
Projet de Fin d'Etudes
Organisation de l'espace de travail en Réalité
Augmentée

Auteurs :

Maxime DUMONTEIL
Loup GERMON
Maxime MEYRAT
Alex PEPI

Clients :

Pascal DESBARATS
Arnaud PROUZEAU

Résumé

Notre projet est un prototype d'application de visualisation de documents (images et textes) qui propose une organisation automatique de ses documents en fonction de certaines règles. L'utilisateur peut également modifier manuellement cette organisation. Elle a été implémentée en utilisant la plateforme de développement Unity et fonctionne sur le casque de réalité augmentée HoloLens 1 de Microsoft. Notre application supporte actuellement deux types de documents qui sont des fichiers image (JPG, JPEG et PNG) et des fichiers textes (TXT). Notre application dispose d'un menu permettant de sélectionner et d'afficher les fichiers. Pour les deux types de documents, le niveau de détail de l'affichage de l'objet, correspondant à un fichier, évolue en fonction de la distance à la tête de l'utilisateur. Il est aussi possible d'ancrer la position un objet soit dans l'environnement, soit dans la vue de l'utilisateur, c'est-à-dire que l'objet suit les mouvements de la tête de l'utilisateur et reste toujours visible à l'écran. Pour ce prototype, nous avons développé deux méthodes principales de visualisation décrivant une stratégie d'organisation automatique d'ouverture et d'affichage des fichiers. La première est basée sur des zones étiquetées. L'utilisateur peut placer un objet zone lié à une étiquette spécifique, autour de laquelle les documents seront automatiquement placés selon leur propre étiquette. La seconde est basée sur l'utilisation d'une carte de chaleur liée au regard de l'utilisateur. Plus une zone est regardée, plus elle suscite de l'intérêt, mettant ainsi à jour la carte de chaleur de l'environnement. Les documents sont alors automatiquement placés suivants les zones les plus intéressantes. Avec ceci, un système de forces physiques est utilisé à l'affichage des éléments afin qu'ils se placent automatiquement autour de la zone d'intérêt.

Table des matières

1	Introduction	1
2	Etat de l'art	2
3	Implementation	4
3.1	Utilisation du casque HoloLens 1	4
3.2	Architecture du projet	4
3.3	Coeur de l'application	5
3.3.1	Menu d'affichage des fichiers	5
3.3.2	Niveau de détail	6
3.3.3	Ancrage	6
3.4	Mode de visualisation n°1 : Etiquetage	6
3.5	Mode de visualisation n°2 : Carte du regard	9
3.6	Complément d'affichage : système de forces	10
3.7	Problèmes non corrigés	11
3.8	Fonctionnalités non implémentées	11
4	User Story	13
4.1	Utiliser le menu de selection de fichier	13
4.2	Ouvrir et afficher un seul fichier texte	13
4.3	Ouvrir et afficher un seul fichier image	14
4.4	Gérer l'ancrage des objets dans l'espace	14
4.5	Gérer le niveaux de détails d'un objet	14
4.6	Gérer l'étiquetages de zones et de fichiers	15
4.7	Création d'une carte de chaleur pour afficher des fichiers aux endroits les plus regardés par l'utilisateur	15
4.8	Implémenter un système de forces pour améliorer la disposition des fichiers	16
5	Tests	17
5.1	Tests unitaires	17
5.1.1	Tests de chargement d'un texte	17
5.1.2	Tests de chargement d'une image	17
5.1.3	Tests de fermeture d'un fichier (texte et image)	17
5.1.4	Tests de changement vers la page suivante pour un fichier texte	18
5.1.5	Tests de changement vers la page précédente pour un fichier texte	18
5.1.6	Tests de création d'une zone étiquetée	18
5.1.7	Tests de changement vers la scène de sélection de fichier	19
5.1.8	Tests de sélection d'un fichier	19
5.1.9	Tests de visualisation d'un fichier	19
5.1.10	Tests du changement du niveau de détail d'un texte	20
5.1.11	Tests du changement du niveau de détail d'une image	20
5.2	Tests MRTK	20
5.2.1	Tests de l'ancrage du menu	20
5.2.2	Tests du déplacement du menu	21
5.2.3	Tests de redimensionnement du menu	22
5.2.4	Tests de la rotation du menu	23
5.2.5	Tests d'affichage de la liste des fichiers accessibles	24
5.2.6	Tests de sélection des fichiers	25
5.2.7	Tests d'ouverture et d'affichage de fichiers via le menu	26
5.2.8	Tests de fermeture de fichiers	27
5.2.9	Test de déplacement d'un objet correspondant à un fichier	28
5.2.10	Tests de rotation d'un objet correspondant à un fichier	29
5.2.11	Tests de redimensionnement d'un objet correspondant à un fichier	30
5.2.12	Tests du changement de page d'un fichier texte	30
5.2.13	Test du mode de visualisation n°2 avec le système de forces	31
6	Résultats	33
7	Conclusion	34

1 Introduction

Les interfaces de réalité augmentée promettent de fournir des accès efficaces et flexibles aux informations numériques, tout en les intégrant dans le monde réel. La possibilité d'organiser librement les éléments de l'interface virtuelle dans l'espace qui nous entoure a le potentiel de faciliter à la fois la productivité, mais aussi les interactions. Cette technologie permet l'exploitation des espaces numériques personnalisés au-delà de ce qui est réalisable avec les ordinateurs actuels. Par exemple, il est même possible d'interagir en coopération sur le même espace de travail via la réalité augmentée. En mélangeant le monde réel et virtuel, il est alors possible de créer un grand nombre d'applications sur des multitudes d'appareils comme les téléphones, les casques ou lunettes de réalité augmentée. Dans ce cadre, les appareils concernés sont les casques de réalité comme ceux de Microsoft (HoloLens) ou de Lynx.

Comme dit précédemment, plusieurs problèmes sont à prendre en compte lors de la disposition de ces fichiers, il faut être capable de les disposer selon la considération de l'environnement ou éviter que ceux-ci s'obstruent entre eux par exemple. Malgré ces contraintes, l'utilisateur doit avoir à sa disposition un espace de travail organisé selon des algorithmes permettant de gérer ces problèmes, pour ensuite pouvoir interagir sur l'organisation de cet espace de travail comme bon lui semble. Ainsi, cette disposition d'éléments doit être automatique et intuitive pour servir au mieux l'utilisateur. On peut alors se poser la question, quelles stratégies seraient intéressantes à mettre en place pour proposer un placement adéquat de fichiers dans un espace de travail en réalité augmentée ?

Le principal objectif de ce projet est de réaliser un prototype d'application de visualisation de documents tels que des images ou du texte. Notre application propose une organisation automatique de ses documents suivant des stratégies que nous avons établies. L'utilisateur a la possibilité par exemple de déplacer les documents où il souhaite après l'agencement automatique ou d'ouvrir de nouveaux fichiers. L'implémentation de ce prototype a été réalisée en utilisant la plateforme de développement Unity et le langage C# et a été déployé sur un casque de réalité augmentée HoloLens 1 de Microsoft.

Nous allons donc proposer notre solution à ce problème dans ce rapport. En premier lieu, nous citerons l'état de l'art à propos de ce sujet et nous mettrons en évidence les sources dont nous nous sommes inspirés. Dans un deuxième temps, nous décrirons notre propre implémentation qui précisera le cœur de l'application et les modes de visualisation sur lesquels nous nous sommes principalement focalisés. Par la suite, viendront les User Stories, c'est-à-dire les descriptions de l'ensemble des besoins nécessaires au fonctionnement de l'application. Puis, nous détaillerons tous les tests que nous avons implémentés pour vérifier les implémentations de nos solutions. Pour finir, nous évoquerons les résultats de notre application via des avis recueillis auprès de quelques utilisateurs.

2 Etat de l'art

Nous allons maintenant passer en revue nos recherches à propos de l'état de l'art et sur l'existant à propos de sujets s'approchant de notre problématique, même s'ils n'ont pas forcément choisi des techniques et des supports similaires. Généralement, les articles se focalisent sur des problématiques d'affichage plutôt des solutions de placement efficace dans l'environnement comme ce que l'on cherche principalement à faire.

Tout d'abord, il peut être intéressant de préciser que la problématique de visualisation d'élément dans un environnement virtuel en 3D est un sujet qui date des prémices du domaine. Celui-ci a bien évolué ces dernières années avec le développement récent des technologies. D'autre part, beaucoup d'articles traitant d'environnement 3D se concentrent plutôt sur la réalité virtuelle (RV). Des besoins peuvent être partagés entre RV et RA, ces articles peuvent donc nous intéresser, mais la différence de support apportera des contraintes importantes comme l'occultation du monde réel en RA. On peut citer un premier article [9] qui présente une solution d'affichage interactif (nommé "Focus and Context") pour les applications de réalité augmentée (RA) en affichant des informations contextuelles et en prenant en compte bien sur les contraintes d'affichage comme l'occultation.

Tout d'abord, plusieurs articles cherchent en particulier à mettre en place une solution prenant en compte la tâche que l'utilisateur est en train de faire pour l'aider via l'interface virtuelle. Un premier article [10] traite de la possibilité de disposer des éléments virtuels plus ou moins proches de l'utilisateur, en modifiant aussi la précision des informations affichées ou la disposition dans l'environnement en fonction de son utilité pour la tâche en cours. Leur idée part du principe que, suivant la charge cognitive de l'action en cours, leur programme va choisir quoi afficher, comment et quand. Ainsi, le type d'informations affichées va changer et le niveau de détails de l'application en question aussi. Dans la même perspective, un autre article [4] va chercher à adapter l'interface virtuelle aux éléments physiques en associant automatiquement, les deux de manière optimale pour afficher les objets virtuels proches d'un objet physique auquel ils sont liés.

De plus, afficher des éléments virtuels dans un environnement réel apporte plusieurs problèmes que l'on ne peut négliger comme l'occultation de l'environnement physique. Pour résoudre ce problème, un article [7] va adapter la disposition des objets virtuels à l'environnement réel à l'aide d'un lot de contraintes et de règles. Celles-ci précisent les relations entre les éléments de l'application entre eux et avec l'environnement physique dans lequel ils sont placés. De ce fait, lorsqu'un utilisateur entre dans un nouvel environnement, une disposition de l'application cohérente est créée en temps réel et de manière optimale.

Un autre article [5] présente un système de placement automatisé en analysant la géométrie des surfaces physiques et le comportement visuel de l'utilisateur au fil du temps, afin d'optimiser le placement du contenu. Dans une approche similaire, un article [6] présente un système qui enregistre et analyse empiriquement le comportement des utilisateurs dans un espace et suggère automatiquement des positions et des tailles pour les éléments virtuels. Pour déterminer les emplacements possibles, l'application calcule des cartes de chaleur volumétriques décrivant la géométrie des structures réelles, les occlusions, la position dans le champ de vision de l'utilisateur et l'angle de vue afin de déduire des emplacements d'affichage valides et d'optimiser les positions des éléments.

En outre, d'autres travaux [3] cherchent à définir une solution de visualisation efficace de données sous la forme de diagramme et de graphe en 3D. Les auteurs ont aussi cherché à trouver les attributs visuels pratiques à la visualisation de ce type de structure, particulièrement dans des cas d'utilisations ou des environnements complexes. Ils ont réfléchi à savoir comment associer les données correctement à la structure. Les auteurs présentent aussi une analyse des contraintes et des nécessités pour la visualisation de graphe 3D en RA, et ainsi définir plusieurs styles de visuels.

Toujours dans le thème de la visualisation en RA, un article [12] propose une application pour que les développeurs suivent dynamiquement la performance et l'évolution d'un système continuellement. Ce n'est pas les mêmes informations que celles que l'on souhaite afficher, mais ces recherches sont aussi intéressantes pour avoir d'autres utilisations pertinentes en RA, surtout que leurs retours d'utilisation révèlent une certaine facilité d'utilisation.

Un autre article [2] décrit une gestion de la vue pour les interfaces RA interactives par l'utilisation de contraintes visuelles sur l'affichage et la vue. On peut retrouver par exemple la localisation ou l'occultation des objets les uns par les autres. Pour se faire leur programme modifie les propriétés de l'objet sélectionné, comme la position, la taille ou la transparence, pour respecter les contraintes, mais aussi réduire les discontinuités visuelles.

Une autre recherche [16] traite de la gestion de la quantité de données à afficher. Leur solution présente un affichage adaptatif de la densité d'information en RA afin d'équilibrer les informations présentées par rapport à l'encombrement potentiel des éléments dans l'environnement. En regroupant hiérarchiquement les composants affichables, ils créent une structure avec différents niveaux de détail en fonction, entre autres, des préférences définies par l'utilisateur afin d'éviter l'encombrement de la scène. Cette solution propose un affichage avec une

meilleure vue d'ensemble des données et permet une accessibilité plus facile aux éléments pertinents.

Beaucoup d'articles abordent des sujets similaires, mais dans le cadre de la réalité virtuelle. De ce fait, l'environnement n'est pas géré de la même manière, mais on peut tout de même retrouver des conventions de conception similaires. D'autres traitent de la visualisation de données en RA, mais se concentrent plutôt sur la gestion d'environnement collaboratif ce qui n'est pas une fonctionnalité que l'on a cherché à intégrer même si cela peut être une évolution de notre projet.

