# Dimension Swap

Bienvenue dans notre toute nouvelle application immersive, conçue pour vous transporter dans des mondes extraordinaires, sans quitter votre navigateur. Plongez dans une expérience captivante où la réalité étendue (XR) rencontre la magie du web, tout cela sans nécessiter de logiciels complexes comme Unity.

Imaginez-vous assis devant votre ordinateur ou un autre appareil, où vous pouvez prendre un objet en 3D et le faire tourner sous tous les angles, presque comme si vous le teniez entre vos mains. Vous pouvez l'inspecter minutieusement, apprécier ses détails sous tous les éclairages, le faire pivoter et le retourner à votre guise. Une exploration interactive qui donne vie à la réalité virtuelle, directement depuis votre écran.

Mais ce n'est pas tout. Lorsque vous prenez votre téléphone portable en main, l'application vous emmène encore plus loin. Activez le mode "AR" (réalité augmentée) et transformez votre environnement en une scène où vous pouvez positionner l'objet 3D, le voir intégré harmonieusement dans votre monde réel. Vous avez désormais la liberté de vous déplacer autour de lui, de l'admirer sous tous les angles comme s'il était vraiment là.

Et pour une immersion totale, enfilez un casque de réalité virtuelle (VR). Entrez dans le monde virtuel et interagissez avec l'objet en utilisant des contrôleurs spéciaux. Vous pouvez le déplacer, l'agrandir, le manipuler de toutes les manières possibles. Vous pouvez même vous téléporter autour de lui, découvrant chaque détail comme si vous y étiez vraiment.

Notre application est conçue pour repousser les limites de l'exploration interactive en ligne, vous offrant une gamme complète d'expériences XR, que vous soyez sur un ordinateur, un téléphone ou un casque VR. Plongez dans l'avenir du web XR, où les frontières entre la réalité et le virtuel s'estompent pour vous offrir une expérience véritablement hors du commun.

## Choix et justification

### Technologies et librairies

En tant que projet web, le premier choix réalisé a été d'utiliser React. En effet, React permet de démarrer un projet rapidement tout en le gardant léger et facile à comprendre. C'est parfait pour un prototype ou un projet de test.

Ensuite, pour permettre la navigation au sein de l'application, nous avons utilisé React Router Dom, car c'est une librairie très facile à prendre en main et les routes sont rapides à comprendre.

Au sein de React, il nous fallait une technologie qui permette de prendre en charge facilement la 3D dans un navigateur web. C'est assez naturellement que nous nous sommes tournés vers Three.js, qui permet énormément de choses et dispose surtout déjà d'une grande communauté et de nombreux utilisateurs.

Cependant, Three.js peut se montrer assez complexe à utiliser et surtout, c'est une librairie prévue pour des projets web classiques. Pour en faciliter l'utilisation avec React, nous avons utilisé React Three Fiber.

React Three Fiber est donc une surcouche de Three.js qui permet d'ajouter une prise en charge de la 3D à React, en plus de permettre une implémentation simple de la XR (réalité augmentée et/ou virtuelle).

En ce qui concerne le style général de l'application, nous avions peu de temps à accorder à la mise en page de manière générale. Le choix s'est donc porté vers MUI.

### Réalité augmentée

L'objectif de la réalité augmentée est de pouvoir visualiser un modèle dans un environnement réel.

Pour ce faire, on doit pouvoir placer le modèle où l'on veut (ce qui inclut de pouvoir le repositionner une fois placé). Afin de garantir une certaine précision dans la visualisation, il était aussi important de pouvoir le faire tourner et l'agrandir/rétrécir.

Le positionnement a été réalisé via des HitTest. Ces HitTest servent à détecter les surfaces visibles par la caméra utilisée en AR afin d'y placer des objets.

React Three.js permet donc de lancer des hit tests en permanence. Chaque hit test lancé renvoie une position. On place donc un réticule (simple cercle) à l'emplacement reçu et on ajoute à ce réticule un événement au toucher. À chaque fois qu'on appuie sur ce réticule, le modèle est déplacé à l'endroit de ce dernier.

