

Potion Cooking

Date : 27/11/2018

Auteur : Floreau Luca

Analyse

Scénario :

Dans un univers médiéval fantastique, il existe, au fond d'une forêt, un stand de restauration rapide de potion magique. Les clients viennent demander une potion magique et ils sont servis dans les plus bref délai.

Principe de jeu:

Vous incarnez le sorcier fabricant les potions. Il vous est donc demandé d'accueillir les clients, de choisir les ingrédients, de gérer la cuisson, le remplir les récipients puis de servir les clients. Pour cela, vous avez accès à 4 fenêtres : Ingrédients, Cuisson, Préparation et Accueil. Les clients possèdent une limite d'attente avant de s'énerver et de partir.

Conditions de victoire :

- **Principale** : servir plus de la moitié des clients.
- **Secondaire** : servir tous les clients le plus rapidement possible.

Condition de défaite:

- **Principale** : servir moins de la moitié des clients

Cahier des charges :

Le projet devait être en 2D et il était à rendre avant le mardi 27 novembre 2018 à 23:59.
Voici les étapes les plus importantes de la création:

Gameplay:

- Glisser Déposer des items et transfère d'information entre les objets
- Utilisation d'une recette par le chaudron pour transformer plusieurs ingrédients en un seul
- Apparition des clients à intervalle régulier
- Affichage de la demande de client
- Gestion des menus

Graphisme :

- Intégration de l'interface utilisateur
- Création des sprites
- Animation recette
- Animation liquide
- Intégration des particules
- Intégration du son

Autres:

- Phase de test
- Rédaction de la documentation

Mécaniques:

Joueur:

Caméra:

Le joueur a accès à 4 fenêtres de vues entre lesquelles il peut naviguer grâce aux panneaux à l'écran ou grâce aux touches 1, 2, 3, 4, Espace du clavier

Controls:

Il peut saisir les objets déplaçables, accueillir les clients en cliquant dessus, avoir accès au détail de la commande en cliquant dessus

Le joueur peut aussi mettre le jeu en pause grâce au bouton disponible ou avec la touche P.

Il peut enfin quitter le jeu à tout moment avec la touche échap.

Objets:

Objets déplaçables :

Les ingrédients, la louche et les récipients sont déplaçables. Si les objets déplaçables sont lâchés sur la table, ils y resteront. Si les objets déplaçables sont lâchés sur leur récipient correspondant, ils transféreront leurs informations au récipient. Sinon l'objet sera détruit ou remis à sa place.

Chaudron:

Une fois que ce récipient possède 3 ingrédients, il vérifie si les trois ingrédients correspondent à une recette, si oui il remplace les trois ingrédients par la potion correspondante.

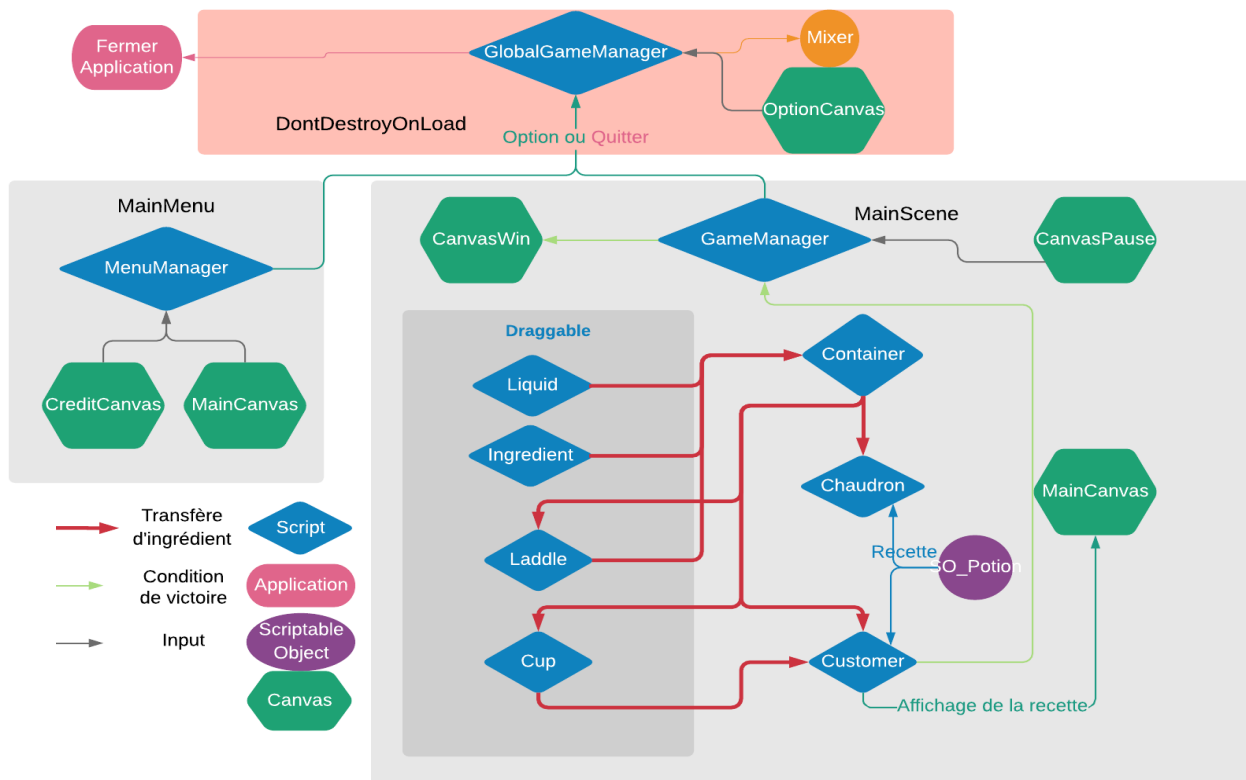
Clients:

A leur arrivé, les clients attendent de passé leur commande. Une fois accueilli, une recette s'affiche avec la recette de la commande. Une bulle s'affiche aussi au dessus de la tête des clients comprenant l'image de la commande et une jauge d'attente. Si le client attend trop longtemps, il partira sans sa commande. Si il est servi avec ce qu'il a demandé, il partira satisfait.

Réalisation :

Explication du code :

Le code est représenté ci-dessous sous forme de schéma:



La navigation entre les scènes se fait par des boutons. Le GlobalGameManager gère les options du mixer car elles doivent être synchronisées entre les scènes. A la fin, le canvas de fin s'affiche permettant de relancer la scène ou de changer de scène. L'affichage des recettes est géré par le client. Le système de Pause interagit avec le jeu et il stop toutes interactions avec la scène. La caméra est gérée par le CameraScript.

Justification des choix:

Transfère des informations entre objets:

Je devais choisir comment circulaient les informations de contenance à travers les différents objets. J'ai décidé d'utiliser des listes de sprites. La liste est compatible avec les objets pouvant contenir plusieurs ingrédients. Les sprites permettent d'afficher sans donnée externe la contenance sur les objets. Cependant, il est nécessaire dans certaines situations de se référer au nom du sprite pour les détecter.

Affichage de la recette:

Le client étant un prefab, je ne pouvais lui donner un serializable panel. Comme le nom était assez explicite, j'ai décidé de rechercher le panel par son nom plutôt que d'ajouter un tag uniquement pour cette fonction.

Conclusion des phases de test :

La dernière semaine, comme je possédais un jeu jouable, j'ai pu faire tester mon jeu à mon entourage. La compatibilité mobile aidant à faire tester n'importe où par n'importe qui.

Corrigé :

- Correction des différentes zones de détections
- Corrections des animations de recettes
- Ajout de particule pour visualisé la cuisson
- Ajouts d'une jauge d'énervement
- Ajouts d'une bulle afficher la commande de chaque clients
- Ajouts d'indication sur l'emplacement de la caméra
- Changer le fonctionnement du chaudron

Non Corrigé:

- Mise en évidence de la louche
- Plusieurs langues
- Impossibilité de remplir les verres sur la table

Conclusion:

Non réalisés :

L'ajout d'un déplacement latéral continue de la caméra a été envisagée mais comme il n'apportait rien au gameplay et qu'il demandait beaucoup de changement, il a été annulé.

Un test d'ajout d'une ambiance lumineuse a aussi été envisagé mais annulé comme non nécessaire et par manque de temps.

Ajouts possibles :

- Ajouts d'argents donné proportionnellement au temps d'attente
- Ajouts d'ingrédients et de recettes
- Ajout d'un tutoriel
- Compatibilité manette + déplacement latéral continue de la caméra

Impression personnel :

Ce projet m'a permis de grandement m'améliorer sur Unity et sur Illustrator. Il m'a aussi permis de mettre en pratique les mécaniques apprises en classe.

C'était la première fois que je produisais un jeu sur une tel durée. Je me suis rendu compte que de l'importance et de la quantité des détails.

Crédits :

Outils:

- Illustrator CC 2018, Photofiltre et <https://icones8.fr> pour les images et <https://fontmemes.com> pour les polices de textes
- <https://www.sound-fishing.net> et <https://www.musicscreen.be> pour le sons
- Unity 2018.2.13f1

Personne:

Je remercie ma famille et mes amis d'avoir tester le jeu pour me donner leurs impressions. Je remercie aussi Elias Farhan, Yann Piller et les étudiants Games Programming seconde année pour m'avoir donné de nombreux conseils sur le développement.