Un autre article [15] propose une visualisation combinant un dispositif RA et un grand écran interactifs afin de permettre l'exploration et l'analyse de données. L'utilisation d'écrans permet d'exploiter plus d'espace d'affichage, ce qui a d'après les auteurs du potentiel pour afficher des données autrement et efficacement. Les auteurs analysent comment leur solution peut permettre un travail collaboratif entre plusieurs utilisateurs. Sur cette base, ils présentent plusieurs exemples d'applications exploitant leur visualisation qui semble être utile pour explorer et comprendre des données variées.

La dernière composante de notre programme (que l'on décrira par la suite dans le chapitre "Implémentation") se base sur un système physique pour gérer la position et les interactions des éléments dans l'environnement. Ainsi, nous pouvons citer un premier article [8] présentant une méthode pour déterminer les emplacements intéressants des annotations pour des modèles complexes. Elle utilise des forces de répulsion et d'attraction entre les éléments et leurs étiquettes.

Dans un autre cas, un article [14] cherche à rendre explicites les liens entre les éléments représentant des données et leur contexte dans l'environnement (réel ou virtuel). Cette recherche examine les contraintes pour représenter ces liens afin de minimiser les problèmes d'affichage comme l'occultation ou l'encombrement. Les auteurs présentent un algorithme réalisant de tels agencements et démontrant des exemples d'utilisation pratiques.

D'autre part, on peut tout d'abord retrouver le projet d'IBM Immersive Insights [1] de visualisation de données en RA (utilisant un casque Microsoft HoloLens 1). Initialement conçu pour des Data Scientist, ce projet propose à tous les utilisateurs d'améliorer l'expérience en RA, explorer leurs données et le processus d'analyse des données. C'est donc un projet traitant surtout de visualisation de beaucoup de données simultanément, avec plusieurs fonctionnalités d'affichage pour l'expérience utilisateur. Les auteurs ont examiné des problèmes de visualisation en RA tel que trouver les dimensions et les mesures idéales pour l'interface.

Pour un autre exemple d'application de visualisation sur HoloLens, nous pouvons retrouver un projet expérimental de visualisation de données liées au COVID-19 [18]. Ce projet explore comment représenter ces données variées efficacement dans l'environnement, en utilisant le package MRTK de Microsoft.

Nous avons bien sûr utilisé la documentation de Unity [17] et le package "Mixed Reality Tool Kit" (MRTK) [13], un ensemble de composants et de fonctionnalités, tel un framework extensible, développé par Microsoft, afin de permettre le développement d'applications de réalité mixte interplateformes dans Unity. Ces composants permettent aussi de faire du prototypage rapide via une simulation dans l'éditeur permettant de tester le programme. Pour utiliser ces composants nous avons récupéré les paquets Unity depuis le dépôt GitHub [11] plutôt qu'avec l'exécutable délivré Microsoft qui peut les installer directement sur notre projet Unity (mais cela nous a amené des problèmes entre nos versions).

3 Implementation

Dans cette partie, nous cherchons à décrire notre solution et son implémentation. Dans un premier temps, afin de bien comprendre nos explications, nous allons détailler l'utilisation de l'appareil, quels sont les commandes du casque de RA. Ensuite, nous présenterons l'architecture et la hiérarchie des fichiers de notre projet. Pour décrire notre programme, nous allons tout d'abord expliquer le coeur de notre application, puis nous présenterons un premier mode de visualisation, l'étiquetage, puis un deuxième mode, la carte de chaleur du regard. Par la suite, nous décrirons le complément à l'affichage des éléments : le système de forces. En outre, nous évoquerons les problèmes que nous avons détecté dans notre solution, mais que nous n'avons pas pu corriger. Enfin, en dernier lieu, nous décrirons les fonctionnalités que nous n'avons pas pu ajouté par manque de temps ou par choix.

3.1 Utilisation du casque HoloLens 1

En réalité mixte, les hologrammes et éléments virtuels se mélangent avec votre environnement physique. Même lorsque les hologrammes sont autour de vous, vous pouvez toujours vous déplacer librement et interagir avec d'autres personnes et objets. Pour ce faire, vous pouvez utiliser vos mains pour manipuler des fenêtres holographiques, les menus et les boutons. Pour pouvoir pointer et utiliser le curseur, vous utiliserez les mouvements de la tête. Ensuite, pour interagir avec les objets et les menus, vous pouvez utiliser les deux principales gestuelles reconnus par l'appareil :

- "Flower gesture" : Cette action permet d'ouvrir le menu "Démarrer", c'est-à-dire le menu principal du HoloLens, duquel vous pourrez parcourir les applications et autres options de l'appareil.
Explication de la gestuelle :
 - Maintenez votre main devant vous pour qu'elle se trouve dans le champ des capteurs de l'appareil.
 - Rassemblez tous vos doigts, puis ouvrez votre main (telle une fleur).
- "Air tap gesture" : Pour sélectionner et cliquer sur une interface virtuelle, il faut appuyer dessus en positionnant au préalable le curseur sur l'objet à l'aide des mouvements de la tête.
Explication de la gestuelle :
 - Examinez l'hologramme que vous souhaitez sélectionner.
 - Pointez l'index vers le haut en direction du plafond.
 - Baissez votre doigt, puis lèvez-le rapidement (comme si vous appuyez dans l'air).
- "Pinch gesture" : Pour sélectionner et conserver la sélection en cliquant sur un élément virtuel, il faut appuyer dessus en positionnant au préalable le curseur sur l'objet à l'aide des mouvements de la tête, puis de conserver le mouvement de sélection actif.
Explication de la gestuelle :
 - Examinez l'hologramme que vous souhaitez sélectionner.
 - Pointez l'index vers le haut en direction du plafond.
 - Baissez votre doigt, puis conservez l'index vers le bas (comme si le doigt restait appuyé sur une touche).

En plus de ces fonctionnalités, il est aussi possible d'utiliser la voix pour sélectionner un objet virtuel. Ce n'est pas une option que l'on a décidé d'exploiter dans notre programme afin de plutôt se concentrer sur notre solution. Néanmoins, cette fonctionnalité s'utilise en disant un des mots reconnus par l'appareil (toujours en anglais cependant) tel que "menu", "back" ou "select" pour sélectionner un objet si on a positionné le curseur avant par exemple.

3.2 Architecture du projet

L'architecture du code et des fichiers de notre projet suit celle d'un projet Unity standard. Ainsi la majorité des fichiers que nous avons modifié et créé se trouve dans le répertoire "Assets". Le répertoire "Packages" correspond à l'ensemble des paquets "personnalisés" que nous avons importé, comme MixedRealityToolkit Foundation. Alors que le répertoire "ProjectSettings" correspond à l'ensemble des paramètres de notre projet. Plus précisément, il contient un profil pour le casque HoloLens 1 qui permet d'éviter d'afficher le maillage de l'environnement ou encore la console avec le nombre d'images par secondes en temps réel.

Ainsi, dans le répertoire "Assets", on retrouve principalement le dossier "MRTK" qui contient tous les éléments de l'ensemble des composants fournis par Microsoft. On peut y retrouver les fichiers de tests de "MRTK" ou encore les prefabs offerts par le paquet. Ils correspondent à des ensembles d'éléments et objets d'Unity formant un modèle que l'on peut ainsi facilement utiliser ou dupliquer dans un autre objet ou une autre scène.

Le répertoire "Resources" regroupe les prefabs que nous avons créé nous-même. On peut donc y retrouver par exemple le modèle pour l'élément virtuel correspondant à un texte ou à une image. Mais aussi les éléments des listes déroulantes des menus, ou encore les zones étiquetés.

Dans le répertoire "Scenes" se trouve les fichiers définissant les scènes Unity. "MainScene" correspond à la scène d'affichage des fichiers et "TagScene" correspond, quant à elle, à la partie étiquetage de l'espace.

Ensuite, nous avons mis en place un dossier "Scripts" pour regrouper tous les scripts dont nous avons besoin lors des interactions utilisateurs. Cependant, il est à noter que certains scripts se trouvent directement à la racine du repertoire "Assets" dû à la nécessité que ceux-ci aient accès aux fichiers du paquet "MRTK".

Par la suite, tous les fichiers à afficher chargés par l'application se situent dans le dossier "StreamingAssets". Ce dossier est directement pris en charge par Unity et n'est pas combiné lors de la compilation du projet. En effet, ces fichiers sont placés dans le système de fichiers normal de l'HoloLens pour les rendre accessibles via un chemin d'accès. Il y a donc à l'intérieur de ce dossier, tous les fichiers textes (TXT) et images (JPG, JPEG, PNG). Egalement, le fichier JSON gérant les liens entre fichiers et étiquettes est présent dans "StreamingAssets".

Enfin, nous avons créé le dossier "Tests" pour contenir l'ensemble des scripts de nos tests unitaires. Les scripts de tests "MRTK" sont quant à eux situés dans le dossier "PlayModeTests" du dossier "Tests" inclus dans le dossier "MRTK".

3.3 Coeur de l'application

Notre application comporte des fonctionnalités que l'on retrouve dans tous les modes de visualisation. D'après nous, celles-ci sont essentielles au bon fonctionnement de l'application et de l'affichage des éléments virtuels. Donc peu importe les interactions que l'on souhaite implémenter, ces composants seront systématiquement présents. Ainsi le cœur de notre application regroupe :

- Sélectionner un fichier parmi une liste.
- Charger un fichier.
- Afficher un fichier.
- Modifier manuellement l'agencement des éléments affichés dans l'espace.

Avec ces éléments, on y ajoute deux fonctionnalités qui d'après nous permettent une amélioration de l'expérience utilisateur : une gestion du niveau de détails et l'ancrage de la vue.

3.3.1 Menu d'affichage des fichiers

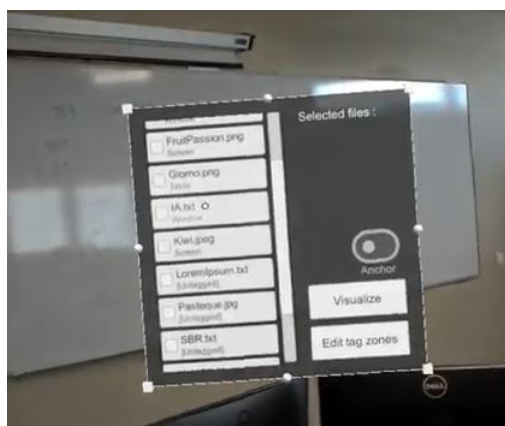


FIGURE 1 – Menu de sélection des fichiers

Pour pouvoir ouvrir un fichier et afficher un élément virtuel, nous avons choisi, pour les deux modes de visualisation, d'avoir un menu sur une surface positionnée devant l'utilisateur (et fixé à l'ouverture de l'application). Cette surface, comme pour les autres éléments, peut être déplacée, réorientée ou encore agrandie ou diminuée. Ce menu est composé d'une liste des fichiers accessibles (sur la gauche du menu 1) et d'un bouton permettant d'afficher les fichiers sélectionnés au préalable (nommé "Visualize", en bas à droite sur la surface 1). Au-dessus de ce bouton, on peut retrouver une liste textuelle affichant la liste des fichiers actuellement sélectionnés. Le

reste des composants de ce menu dépendant du mode de visualisation. Par la suite, lorsque l'utilisateur sélectionne un ou plusieurs fichiers et appuie sur le bouton "Visualize", les éléments des fichiers s'ouvriront suivant le mode de visualisation utilisé.

3.3.2 Niveau de détail

Afin d'améliorer l'expérience utilisateur, lorsqu'un élément est affiché, les détails des informations représentés sont gérés automatiquement. C'est-à-dire que suivant la distance entre l'utilisateur et l'élément virtuel, la taille de la police augmentera ou diminuera pour un texte ou un rognement s'appliquera pour une image par exemple. Notons que la taille de l'objet correspondant au fichier ne change pas, mais bien l'affichage du contenu qui s'adaptera. Cette fonctionnalité permet d'après nous de pouvoir réduire et optimiser la charge d'informations que l'on affiche. Ainsi, la fatigue visuelle est réduite pour l'utilisateur puisque celui-ci a besoin de moins se concentrer pour discerner le contenu des éléments.

