

TP Réalité Immersive

Simulation d'emballage de peluches dans une usine

Contenu de l'archive :

- Les dossiers Packages, ProjectSettings, et Assets du projet Unity, contenant le projet lui-même avec tout le code commenté
- Ce fichier README
- Un dossier build contenant un exécutable du projet

Contexte :

Nous sommes dans la nouvelle usine construite par Bernard Lhermite pour trier et emballer les produits (des peluches) qu'il vendra à Noël dans sa nouvelle boutique en ligne permettant d'offrir aux autres des cadeaux pré-emballés.

La simulation doit servir à entraîner les opérateurs à effectuer la série de tâches répétitives pour emballer et trier les peluches.

Description de la scène :

Vous démarrez en tant qu'opérateur dans le SAS de sécurité, devant la porte de l'atelier d'emballage. En ouvrant la porte, vous découvrez 3 bacs marqués d'une couleur différente, un distributeur à objets, et un tapis roulant menant à une machine d'emballage automatique.

Distributeur à objets :

Un gros **bouton rouge** permet de faire sortir une par une les peluches de ce distributeur.

La peluche que vous recevez est aléatoire, parmi **6 possibilités** : ours, lapin, singe, pingouin, cochon, et mouton.

Machine d'emballage :

Elle emballe automatiquement les peluches qui arrivent jusqu'à elle via le tapis roulant. Le **paquet cadeau** peut alors être récupéré dans le trou à l'extrémité de la machine.

Bacs de couleur :

Il y a 3 bacs de couleur différente, chacune correspondant à un type de peluche. Ils doivent réceptionner les peluches **emballées**. Attention à bien trier les paquets cadeau dans le bac correspondant à la peluche emballée, sinon l'alarme sonnera et vous serez pénalisé.

Notons qu'il y a 6 types de peluches pour 3 bacs, c'est parce que Bernard Lhermite a décidé au dernier moment de ne pas commercialiser 3 de ces 6 types de peluches, mais elles sont toujours acheminées à l'usine (malgré ses qualités de manager, Bernard Lhermite n'est pas très bon en logistique...).

À vous de trouver les **intrus** (ce n'est pas bien compliqué...), et de vous en débarrasser en les jetant à la **poubelle** situés à gauche du distributeur.

Description technique :

En partant des fonctionnalités demandées dans le TP de base, nous avons donc décidé de créer une expérience complète de type serious game.

Les interactions de base présentes dans ce travail sont : déplacement du personnage, téléportation du personnage, génération aléatoire d'objets et saisie d'objets, tri d'objets dans des réceptacles, affichage d'un compteur sur un tableau d'affichage.

Nous avons aussi développé d'autres fonctionnalités : le tapis roulant, la machine d'emballage, une porte coulissante, un système d'alarme, et un vrai système de jeu avec temps limité et écran de fin de jeu. Le tout dans un environnement un minimum cohérent, pour que ce soit plus crédible. La plupart des assets utilisés pour le décor proviennent de l'Asset Store.

Tapis roulant :

Il fait avancer les objets posés dessus. Pour cela, nous appliquons au rigidbody du tapis roulant deux transformations inverses qui, cumulées, déplacent les objets au-dessus, sans déplacer le rigidbody du tapis lui-même.

L'animation du tapis est réalisée en décalant l'offset de la texture à chaque frame.

Machine d'emballage :

Elle contient un trigger qui, lorsque déclenché par un gameObject (peluche), transforme celui-ci en un autre objet (paquet cadeau). Le tag de l'objet est modifié pour indiquer qu'il est emballé.

Les réceptacles, de leur côté, vérifient que l'objet posé dessus est emballé.

Porte coulissante :

L'ouverture / fermeture de la porte est effectuée sur un temps donné (en secondes) grâce au système de **coroutine**.

Alarme :

Lorsque l'on **pose sur un réceptacle un objet qui n'est pas destiné à aller sur ce réceptacle**, une alarme se déclenche, ainsi qu'une animation de jeux de lumière dans la scène.

Game manager :

Le tout repose sur un script GameManager associé à un objet arbitraire dans la scène. Il gère quelques variables globales et fonctionnalités, et surtout gère le temps de jeu et les points.

Lorsque le temps est écoulé, un **changement de scène** est effectué, le joueur se retrouve alors dans la scène de fin de jeu, où il peut voir son score final s'afficher, et quitter l'application via un bouton.

Le jeu est aussi **interrompu si le joueur déclenche 10 fois l'alarme**, et la scène finale affiche qu'il a échoué.

Pistes d'amélioration :

Une amélioration possible serait d'ajouter plus d'effets sonores à la simulation lors des interactions.

De plus, les effets sonores et les animations liés aux coroutines ont un problème dans la build. Tout est plus lent que dans l'inspecteur Unity, les durées définies en secondes ne semblent donc pas respectées. Nous n'avons pas encore trouvé la raison de ce problème.

Enfin, il faudrait ajouter un menu avant de commencer la simulation, et la possibilité de recommencer après avoir terminé une partie.