



RV01 - Réalité Virtuelle - A20

Rapport de fin de projet

DodgeBall VR

Participants : Robin ROSSEEUW, Maxime GRANDIDIER

Superviseurs : Indira THOUVENIN, Baptiste WOJTKOWSKI,
Luca PELISSERO-WITOSLAWSKI

I - Sujet

1. Choix du sujet

Dans le cadre de ce cours sur la réalité virtuelle, nous avons choisi de réaliser un projet avec un principe simple qui allait nous permettre d'explorer les techniques pour créer des expériences de réalité virtuelle sans nous heurter à des difficultés chronophages qui n'auraient pas ajouté à notre formation. C'est pour cela que nous avons opté pour un projet avec une balle comme élément central de l'interaction : une simulation de balle au prisonnier à 360 degrés. Grâce à ce choix, nous avons évité des problèmes de modélisation car il est très simple de modéliser une balle et de lui associer une physique dans le moteur Unity, l'enjeu était donc de pousser au maximum cette interaction pour la rendre réaliste.

L'idée nous est venue en cherchant la simplicité : nous avons exploré des idées de jeux simples auxquels nous avons pu jouer dans notre vie. À partir de cette idée, nous avons exploré rapidement ce que nous pourrions faire sur le principe de la balle au prisonnier. Le principe de la balle au prisonnier demande aux joueurs d'être attentifs à tout leur entourage pour esquiver des balles provenant de face ou bien de derrière eux. Il nous a paru pertinent de partir de ce principe mais de l'adapter à un jeu seul contre tous où le joueur est encerclé par ses adversaires.

Notre but était de créer une expérience intense dans laquelle le joueur doit rester à l'affût et en mouvement pour ne pas se faire toucher et réussir à attraper puis renvoyer des balles sur ses adversaires afin de remporter la victoire. Pour que l'immersion soit bonne et permette au joueur de faire l'expérience de ces sensations, il était essentiel de gérer la physique des lancers de façon à ce que le joueur aie l'impression de vraiment lancer et esquiver des balles. Nous expliquerons cet aspect en profondeur dans la suite de ce rapport. Nous avons d'ailleurs préféré nous concentrer sur les explications des différentes interactions et fonctionnalités plutôt que d'ajouter des visuels à notre rapport, étant donné la limite de page, gardant donc les illustrations pour la vidéo de présentation jointe à ce rapport.

2. Principe

Comme nous l'avons expliqué dans la présentation du sujet, le jeu est basé sur un principe très simple. Au début d'une partie, le joueur se retrouve au centre d'un gymnase, encerclé par un nombre de robots variable selon la difficulté.

Ces robots vont, à tour de rôle, tirer des balles sur le joueur. Ce dernier peut esquiver ou attraper les balles avec ses mains. S'il réussit à attraper une balle, il peut alors la lancer sur l'un des robots. Si le joueur réussit son lancer et que le robot est touché, alors celui-ci est éliminé pour le restant de la partie.

Le joueur remporte la partie lorsqu'il ne reste plus aucun robot debout. Cependant, le joueur peut perdre la partie s'il se fait toucher trop de fois par des balles lancées par ses adversaires.

Les robots lancent parfois des balles enflammées sur le joueur. Ces balles sont spéciales car elles ne peuvent être attrapées et doivent impérativement être esquivées. Si le joueur tentait de les attraper, il perdrait à la place une vie.

Le jeu repose donc sur l'adresse du joueur ainsi que sa capacité à être conscient des adversaires hors de son champ de vision.