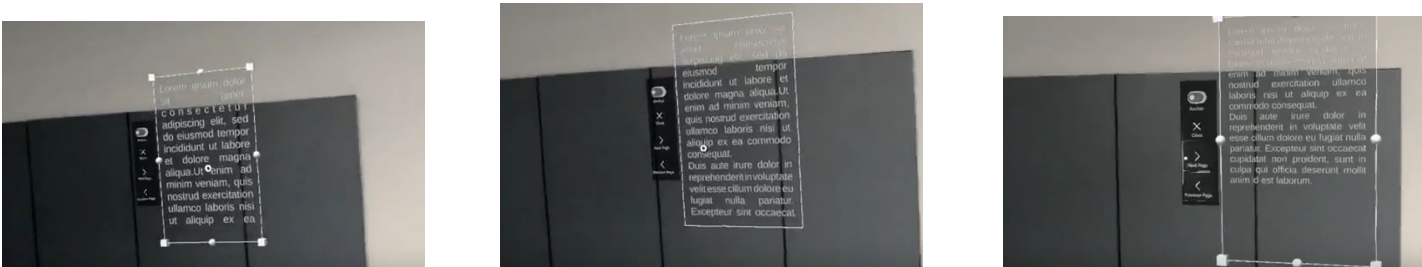


FIGURE 2 – Différents niveaux de détails (pour un fichier texte)



FIGURE 3 – Différents niveaux de détails (pour un fichier image)

3.3.3 Ancrage

D'autre part, les éléments virtuels sont soit ancrés dans l'environnement ("world-anchored"), soit ancrés par rapport à la vue ("view-anchored"), afin de pouvoir équilibrer entre un accès rapide et une aisance pour l'utilisateur via une position fixe dans l'environnement. Si l'élément est ancré dans l'environnement, c'est une gestion habituelle de la position, donc la position et les transformations de l'élément, ne seront pas automatiques, mais fixes (néanmoins bien entendu modifiables par l'utilisateur). Au contraire, si un élément est ancré par rapport à la vue, il va suivre les mouvements de la tête et du casque en restant à une distance fixe devant l'utilisateur. Ce mode d'affichage permet de toujours avoir l'élément dans le champ de vision et peut représenter une alternative d'automatisation du placement des éléments qui permet, par exemple, à l'utilisateur d'accéder et de déplacer les éléments sans être interrompu ou gêné. Dans l'interface, ce mode d'ancrage peut être activé pour chaque élément virtuel indépendamment via un bouton pour passer de l'ancrage dans l'environnement à celui sur la vue, et inversement.

3.4 Mode de visualisation n°1 : Etiquetage

Ce mode de visualisation va permettre (comme le suivant) de définir l'agencement, donc la position, la rotation et les autres transformations d'un élément lorsque l'on ouvre un fichier. Dès lors, cette première proposition va se baser sur un principe d'étiquettes (inspiré de [10]). Afin d'indiquer au programme la nature de l'environnement réel, l'utilisateur va étiqueter les objets réels et ainsi les fichiers, qui porte eux aussi des étiquettes,

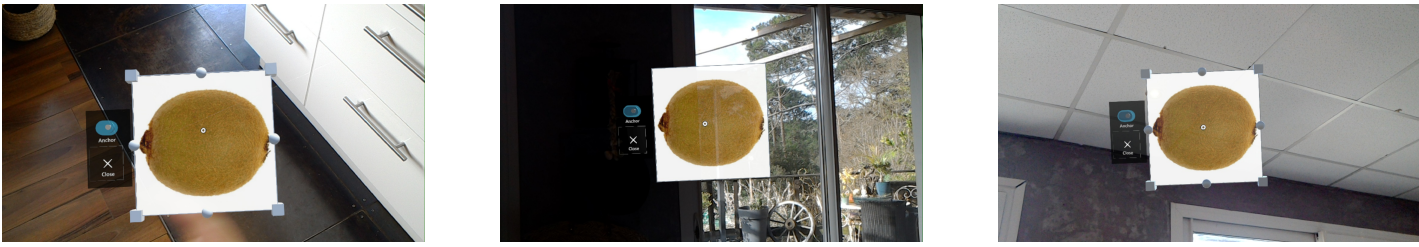


FIGURE 4 – Exemples d'un ancrage d'un élément suivant la vue

s'ouvriraient proche des éléments réels ayant une étiquette similaire.

Néanmoins, l'algorithme nécessite quelques prérequis pour être fonctionnel. Premièrement, tous les fichiers doivent être associés à une étiquette pour un fonctionnement optimal de ce mode de visualisation. Dans le cas contraire, notre solution regroupe tous les fichiers non labellisés. Ceci permet de les rendre accessibles et utilisables, mais il y a alors une perte d'intérêt, le but étant avant tout de classer les éléments. Par la suite, il est aussi nécessaire que l'utilisateur labellise les objets réels. Pour cette interaction, dans notre programme, elle se manifeste par un premier menu qui s'affiche au lancement de l'application et qui va permettre de placer dans l'environnement les zones étiquetées, autour desquelles les fichiers correspondants seront affichés à l'ouverture. En conséquence, lorsqu'un fichier est ouvert, sa position à l'ouverture est ajustée en respectant les règles qu'on lui impose, c'est à dire le lien entre l'étiquette du fichier et celle de la zone d'affichage (qui dans l'article correspond à un objet réel).

Dans les nécessités qu'implique cette solution, on peut en déduire certaines que l'on traduit en inconvénients. Par exemple, on peut retrouver le fait que beaucoup d'entrées utilisateurs sont nécessaires, ce qui ajoute beaucoup d'interactions et donc beaucoup de sources d'erreurs ainsi que de compréhension de l'interface pour l'utilisateur. Enfin, nous pouvons aussi citer la contrainte qui exige un prérequis de traitement pour les fichiers puisque l'étiquetage est obligatoire. Pour ce dernier problème, il est imaginable de le résoudre avec des solutions d'intelligence artificielle pour catégoriser les images ou autres fichiers, mais ce n'était pas le but de notre projet. Cette solution a toutefois aussi des avantages comme un lien sémantique direct entre les objets réels et l'environnement virtuel, ce qui permet de justifier l'intérêt de cette proposition.

Ainsi, pour implémenter cette solution, nous avons décidé d'utiliser un fichier JSON pour stocker tous les liens entre les étiquettes et les fichiers. Dans cette optique, le fichier JSON est composé d'une liste où chaque fichier est renseigné et est relié avec une seule étiquette. Cette liste est écrite en dur dans le fichier JSON et nécessite que l'utilisateur modifie ce dernier pour changer les étiquettes.

Le menu permettant à l'utilisateur de positionner les zones étiquetées est directement dépendant des étiquettes du fichier JSON. Chaque étiquette unique comprise dans le fichier donne lieu à un bouton spécifique dans le menu. Une fois un des boutons pressés, une zone liée à l'étiquette du bouton apparaît à une distance fixe face à l'utilisateur. Ce dernier peut alors positionner la zone comme bon lui semble. Deux boutons additionnels se situent sur le menu. Un nommé "Remove all" permettant de supprimer toutes les zones étiquetées placées précédemment. Ainsi qu'un bouton "Done!", permettant de valider l'emplacement des zones et de passer au menu de sélection des fichiers.

En ce qui concerne le menu de sélection de fichiers, une liste déroulante est disponible sur celui-ci affichant tous les fichiers et leurs étiquettes. Les étiquettes sont basées sur le fichier JSON et comme évoqué précédemment, les fichiers sans étiquette ont une étiquette arbitraire "[Untagged]" qui leur sont attribuée.

Une fois plusieurs fichiers sélectionnés, lorsque l'utilisateur souhaite les afficher, ceux-ci se positionnent autour de la zone liée à leur étiquette à une distance relative à leurs tailles. Egalement, ils sont positionnés en suivant la même rotation que la zone. Chaque zone peut contenir un maximum de quatre fichiers positionnés sur les côtés haut, bas, gauche et droit par rapport à la face étiquetée de la zone. Le premier fichier se place sur le côté gauche et les suivants se placent suivant l'ordre horaire.

Si plusieurs zones liées à une même étiquette sont positionnées, c'est la zone la plus proche de l'utilisateur qui sera remplie en première. Puis une fois pleine, la deuxième zone la plus proche avec la même étiquette sera remplie et ainsi de suite.

Lorsque l'utilisateur ferme, ancre ou déplace un fichier d'une zone, celui-ci n'est plus considéré comme appartenant à cette zone. Il est alors possible d'y replacer des éléments en ouvrant de nouveaux fichiers.

Si l'utilisateur souhaite ouvrir et afficher un fichier dont l'étiquette n'a pas été définie par une zone dans l'environnement précédent, ou si toutes les zones liées à une étiquette sont pleines, le programme affiche l'élé-

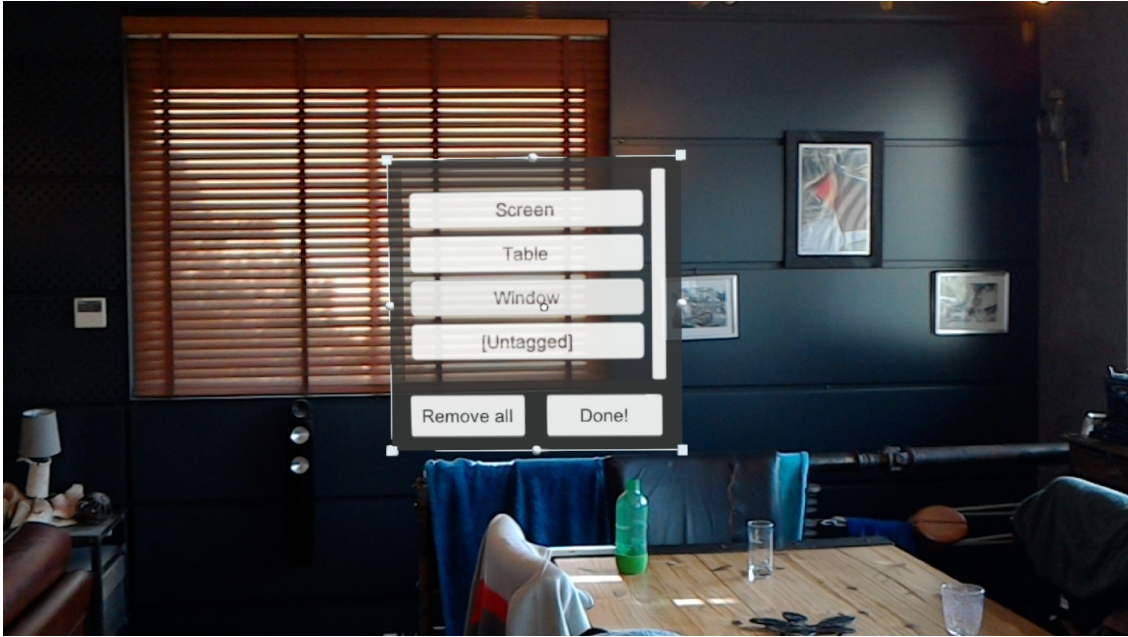


FIGURE 5 – Menu de sélection des zones d'étiquetages.

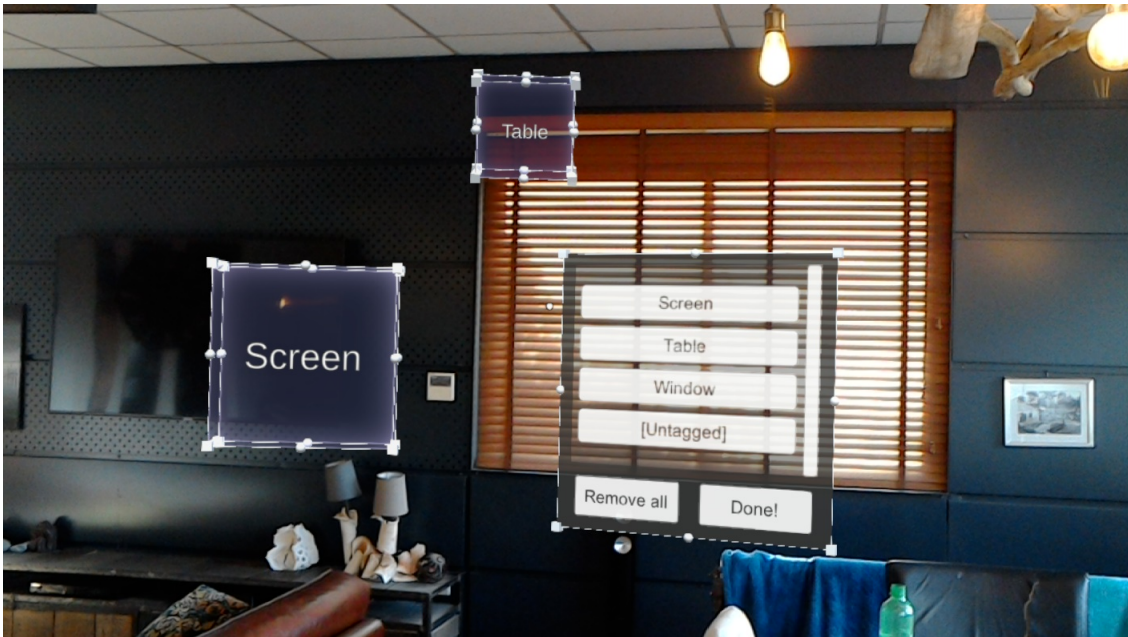


FIGURE 6 – Exemple de zones d'étiquetage.

ment devant l'utilisateur tourné dans sa direction. Tout ceci, de la manière la plus simple possible pour qu'il puisse quand même avoir accès à tous les documents.

Enfin, le menu d'affichage des fichiers contient un bouton "Edit tag zones". Celui-ci permet à l'utilisateur de retourner au menu de positionnement des zones étiquetées. Toutes les zones précédemment créées sont alors réaffichées et l'utilisateur peut donc les déplacer ou créer de nouvelles zones. Une fois les modifications faites, l'utilisateur peut à nouveau appuyer sur le bouton "Done!", ce qui le fera retourner au menu d'affichage des documents. Il est cependant important de noter que si les zones étiquetées sont modifiées, tous les documents affichés seront automatiquement fermés pour éviter des problèmes de positionnement.

