

BOULANGERIE

Kit Pédagogique

LA BOULANGERIE

Simulation de production et gestion de ressources en temps réel.

Vous devez créer cette application avec VueJS.

OBJECTIFS

L'objectif de cet exercice est de créer une simulation de production et de gérer des commandes entrantes associées à cette production.

CONTEXTE

La boulangerie

Le boulanger est LE spécialiste des baguettes.

Pour fabriquer ses baguettes, le boulanger a besoin de farine.

Pour produire sa farine, le boulanger doit acheter des moulins.

Pour acheter des moulins, le boulanger a besoin d'or.

Lorsque la boulangerie est ouverte, le boulanger crée des baguettes à la chaîne.

Pour augmenter la cadence de production, le boulanger doit améliorer sa boulangerie.

Pour améliorer sa boulangerie, le boulanger a besoin d'or.

La boulangerie génère des faibles coûts de maintenance (en or) indexés sur son niveau.

Pour gagner de l'or, le boulanger vend ses baguettes.

Augmenter le niveau de la boulangerie augmente également sa popularité. La qualité des baguettes s'améliore et leur prix augmente en conséquence.

Les commandes

Le boulanger vend ses baguettes à la commande.

Le boulanger gère ses commandes dans une file d'attente.

La file d'attente peut contenir au maximum 10 commandes.

Pour chaque commande :

- un nombre de baguettes est défini
- le prix unitaire de la baguette est fixé par le demandeur
- un temps limite est défini
 - la commande doit être livrée avant l'expiration du délai
- le nombre de baguettes et le prix unitaire permettent de calculer le montant total de la commande

Le boulanger peut accepter ou refuser une commande.

Une commande acceptée ne peut plus être annulée.

Les commandes sont traitées dans l'ordre dans lequel elles ont été acceptées.

Une commande est livrée lorsque la boulangerie dispose de suffisamment de baguettes pour satisfaire la commande.

Lorsqu'une commande est livrée, le boulanger gagne le montant total de la commande en or.

Formules (prix, production, commandes...)

Soit :

- `BAKERY_LEVEL` = niveau de la boulangerie
- `NB_MILLS` = nombre de moulins possédés

Boulangerie

- Coût de maintenance par seconde (en or): $0.05 * \text{BAKERY_LEVEL} * \text{NB_MILLS}$
- Prix d'amélioration de base (en or): 100
- Chaque amélioration augmente le prix de la suivante selon cette formule:
 - $(\text{nouveauPrix} = \text{prixActuel} * 1.5)$

Notoriété et commandes

Le nombre de baguettes pour chaque commande est indexé sur le niveau de la boulangerie. Il est déterminé au hasard à la création de la commande et calculé selon ces règles:

- minimum: 5
- maximum: $30 * \text{BAKERY_LEVEL}$

Le prix unitaire est également indexé sur le niveau de la boulangerie. Il est déterminé au hasard à la création de la commande et calculé selon les règles suivantes:

- minimum: $\text{BAKERY_LEVEL} / 100$
- maximum: $30 * \text{BAKERY_LEVEL} / 100$

Moulins

- Prix d'achat de base (en or): 80
- Chaque moulin acheté augmente le prix du suivant selon cette formule:
 - $(\text{nouveauPrix} = \text{prixActuel} * 1.5)$

Farine

- Production par seconde: NB_MILLS
 - ex: une boulangerie possédant 5 moulins produit 5 unités de farine par seconde
- Consommation par seconde: BAKERY_LEVEL + 1
 - ex: une boulangerie de niveau 3 consomme 4 unités de farine par seconde

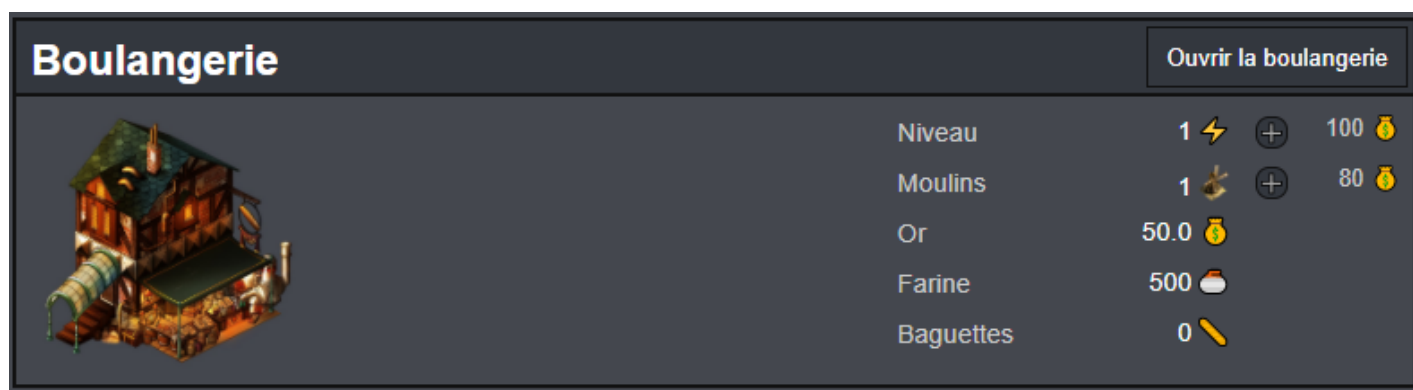
Baguettes

- Production par seconde : BAKERY_LEVEL
 - ex: une boulangerie de niveau 3 fabrique 3 baguettes par seconde