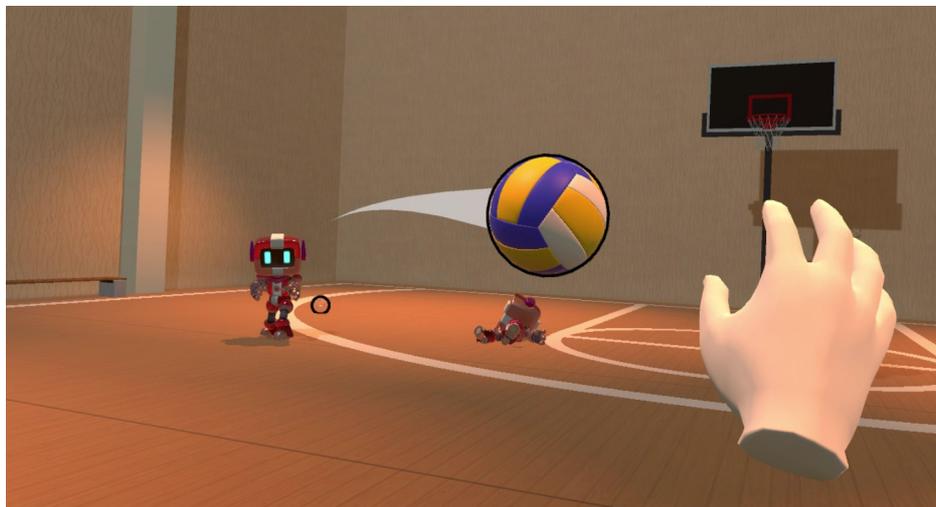


Illustration du jeu DodgeBall VR

3. Interactions

Pour parler des interactions possibles entre le joueur et le monde dans lequel il évolue, il est important de définir comment le joueur est incarné dans ce monde. Dans notre cas, il s'agit simplement d'une paire de mains que le joueur contrôle avec ses manettes, ainsi qu'une tête qui suit les mouvements du casque porté par le joueur dans le monde réel. Le joueur n'a ainsi pas de corps complet et ne peut voir que ses mains ainsi que son champ de vision. Au niveau de la représentation du corps dans la logique du jeu, il s'agit simplement d'un collider en forme de capsule qui va permettre de détecter si le joueur se fait toucher par les balles qui lui sont lancées dessus. Il nous est maintenant beaucoup plus aisé de définir les différentes interactions possibles pour le joueur.

Attraper une balle est la première interaction. Il s'agit pour le joueur d'appuyer sur la gâchette de sa manette au moment où une balle est suffisamment proche de sa main virtuelle. Dans le jeu, cet intervalle de temps est représenté visuellement : quand elle peut être saisie, une balle a un contour qui s'illumine en jaune. Lorsque le joueur saisit une balle, il ressent également la manette correspondante qui vibre pour remplacer la sensation de toucher et donner une confirmation tactile de la réussite de son action. Cette paire de retours artificiels (le contour qui s'illumine et la vibration) compensent l'absence des sens accessibles dans notre réalité physique. Une fois attrapée, la balle reste accrochée à la main tant que le joueur maintient la gâchette enfoncée.

Une fois une balle attrapée, le joueur peut la **lancer ou la lâcher**. Ces actions sont réalisées en lâchant la gâchette utilisée précédemment pour attraper la balle, pour remplacer le mouvement d'une main qui s'ouvre. Il y a cependant deux possibilités : le joueur peut simplement laisser la balle tomber en ne faisant pas de geste rapide. Dans ce cas, la balle se retrouve affectée par la gravité du monde virtuel et va tomber d'une façon très similaire à ce à quoi on peut s'attendre dans la réalité. La seconde possibilité est celle qui nous intéresse le plus, celle où le joueur va faire un geste de lancer, qui sera donc plus rapide et violent. Dans ce cas, au moment où la balle est lâchée par le joueur, elle sera dotée de la vélocité du mouvement du joueur, mais également d'un ajout artificiel pour la faire partir plus vite. Cet ajout permet de combler l'écart entre le virtuel et la réalité. En effet, sans celui-ci, on se retrouvait avec des lancer très faibles malgré des gestes très violents et puissants. Cela causait une dissonance cognitive importante pour les joueurs, brisant l'immersion. Nous avons donc fait cet ajout à force d'essais et d'erreurs pour arriver à un résultat crédible pour le joueur.

Ce qui fait la différence entre un lancer et un lâcher est la vélocité de la manette au moment où la gâchette est relâchée par le joueur. En dessous d'un certain seuil, la balle est lâchée, et au-dessus de celui-ci, elle est lancée avec cette force supplémentaire.

En plus de pouvoir attraper les balles, le joueur peut également les **esquiver**. C'est même nécessaire si la balle qui lui est lancée est enflammée ! L'esquive est une interaction très simple. Comme nous l'avons mentionné précédemment, le collider du joueur est basé sur la position de sa tête, pour esquiver une balle, il lui suffit donc de déplacer sa tête hors de la trajectoire de la balle.

Dans le cas où le joueur échoue à esquiver une balle et se fait toucher, il ressent alors ses manettes vibrer, sa vision devient rouge pendant une fraction de seconde et il entend un son de douleur provenant de son personnage en plus d'entendre le bruit de la balle qui entre en collision avec lui.