3.5 Mode de visualisation n°2 : Carte du regard

Comme le mode de visualisation précédent, celui-ci définit une autre solution pour l'agencement d'un élément lorsque l'on ouvre un fichier. Cette deuxième option se base sur une optimisation du placement des éléments par rapport aux zones où l'utilisateur regarde le plus.

Pour faire ceci, on va chercher à sauvegarder la position du pointeur de l'utilisateur pour repérer les zones de l'environnement qu'il va observer. Via une carte de l'environnement, on va attribuer un poids, une valeur à ces zones consultées afin de pouvoir, lorsqu'un élément sera affiché à l'ouverture d'un fichier, positionner l'élément suivant cette carte.

Cette solution a aussi quelques inconvénients tels que les calculs nécessaires qui peuvent s'avérer être coûteux. Cette complication nous amène à expliquer un autre problème que nous avons volontairement évité. A l'origine, cette solution est inspirée par l'article HeatSpace [6] qui met à jour constamment la carte de l'environnement. Dans notre programme, ce n'est pas le cas car, avec notre implémentation, cela provoquerait un biais : les zones ayant déjà un poids, le conserveraient au cours du temps puisque l'utilisateur garderait son regard dessus étant donné que les fichiers y seraient placés.

Toutefois cette solution à l'avantage d'une recherche d'optimisation dans le placement des fichiers à l'ouverture avec une approche différente du mode de visualisation précédent.

Ainsi, pour implémenter cette solution, nous sommes partis d'un package implémenté par Alan Zucconi [19], décrivant un script C# et un shader, permettant de créer une carte de chaleurs sur une carte de la ville de Londres avec des positions et valeurs aléatoires.

Le script contient deux tableaux de vecteurs de taille 4, que l'on passe au Shader, qui sont les positions des puits de chaleurs, et un tableau contenant les propriétés de ces puits, ainsi qu'un entier qui correspond au nombre de points maximum que l'on peut enregistrer. Dans notre cas, nous n'utilisons pas le Shader car nous avons décidé de représenter les zones regardées par l'utilisateur par des sphères. Ainsi, nous avons 2 tableaux de vecteurs de taille 3 et 2, qui contiennent respectivement les positions et les propriétés des puits de chaleur. Le tableau des propriétés contient des vecteurs qui correspondent au rayon du puits et son intensité. Nous avons fixé le rayon des puits (autrement dit des sphères) à 0.05 m à l'initialisation, l'intensité, quant à elle, est augmentée de 0.1 lorsque l'utilisateur passe de plus en plus de temps à regarder la même zone.

Le tableau des positions contient le centre des zones de rayon 0.05 que l'utilisateur a regardé dans son environnement. Nous avons ajouté un tableau contenant les directions dans lesquelles les fichiers vont être affichés. Il s'agit de la direction inverse de la normale du point vers lequel regarde l'utilisateur. Puis il y a aussi un autre tableau contenant l'ensemble des GameObjects qui modélisent les sphères qui sont vertes à l'origine, et deviennent rouges et grossissent en volume si l'utilisateur garde son regard dans un rayon de 0.25 m par rapport au centre de cette zone. Pour trouver la zone la plus observée, il suffit de trouver l'intensité maximum, et de récupérer les coordonnées du centre de cette zone grâce à l'indice correspondant.

Dans l'application, les processus de scan et d'affichage sont les suivants :

l'application se lance, l'utilisateur lance le scan en appuyant sur le bouton start sous ses yeux et dispose de 20 secondes pour regarder autour de lui les zones dans lesquelles il souhaite placer des fichiers. L'utilisateur aperçoit devant lui les puits se former qui sont des sphères de couleur verte, plus il reste au même endroit, plus le puits devient rouge et grandit en taille, indiquant que cette zone a fortement été regardée. Lorsque la sphère atteint un rayon de 0.35, le volume de la sphère ne change plus et uniquement la couleur change. A la fin du décompte (présent sous le curseur en bas), le menu de sélection s'affiche et l'utilisateur peut alors sélectionner les fichiers que bons lui semble. Les fichiers choisis sont ensuite disposés autour du centre des trois zones ayant été les plus soumises au regard de l'utilisateur. L'ensemble des sphères indiquant les zones regardées disparaissent et une sphère blanche est placée au centre des zones élues, et les fichiers sont soumis à des forces ressorts entre eux et par rapport à ces sphères pour éviter qu'ils se confondent les uns avec les autres. Les fichiers sont répartis équitablement entre chacune des zones. Par exemple si 10 fichiers sont ouverts alors la première zone aura 3

fichiers, la seconde aussi et la dernière en aura 4.



FIGURE 7 – Exemple de l'étape de scan

3.6 Complément d'affichage : système de forces

Afin de compléter l'affichage, nous avons décidé d'ajouter un système physique pour gérer la position et les interactions des éléments dans l'environnement. Lorsqu'un fichier est ouvert, dans le mode de visualisation selon la carte du regard, celui-ci va avoir des forces de répulsion et d'attraction appliquées sur lui afin qu'il se place de manière automatique dans l'espace par rapport aux autres éléments dans l'environnement.

Le script C# *ApplyForce* est chargé d'appliquer les forces ressorts entre les *GameObjects* représentant les fichiers, mais aussi les forces ressorts entre le centre de la zone concerné et les fichiers un à un. Dans ce script, nous agissons en temps réel, sur les interactions entre le centre et les fichiers dans un premier temps. Pour ce faire, on fixe une distance de 0.5 souhaitée entre le centre et les différents fichiers (*distanceDesiredFromCenterMax*), ensuite, on calcule, pour chacun des fichiers, la distance (*distance*) entre le centre de la zone et le centre du *GameObject* représentant le fichier. Pour les calculs de distances, on considère uniquement les coordonnées sur les axes x et y, pour éviter que les fichiers se retrouvent l'un devant l'autre sur l'axe z. La force à appliquer sera pondérée par la différence normalisée indiquant la direction dans laquelle le fichier doit bouger. Ensuite, la constante (*connectedForce*) est fixée à 1.0. On calcule selon la formule suivante la force à appliquer à ce fichier :

$$appliedForce = \frac{difference}{\|difference\|} \times \frac{connectedForce}{distance} \times mode$$

On utilise *mode* qui est une variable valant -1, 0 ou 1. On définit alors le mode de la force selon ces valeurs comme il suit :

$$\begin{aligned} mode &= -1, \text{ répulsion} \\ mode &= 0, \text{ neutre (aucun mouvement)} \\ mode &= 1, \text{ attraction} \end{aligned}$$

Par défaut, le mode est sur neutre. Lorsque la distance entre le centre et le fichier dépasse *distanceDesiredFromCenterMax* alors le mode est mis sur attraction. Dans le cas contraire, le mode est placé sur répulsion. On utilise ensuite ce calcul de force pour stocker une vitesse que l'on va ajouter à elle-même au cours du temps. Elle s'exprime de la manière suivante :

$$velocity = appliedForce \times \Delta_{time}$$

Δ_{time} correspond au nombre de secondes entre la frame précédente et celle actuelle, il permet de visualiser l'accélération. De plus, pour avoir une visualisation de ces déplacements un peu plus lisse, on divise cette vitesse par un facteur 10 lorsqu'on l'applique à la position du fichier.

Concernant les interactions entre les objets représentant les fichiers, il s'agit des mêmes formules. Néanmoins, il y a une certaine subtilité pour éviter au maximum les collisions, nous utilisons deux variables qui sont *desiredConnectedDistance* et *desiredConnectedDistanceMax*. On agit sur le mode ensuite selon la distance courante entre deux objets et les deux variables citées auparavant. Voici la relation qui opère :

$$\begin{aligned} \text{Si } distance &\leq desiredConnectedDistance, \text{ alors } mode = -1 \\ \text{Si } distance &\geq desiredConnectedDistanceMax, \text{ alors } mode = 1 \end{aligned}$$

Ainsi on maintient forcément une distance entre les fichiers, sans qu'ils s'éloignent très loin l'un de l'autre. De plus, si le nombre de fichiers à afficher au fil du temps est égal ou dépasse le nombre de 10 pour une zone alors on augmente les variables *desiredConnectedDistance*, *desiredConnectedDistanceMax* et *distanceDesiredFromCenterMax* de 0.3, afin d'éviter que les fichiers s'obstruent entre eux et entraînent donc une certaine illisibilité. Ainsi, avec ce système de forces, nous itérons en permanence sur les objets présents dans la scène et mettons à jour en conséquence les forces afin d'un obtenir un affichage des fichiers dans l'espace qui soit convenable.

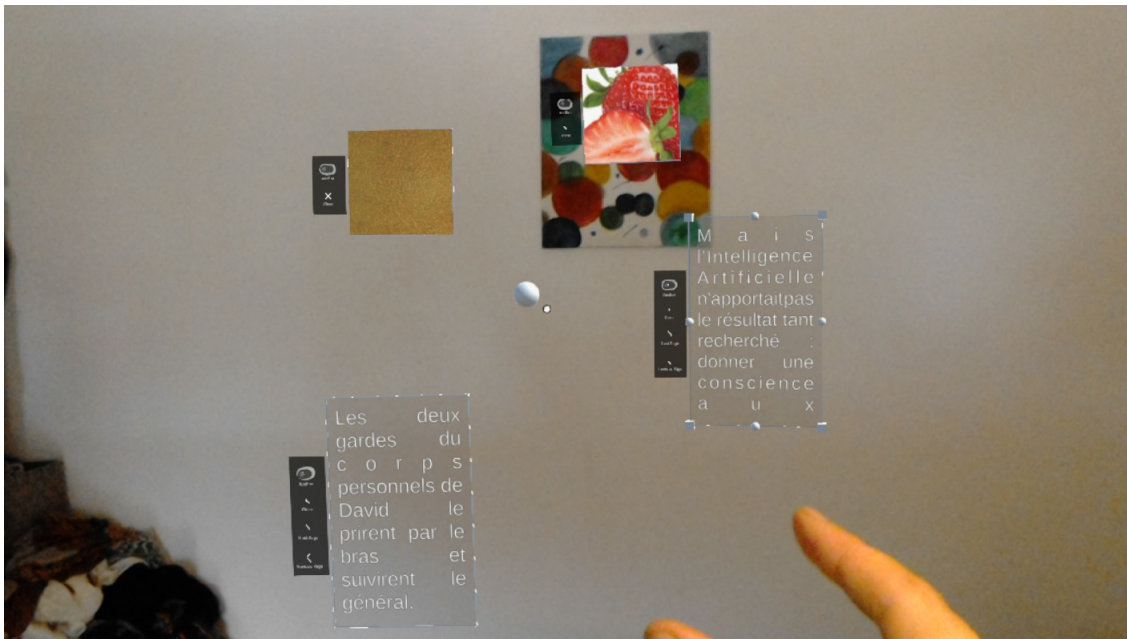


FIGURE 8 – Fichiers en gravitation dans une des zones choisies

3.7 Problèmes non corrigés

Dans l'état actuel du programme, il nous reste malheureusement quelques problèmes que nous avons pu détecter, mais pas pu corriger. Premièrement, nous avons remarqué qu'il y a une complication avec certaines opérations de transformations des éléments. Ce problème se manifeste surtout sur l'élément du menu de sélection de fichiers, commun aux deux modes de visualisation. Lorsque l'on souhaite orienter, agrandir ou réduire la taille du menu, celui-ci se déforme. En effet, le menu est étiré verticalement ou horizontalement et dépasse du contour représentant ses limites. Des déformations peuvent aussi se manifester lorsque l'élément est très proche de l'utilisateur.

Un deuxième problème que nous avons pu détecter est le déplacement d'un fichier en cliquant sur certains des boutons de son sous-menu. En effet, nous avons mis en place un script pour empêcher les menus de se déplacer lorsque l'utilisateur clique sur ceux-ci. Cependant, dans certaines circonstances que nous n'avons pas pu identifier, le script ne fait pas effet avec certains des boutons des sous-menus des fichiers. N'ayant pas pu déterminer les circonstances exactes, nous n'avons pas pu corriger efficacement ce problème.

3.8 Fonctionnalités non implémentées

Durant le projet, nous avons eu plusieurs idées de fonctionnalités ou de solutions que nous n'avons pas pu implémenter par manque de temps ou de moyen. Néanmoins certaines idées restent intéressantes ou pertinentes à citer d'après nous.

Dans un premier temps, nous avons eu l'idée d'ajouter des indications dans l'interface pour indiquer la présence d'éléments en dehors du champ de vision de l'utilisateur. Le champ de vision est un problème important de l'appareil, en particulier sur le HoloLens 1, qui est petit et peut poser des difficultés de compréhension pour l'utilisateur. Ainsi, nous avons eu l'idée d'utiliser des symboles, par exemple des flèches, pour indiquer la direction des éléments hors du champ de vision. Cependant, afin de simplifier l'interface, il y aurait eu une seule flèche suivant la position de l'utilisateur si un ou plusieurs éléments s'y trouvent à droite, à gauche, en bas ou au-dessus du centre de l'écran. Il n'y aurait donc pas une indication par élément hors du champ de vision, mais plutôt une seule par bord du champ de vision. Également, à ce symbole, nous aurions ajouté un nombre indiquant la quantité d'éléments situés sur chaque bord. Tout ceci, dans le but de ne pas surcharger les informations affichées à l'utilisateur.