La raison pour laquelle nous avons opté pour cette mécanique est le fait que lorsque l'on passe en mode AR, l'environnement est calculé selon un plan ayant pour origine la position de la caméra lorsque l'on est entré dans le mode.

Le problème est que même si nous nous déplaçons, la caméra reste à cette position d'origine. Nous n'avons donc pas trouvé de moyen pour connaître notre position dans ce plan. Les hit tests nous permettaient de la récupérer via le réticule.

Une fois l'objet placé, il nous faut pouvoir choisir sa rotation afin de le faire regarder dans différentes directions. La première solution était de permettre une sorte de contrôle orbital : permettre de glisser son doigt sur le modèle afin de lui imposer une rotation en fonction de nos mouvements.

Cependant, même si React Three Fiber permet l'AR, le web n'est pas conçu pour ça. Les actions disponibles sont donc relativement limitées, tout comme les événements disponibles dans ce mode.

Nous n'avons donc pas réussi à déclencher un événement de "glissement" avec ce que nous proposait la bibliothèque.

La solution d'utiliser une librairie en plus a été envisagée : Use Gesture. Cette librairie permet de capter ce genre de mouvements au sein d'un composant React. Cependant, le Canvas nécessaire à l'utilisation de React Three Fiber se met en avant-plan lorsqu'on passe en mode AR, rendant la détection de ces mouvements impossible.

La solution implémentée a donc été un simple événement de clic sur le modèle. À chaque clic, le modèle tourne d'un degré prédéfini dans une direction prédéfinie. Ce qui permet de maintenir un modèle "droit" par rapport au sol et de le faire regarder dans la direction que l'on souhaite avec une simple interaction.

En ce qui concerne la redimension du modèle, nous voulions également utiliser des mouvements populaires pour cette fonctionnalité, mais cette idée a été compromise pour les mêmes raisons que la fonctionnalité précédente.

Nous avons donc pensé à mettre deux boutons par-dessus le canvas, l'un permettant d'agrandir le modèle, l'autre permettant de le rétrécir.

Cependant, comme dit précédemment, le canvas se met au premier plan lorsque l'on passe en mode AR et nous n'avons pas réussi à placer les boutons par-dessus.

La solution qui a été mise en place consiste donc à utiliser deux autres réticules. Fonctionnant aussi avec des HitTest, leur position est définie par ces derniers auxquels nous appliquons un offset pour les placer en bas de l'écran. Leurs fonctions sont exactement les mêmes que celles imaginées pour les deux boutons précédents : l'un agrandit le modèle, l'autre le rétrécit.

Réalité virtuelle

L'objectif de la réalité virtuelle est le même que celui de la réalité augmentée, mais elle permet la visualisation dans un environnement neutre.

Il faut donc également pouvoir voir le modèle sous différentes formes, l'agrandir, le tourner,…

À l'origine, nous voulions que toutes ces actions soient intuitives en utilisant les mêmes actions que celles imaginées pour l'AR, mais avec les raycasts lancés par les manettes d'un casque VR.

Cependant, comme pour l'AR, la bibliothèque ne permet que des actions et événements simples. Nous n'avons donc pas trouvé comment gérer chaque bouton d'une manette séparément afin d'y associer différentes actions.

React Three Fiber disposait quand même d'une fonctionnalité adaptée à la VR : le Raygrab.

Le Raygrab est une interaction fournie par la bibliothèque et qui permet de saisir un objet en réalité virtuelle afin de le déplacer.

Même si l'idée avait l'air intéressante, nous nous sommes rapidement rendu compte que le manque de possibilités lié au Raygrab rendait son utilisation compliquée à maîtriser pour l'utilisateur.

Afin de permettre un contrôle le plus précis possible pour l'utilisateur, le choix s'est donc dirigé vers un menu avec lequel l'utilisateur peut interagir quand il est en VR. Ce menu dispose de plusieurs boutons et chaque interaction effectue une action précise sur le modèle disposé devant l'utilisateur (rotation, redimension, position,…).

Une fois que le modèle correspond aux attentes de l'utilisateur, ce dernier peut se téléporter tout autour de lui afin de le visualiser sous différents angles.