4. Autres éléments d'immersion

Afin de compléter l'immersion du joueur, nous avons ajouté d'autres éléments qui nous paraissaient pertinents. En premier, nous avons découpé les lanceurs des robots en plusieurs phases. Sans cela, les robots lanceraient instantanément des balles sur le joueur, sans lui donner la moindre indication que cela va se produire et rendant toute réaction très difficile. En ajoutant à cela le fait que le joueur ne voit pas certains lanceurs, il fallait absolument implémenter un ou plusieurs systèmes permettant au joueur d'avoir des points de repère dans l'univers virtuel ainsi que du temps pour se préparer. Nous avons choisi d'utiliser des indices visuels et sonores pour résoudre ce problème. D'abord, les robots produisent un son lorsqu'ils s'appêtent à lancer et une balle, avertissant le joueur qu'une balle va lui être tirée dessus. En vérité, ce son est spatialisé, ce qui indique également au joueur la direction dans laquelle se tourner pour voir le robot qui va lancer la balle. Enfin, le robot a un délai entre ce son et le début de son animation de lancer, pour laisser au joueur le temps d'intégrer l'information sonore et d'agir. Le robot a ensuite une petite animation pour projeter la balle.

Enfin, nous avons voulu compléter l'immersion du joueur au niveau sonore. Pour cela nous avons décidé d'ajouter de la musique ainsi que des bruitages personnalisés. Pour avoir des musiques et sons remplissant notre objectif, nous avons fait appel au studio Long Night Records. Nous avons collaboré avec eux afin qu'ils produisent une musique correspondant au thème et à l'énergie que nous voulions insuffler au jeu. Ils nous ont également produit des éléments de sound design qui ont élevé la qualité des bruitages de notre jeu, renforçant grandement l'immersion du joueur.

II - Développement

1. Outils utilisés

Pour organiser le développement de notre projet à deux, nous avons majoritairement utilisé l'application Discord comme moyen de communication et de réunion. Les fonctionnalités de partage d'écran nous permettant de collaborer en temps réel sur le projet lorsque nous en avons eu besoin. La répartition des tâches a été réalisée avec un diagramme de Gants que vous trouverez en Annexe 1 ce rapport. Nous avons également procédé à une répartition plus fine des tâches élémentaires avec un fichier excel de suivi de projet. Le moteur de jeu que nous avons utilisé pour développer Dodgeball VR est Unity. Notre IDE de scripting est Microsoft Visual Studio Community. Pour que notre collaboration se passe au mieux, nous avons choisi d'utiliser Unity Collab, un plugin de versioning disponible sur Unity et lié à Git. Cela nous a permis de travailler de façon très aisée sans rencontrer de conflits liés à des ajouts simultanés. Enfin, pour tester le jeu, nous avons pu utiliser le casque de réalité virtuelle personnel de Robin : un Oculus Quest 2.

2. Fonctionnalités

Pour chaque fonctionnalité que nous allons présenter, il est important de prendre en compte que toutes les interactions en réalité virtuelle ont été codées pour le projet. Nous n'avons pas utilisé de bibliothèque existante. Nous avons réalisé tous nos scripts.

- Le jeu commence par un menu navigable permettant de lancer une partie ou d'accéder aux options. Les options permettent de sélectionner la difficulté de la partie. Cela va changer les paramètres de jeu suivant : le nombre de robots, la vitesse de leurs lancers et la fréquence d'apparition des balles enflammées. Les choix sont présentés sous la forme de balles statiques avec un label flottant en dessous. Pour les sélectionner, le joueur doit les attraper de la même façon que les balles qui lui sont lancées dessus.
- Le joueur peut également mettre pause sa partie en appuyant sur un bouton, stoppant toutes les interactions en cours sauf le déplacement de sa tête et de ses mains pour ne pas lui faire ressentir de l'inconfort lié à un décalage sensorimoteur.
- Comme nous l'avons expliqué précédemment, le joueur est incarné seulement par ses mains, visibles et animées, et sa tête, invisible mais disposant d'un collider en forme de capsule.
- Pour animer les mains de l'utilisateur, nous avons récupéré sur internet un modèle 3D de main disposant d'un squelette, que nous avons ensuite animé via les outils d'animations intégrés à Unity. Il nous a donc ensuite suffi de lier cette animation à l'appui des gâchettes.

