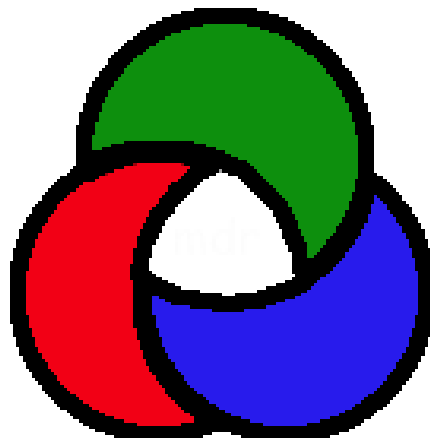


# Projet HMIN320

## *“Les Couleurs”*



[Code source du jeu](#)

[Video trailer du jeu](#)

[Diapo du projet](#)

[Exécutable du projet](#)

# Sommaire

<b>Notre projet</b>	<b>2</b>
<b>Mode d'emploi</b>	<b>3</b>
<b>Rapport avec le cours</b>	<b>5</b>
<b>Notre fierté</b>	<b>6</b>
<b>Les difficultés</b>	<b>6</b>
<b>Les références</b>	<b>8</b>
<b>Outils/Librairies externes</b>	<b>9</b>

# Notre projet

Pour notre projet, nous avons décidé de profiter de l'accès au matériel de réalité virtuelle fourni dans le cadre de cette matière pour développer un jeu demandant des réflexions de game design et de programmation très différentes de ce que l'on a l'habitude de faire avec une manette ou un combo clavier et souris.

Dans notre jeu, le joueur peut lancer des projectiles de couleurs qui ont différents effets. Il devra faire preuve de dextérité et de repérage visuel pour trouver les différents objets avec lesquels il doit interagir pour avancer dans le jeu. Le but du jeu est d'arriver à monter tout en haut du pilier.



*Le pilier à escalader*

*Les Couleurs* est un jeu adaptant le principe des jeux de plateforme à la VR. Pour régler le problème de motion sickness propre aux déplacements dans un jeu en réalité virtuelle, nous avons remplacé les traditionnels mouvements par une téléportation. En effet, le joueur se retrouve téléporté à l'emplacement des projectiles de couleur verte qu'il lance.

Par ce procédé, la réactivité demandée par les jeux de plateformes traditionnels est remplacée par une demande de dextérité : le joueur doit doser ses mouvements pour viser les zones qu'il souhaite atteindre.

Nous avons voulu pousser ce principe en ajoutant des éléments de couleurs (les ballons et les prismes) que le joueur doit viser pour faire apparaître des plateformes nécessaires à sa progression.

# Mode d'emploi

Pour jouer à notre jeu, il suffit de brancher un casque HTC Vive à l'ordinateur, de lancer SteamVR puis de lancer l'exécutable *Les Couleurs.exe*. Le lien vers le téléchargement du jeu est disponible [ici](#).

Le joueur va pouvoir interagir avec deux types d'objets pour progresser : les ballons et les prismes.



*Ballons*



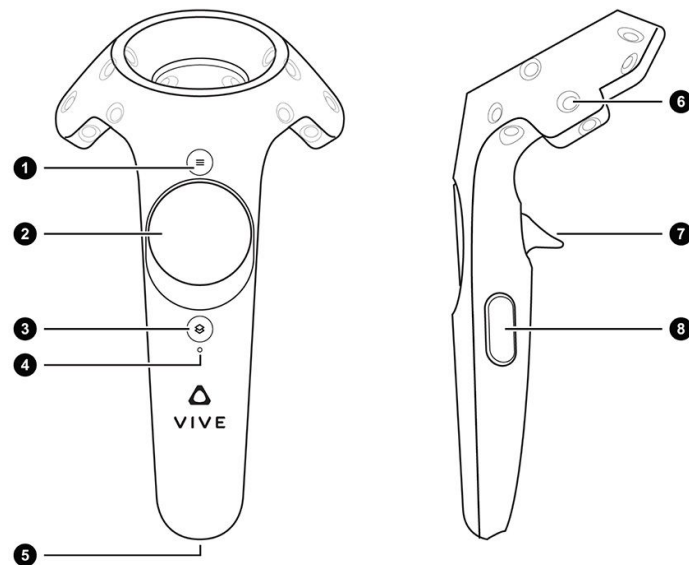
*Prisme*

L'utilisateur devra lancer des projectiles de la même couleur que les ballons pour les détruire, ce qui fera apparaître une plateforme nécessaire à sa progression. Dans le cas où il envoie un projectile de la mauvaise couleur, le projectile rebondira simplement sur le ballon.

