Rapport du TP de réalité virtuelle ZZ2 F1

BOISSET Florian

MARCHAND Elisa

# Rappel du sujet :

Durée : 6h (3x 2h)

En binômes, vous devrez créer un jeu (ou un logiciel) qui permette à un utilisateur de suivre des étapes pour créer ou modifier quelque chose en appliquant les notions vues en cours. Un rapport de TP devra accompagner le rendu final.

Notions à appliquer :

* Collider
* Raycast
* Rigidbody (physique, collision et trigger)
* Script
* Prefab (utilisation et instanciation)
* Asset 3D
* UI

Bonus :

* Animation
* Particules et effets spéciaux (ex : post process)

# Choix du projet et comment ça marche :

**ATTENTION** : notre projet à comme version : **2022.3.13f1**

Pour notre projet, nous avons décidé de suivre l’exemple proposé lors de la présentation du TP. Nous avons donc créé un jeu de fabrique de burger. Nous nous sommes aussi inspirés du jeu Overcooked pour ce qui est du timer et de la prise de commande.

Le but de notre jeu est de faire le plus de commande dans un temps imparti. Pour cela, le joueur doit sélectionner les aliments dans un ordre définis puis apporter le burger à la table qui à la fiche associée au plat. Chaque table est indépendante, elle a sa propre recette, son propre timer et son propre score.

Le joueur se déplace avec les flèches directionnelles et avec la souris, il peut régler l’angle de la caméra. Pour récupérer les composants du burger, le joueur doit se placer en dessous du tube dans le faisceau rouge et faire un click souri. Pour finir la prise de la commande, le joueur doit apporter le plat à la bonne table puis appuyer sur la touche « e » pour valider le rendu.

Pour ce projet, nous avons dans un premier temps décidé de faire les attendus obligatoires. Puis au fils de l’avancer du projet, nous avons complexifié le projet en rajoutant les bonus du TP, un timer et plusieurs commandes à gérer en même temps.

Nous avons réalisé notre projet à l’aide d’un github afin de pouvoir travailler facilement et efficacement sur plusieurs machines à la fois. Nous avions chacun notre branche sur le projet nous l’envoyions sur le master lorsque que l’objectif en cours était atteint.

Rendu Final :

Une image contenant capture d’écran, ciel, Jeu PC, Logiciel de jeu vidéo

Description générée automatiquement -> affichage des recettes

Une image contenant capture d’écran, bâtiment, sol, art

Description générée automatiquement -> zone de fabrique Une image contenant capture d’écran, chaise, meubles

Description générée automatiquement -> rendu du burgerUne image contenant capture d’écran, Jeu PC, Logiciel de jeu vidéo, Modélisation 3D

Description générée automatiquement -> fenêtre de fin de partieUne image contenant bâtiment, capture d’écran, brique

Description générée automatiquement -> recupération d’aliment

# Partie Florian :

Mon objectif premier à été de mettre en place la façon dont le jeu allait fonctionner, chose qui s’est très rapidement faite : des raycasts allaient détecter le burger en cours et allaient placer l’ingrédient correspondant dessus. Une fois le burger finit, il n’y a plus qu’a le servir à la table.

Implémentation :

* Les raycasts ont rapidement été mit en place sans soucis.
* Le positionnement des ingrédients les uns sur les autres a posé problème :
  + Les ingrédients se mettaient les uns dans les autres
    - Solution : ajout d’un collider box aux prefabs des ingrédients.
  + Ils ne s’empilaient pas en leur centre => la position était donnée dans le repère global
    - Solution : passage au repère du parent.
  + Deux ingrédients pouvaient parfois se retrouver à la même position hiérarchique (avec le même parent) et donc ils se retrouvaient l’un dans l’autre
    - Solution : limite sur le nombre de fils.
  + En somme : Chaque ingrédient devient le fils de l’ingrédient en dessous et sa position est celle de son parent + la taille de son collider sur l’axe vertical.
* Mise en place des tables (création de préfabs avec le prefab de la table et de la chaise)
* Collider box pour la zone de rendu + appuie sur une touche.
* Ajout de TextMeshPro pour afficher la commande désirée pour chaque table + mise à jour de la commande régulièrement avec un Timer aléatoire en fonction du nombre d’ingrédient commandés.
* Ajout des particules autour des raycasts.

# Partie Elisa :

Au tout début du projet, j’ai mis en place la vérification du burger. Pour cela, j’ai dû mettre en place une fonction qui permet de savoir le nombre d’élément qui compose le plat et une recette prédéfinie pour pouvoir vérifié que cela fonctionne correctement. Puis, j’ai créé une énumération avec tous les aliments disponibles pour pouvoir créer les recettes de manière aléatoire avec un Random. La vérification est lancer lorsque le burger est donné au client (collider en triger avec touche enfoncé) après cela, l’objet est détruit et une nouvelle fiche recette est affiché. Chaque table à son propre score calculé lors de la vérification.

J’ai implémenté la fin du jeu en m’inspirant du travail de Florian, au bout d’un certain temps le jeu s’arrête grâce à un timer. Le score final apparait à l’écran avec deux boutons (un pour rejouer et un pour quitter le jeu). Le score final est les sommes du score de chaque table.

Je me suis aussi occupée de l’affichage des indications pour le joueur, par exemple, quand le joueur est à côté d’une table (dans un collider en trigger) la phrase « touche e pour donner le burger » apparait à l’écran.

Pour rendre la scène plus vivante, j’ai mis en place des PNJ avec une animation. Le PNJ est un préfab importer de Unity Asset Store.

# Conclusion :

Nous avons eu quelques difficultés pour le système de vérification qui n’est toujours pas parfait. Finalement, après avoir essayé avec différents types de données et fait un appel à l’aide à des ZZ3 F1, nous n’avons pas pu corriger ce problème qui mène à un crash du jeu sur beaucoup d’exécutions.

Le reste du projet c’est bien déroulé.

Pour améliorer notre projet nous pourrions : améliorer le système de vérification, rajouter du bruitage ou des sons, rajouter des niveaux plus simple ou plus complexe.