Par la suite, nous avons aussi souhaité ajouter une nouvelle gestuelle, similaire à la "Flower gesture", afin d'afficher ou désafficher le menu de sélection de fichiers. Malgré le fait que l'ajout d'une nouvelle gestuelle soit possible sur le casque HoloLens 1, cela implique d'utiliser un algorithme de Machine Learning. Il est possible de détecter une nouvelle gestuelle via des réseaux de neurones de Deep Learning. Le problème est qu'il faut entraîner ce réseau de neurones, sur une base de données importante. Il faut alors également créer cette base de données. Par conséquent, il faut accorder un certain temps d'entraînement pour permettre au réseau de détecter la nouvelle gestuelle. Enfin, il faut tester le réseau pour voir sa précision sur la gestuelle que l'on souhaite reconnaître. Cela demande un certain temps à consacrer. De plus, il faut permettre cette reconnaissance en temps réel et donc sûrement passer par un langage tel que le C++. Il existe également certaines API développées que nous aurions pu essayer. Cependant, ce n'est pas réellement l'objectif principal de ce projet et nous n'avons pas décidé de nous pencher sur ce problème plus longtemps.

Afin d'améliorer l'efficacité et l'utilité de l'ancrage d'un élément suivant la vue, nous aurions souhaité mettre en place une grille de placement. Celle-ci permettrait, lorsque plusieurs éléments sont ancrés suivant la vue, de ne pas les afficher les uns derrière les autres comme actuellement, mais suivant un quadrillage calculé par rapport aux zones de la vision périphérique que l'on remarque le plus. Cette solution est inspirée de l'article "Context-Aware Online Adaptation of Mixed Reality Interfaces" [10] qui adoptait aussi une grille de placement dans sa solution pour positionner les éléments efficacement. Cette fonctionnalité n'a néanmoins pas été ajoutée car, dans l'état de notre projet, le développement des modes de visualisation dans l'espace monde était prioritaire d'après nous.

Enfin, une fonctionnalité importante d'un tel projet est la gestion de l'occultation de l'environnement physique par les éléments virtuels. Les objets virtuels ne doivent normalement pas cacher des éléments réels ou des personnes. Cette problématique fait partie des éléments intéressants à résoudre dans le cas d'application de RA. Toutefois, de par les choix de développement que nous avons fait durant ce projet, nous avons préféré nous concentrer sur la visualisation et la stratégie d'ouverture et d'affichage des éléments plutôt que la gestion de cas.

4 User Story

Les "User Stories" sont des descriptions de besoins essentiels ou de fonctionnalités nécessaires au fonctionnement souhaité de l'application. Elles sont décomposées en tâches qui doivent être réalisées pour parvenir au besoin évoqué. Toutes les fonctionnalités sont ordonnées par ordre de priorité. En fonction de leur importance, elles sont notées de 1 à 3 (1 étant le niveau avec le plus de priorité). La priorité est déterminée par rapport à ce que nous jugeons être le plus important dans l'application. Pour ces besoins, la machine utilisée correspond à un casque HoloLens de première génération sur lequel l'application peut être déployée et lancée.

4.1 Utiliser le menu de selection de fichier

Priorité : 1

Description :

Un utilisateur ouvre l'application depuis le menu principal de l'appareil. Sans considérer le mode de visualisation (qui peut impliquer un comportement différent de l'application au démarrage), une surface représentant le menu de sélection de fichier s'affiche devant l'utilisateur. L'utilisateur peut cliquer, à l'aide de la gestuelle "Air Tap", sur les éléments de la liste des fichiers pour les sélectionner, sur le bouton de visualisation pour afficher les fichiers sélectionnés ou sur le bouton d'ancrage (nommé "anchor") pour changer le mode d'ancrage de l'objet.

Liste des tâches associées :

- Afficher un objet représentant le menu de sélection de fichiers.
- Permettre d'avoir toujours le menu de sélection ouvert.
- Adapter la liste des fichiers sélectionnables suivant ceux disponible dans l'application.
- Ajouter un message pour prévenir si aucun fichier n'est sélectionné.
- Sélectionner et charger un fichier (au format quelconque).
- Sélectionner et charger plusieurs fichiers (au format quelconque) en même temps.
- Ajouter un bouton pour gérer le mode d'ancrage de l'objet (dans le menu d'options)

Risques :

- Le menu de sélection ne s'affiche pas.
- La liste des fichiers est incorrect (manque de fichiers, doublons involontaires, etc...).
- Les fichiers sélectionnés ne sont pas ceux affichés (lorsque l'on appuie sur le bouton de visualisation).
- Les fichiers ne sont pas sélectionnés lorsque l'on clique sur les éléments correspondants dans la liste.

4.2 Ouvrir et afficher un seul fichier texte

Priorité : 1

Description :

Un utilisateur ouvre l'application depuis le menu principal de l'appareil. Il sélectionne un fichier (au format TXT) à afficher, dans la liste déroulante dans le menu de l'application. Un objet apparaît contenant le fichier texte sélectionné (en se positionnant suivant le mode de visualisation choisi ou devant l'utilisateur par défaut).

Liste des tâches associées :

- Charger un fichier texte dans un objet.
- Afficher un objet représentant le texte chargé.
- Permettre de redimensionner l'objet du fichier texte.
- Permettre de déplacer l'objet du fichier texte.
- Formater le texte pour l'afficher.
- Afficher un menu pour les boutons d'options d'affichage.
- Ajouter une option de fermeture de fichier (dans le menu d'options).
- Ajouter un bouton de changement de page (dans le menu d'options).
- Ajouter un bouton pour gérer l'ancrage de l'objet (dans le menu d'options).

Risques :

- Le fichier ne se charge pas correctement.
- Le fichier chargé n'est pas le bon.
- L'objet contenant le fichier ne s'affiche pas dans l'environnement.
- Les effets des options d'affichage ne s'appliquent pas sur le bon objet dans l'environnement.

4.3 Ouvrir et afficher un seul fichier image

Priorité : 1

Description :

Un utilisateur ouvre l'application depuis le menu principal de l'appareil. Il sélectionne un fichier (au format JPG ou PNG) à afficher, dans la liste déroulante dans le menu de l'application, à l'aide de la gestuelle "Air Tap". Un objet apparaît contenant le fichier image sélectionné (en se positionnant suivant le mode de visualisation choisi ou devant l'utilisateur par défaut).

Liste des tâches associées :

- Charger un fichier PNG ou JPG .
- Afficher un objet représentant l'image.
- Permettre de redimensionner l'objet contenant l'image.
- Permettre de déplacer l'objet contenant l'image.
- Positionner l'objet contenant l'image correctement.
- Afficher un menu pour les boutons d'options d'affichage.
- Ajouter une option de fermeture de fichier (dans le menu d'options).
- Ajouter un bouton pour gérer l'ancrage de l'objet (dans le menu d'options).

Risques :

- Le fichier ne se charge pas correctement.
- Le fichier chargé n'est pas le bon.
- L'objet contenant le fichier ne s'affiche pas dans l'environnement.
- Les effets des options d'affichage ne s'appliquent pas sur le bon objet dans l'environnement.

4.4 Gérer l'ancrage des objets dans l'espace

Priorité : 2

Description :

- **Scénario 1** Un utilisateur ouvre l'application depuis le menu principal de l'appareil. L'utilisateur clique, à l'aide de la gestuelle "Air tap", sur le bouton d'ancrage (nommé "anchor") dans le menu. Le mode d'ancrage de la vue est activé pour l'objet du menu. Celui-ci suit les déplacements de la tête de l'utilisateur pour rester devant lui, dans son champ de vision.
- **Scénario 2** Un utilisateur ouvre l'application depuis le menu principal de l'appareil. Il sélectionne un fichier à afficher, dans la liste déroulante dans le menu de l'application, à l'aide de la gestuelle "Air Tap". Un objet apparaît contenant le fichier sélectionné (en se positionnant suivant le mode de visualisation choisi ou devant l'utilisateur par défaut). L'utilisateur clique, à l'aide de la gestuelle "Air Tap", sur le bouton d'ancrage (nommé "anchor") qui se trouve dans le menu des options d'affichage. Le mode d'ancrage de la vue est activé pour cet objet. L'objet suit les déplacements de la tête de l'utilisateur pour rester devant lui, dans son champ de vision.

Liste des tâches associées :

- Afficher un objet dans le mode d'ancrage dans l'environnement (position fixe).
- Afficher un objet dans le mode d'ancrage suivant la vue de l'utilisateur.
- Permettre de passer d'un mode d'ancrage à l'autre.

Risques :

- Le mode d'ancrage suivant la vue ne fonctionne pas.
- Le mode d'ancrage ne peut plus changer.
- Le mode d'ancrage ne s'applique pas au bon objet.

4.5 Gérer le niveaux de détails d'un objet

Priorité : 2

Description :

- **Scénario 1** Un utilisateur ouvre l'application depuis le menu principal de l'appareil. Il sélectionne un fichier texte à afficher, dans la liste déroulante dans le menu de l'application, à l'aide de la gestuelle "Air Tap". Un objet apparaît contenant le fichier sélectionné (en se positionnant suivant le mode de visualisation choisi ou devant l'utilisateur par défaut). L'utilisateur se déplace (ou déplace l'objet) ce qui va adapter la taille du texte en conséquence.

- **Scénario 2** Un utilisateur ouvre l'application depuis le menu principal de l'appareil. Il sélectionne un fichier image (JPG ou PNG) à afficher, dans la liste déroulante dans le menu de l'application, à l'aide de la gestuelle "Air Tap". Un objet apparaît contenant le fichier sélectionné (en se positionnant suivant le mode de visualisation choisi ou devant l'utilisateur par défaut). L'utilisateur se déplace (ou déplace l'objet) ce qui va adapter la taille de l'image en conséquence, en rognant l'image par exemple.

Liste des tâches associées :

- Adapter la taille du texte en fonction de la distance.
- Mettre à jour les pages lorsque le niveau de détail change.
- Adapter la taille d'une image en fonction de la distance.
- Permettre de passer d'un mode d'ancrage à l'autre.

Risques :

- Le niveaux de détails de l'objet ne change pas.
- Le niveaux de détails de l'objet ne reviens pas à un état normal.
- Le niveaux de détails de l'objet à des valeurs extrêmes.

4.6 Gérer l'étiquetages de zones et de fichiers

Priorité : 1

Description :

- **Scénario 1** L'utilisateur ouvre l'application avec la gestuelle "Air Tap" depuis le menu principal du HoloLens. Il choisit ensuite les zones étiquetées qu'il souhaite placer en les sélectionnant depuis le menu avec la gestuelle "Air Tap". Ces zones apparaissent devant lui et il peut alors les positionner où il souhaite dans l'espace avec un "Pinch". Une fois cela fait, il peut passer au menu d'affichage des fichiers en appuyant sur le bouton "Done" avec la gestuelle "Air Tap". Enfin, il ouvre des fichiers image ou texte liés aux zones étiquetées précédemment placées 4.2 4.3. Ces fichiers s'ouvrent autour des zones correspondantes.
- **Scénario 2** L'utilisateur ouvre l'application avec la gestuelle "Air Tap" depuis le menu principal du HoloLens. Il choisit ensuite de ne placer aucune zone étiquetée. Il passe alors au menu d'affichage des fichiers en appuyant sur le bouton "Done" avec la gestuelle "Air Tap". Enfin, il ouvre des fichiers image ou texte, qui s'ouvrent devant lui puisqu'aucune zone correspondante n'a été placée 4.2 4.3.

Liste des tâches associées :

- Créer une scène dédiée à l'étiquetage de l'environnement.
- Mettre en place un menu pour sélectionner des zones étiquetées à placer dans l'environnement.
- Etiqueter les fichiers chargés.
- Afficher les fichiers chargés à côté des zones étiquetées.
- Créer un bouton pour modifier les zones étiquetées.

Risques :

- Le menu de sélection de zone ne s'affiche pas.
- Les zones étiquetées ne s'affichent pas lorsqu'on les sélectionne.
- Les fichiers ne s'ouvrent pas autour des zones alors qu'elles sont placées.
- Les fichiers se placent aux mauvais endroits autour des zones.

4.7 Création d'une carte de chaleur pour afficher des fichiers aux endroits les plus regardés par l'utilisateur

Priorité : 1

Description :

- **Scénario 1** L'utilisateur ouvre l'application avec la gestuelle "Air Tap" depuis le menu principal du HoloLens. Un bouton "Start" apparaît devant lui. L'utilisateur clique dessus avec la gestuelle "Air Tap". L'utilisateur dispose de 20 secondes à présent pour regarder autour de lui, lorsqu'il regarde une zone une petite sphère de couleur verte apparaît. Plus son regard est posé aux alentours ou dessus plus cette sphère s'agrandit et rougit. Après ces 20 secondes, le menu principal apparaît et l'utilisateur peut choisir dans la liste déroulante les fichiers qu'il souhaite disposer aux niveaux des trois zones les plus regardées. Lorsqu'il confirme son choix via la gestuelle "Air Tap" sur le bouton "Visualize", les fichiers apparaissent autour des zones concernées, en gravitation. Les fichiers sont répartis équitablement autour des centres des zones élues.