Les prismes, quand à eux changent de couleur quand on les touche avec un projectile. Lorsqu'ils sont rouges, ils font apparaître des plateformes rouges et font disparaître les plateformes bleues. Dans le cas où ils sont bleus, ils font apparaître les plateformes bleues et disparaître les plateformes rouges. Le joueur peut se téléporter sur ces plateformes.

Pour éviter la frustration du joueur, nous avons limité les téléportations à certains sols. De plus, si une orbe touche un sol qui est trop loin du joueur, la téléportation n'aura pas lieu afin d'éviter que le joueur rate un lancer, que le projectile tombe tout en bas du pilier et qu'il doive recommencer l'ascension depuis le début.

Pour lancer des projectiles, le joueur doit utiliser les manettes du HTC Vive.

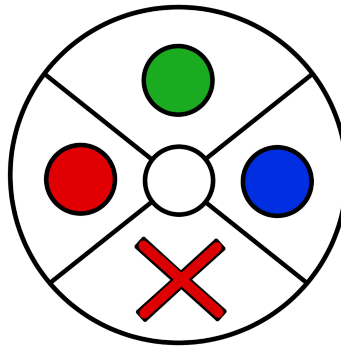


*Schéma manette HTC Vive (crédit : [Manuel Unity](#))*

Pour cela, il doit rester appuyé sur la touche 7 de la manette (Trigger), faire un mouvement de bras et relâcher le bouton 7 pendant le mouvement. Si la touche 7 n'est pas maintenue assez longtemps, le projectile sera détruit avant d'être envoyé pour obliger les joueurs à viser plutôt que de tirer très rapidement des projectiles en espérant toucher leur objectif.

Le joueur peut ensuite choisir le projectile à lancer grâce au bouton 2 (Trackpad). En posant simplement son doigt sur le trackpad, sans enfoncer le bouton. Le joueur va changer le type de projectile qu'il va pouvoir lancer en fonction de la position de son doigt. Il n'est pas obligé de relâcher la pression avec le bouton 2 pour choisir une nouvelle couleur : il lui suffit de faire glisser son doigt sur une nouvelle zone. Une fois la couleur choisie, il peut relâcher la pression sur le trackpad et tous les projectiles créés par cette manette seront des projectiles de la couleur choisie. Les deux manettes sont indépendantes ; il est donc possible de sélectionner un projectile avec sa main droite et un autre type de projectile avec sa main gauche..

Comme l'indique notre interface utilisateur, appuyer vers la gauche du bouton 2 permettra d'utiliser des projectiles de couleur rouge, permettant d'interagir avec les ballons de couleur rouge et les prismes, appuyer vers la droite permet de lancer des projectiles bleus, permettant d'interagir avec les ballons bleus et les prismes et appuyer vers le haut permet de préparer des projectiles verts servant à se téléporter. Enfin, si l'utilisateur appuie sur le bas du bouton 2, l'orbe qu'il était en train de charger se verra détruite sans qu'il n'ait besoin de la lancer, permettant ainsi de rapidement changer la couleur de ses projectiles en cas d'erreur.



*Notre interface utilisateur, enfante des manettes*

Pour permettre aux droitiers comme aux gauchers de profiter de notre jeu, les manettes de la main droite et de la main gauche fonctionnent de la même manière.

## Rapport avec le cours

Le projet a été développé sous Unity 3D (comme les travaux pratiques donnés par Mme Nancy Rodriguez), nous permettant à tous les 3 de découvrir le développement de jeux en Réalité Virtuelle et plus particulièrement l'utilisation de librairies comme "[VRTK](#)" ou "[Steam VR](#)".