- Quand le joueur bouge sa tête, le collider va donc bouger avec, ce qui lui permet d'esquiver les balles lancées.
- Pour attraper les balles, nous avons défini une zone cubique autour du prefab de balle qui est un trigger, tout en donnant un tag à la balle. Nous avons ensuite géré la détection de collision avec ce trigger et ce tag au niveau des mains du joueur, lorsqu'il appuie sur la gâchette. Si les conditions sont remplies, nous désactivons la gravité de la balle et la faisons suivre la main du joueur à toutes les frames, grâce à une mise à jour de son Transform dans la méthode Update.
- Pour qu'un joueur perde une vie, il faut qu'il y ait collision entre le collider d'une balle lancée par un adversaire et celui attaché à la caméra, impossible de perdre de vie à cause d'une balle déjà attrapée ou renvoyée car nous détectons la différence en modifiant un booléen.
- Comme nous l'avons dit, une fois attrapée, la balle suit le mouvement de la manette du joueur tant qu'il garde la gâchette enfoncée. Tant que la balle est tenue, la vitesse de la manette sur les dernières frames est sauvegardée parce qu'au moment de lancer la balle, en lâchant la gâchette, une moyenne des vitesses angulaires et linéaires est faite avant de transférer cette vitesse à la balle. Cela évite les problèmes où la gâchette est relâchée un peu après la fin du mouvement, ce qui fait partir la balle dans une direction non désirée.
- Les adversaires du joueur sont des robots animés dont nous avons pris le squelette ainsi que le modèle sur une source libre de droit. Nous avons également ajouté quelques animations à celles existantes. Au commencement d'une partie, les robots sont créés selon les paramètres de la partie. Ils apparaissent dans un cercle de rayon fixe autour du joueur, à une position aléatoire sur la périphérie du cercle. Durant l'initialisation, les valeurs aléatoires sont arrondies et gardées en mémoire pour éviter la superposition de robots, une condition détecte cela. Chaque robot est également source de ses propres bruitages.
- La procédure de lancer d'une balle pour un robot est en plusieurs étapes:
 - Le robot se tourne vers le joueur.
 - La balle est instanciée dans la main du robot, et ce dernier joue une animation ainsi qu'un son spatialisé pour que le joueur le repère.
 - Le type de lancer (en cloche ou droit) est déterminé aléatoirement.
 - La direction, le décalage par rapport au joueur et la force sont passées à la balle.
 - Le robot attend ensuite pour donner au joueur le temps de réagir.
 - La balle est lancée puis le robot efface toutes ses références à la balle.
 - Les variables aléatoires du prochain tir sont ensuite déterminées.
- Le tableau des scores compte le nombre de robots toujours en vie. Il représente également la vie du joueur par des cœurs qui s'éteignent lorsque le joueur se fait toucher par une balle. Le tableau sert également à afficher le temps passé dans une partie. Il est donc réinitialisé à chaque début de partie.

- Pour modéliser le gymnase nous avons simplement mis 6 plans pour former une boîte rectangulaire. Nous avons ensuite rajouté des cages de handball ainsi que des paniers de basket récupérés sur internet. Il nous a enfin suffi de rajouter, grâce à un matériel, les marquages au sol, des colonnes qui donnent l'impression de soutenir le tout et un éclairage de plafond coloré pour avoir notre rendu final.
- Lorsqu'un joueur est touché, un son est joué selon le type de balle, normale ou enflammée. Il y a également la vibration des manettes pour donner un retour sensoriel supplémentaire et enfin l'écran prend une teinte rouge pour compléter la sensation.
- [VARIATIONS DE GAMEPLAY À COMPLÉTER QUAND IMPLÉMENTÉ]
- Les différents bruitages du jeu ont été fournis par le studio Long Night Records et implémentés dans le jeu. Ils ont également composé plusieurs musiques. Une pour le menu, une qui se lance au début d'une partie et une pour chaque fin de partie (victoire/défaite).

3. Mécanique de lancer

Nous souhaitons maintenant revenir sur ce qui compose le cœur de notre jeu : le lancer du joueur. En effet, c'est l'interaction centrale sur laquelle tout repose. Si la sensation procurée par le lancer n'est pas satisfaisante et en adéquation avec les attentes du joueur, l'immersion sera brisée et l'expérience un échec. Pour cela, nous avons passé le plus de temps à rendre cette action aussi crédible que possible. Nous avons également associé nos idées et réflexions personnelles à des recherches et sources extérieures, principalement cet article publié en ligne traitant précisément de notre sujet : *How to Improve Throwing Physics in VR*¹.