- **Scénario 2** L'utilisateur effectue l'ensemble du scénario précédent. Néanmoins, lors de l'affichage des fichiers, il se rend compte que ces zones ne lui conviennent pas. Il clique alors sur le bouton "Scan", qui relance le décompte des 20 secondes, et il peut désormais regarder les zones qui lui semblent plus favorables. Evidemment, les fichiers ouverts précédemment restent à leur position. Après la fin du décompte, le menu de sélection de fichiers réapparaît et il peut alors choisir les fichiers à afficher qui seront disposés sur les trois nouvelles zones les plus regardées.

Liste des tâches associées :

- Créer une carte de chaleur (heatmap) contenant les positions et les intensités de chacun des points regardés par l'utilisateur.
- Afficher un retour visuel pendant la période de scan, qui est l'apparition de sphères vertes au départ, et qui grossissent et rougissent si l'utilisateur pose son regard longtemps dessus.
- Mettre à jour les intensités de chaque zone.
- S'assurer de la disposition des fichiers, équitablement répartis sur les trois zones, autour des centres de ces zones les plus regardées. Les fichiers s'organisent autour du centre de leur zone selon un système de forces d'attraction et de répulsion. Les fichiers se repoussent et s'attirent aussi entre eux.
- Ajouter un bouton pour relancer un scan de l'environnement.

Risques :

- Lors du scan, les sphères ne grossissent pas et ne change pas de couleur en conséquence.
- Les fichiers ne se disposent pas autour des trois zones les plus regardées.

4.8 Implémenter un système de forces pour améliorer la disposition des fichiers

Priorité : 1

Description :

- **Scénario 1** L'utilisateur effectue l'ensemble du scénario 1 précédent 4.7. Pour chaque zone, les fichiers sont repoussés du centre jusqu'à une certaine distance où ils sont attirés par ce centre. Pour chaque fichier de la même zone, les fichiers se repoussent entre eux jusqu'à une certaine distance et s'attirent entre eux.

Liste des tâches associées :

- Etablir une relation pour calculer la force ressort à appliquer entre deux éléments.
- Ajouter une vitesse à chacun des objets représentant les fichiers pour être ensuite ajoutée à la position de ces objets, leur permettant de bouger.
- Etablir le système de forces entre les fichiers, pour l'ensemble des fichiers d'une zone.
- Etablir le système de forces entre les fichiers et le centre d'une même zone.

Risques :

- Malgré le fait que les fichiers se repoussent, ils peuvent s'installer l'un devant l'autre en profondeur empêchant une bonne visualisation.
- Si un très grand nombre de fichiers est ouvert, les fichiers se retrouvent assez loin de la zone fixée de base.
- Si deux zones choisies par l'utilisateur se trouvent assez proches, il se peut que les fichiers d'une zone se retrouvent sur celle de l'autre surtout lorsqu'un grand nombre de fichiers sont ouverts.

5 Tests

Dans cette partie, nous cherchons à décrire dans un premier temps les tests unitaires, qui vont tester en priorité les méthodes que l'on a écrit dans nos scripts, dans une deuxième partie nous décrivons les test MRTK, c'est à dire les tests suivant un scénario qui nécessitent donc de simuler l'application.

5.1 Tests unitaires

5.1.1 Tests de chargement d'un texte

- **Données :** un casque HoloLens et l'application développée déployée sur ce casque.
- **Résultats attendus :**
 - positif : le texte est chargé et affiché.
 - négatif : le texte n'est pas chargé.
- **Scénario positif :**
 - Un objet pouvant accueillir un texte est créé.
 - Le script charge le fichier texte comme texte de l'objet.
 - Le texte de l'objet n'est pas nul.
- **Scénario négatif :**
 - Un objet pouvant accueillir un texte est créé.
 - Le script ne charge pas le fichier texte comme texte de l'objet.
 - Le texte de l'objet est nul.

5.1.2 Tests de chargement d'une image

- **Données :** un casque HoloLens et l'application développée déployée sur ce casque.
- **Résultats attendus :**
 - positif : l'image est chargée et affichée.
 - négatif : l'image n'est pas chargée.
- **Scénario positif :**
 - Un objet pouvant accueillir une image est créé.
 - Le script charge le fichier image comme texture de l'objet.
 - La texture de l'objet n'est pas nulle.
- **Scénario négatif :**
 - Un objet pouvant accueillir une image est créé.
 - Le script ne charge pas le fichier image comme texture de l'objet.
 - La texture de l'objet est nulle.

5.1.3 Tests de fermeture d'un fichier (texte et image)

- **Données :** un casque HoloLens, l'application développée déployée sur ce casque et un fichier ouvert dans l'application.
- **Résultats attendus :**
 - positif : le fichier est fermé.
 - négatif : le fichier n'est pas fermé.
- **Scénario positif :**
 - Un objet pouvant être fermé est créé.
 - L'objet est affecté au script de fermeture.
 - La fonction de fermeture est exécutée.
 - L'objet est correctement détruit et donc fermé.
- **Scénario négatif :**
 - Un objet pouvant être fermé est créé.
 - L'objet est affecté au script de fermeture.
 - La fonction de fermeture est exécutée.
 - L'objet n'est pas détruit et donc pas fermé.

5.1.4 Tests de changement vers la page suivante pour un fichier texte

- **Données :** un casque HoloLens, l'application développée déployée sur ce casque et un fichier texte ouvert dans l'application.
- **Résultats attendus :**
 - positif : la page actuelle du fichier texte est la suivante.
 - négatif : la page actuelle du fichier texte n'est pas la suivante.
- **Scénario positif :**
 - Un objet texte est créé.
 - La page actuelle est sauvegardée.
 - La fonction de changement à la page suivante est exécutée.
 - La page actuelle est supérieure à celle sauvegardée.
- **Scénario négatif :**
 - Un objet texte est créé.
 - La page actuelle est sauvegardée.
 - La fonction de changement à la page suivante est exécutée.
 - La page actuelle est inférieure ou égale à celle sauvegardée.

5.1.5 Tests de changement vers la page précédente pour un fichier texte

- **Données :** un casque HoloLens, l'application développée déployée sur ce casque et un fichier texte ouvert dans l'application.
- **Résultats attendus :**
 - positif : la page actuelle du fichier texte est la précédente.
 - négatif : la page actuelle du fichier texte n'est pas la précédente.
- **Scénario positif :**
 - Un objet texte est créé.
 - La page actuelle est sauvegardée.
 - La fonction de changement à la page précédente est exécutée.
 - La page actuelle est inférieure à celle sauvegardée.
- **Scénario négatif :**
 - Un objet texte est créé.
 - La page actuelle est sauvegardée.
 - La fonction de changement à la page précédente est exécutée.
 - La page actuelle est supérieure ou égale à celle sauvegardée.

5.1.6 Tests de création d'une zone étiquetée

- **Données :** un casque HoloLens, l'application développée déployée sur ce casque.
- **Résultats attendus :**
 - positif : la zone étiquetée est créée.
 - négatif : la zone étiquetée n'est pas créée.
- **Scénario positif :**
 - Un objet pouvant gérer les zones étiquetées est créé.
 - On compte le nombre de zones étiquetées.
 - La fonction de génération d'une zone étiquetée est exécutée.
 - Le nombre de zones étiquetées a augmenté.
- **Scénario négatif :**
 - Un objet pouvant gérer les zones étiquetées est créé.
 - On compte le nombre de zones étiquetées.
 - La fonction de génération d'une zone étiquetée est exécutée.
 - Le nombre de zones étiquetées n'a pas augmenté.

5.1.7 Tests de changement vers la scène de sélection de fichier

- **Données :** un casque HoloLens, l'application développée déployée sur ce casque.
- **Résultats attendus :**
 - positif : la scène actuelle est celle de sélection de fichier.
 - négatif : la scène actuelle n'est pas celle de sélection de fichier.
- **Scénario positif :**
 - Un objet pouvant gérer les zones étiquetées est créé.
 - La fonction de changement de scène est exécutée.
 - La scène actuelle est égale à la scène de sélection de fichier.
- **Scénario négatif :**
 - Un objet pouvant gérer les zones étiquetées est créé.
 - La fonction de changement de scène est exécutée.
 - La scène actuelle n'est pas égale à la scène de sélection de fichier.

5.1.8 Tests de sélection d'un fichier

- **Données :** un casque HoloLens, l'application développée déployée sur ce casque.
- **Résultats attendus :**
 - positif : le fichier est sélectionné.
 - négatif : le fichier n'est pas sélectionné.
- **Scénario positif :**
 - Un objet pouvant gérer les fichiers est créé.
 - Un fichier est créé.
 - Le nombre de fichiers créés est sauvegardé.
 - La fonction de sélection de fichier est exécutée avec le fichier.
 - Le nombre de fichiers créés est supérieur au nombre sauvegardé.
- **Scénario négatif :**
 - Un objet pouvant gérer les fichiers est créé.
 - Un fichier est créé.
 - Le nombre de fichiers créés est sauvegardé.
 - La fonction de sélection de fichier est exécutée avec le fichier.
 - Le nombre de fichiers créés est inférieur ou égal au nombre sauvegardé.

5.1.9 Tests de visualisation d'un fichier

- **Données :** un casque HoloLens, l'application développée déployée sur ce casque.
- **Résultats attendus :**
 - positif : le fichier est affiché.
 - négatif : le fichier n'est pas affiché.
- **Scénario positif :**
 - Un objet pouvant gérer les fichiers est créé.
 - Un fichier est ajouté à la liste des fichiers sélectionnés.
 - Le nombre de fichiers créés est sauvegardé.
 - La fonction de visualisation est exécutée avec le fichier.
 - Le nombre de fichiers créés est supérieur au nombre sauvegardé.
- **Scénario négatif :**
 - Un objet pouvant gérer les fichiers est créé.
 - Un fichier est ajouté à la liste des fichiers sélectionnés.
 - Le nombre de fichiers créés est sauvegardé.
 - La fonction de visualisation est exécutée avec le fichier.
 - Le nombre de fichiers créés est inférieur ou égal au nombre sauvegardé.

5.1.10 Tests du changement du niveau de détail d'un texte

- **Données :** un casque HoloLens, l'application développée déployée sur ce casque et un fichier texte ouvert dans l'application.
- **Résultats attendus :**
 - positif : le niveau de détail a changé.
 - négatif : le niveau de détail n'a pas changé.
- **Scénario positif :**
 - Un objet pouvant accueillir un texte est créé.
 - Le script charge le fichier texte comme texture de l'objet.
 - La taille de la police du texte est sauvegardée.
 - La caméra principale est éloignée du fichier texte.
 - La taille de la police a augmenté par rapport à la taille sauvegardée.
- **Scénario négatif :**
 - Un objet pouvant accueillir un texte est créé.
 - Le script charge le fichier texte comme texture de l'objet.
 - La taille de la police du texte est sauvegardée.
 - La caméra principale est éloignée du fichier texte.
 - La taille de la police est inférieure ou égale par rapport à la taille sauvegardée.

5.1.11 Tests du changement du niveau de détail d'une image

- **Données :** un casque HoloLens, l'application développée déployée sur ce casque et un fichier image ouvert dans l'application.
- **Résultats attendus :**
 - positif : le niveau de détail a changé.
 - négatif : le niveau de détail n'a pas changé.
- **Scénario positif :**
 - Un objet pouvant accueillir une image est créé.
 - Le script charge le fichier image comme texture de l'objet.
 - La taille de l'image est sauvegardée.
 - La caméra principale est éloignée du fichier image.
 - La taille de l'image a augmenté par rapport à la taille sauvegardée.
- **Scénario négatif :**
 - Un objet pouvant accueillir une image est créé.
 - Le script charge le fichier image comme texture de l'objet.
 - La taille de l'image du texte est sauvegardée.
 - La caméra principale est éloignée du fichier image.
 - La taille de l'image est inférieure ou égale par rapport à la taille sauvegardée.

5.2 Tests MRTK

L'ensemble MRTK dispose d'une librairie de tests, permettant d'écrire des tests sur l'application comme si l'on s'en servait sur le HoloLens. Il est par exemple possible de créer une main virtuelle (comme si c'était la notre que nous utilisions), de la faire pointer à un endroit précis dans la scène, ou d'effectuer les diverses actions possibles telles que le clic, le pincement (Pinch) et le déplacement etc. Il est aussi possible de simuler un déplacement du regard en faisant bouger la caméra principale dans l'éditeur. Ainsi avec l'ensemble de ces fonctionnalités, il est possible de créer des multitudes de scénarios pour vérifier le bon fonctionnement de notre application.

5.2.1 Tests de l'ancrage du menu

- **Données :** un casque HoloLens et l'application développée déployée sur ce casque.
- **Résultats attendus :**
 - positif : le menu suit le regard de l'utilisateur après avoir cliqué sur le toggle switch portant le nom "anchor".

— négatif : le menu ne suit pas le regard de l'utilisateur.

