








Machine à café++



Raphaël Romanet

Lucas Aubron

Sommaire:

Fonctionnalités livrées	3
Justification de nos choix	3
Choix de la boisson des options et paiement 	4
Boissons et options en général 	4
Option soupe (Uniquement implémenté sur le statechart) 	4
Option Iced Tea 	5
Option ajout d'une tasse personnelle 	5
Option barre de chargement  (Non implémentée)	5
Gestion du temps dans la machine 	5
Utilisation de LTSA	6
Prise de recul	6

I. Fonctionnalités livrées

Actuellement, notre machine à café++ répond à toutes les attentes du MVP, y compris les nouvelles spécifications ajoutées en cours, telles que l'offre promotionnelle concernant le 11eme achat ou bien les options telles que l'ajout de vanille à la fin de la préparation.

En ce qui concerne les extensions, nous en avons implémenté 3 :

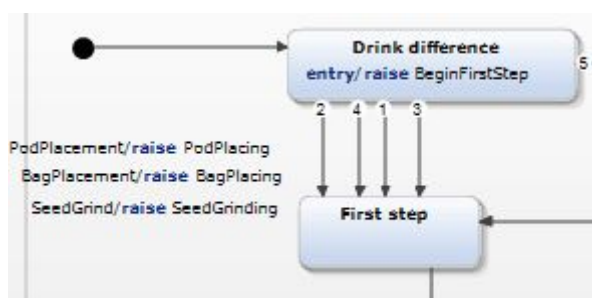
- La gestion de la soupe (Uniquement sur le stateChart)
- La gestion de l'ice tea
- La détection d'une tasse ajoutée par l'utilisateur (Uniquement sur le stateChart)

Pour ce qui est des options comme étant uniquement sur le statechart, nous les avons mises sur un statechart à part qui sera dans le dossier rapport. Ce statechart représente notre machine à son état final avec toutes les options implémentés.

II. Justification de nos choix

Tout d'abord, nous avons choisi de partir sur une machine dite "event-driven" car nous pensons qu'une machine à café est mieux représentée par une suite d'événements logiques plutôt que gérée par une horloge effectuant des opérations à chaque période donnée (200ms par défaut).

Finalement, nous avons décidé d'utiliser notre machine comme d'un garde fou pour notre code. Notre machine va envoyer des checkpoints tels que *"BeginFirstStep"* indiquant au code de démarrer l'étape une de la boisson précédemment sélectionnée par l'utilisateur. Le code va alors renvoyer un événement correspondant à la première étape de la boisson choisie.



Dans cet exemple, si nous avons choisie la boisson *"Thé"* notre code aurait alors renvoyé l'événement *"BagPlacement"* signifiant que la machine a bien effectué l'événement de la première étape, placer un sachet de thé. Cela nous permet d'éviter que notre machine n'ait pas trop de responsabilités, tout en mettant en place une bonne synergie d'envoi d'événements d'entrées et de sorties entre notre code et notre machine.

Cette implémentation donne un poids plus important au code utilisé et qui requiert donc une plus grande rigueur lors de son écriture et qui rend plus difficiles les contrôles possibles sur la machine.

1. Choix de la boisson des options et paiement

Tout d'abord lorsque nous arrivons dans notre premier état composite qui va gérer le choix de la boisson, des options et du paiement, la machine va demander au code de désactiver toutes les boissons ou options qui ne pourront pas être préparées par manque d'ingrédients. C'est selon nous la méthode la plus efficace puisque l'utilisateur est directement informé de quelles sont les boissons dont il a la possibilité d'acheter et cela nous évite une gestion supplémentaire à l'intérieur de la machine puisque les boissons non faisables ne sont tout simplement pas sélectionnables.

On retrouve le même système pour le choix des options. En effet, pour sélectionner une option, l'utilisateur doit tout d'abord sélectionner une boisson et seulement les options disponibles pour cette boisson, si elles ont assez d'ingrédients, vont s'afficher. Comme précédemment, nous faisons d'une pierre deux coups, la machine n'ayant pas à gérer des cas compliqués mais seulement envoyer quelle est la boisson et option qui a été sélectionnée, sans erreur possible.

2. Boissons et options en général

Concernant les boissons, nous avons essayé de factoriser au maximum, en regroupant les parties des recettes pouvant être effectuées de manière parallèle ensemble. Par exemple, chaque recette commence par une étape qui possède 2 ou plus actions qui peuvent être effectuées en parallèle. Nous avons appelé ça "step1" et notre code s'occupe d'effectuer les bonnes actions pour la "step1" de chaque boisson. Les actions parallèles étant évidemment des états orthogonaux, leur donnant leur pseudo parallélisme. Ce fonctionnement est universel pour nos boissons et extensions, sauf pour la soupe, qui sera détaillée plus bas.