Nous avons aussi pu prendre en main l'HTC Vive, comprendre comment fonctionnent les contrôles pour pouvoir choisir les plus intuitifs et les plus faciles d'accès, et s'en servir dans le jeu.

## Notre fierté

Tout d'abord, ce projet a été l'occasion d'expérimenter un matériel nouveau avec lequel les normes de game design telles que nous les connaissons sont à redéfinir sur bien des points. Nous sommes fier d'avoir pu créer un jeu sous ce support qui a été aussi apprécié par les personnes qui l'ont testé lors de sa dernière semaine de développement.

Nous sommes heureux d'avoir pu apprendre à utiliser l'outil ProBuilder d'Unity qui nous a permis de réaliser la très grande majorité des décors de notre jeu, en plus du monde en lui même.

Ce projet fût notre premier où nous étions directement en contact avec des joueurs, ce qui nous a permis de prendre en compte leurs retours et ainsi améliorer notre jeu. Nous sommes fier du chemin qu'a parcouru notre jeu grâce à l'écoute des critiques de nos joueurs.

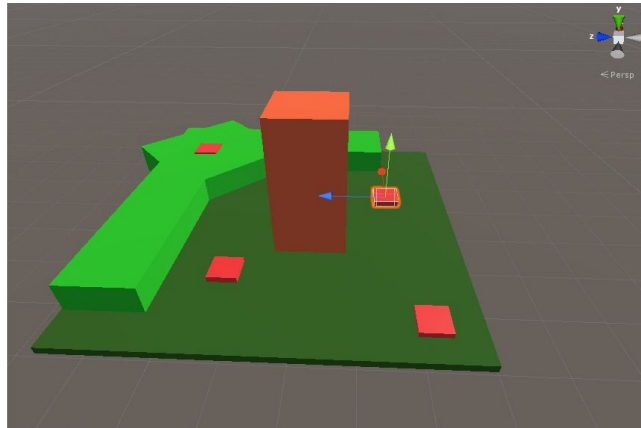
## Les difficultés

Nous avons rencontré plusieurs difficultés liés à notre inexpérience des jeux en réalité virtuelle. Notre premier problème s'est posé dès la conception du jeu et concernait la motion sickness.

En effet, nous étions au courant que l'utilisation de touches pour déplacer un avatar, à la manière d'un joystick dans un jeu classique, posait de gros problèmes de motion sickness. Pour palier à ce problème, nous avons décidé que notre joueur serait statique et devrait tourner sur lui même pour se protéger d'ennemis pouvant venir de derrière lui. Seulement, le jeu nous a très vite semblé très creux en plus de souligner un autre problème du casque : les câbles.

Obliger le joueur à tourner sur lui même aussi souvent l'amener à s'enrouler dans les câbles rattaché au casque, le sortant ainsi de l'immersion du jeu. Pour palier à ces problèmes, nous avons décidés d'utiliser le système de téléportation que nous avons pu expérimenter dans la plupart des jeux en réalité virtuelle auxquels nous avons pu jouer mais plutôt que de le placer sur un simple bouton, nous avons décidé d'utiliser le système de projectiles que nous avons déjà mis en place que nous trouvions plus ludique.

Nous sommes donc arrivé à une deuxième itération du projet dans lequel, à la manière d'un tower defense, les joueurs ne sont plus attirés vers le joueur mais vers des zones que le joueur devait défendre. Il devait alors se déplacer de zones en zones avec les projectiles de téléportation dans le but d'atteindre les ennemis cachés dans les angles morts du joueur, créés par un pilier central.



*1er prototypage de la carte avec en vert le sol, en rouge les zones à défendre, en orange le pilier central*

Seulement, les joueurs auxquels nous avons fait tester le jeu ne se souciaient pas du tout des zones à défendre et préféraient s'amuser avec les déplacements, essayant d'atteindre le haut du pilier central malgré plusieurs tentatives pour rendre l'objectif compréhensible. Au vu de ces résultats, nous avons ravalé notre orgueil et décidé que si les joueurs préféraient les déplacements, c'est que la force de notre jeu n'était simplement pas là où nous l'espérions.