- **Scénarios positifs :**

- **Scénario 1**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur en étant ancrée en mode "monde".
 - L'utilisateur clique sur le toggle switch avec le nom "anchor" via la gestuelle "Air tap".
 - Le toggle switch est maintenant de couleur bleue ce qui signifie que le mode ancrage est actif.
 - L'utilisateur se déplace dans la pièce.
 - Le menu se trouve en permanence devant le regard de l'utilisateur.

- **Scénario 2**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur en étant ancrée en mode "monde".
 - L'utilisateur clique sur le toggle switch avec le nom "anchor" via la gestuelle "Air tap".
 - Le toggle switch est maintenant de couleur bleue ce qui signifie que le mode ancrage est actif.
 - L'utilisateur se déplace dans la pièce.
 - Le menu se trouve en permanence devant le regard de l'utilisateur.
 - L'utilisateur décide de désactiver le mode "ancrage" en cliquant avec la gestuelle "Air tap" sur le toggle switch "anchor".
 - L'utilisateur bouge le regard ou se déplace dans la pièce.
 - Le menu ne suit plus le regard de l'utilisateur et reste à la dernière place occupée.

- **Scénario 3**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur en étant ancrée en mode "monde".
 - L'utilisateur clique sur le toggle switch avec le nom "anchor" via la gestuelle "Air tap".
 - Le toggle switch est maintenant de couleur bleue ce qui signifie que le mode ancrage est actif.
 - L'utilisateur décide d'effectuer une autre action sur le menu (dérouler la liste, sélectionner des fichiers etc).
 - L'utilisateur déplace son regard en dehors du menu.
 - Le menu suit le regard de l'utilisateur.

- **Scénarios négatifs :**

- **Scénario 1**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur en ancrage "monde".
 - L'utilisateur décide de ne pas déclencher le mode "ancrage".
 - L'utilisateur se déplace dans son environnement, ou tourne le regard.
 - Le menu ne suit pas le regard de l'utilisateur.

- **Scénario 2**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur en ancrage "monde".
 - L'utilisateur décide de se placer à l'arrière du menu principal de l'application.
 - L'utilisateur clique sur le bouton d'ancrage depuis cette position via la gestuelle de "Air tap".
 - Le bouton "anchor" ne change pas de couleur.
 - L'utilisateur se déplace dans la pièce.
 - Le menu ne suit pas le regard de l'utilisateur.

5.2.2 Tests du déplacement du menu

- **Données :** un casque HoloLens et l'application développée déployée sur ce casque.

- **Résultats attendus :**

- positif : le menu a été déplacé selon le souhait de l'utilisateur.
 - négatif : le menu n'a pas été déplacé malgré le souhait de l'utilisateur car il a effectué la mauvaise

action pour atteindre ce résultat.

- **Scénarios positifs :**

- **Sénario 1**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
- L'utilisateur décide de sélectionner le menu en plaçant son regard sur le fond de ce dernier et le sélectionne avec la gestuelle "Pinch".
- L'utilisateur déplace le menu qui a été sélectionné où il le souhaite.
- Le menu a été déplacé à la place voulue.
- L'utilisateur désélectionne le menu en relâchant la gestuelle "Pinch".
- L'utilisateur bouge le curseur via son regard.
- Le menu est resté à la dernière position occupée.

- **Sénario 2**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
- L'utilisateur décide de se placer à l'arrière du menu.
- L'utilisateur sélectionne l'arrière du menu avec la gestuelle "Pinch".
- L'utilisateur déplace le menu avec sa main.
- Le menu a été déplacé à la place voulue.

- **Scénarios négatifs :**

- **Scénario 1**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
- L'utilisateur décide de déplacer son regard pour faire bouger le pointeur.
- Le menu n'a pas été déplacé.

- **Scénario 2**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur en ancrage "monde".
- L'utilisateur décide de regarder à côté du menu, dans le vide, et effectue la gestuelle de "Pinch".
- L'utilisateur déplace son regard, afin de bouger le curseur.
- Le menu n'a pas été déplacé.

- **Scénario 3**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur en ancrage "monde".
- L'utilisateur décide de regarder un bouton ou le fond du menu, et effectue la gestuelle "Pinch".
- Le menu n'a pas été déplacé.

5.2.3 Tests de redimensionnement du menu

- **Données :** un casque HoloLens et l'application développée déployée sur ce casque.

- **Résultats attendus :**

- positif : le menu est agrandi selon le souhait de l'utilisateur.
- négatif : le menu ne change pas de taille malgré l'action de l'utilisateur.

- **Scénarios positifs :**

- **Sénario 1**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
- L'utilisateur décide de sélectionner un coin du menu afin de l'agrandir avec la gestuelle "Pinch".
- L'utilisateur déplace ce coin plus ou moins loin de sa position initiale.
- Le menu est ajusté à sa nouvelle taille.

- **Sénario 2**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".

- L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
- L'utilisateur décide de se placer à l'arrière du menu.
- L'utilisateur sélectionne un coin du menu avec la gestuelle "Pinch".
- L'utilisateur déplace ce coin plus ou moins loin de sa position de départ.
- Le menu est redimensionné à sa nouvelle taille.

- **Scénario négatif :**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
- L'utilisateur décide de sélectionner un des coins du menu sans la gestuelle "Pinch".
- L'utilisateur essaie de déplacer le coin sélectionné plus ou moins loin de la position initiale.
- Le menu ne change pas de taille.

5.2.4 Tests de la rotation du menu

- **Données :** un casque HoloLens et l'application développée déployée sur ce casque.

- **Résultats attendus :**

- positif : le menu est orienté selon le souhait de l'utilisateur.
- négatif : le menu ne change pas d'orientation malgré l'action de l'utilisateur.

- **Scénarios positifs :**

- **Sénario 1**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
- L'utilisateur décide de sélectionner la sphère sur une arête du menu afin d'activer l'action de rotation de l'objet.
- L'utilisateur déplace cette sphère autour de l'objet à un angle plus ou moins différent de sa position initiale.
- Le menu est ajusté à sa nouvelle orientation.

- **Sénario 2**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
- L'utilisateur décide de se placer à l'arrière du menu.
- L'utilisateur décide de sélectionner la sphère sur une arête du menu afin d'activer l'action de rotation de l'objet.
- L'utilisateur déplace cette sphère autour de l'objet à un angle plus ou moins différent de sa position initiale.
- Le menu est ajusté à sa nouvelle orientation.

- **Scénario négatif :**

- **Sénario 1**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
- L'utilisateur décide de sélectionner la sphère sur une arête du menu afin d'activer l'action de rotation de l'objet mais sans activer la gestuelle "Pinch".
- L'utilisateur déplace cette sphère autour de l'objet à un angle plus ou moins différent de sa position initiale.
- L'orientation du menu n'a pas changé.

- **Sénario 2**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
- L'utilisateur décide de sélectionner la sphère sur une arête du menu afin d'activer l'action de rotation de l'objet en activant la gestuelle "Pinch".
- L'utilisateur ne déplace pas le curseur qui sélectionne toujours le bord autour de l'objet.
- L'orientation du menu n'a pas changé.

- **Sénario 3**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.

- L'utilisateur décide de sélectionner la sphère sur une arête du menu afin d'activer l'action de rotation de l'objet en activant la gestuelle "Pinch".
- L'utilisateur décide de relâcher la sphère puis de sélectionner un coin du menu afin d'activer l'action de redimension de l'objet à l'aide de la gestuelle "Pinch".
- L'utilisateur déplace le curseur pour déplacer le coin.
- L'orientation du menu n'a pas changé.

5.2.5 Tests d'affichage de la liste des fichiers accessibles

- **Données :** un casque HoloLens et l'application développée déployée sur ce casque.
- **Résultats attendus :**
 - positif : la liste est déroulable, l'ensemble des fichiers est visible dans cette liste.
 - négatif : la liste ne défile pas si l'utilisateur ne place pas le curseur sur la scrollbar associée à la liste.
- **Scénarios positifs :**
 - **Sénario 1**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur décide de placer le pointeur avec son regard, sur la scrollbar.
 - L'utilisateur attrape la scrollbar avec la gestuelle "Pinch".
 - L'utilisateur descend (ou monte) la scrollbar en conservant cette gestuelle et en bougeant sa main vers le bas (ou le haut).
 - La liste défile et contient le nom des fichiers sélectionnables.
 - **Sénario 2**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur décide de placer le pointeur avec son regard, sur la scrollbar.
 - L'utilisateur attrape la scrollbar avec la gestuelle "Pinch".
 - L'utilisateur descend (ou monte) la scrollbar en conservant cette gestuelle et en bougeant sa main vers le bas (ou le haut).
 - La liste défile et contient le nom des fichiers sélectionnables.
 - L'utilisateur décide de lâcher la gestuelle "Pinch".
 - La liste ne défile plus.
 - **Sénario 3**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur décide de placer le pointeur avec son regard, sur la scrollbar.
 - L'utilisateur attrape la scrollbar avec la gestuelle "Pinch".
 - L'utilisateur descend (ou monte) la scrollbar en conservant cette gestuelle et en bougeant sa main vers le bas (ou le haut).
 - La liste défile et contient le nom des fichiers sélectionnables.
 - L'utilisateur décide de lâcher la gestuelle "Pinch".
 - La liste ne défile plus.
 - L'utilisateur décide de déplacer le menu, ou de l'incliner.
 - L'utilisateur décide de sélectionner des fichiers dans la liste.
 - L'utilisateur place le curseur sur le bouton pour visualiser les fichiers et applique la gestuelle "Pinch".
 - Les fichiers apparaissent devant lui.
 - La scrollbar du menu n'a pas été réinitialisée à sa position de départ et n'a pas bougé.
- **Scénario négatif :**
 - **Sénario 1**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
 - L'utilisateur place le curseur sur la scrollbar sans effectuer la gestuelle "Pinch".
 - L'utilisateur déplace sa main vers le bas (ou le haut) pour parcourir la liste.
 - La liste reste figée et ne défile pas.
 - **Sénario 2**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
- L'utilisateur place le curseur ailleurs que sur la scrollbar et effectue la gestuelle "Pinch".
- L'utilisateur déplace sa main vers le bas (ou le haut) pour parcourir la liste.
- La liste reste figée et ne défile pas.

5.2.6 Tests de sélection des fichiers

- **Données :** un casque HoloLens et l'application développée déployée sur ce casque.
- **Résultats attendus :**
 - positif : les fichiers sélectionnés dans le menu sont cochés dans la liste des fichiers et le nom du fichier s'affiche dans une liste résumant tous ceux sélectionnés.
 - négatif : les fichiers non-sélectionnés dans le menu ne sont pas cochés dans la liste des fichiers et le nom du fichier ne s'affiche pas dans une liste résumant tous ceux sélectionnés.
- **Scénarios positifs :**
 - **Sénario 1**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur décide de sélectionner, à l'aide de la gestuelle "Pinch", un fichier en déplaçant le curseur sur la liste des fichiers accessibles.
 - La case à cocher, à côté du nom du fichier dans la liste, se coche automatiquement.
 - Le nom du fichier apparaît dans la liste résumant les fichiers sélectionnés situés à droite dans le menu.
 - **Sénario 2**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur décide de sélectionner, à l'aide de la gestuelle "Pinch" un fichier en déplaçant le curseur sur la liste des fichiers accessibles.
 - La case à cocher, à côté du nom du fichier dans la liste, se coche automatiquement.
 - Le nom du fichier apparaît dans la liste résumant les fichiers sélectionnés situés à droite dans le menu.
 - L'utilisateur décide de sélectionner, à l'aide de la gestuelle "Pinch", un deuxième fichier en déplaçant le curseur sur la liste des fichiers accessibles.
 - La case à cocher, à côté du nom du deuxième fichier dans la liste, se coche automatiquement.
 - Le nom du deuxième fichier apparaît dans la liste, dans laquelle il y a déjà le nom du premier fichier, résumant les fichiers sélectionnés situés à droite dans le menu.
 - **Sénario 3**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur décide de sélectionner, à l'aide de la gestuelle "Pinch" un fichier en déplaçant le curseur sur la liste des fichiers accessibles.
 - La case à cocher, à côté du nom du fichier dans la liste, se coche automatiquement.
 - Le nom du fichier apparaît dans la liste résumant les fichiers sélectionnés situés à droite dans le menu.
 - L'utilisateur décide de sélectionner, à l'aide de la gestuelle "Pinch", une nouvelle fois le même fichier en déplaçant le curseur sur la liste des fichiers accessibles.
 - La case à cocher du fichier dans la liste est désormais décoché automatiquement.
 - Le nom du fichier n'est plus dans la liste résumant les fichiers sélectionnés, situés à droite dans le menu.
 - **Sénario 4**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur décide de sélectionner, à l'aide de la gestuelle "Pinch" un fichier en déplaçant le curseur sur la liste des fichiers accessibles.
 - La case à cocher, à côté du nom du fichier dans la liste, se coche automatiquement.
 - Le nom du fichier apparaît dans la liste résumant les fichiers sélectionnés situés à droite dans le menu.

