

# Finite State Machine

Contrôleur de machine à boissons

## Rapport

Bastien Achard  
Julien Madrias

## Implémentation :

Lors de ce projet, nous avons développé un contrôleur de machine à boissons qui répond à toutes les spécifications abordées dans le MVP, ainsi qu'à certaines extensions. En ce qui concerne les extensions, nous avons intégré à notre contrôleur de machine à boissons les extensions suivantes :

- La gestion de la soupe
- La gestion de l'avancement de la préparation
- La détection des gobelets

## I. Choix de conception

### A. Choix sur la gestion de la préparation des recettes

Le système de gestion de la préparation des recettes est une partie essentielle de notre contrôleur de machine à boissons. C'est pour cela que nous avons mis beaucoup de temps à réfléchir à la meilleure solution de conception.

Nous sommes d'abord partis sur l'idée d'un système où la gestion de la préparation était quasi totalement dans le code. C'est-à-dire que nous pensions au fait que la FSM ne contienne que les étapes importantes de la préparation des recettes, tel que le lancement de la préparation qui gère la récupération des ingrédients et le démarrage du chauffage de l'eau. La fin de la préparation qui gère l'ajout ou non du gobelet et la fin du chauffage de l'eau, le versement de la préparation et l'ajout de supplément. Cette solution a certains avantages comme la réutilisation puisque les étapes sont communes à toutes les recettes, l'ajout de nouvelles recettes ne nécessitent pas de modification de la FSM. Mais en contrepartie cette idée n'utilise presque pas les fonctionnalités des FSM, puisque le système de parallélisme des événements dans la préparation est fait dans le code.

Ensuite nous avons pensé à un système où toutes les étapes de la préparation se faisaient dans la FSM et donc que les recettes étaient dans la FSM. Cette solution ne nous paraissait pas envisageable d'un point de vue réutilisation puisque si une recette devait être modifier ou qu'il y aurait de nouvelles recettes, nous devrions alors modifier la FSM.

Finalement nous avons choisi une solution qui nous semble un bon compromis puisque la FSM gère le parallélisme des événements ainsi que la gestion du temps entre les différents événements. Mais tout en ayant les recettes dans le code ce qui permet d'avoir une meilleure réutilisation puisque nous n'avons plus besoin de modifier la FSM. Cependant, notre solution finale n'est pas parfaite pour autant, par exemple, dans la FSM nous ne possédons pas d'information sur la recette en cours pour pouvoir ainsi changer les messages en fonction des options sélectionnées.

## B. Choix sur la gestion du temps des préparations

La gestion du temps des préparations est aussi une partie très importante de notre contrôleur de machine à boissons. Pour répondre à ce choix de conception, nous avons réfléchi à deux solutions.

La première était de gérer tout le système de gestion du temps des préparations dans le code où chaque recettes calculaient le temps de préparation de chaque étape pour ensuite lancer des threads afin de mettre en pause le code. Cette solution n'était pas très fonctionnelle puisqu'elle gérait le système de parallélisme assez mal et de plus elle n'utilisait pas du tout la fsm.

La deuxième solution, celle que nous avons choisie, utilise la FSM pour gérer le temps d'attente entre les étapes de la préparation d'une recette. Ce temps est calculé dans les différentes recettes (dans le code) et envoyé à la FSM par l'initialisation de variables. Cette solution permet d'avoir une gestion du temps beaucoup plus simple puisque si nous voulons rajouter une recette nous avons juste besoin d'implémenter la méthode de calcul du temps.

## C. Choix sur la gestion des sélections

Pour le système de sélection des boissons nous avons trouvé trois solutions. La première solution est de laisser le code tout gérer. La deuxième solution est de mettre chaque boissons dans la machine avec un état par boisson. Et la dernière solution est celle que nous avons choisie, à savoir une implémentation mixte des deux solutions précédentes.

Que ce soit le code ou la machine qui ait tout le système de choix à gérer, cette solution aurait apporté trop de complexité et des difficultés à implémenter de nouvelles boissons. Notre solution mixte permet de séparer les boissons actuelles et futures (par exemple l'iced tea) en catégories influant sur les options liées aux boissons comme la température. Ainsi ajouter une nouvelle boisson ne multiplie pas le nombre d'états et de transitions dans la machine et limite la duplication de code.

## II. V&V

Nous avons décidé de modéliser la partie Sélection de notre machine afin de vérifier que le paiement en CB bloque le paiement en espèce comme nous en avons fait le choix dans notre implémentation.

Cette question permet de vérifier une *safety property*.

## III. Conclusion

Selon nous, le point essentiel de notre projet qui devrait être à améliorer et qui serait à retravailler est la vérification du paiement qui se fait entièrement dans le code, si cette vérification se faisait directement dans la machine il serait facile de rajouter des conditions au lancement de la préparation.