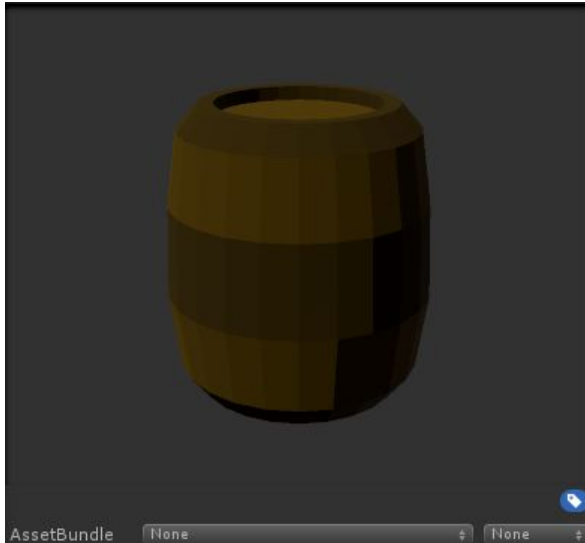
Nous avons donc redéfini le projet pour obtenir le résultat qu'on lui connaît aujourd'hui : un jeu de plateforme, sans ennemis ni zones à défendre dans lequel il faut monter en haut du pilier central qui attirait tant nos joueurs. Cependant, après avoir terminé le level design de notre jeu, nous nous sommes rendu compte d'un nouveau problème de taille.

En effet, la taille réelle du joueur influence directement la difficulté du jeu. Notre groupe étant composé de 3 personnes de taille similaire, nous ne nous en sommes rendu compte que lors de session de playtest où des personnes faisant moins d'environ 1m60 étaient incapable d'atteindre certaines zones avec leurs projectiles car les projectiles ne partaient pas suffisamment haut du sol pour les atteindre. Nous avons donc dû modifier tout notre level design, en ajoutant ou rapprochant des plateformes, pour que le jeu soit finissable pour des personnes de plus petite taille.

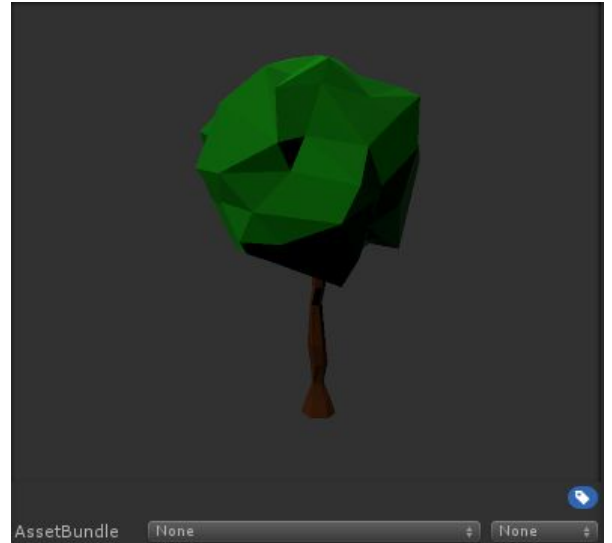


# Les références

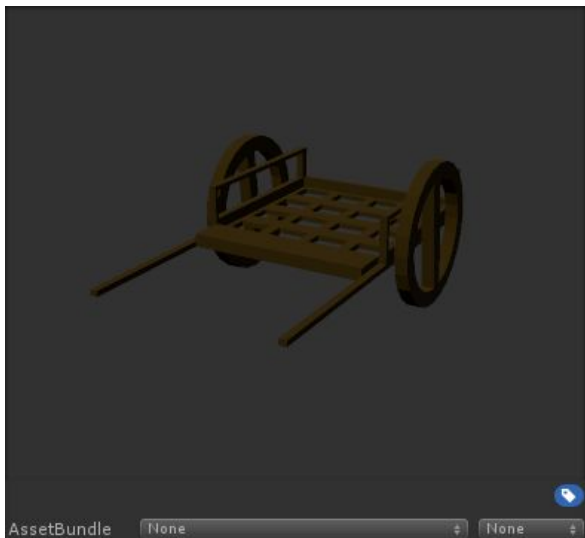
Quelques modèles 3D du jeu ont été repris de l'assets store de Unity quand ceux-ci correspondaient à la direction artistique que nous souhaitions donner au jeu. Les autres modèles 3D ont été réalisés par nos soins.