_____ Nous avons obtenu notre meilleur résultat après avoir implémenté l'utilisation de la vitesse angulaire en plus de la vitesse linéaire pour se rapprocher autant que possible de la physique d'un véritable lancer ainsi que la protection contre les erreurs de dernier instant du joueur en gardant en mémoire les vitesses prises par la manette. Mais nous avons également rajouté une dichotomie dans les lancers : Nous avons observé que lorsque nous faisons un geste à pleine puissance, le lancer ne suivait pas et paraissait faible. Alors nous avons résolu ce problème via une astuce. Lorsque la vitesse de la manette dépasse un certain seuil, alors, en plus de transférer cette vitesse à la balle au moment du lancer, une force supplémentaire qui multiplie cette vitesse est ajoutée. Cela nous permet de créer un lancer avec une véritable sensation de puissance. Il nous a fallu cependant essayer plusieurs valeurs de seuil ainsi que de force pour ne pas créer de nouvelles dissonances cognitives avec cet ajout, le rendu final étant pleinement convaincant et immersif.

¹ Lewis, K. *How to Improve Throwing Physics in VR*. url: <https://karllewisdesign.com/how-to-improve-throwing-physics-in-vr/> (visité le 20/11/2020)

Conclusion

Bien que nous n'ayons pas eu le temps d'implémenter les dernières fonctionnalités initialement prévues pour le niveau 3 de réalisation de notre projet, nous considérons que c'est une réussite. En effet les fonctionnalités manquantes sont peu nombreuses et n'enlèvent rien à l'expérience principale. Il s'agissait principalement de variations de gameplay pour rendre l'expérience plus dynamique, comme l'ajout de nouveaux modes de jeu ou encore de super pouvoirs et autres éléments perturbateurs.

En somme, nous avons réussi à créer une expérience immersive grâce à nos choix de modélisation, nos ajouts sonores comme les bruitages ou bien la musique. Mais surtout, l'interaction qui fait le cœur du jeu est satisfaisante et crédible. Tenir, manipuler, lancer et attraper une balle sont toutes des actions qui sont faisables facilement et sans créer de dissonance cognitive, rendant accessible notre jeu à un large public. Cela est possible grâce aux retours haptiques donnés par les manettes, mais également grâce aux métaphores visuelles telles que les contours de balle illuminés. Enfin, le lancer est réactif, il semble aussi puissant que le mouvement qui lui est donné et précis. C'est l'interaction qui aurait pu briser toute immersion si l'effet ne correspondait pas à ce que l'utilisateur se représente, mais à travers nos recherches et notre calibrage, nous avons évité cet écueil. De plus, nos tests réalisés auprès de proches dont de vrais joueurs de balle au prisonnier nous ont prouvé que l'immersion était réussie et proche de la réalité.

Pour finir, nous pensons que ce concept ne doit pas s'arrêter avec ce que nous avons eu le temps de produire, et que nous pouvons le pousser plus loin. Comme nous l'avions initialement envisagé, nous pouvons introduire de nouveaux modes de jeu, comme un mode survie où les robots réapparaissent à l'infini par exemple où le but serait de survivre le plus longtemps pour comparer nos scores, un mode multijoueur où deux joueurs pourraient jouer en coopération ou en s'affrontant, ou bien rajouter des nouveaux effets environnementaux venant modifier les lois de la physiques au sein du jeu.

Bibliographie

1. Cours de RV01 - Indira THOUVENIN, Université de Technologie de Compiègne
2. La réalité virtuelle démystifiée - Indira THOUVENIN, Romain LELONG
3. Lewis, K. *How to Improve Throwing Physics in VR*. url:
<https://karllewisdesign.com/how-to-improve-throwing-physics-in-vr/>

Annexe 1 : Diagramme de Gants

Semaines	21 sept.	12 oct.	19 oct.	26 oct.	2 nov.	9 nov.	16 nov.	23 nov.	30 nov.	7 déc.	14 déc.	21 déc.	4 jan.
Définition du cahier des charges													
Création de la scène et des éléments principaux dans Unity (personnage principal, balle, cibles)													
Développement du lancer et de l'attrapage de balle													
Développement des lancers des entités externes													
Ajout des conditions de victoire/défaite et des collisions balle/joueur, balle/ennemi													
Ajout du menu principal + pause													
Ajout des éléments visuels/sonores													
Ajout de variations de gameplay													
Peaufinage et résolutions des derniers bugs													
Rédaction du rapport et présentation finale du projet													