INTERFACE

La boulangerie

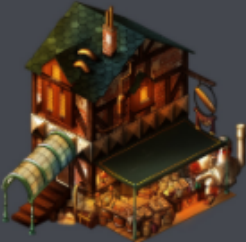
Créer une page web et représentez les données de la boulangerie à l'identique de la capture suivante (la largeur du cadre s'adapte à la largeur de la fenêtre).



Les images se trouvent dans le répertoire assets. Certaines icônes (l'éclair, le sac d'or, la baguette...) ne sont pas des images mais des caractères UTF-8. A vous de trouver la manière de les afficher :) Recherchez sur ce site : <https://www.utf8icons.com/>.

Le bouton "ouvrir la boulangerie" permet de démarrer la production. Lorsque la boulangerie est ouverte, une fonction s'exécute toutes les secondes et les règles définies plus haut s'appliquent (perte d'or, production de farine...).

Boulangerie



Fermer la boulangerie

Niveau	1	⚡	+	100	💰
Moulins	1	🔧	+	80	💰
Or	49.5	💰			
Farine	489	🍞			
Baguettes	11	🥖			

Ajouter un entête de page avec un titre et un tableau affichant le cumul des gains et pertes de ressources.

Le Boulanger

Or gagné	Or dépensé	Farine produite	Baguettes produites
0.00	1.95	39	39

Boulangerie



Fermer la boulangerie

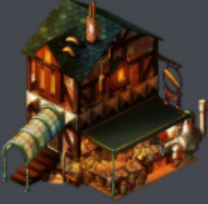
Niveau	1	⚡	+	100	💰
Moulins	1	🔧	+	80	💰
Or	48.1	💰			
Farine	461	🍞			
Baguettes	39	🥖			

Validez le fonctionnement de votre application avec votre formateur avant de passer à la suite.

Les commandes

Sous la boulangerie, ajouter un bloc contenant un tableau de 10 lignes (+ entête du tableau).

Boulangerie



Ouvrir la boulangerie

Niveau	1			100	
Moulins	1			80	
Or	50.0				
Farine	500				
Baguettes	0				

Commandes

0	0	0	0	0	0	En attente...
0	0	0	0	0	0	En attente...
0	0	0	0	0	0	En attente...
0	0	0	0	0	0	En attente...
0	0	0	0	0	0	En attente...
0	0	0	0	0	0	En attente...
0	0	0	0	0	0	En attente...
0	0	0	0	0	0	En attente...
0	0	0	0	0	0	En attente...
0	0	0	0	0	0	En attente...

Chaque ligne de ce tableau correspond à un emplacement pouvant accueillir une commande. Pour chaque commande, le tableau indique :

- Le numéro de commande
- Le nombre de baguettes demandées pour cette commande
- Le prix unitaire de la baguette pour cette commande
- Le prix total de la commande (prix unitaire * nombre baguettes)
- Le temps restant
 - nouvelle commande : aléatoire entre 10 et 60 secondes
 - commande acceptée : le compteur redémarre à 90 secondes
- une colonne d'action dynamique
 - nouvelle commande :
 - 1 bouton pour accepter la commande
 - 1 bouton pour refuser la commande
 - commande acceptée :
 - une barre indiquant la progression de la commande
 - commande annulée :
 - texte "Commande annulée"
 - pas de commande à cet emplacement
 - text "En attente..."

Commandes en attente d’être acceptées/refusées :

Boulangerie



Fermer la boulangerie

Niveau1⚡⊕100💰

Moulins1🌀⊕80💰

Or49.8💰

Farine495🍞

Baguettes5🥖

Commandes

					
1	11	0.26	2.86	6	<div>Y</div> <div>N</div>
2	23	0.12	2.76	14	<div>Y</div> <div>N</div>
5	21	0.25	5.25	16	<div>Y</div> <div>N</div>
9	8	0.06	0.48	10	<div>Y</div> <div>N</div>
3	24	0.19	4.56	11	<div>Y</div> <div>N</div>
7	27	0.21	5.67	11	<div>Y</div> <div>N</div>
10	5	0.18	0.9	15	<div>Y</div> <div>N</div>
8	12	0.29	3.48	13	<div>Y</div> <div>N</div>
4	21	0.13	2.73	10	<div>Y</div> <div>N</div>
6	18	0.06	1.08	13	<div>Y</div> <div>N</div>