3. Option soupe (Uniquement implémenté sur le statechart)

La soupe ayant une recette trop divergente par rapport aux autres recettes, nous avons décidé de l'implémenter différemment. Nous avons considéré que le placement de son bol, étant la première chose à faire dans la recette, était la spécificité de cette recette comme pourrait l'être le sachet de thé pour le thé ou bien le fait de moulinier les grains pour l'expresso. Une fois, cette étape faite, le code signal à la machine que la recette en cours est une soupe pour qu'elle choisisse l'étape spéciale de la soupe, un état parallèle triple. Une fois cet état fait, la soupe rejoint les autres boissons dans la machine, n'ayant pas d'autres spécificités.

4. Option Iced Tea 🍵

Pour ce qui est de l'option Iced Tea, nous avons considéré que pour de la préparation jusqu'à l'infusion on avait un thé banal. Après son infusion, le code va indiquer à la machine, qui n'a aucune notion de thé normal ou iced, de verrouiller la porte. Une fois la porte verrouillée, on va pouvoir refroidir le thé avec l'azote jusqu'à la température souhaitée par l'utilisateur. Une fois cette température atteinte, on déverrouille la porte et l'utilisateur peut récupérer son iced tea.

5. Option ajout d'une tasse personnelle (Uniquement implémenté sur le statechart) 🥤

Pour implémenter cette option, nous avons considéré encore une fois que la machine allait envoyer au code l'événement "une tasse à été insérée" au code, effectuant la réduction au passage, et que ce dernier allait retenir cette information. Ainsi arrivé au moment de poser le gobelet la machine s'interroge pour savoir si une tasse est déjà insérée et le code lui envoie l'événement adapté, soit il y a une tasse et on passe directement à l'état suivant soit on doit effectivement en ajouter un et la machine s'en occupe comme elle l'aurait fait dans le MVP. Cette implémentation nous permet de modifier seulement les parties liées à la pose du gobelet.

6. Option barre de chargement ⌚ (Non implémentée)

Nous n'avons pas implémenté cette option par manque de temps mais nous avons quand même pu réfléchir à une implantation cohérente avec notre machine. Nous aurions pu ajouter à la fin de chaque step, qui représentent les étapes de nos recettes, un état AddPercentage suivi du chiffre de l'étape, ajoutant un pourcentage à la progress bar en fonction de la longueur de l'étape. Cette implémentation à l'avantage d'être facile à ajouter, mais n'est par contre pas très précise. En effet, selon la boisson choisi la longueur d'une étape peut grandement varier ce qui rendrait la progress bar assez imprécise mais suffisamment intéressante pour l'utilisateur.

7. Gestion du temps dans la machine 🕒

Nous avons souhaité donner au minimum la responsabilité de gérer le temps à la machine. En effet, il n'aurait pas été logique d'assigner un temps fixe à des événements tels que le chauffage de l'eau. Par contre, nous avons fait le choix de lui laisser gérer le temps pour 2 événements. Le nettoyage de la machine ainsi que le positionnement du gobelet. Nous estimons que pour ces 2 événements, le temps est fixe et ne peut varier, ce qui nous autorise à gérer le temps de ces événements depuis la machine.

III. Utilisation de LTSA

A l'aide du logiciel LTSA nous avons mis au point trois analyses. Les trois sont des validations. La première permet de tester que lorsqu' on verrouille la porte, on est sûr de la rouvrir plus tard, pour éviter que la machine ne reste bloquée. L'autre validation nous permet de confirmer que peu importe la boisson, on aura bien une eau chaude versée. Nous avons fait ce test suite à l'implémentation complexe de la soupe et son interaction avec les autres boissons. La dernière est une validation de sécurité. On teste que si on utilise de l'azote liquide, on vérifie que plus tard on tombera forcément sur une transition "ouverture des portes" signifiant qu'on aura forcément fermé les portes au préalable.

IV. Prise de recul

Notre implémentation possède un inconvénient majeur: il est très difficile d'évaluer notre machine à l'aide de logiciels LTSA. En effet, les fonctionnalités étant encapsulées dans le code on ne peut pas vraiment tester efficacement notre machine, surtout en ce qui concerne la vérification.