurants werden mit random Werten generiert
    - Werte:
      * Anzahl an verkauften Pizzen
      * momentanes Geld (oder insgesamtes Geld)
      * Nachfrage (beim Verkaufswert von 2,50€)
  + ein Ranking wird erstellt, wobei insgesamt berücksichtigt wird:
    - insgesamte Anzahl an erstellten Pizzen
    - momentanes Geld
    - Pizzen-Nachfrage (beim Verkaufswert von 2,50€)
    - zusammen in einer Funktion
  + jede Sekunde wird das Ranking aktualisiert und mit Effekten werden die Plätze getauscht
  + die Verkaufszahlen ändern sich auch bei den KI-Restaurants
* mögliche Erweiterung:
  + andere Restaurants zu sabotieren
  + andere Restaurants zu kaufen (für hohen Preis → dafür doppeltes Geld generieren im Vergleich zu zuvor)
* Level 5 beginnt erst, wenn der erste Platz am Leaderboard erreicht wurde