Commandes en cours :

Commandes					
					
21	29	0.01	0.29	70	<div><div></div></div> 0/29
27	17	0.03	0.51	69	<div><div></div></div> 9/17
30	19	0.2	3.8	69	<div><div></div></div> 0/19
37	19	0.28	5.32	4	<div><div>Y</div><div>N</div></div>
33	15	0.18	2.7	1	<div><div>Y</div><div>N</div></div>
38	21	0.23	4.83	13	<div><div>Y</div><div>N</div></div>
39	20	0.05	1	12	<div><div>Y</div><div>N</div></div>
36	6	0.08	0.48	2	<div><div>Y</div><div>N</div></div>
34	15	0.02	0.3	4	<div><div>Y</div><div>N</div></div>
35	7	0.2	1.4	0	Commande annulée !

Comme pour la boulangerie, les informations du tableau sont actualisées une fois par seconde.

Lorsque la boulangerie est fermée, le tableau des commandes est réinitialisé.

Validez le fonctionnement de votre application avec votre formateur avant de passer à la suite.

Le journal

Sous le bloc des commandes, ajouter un nouveau bloc qui affichera les 20 derniers événements :

- La boulangerie gagne de l'or
- La boulangerie accepte une commande
- La boulangerie livre une commande

Log

14:35:29: La boulangerie a gagné 0.29 Or
14:35:29: La boulangerie a livré 29 baguettes
14:34:59: La boulangerie a gagné 3.8 Or
14:34:59: La boulangerie a livré 19 baguettes
14:34:40: La boulangerie a gagné 0.51 Or
14:34:40: La boulangerie a livré 17 baguettes
14:34:24: La boulangerie a gagné 0.63 Or
14:34:23: La boulangerie a livré 9 baguettes
14:34:14: La boulangerie a gagné 4.8 Or
14:34:14: La boulangerie a livré 24 baguettes
14:34:12: La commande 21 a été acceptée.
14:34:11: La commande 27 a été acceptée.
14:34:11: La commande 30 a été acceptée.
14:34:11: La commande 28 a été acceptée.
14:34:10: La commande 24 a été acceptée.
14:34:10: La boulangerie a gagné 6.24 Or
14:34:10: La boulangerie a livré 26 baguettes
14:34:09: La commande 25 a été acceptée.

APERÇU DE L'INTERFACE COMPLÈTE :

Le Boulanger

Or gagné	Or dépensé	Farine produite	Baguettes produites
2.10	0.60	12	12

Boulangerie



Niveau 1 ⚡ + 100 🪙

Moulins 1 🪄 + 80 🪙

Or 51.5 🪙

Farine 488 🍞

Baguettes 2 🥖

Fermer la boulangerie

Commandes

📱	🥖	🪙/🥖	🪙🪙🪙	🕒	👤
6	14	0.29	4.06	82	<div><div></div></div> 2/14
11	10	0.16	1.6	12	<input type="button" value="Y"/> <input type="button" value="N"/>
2	22	0.12	2.64	6	<input type="button" value="Y"/> <input type="button" value="N"/>
8	7	0.28	1.96	83	<div><div></div></div> 0/7
7	7	0.11	0.77	4	<input type="button" value="Y"/> <input type="button" value="N"/>
10	26	0.04	1.04	16	<input type="button" value="Y"/> <input type="button" value="N"/>
3	24	0.19	4.56	0	<input type="button" value="Y"/> <input type="button" value="N"/>
4	5	0.17	0.85	1	<input type="button" value="Y"/> <input type="button" value="N"/>
5	12	0.11	1.32	3	<input type="button" value="Y"/> <input type="button" value="N"/>
9	18	0.07	1.26	10	<input type="button" value="Y"/> <input type="button" value="N"/>

Log

14:54:21: La boulangerie a gagné 2.1 Or

14:54:21: La boulangerie a livré 10 baguettes

14:54:17: La commande 8 a été acceptée.

14:54:15: La commande 6 a été acceptée.

14:54:14: La commande 1 a été acceptée.

Centre de Réadaptation de Mulhouse

Validez le fonctionnement de votre application avec votre formateur.

Lorsque vous avez terminé, proposez un système de sauvegarde de l'état actuel de la partie dans le navigateur avec l'api LocalStorage.

- Lors d'une sauvegarde, la partie est mise en pause.
- Au chargement, si une sauvegarde existe, l'état de la partie est rétabli.
- Si l'utilisateur quitte la page (en fermant le navigateur, par exemple), la partie est sauvegardée.

Vue JS

1. Céréales
2. Fruits et Légumes
3. Boulangerie

RESSOURCES ET EXERCICES POUR CONCEPTEURS ET DÉVELOPPEURS DISPONIBLES SOUS LICENSE MIT.

© Copyright MDevoldère pour ARFP · Sitemap · Contribuer · Dépôt GitHub · Font-Awesome