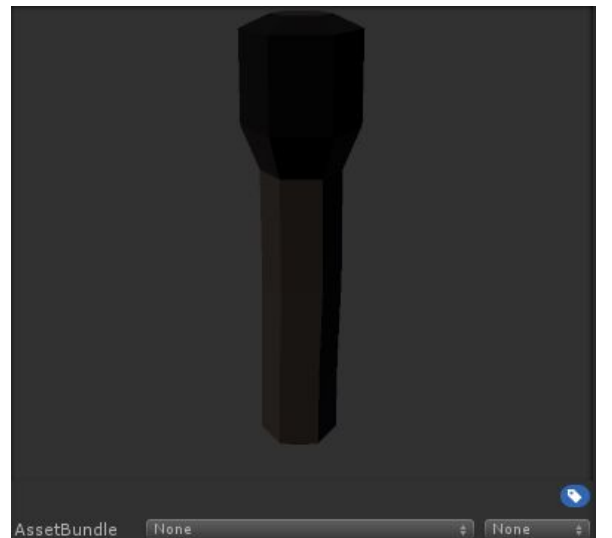
*Tonneau ([credit](#))*



*Arbre 1([credit](#))*



*Charrette ([credit](#))*



*Torche ([credit](#))*

Bien que nous ayant repris le modèle sur l'asset store d'Unity, nous avons néanmoins apporté notre touche à la torche en lui ajoutant des particules et une lumière pour simuler du feu.



*La torche avec nos effets de particules en jeu*

## Outils/Librairies externes

Comme dit plus haut, nous avons utilisé la librairie "[VRTK](#)" qui permet d'intégrer rapidement des éléments comme l'affichage des manettes en jeu, accélérant ainsi la phase de développement d'un jeu en VR. La reconnaissance du Vive par *VRTK* a été facilitée par l'utilisation du Package *SteamVR* de Unity3D.

La première facilité de développement apportée par *VRTK* a été la détection et l'utilisation des contrôles des manettes. En plus de pouvoir faire une abstraction des contrôles pour pouvoir les utiliser quelque soit le casque de Réalité Virtuelle (HTC Vive, Oculus Rift, ...) utilisé, nous avons pu ajouter des event listener sur différentes touches des manettes

```
GetComponent<VRTK_ControllerEvents>().TriggerPressed += new
ControllerInteractionEventHandler(DoTriggerClicked);
GetComponent<VRTK_ControllerEvents>().TriggerReleased += new
ControllerInteractionEventHandler(DoTriggerUnclicked);
GetComponent<VRTK_ControllerEvents>().TouchpadAxisChanged += new
ControllerInteractionEventHandler(ChangeSelect2);

private void DoTriggerClicked(object sender,
ControllerInteractionEventArgs e)
{
    ...
}
```

```

}

private void DoTriggerUnclicked(object sender,
ControllerInteractionEventArgs e)
{
    ...
}

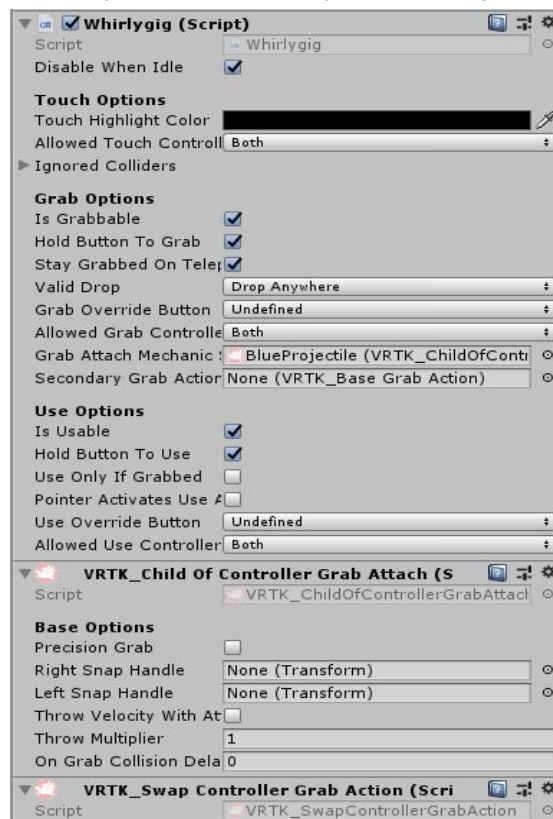
private void ChangeSelect2(object sender,
VRTK.ControllerInteractionEventArgs e)
{
    ...
}

```

*Portion du script de chargement des projectiles dans les controllers*

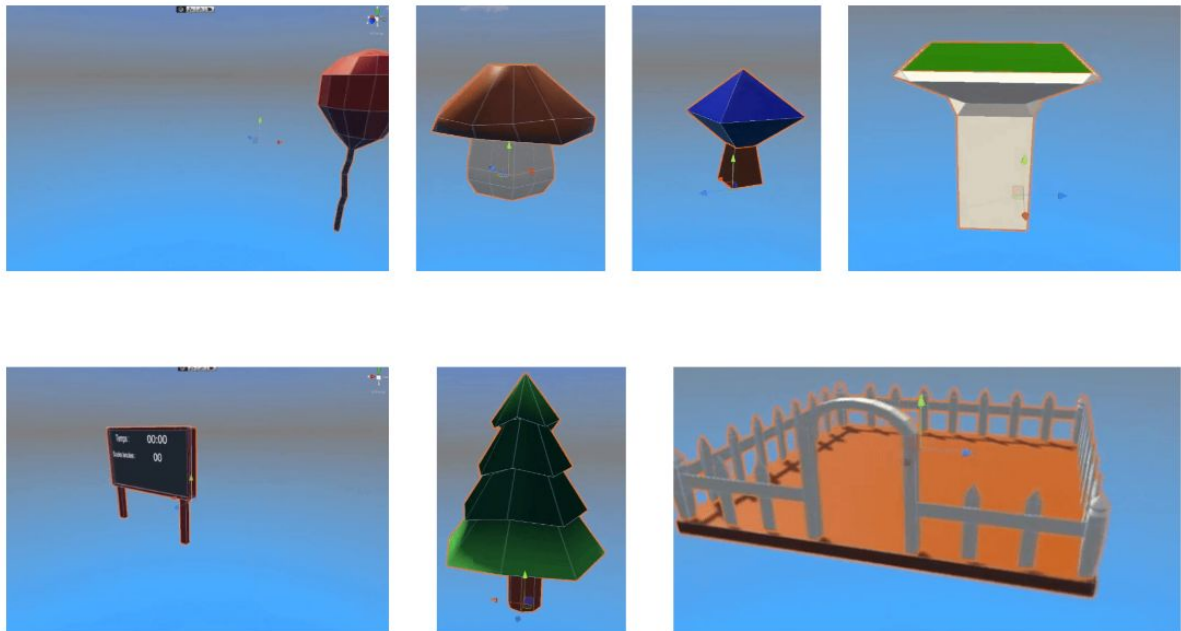
VRTK nous a aussi aidé en rendant automatique la gestion du casque de Réalité Virtuelle branché lors du jeu. La librairie intègre en effet un switch automatique pour adapter l'environnement du joueur en jeu au Vive, à l'Oculus, ...

Enfin, VRTK nous a permis de rendre les projectiles manipulables facilement par le joueur, en leur attachement simplement des composants existants qu'il nous aura suffi de configurer pour qu'ils correspondent au comportement que l'on attend en jeu.



*Composants de VRTK adaptés pour nos projectiles*

Pour réaliser nos décors, nous avons utilisé [ProBuilder](#), qui est un outil permettant de faire de la modélisation 3D directement dans les scènes de Unity ce qui nous a permis de créer nos préfabs sans avoir à utiliser des logiciels à part. Avec cet outil, il est possible d'appliquer des transformations à chaque points, arêtes et faces d'un objet en plus de pouvoir rajouter des faces et des arêtes ce qui nous a permis, à partir de formes simples comme des cubes ou des cylindres, de réaliser tous ces modèles 3D nous même.



*Modèles 3D réalisées pour notre projet*

De plus, ProBuilder nous a permis de réaliser les contours de notre niveau.