- L'utilisateur décide de cliquer sur le bouton de visualisation, en déplaçant le curseur et à l'aide de la gestuelle "Pinch".
- Un nouvel objet, correspondant au fichier sélectionné, s'affiche (en suivant le mode de visualisation correspondant).
- La case à cocher du fichier dans la liste est désormais décoché automatiquement.
- Le nom du fichier n'est plus dans la liste résumant les fichiers sélectionnés, situés à droite dans le menu.

- **Scénario négatif :**

- **Sénario 1**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur décide de placer le curseur sur un fichier dans la liste des fichiers accessibles, sans activer la gestuelle "Pinch".
 - Les éléments du menu (cose à cocher dans la liste et liste des fichiers déjà sélectionnés) n'ont pas été modifiés.

5.2.7 Tests d'ouverture et d'affichage de fichiers via le menu

- **Données :** un casque HoloLens et l'application développée déployée sur ce casque.

- **Résultats attendus :**

- positif : les fichiers sélectionnés dans la liste sont visibles une fois le bouton "visualize" pressé.
- négatif : une pop-up apparaît décrivant à l'utilisateur qu'au moins un fichier doit être sélectionné pour être visualisé.

- **Scénarios positifs :**

- **Sénario 1**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un ou plusieurs fichiers dans la liste via la gestuelle "Pinch".
 - L'utilisateur place le curseur avec son regard sur le bouton visualize et valide avec la gestuelle "Pinch".
 - Les fichiers sélectionnés apparaissent devant les yeux de l'utilisateur.

- **Sénario 2**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un ou plusieurs fichiers dans la liste via la gestuelle "Pinch".
 - L'utilisateur place le curseur avec son regard sur le bouton visualize et valide avec la gestuelle "Pinch".
 - Les fichiers sélectionnés apparaissent devant les yeux de l'utilisateur.
 - L'utilisateur décide de sélectionner le ou les mêmes fichiers qui ont été affichés.
 - L'utilisateur confirme son choix avec le bouton visualize.
 - Les mêmes fichiers sont affichés devant les yeux de l'utilisateur.

- **Sénario 3**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur décide de redimensionner la taille du menu et le changer de position.
 - L'utilisateur sélectionne un ou plusieurs fichiers dans la liste via la gestuelle "Pinch".
 - L'utilisateur place le curseur avec son regard sur le bouton visualize et valide avec la gestuelle "Pinch".
 - Les fichiers sélectionnés apparaissent devant les yeux de l'utilisateur.

- **Scénario négatif :**

- **Sénario 1**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
 - L'utilisateur ne sélectionne aucun fichier dans la liste déroulante.
 - L'utilisateur place son curseur sur le bouton visualize et effectue la gestuelle "Pinch".
 - La pop-up apparaît précisant qu'il n'y a aucun fichiers de sélectionnés.

- **Sénario 2**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
- L'utilisateur décide de se placer à l'arrière de la liste déroulante (et du menu).
- L'utilisateur essaie de sélectionner un fichier avec la gestuelle "Pinch".
- L'utilisateur se replace devant le menu et confirme son choix.
- La pop-up apparaît indiquant qu'aucun fichier n'a été sélectionné.

5.2.8 Tests de fermeture de fichiers

- **Données :** un casque HoloLens et l'application développée déployée sur ce casque.

- **Résultats attendus :**

- positif : une fois le bouton "close" pressé, associé à un certain fichier, le fichier concerné disparaît.
- négatif : le bouton "close" n'a pas été pressé et le fichier reste toujours ouvert.

- **Scénarios positifs :**

- **Sénario 1**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
- L'utilisateur sélectionne un ou plusieurs fichiers dans la liste via la gestuelle "Pinch".
- L'utilisateur place le curseur avec son regard sur le bouton visualize et valide avec la gestuelle "Pinch".
- Les fichiers sélectionnés apparaissent devant les yeux de l'utilisateur.
- L'utilisateur place le pointeur sur le bouton "close" avec son regard.
- L'utilisateur appuie sur ce bouton avec la gestuelle "Pinch".
- Le fichier disparaît, il est fermé.

- **Sénario 2**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
- L'utilisateur sélectionne un ou plusieurs fichiers dans la liste via la gestuelle "Pinch".
- L'utilisateur place le curseur avec son regard sur le bouton visualize et valide avec la gestuelle "Pinch".
- Les fichiers sélectionnés apparaissent devant les yeux de l'utilisateur.
- L'utilisateur décide d'agrandir, de déplacer ou incliner un de ces fichiers.
- L'utilisateur place le pointeur sur le bouton "close" du même fichier avec son regard.
- L'utilisateur appuie sur ce bouton avec la gestuelle "Pinch".
- Le fichier disparaît, il est fermé.

- **Scénario négatif :**

- **Sénario 1**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
- L'utilisateur sélectionne un ou plusieurs fichiers dans la liste via la gestuelle "Pinch".
- L'utilisateur place le curseur avec son regard sur le bouton visualize et valide avec la gestuelle "Pinch".
- Les fichiers sélectionnés apparaissent devant les yeux de l'utilisateur.
- L'utilisateur place le pointeur sur un des fichiers positionnés devant lui.
- L'utilisateur applique la gestuelle "Pinch".
- Le fichier reste ouvert et peut être déplacé.

- **Sénario 2**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
- L'utilisateur sélectionne un ou plusieurs fichiers dans la liste via la gestuelle "Pinch".
- L'utilisateur place le curseur avec son regard sur le bouton visualize et valide avec la gestuelle "Pinch".
- Les fichiers sélectionnés apparaissent devant les yeux de l'utilisateur.
- L'utilisateur place le pointeur sur le bouton close ou un autre d'un des fichiers.
- Le fichier reste présent.

5.2.9 Test de déplacement d'un objet correspondant à un fichier

- **Données :** un casque HoloLens et l'application développée déployée sur ce casque.
- **Résultats attendus :**
 - positif : l'objet a été déplacé selon le souhait de l'utilisateur.
 - négatif : l'objet n'a pas été déplacé malgré le souhait de l'utilisateur car il a effectué la mauvaise action pour atteindre ce résultat.
- **Scénarios positifs :**
 - **Sénario 1**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un fichier dans la liste et clique sur le bouton de visualisation du menu.
 - Un nouvel objet, correspondant au fichier sélectionné, s'affiche (en suivant le mode de visualisation correspondant).
 - L'utilisateur décide de sélectionner l'objet en plaçant le curseur dessus et en le sélectionnant avec la gestuelle "Pinch".
 - L'utilisateur déplace l'objet qui a été sélectionné où il le souhaite, en déplaçant le curseur.
 - L'objet a été déplacé à la position voulue.
 - **Sénario 2**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un fichier dans la liste et clique sur le bouton de visualisation du menu.
 - Un nouvel objet, correspondant au fichier sélectionné, s'affiche (en suivant le mode de visualisation correspondant).
 - L'utilisateur décide de sélectionner l'objet en plaçant le curseur dessus et en le sélectionnant avec la gestuelle "Pinch".
 - L'utilisateur déplace l'objet qui a été sélectionné où il le souhaite, en déplaçant le curseur.
 - L'objet a été déplacé à la position voulue.
 - L'utilisateur désélectionne l'objet en relâchant la gestuelle "Pinch".
 - L'utilisateur bouge le curseur via son regard.
 - L'objet est resté à la dernière position occupée.
 - **Sénario 3**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un fichier dans la liste et clique sur le bouton de visualisation du menu.
 - Un nouvel objet, correspondant au fichier sélectionné, s'affiche (en suivant le mode de visualisation correspondant).
 - L'utilisateur décide de se placer à l'arrière de l'objet.
 - L'utilisateur décide de sélectionner l'objet en plaçant le curseur dessus et en le sélectionnant avec la gestuelle "Pinch".
 - L'utilisateur déplace l'objet qui a été sélectionné où il le souhaite, en déplaçant le curseur.
 - L'objet a été déplacé à la position voulue.
- **Scénarios négatifs :**
 - **Scénario 1**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un fichier dans la liste et clique sur le bouton de visualisation du menu.
 - Un nouvel objet, correspondant au fichier sélectionné, s'affiche (en suivant le mode de visualisation correspondant).
 - L'utilisateur décide de déplacer son regard pour faire bouger le curseur.
 - L'objet n'a pas été déplacé.
 - **Scénario 2**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un fichier dans la liste et clique sur le bouton de visualisation du menu.
 - Un nouvel objet, correspondant au fichier sélectionné, s'affiche (en suivant le mode de visualisation

- correspondant).
- L'utilisateur décide de regarder à côté de l'objet, dans le vide, et effectue la gestuelle de "Pinch".
- L'utilisateur déplace son regard, afin de bouger le curseur.
- L'objet n'a pas été déplacé.

5.2.10 Tests de rotation d'un objet correspondant à un fichier

- **Données :** un casque HoloLens et l'application développée déployée sur ce casque.
- **Résultats attendus :**
 - positif : le menu est orienté selon le souhait de l'utilisateur.
 - négatif : le menu ne change pas d'orientation malgré l'action de l'utilisateur.
- **Scénarios positifs :**
 - **Sénario 1**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un fichier dans la liste et clique sur le bouton de visualisation du menu.
 - Un nouvel objet, correspondant au fichier sélectionné, s'affiche (en suivant le mode de visualisation correspondant).
 - L'utilisateur décide de sélectionner la sphère sur une arête du menu afin d'activer l'action de rotation de l'objet.
 - L'utilisateur déplace cette sphère autour de l'objet à un angle plus ou moins différent de sa position initiale.
 - L'objet est ajusté à sa nouvelle orientation.
 - **Sénario 2**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un fichier dans la liste et clique sur le bouton de visualisation du menu.
 - Un nouvel objet, correspondant au fichier sélectionné, s'affiche (en suivant le mode de visualisation correspondant).
 - L'utilisateur décide de se placer à l'arrière du menu.
 - L'utilisateur décide de sélectionner la sphère sur une arête du menu afin d'activer l'action de rotation de l'objet.
 - L'utilisateur déplace cette sphère autour de l'objet à un angle plus ou moins différent de sa position initiale.
 - L'objet est ajusté à sa nouvelle orientation.
- **Scénario négatif :**
 - **Sénario 1**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un fichier dans la liste et clique sur le bouton de visualisation du menu.
 - Un nouvel objet, correspondant au fichier sélectionné, s'affiche (en suivant le mode de visualisation correspondant).
 - L'utilisateur décide de sélectionner la sphère sur une arête du menu afin d'activer l'action de rotation de l'objet mais sans activer la gestuelle "Pinch".
 - L'orientation de l'objet n'a pas changé.
 - **Sénario 2**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un fichier dans la liste et clique sur le bouton de visualisation du menu.
 - Un nouvel objet, correspondant au fichier sélectionné, s'affiche (en suivant le mode de visualisation correspondant).
 - L'utilisateur décide de sélectionner la sphère sur une arête du menu afin d'activer l'action de rotation de l'objet à l'aide de la gestuelle "Pinch".
 - L'utilisateur ne déplace pas son regard pour faire bouger le curseur.
 - L'orientation de l'objet n'a pas changé.
 - **Sénario 3**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
- L'utilisateur sélectionne un fichier dans la liste et clique sur le bouton de visualisation du menu.
- Un nouvel objet, correspondant au fichier sélectionné, s'affiche (en suivant le mode de visualisation correspondant).
- L'utilisateur décide de sélectionner la sphère sur une arête du menu afin d'activer l'action de rotation de l'objet à l'aide de la gestuelle "Pinch".
- L'utilisateur décide de relâcher la sphère puis de sélectionner un coin de l'objet, afin d'activer l'action de redimensionnement, à l'aide de la gestuelle "Pinch".
- L'utilisateur déplace le curseur pour déplacer le coin.
- L'orientation du menu n'a pas changé.

5.2.11 Tests de redimensionnement d'un objet correspondant à un fichier

- **Données :** un casque HoloLens et l'application développée déployée sur ce casque.
- **Résultats attendus :**
 - positif : le fichier souhaité est agrandi selon le souhait de l'utilisateur.
 - négatif : le fichier ne change pas de taille malgré l'action de l'utilisateur.
- **Scénarios positifs :**
 - **Sénario 1**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un ou plusieurs fichiers à afficher devant lui, dans la liste.
 - L'utilisateur décide de sélectionner un coin d'un des fichiers afin de l'agrandir avec la gestuelle "Pinch".
 - L'utilisateur déplace ce coin plus ou moins loin de sa position initiale.
 - Le fichier concerné est ajusté à sa nouvelle taille.
 - **Sénario 2**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un ou plusieurs fichiers à afficher devant lui.
 - L'utilisateur décide de se placer à l'arrière d'un des fichiers affichés.
 - L'utilisateur sélectionne un coin du fichier concerné avec la gestuelle "Pinch".
 - L'utilisateur déplace ce coin plus ou moins loin de sa position de départ.
 - Le fichier concerné est redimensionné à sa nouvelle taille.
- **Scénario négatif :**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal de l'application apparaît devant les yeux de l'utilisateur.
 - L'utilisateur décide de sélectionner un ou plusieurs fichiers dans la liste, à afficher devant lui.
 - L'utilisateur décide de sélectionner un des coins d'un des fichiers à disposition devant lui sans la gestuelle "Pinch".
 - L'utilisateur essaie de déplacer le coin sélectionné plus ou moins loin de la position initiale.
 - Le fichier concerné ne change pas de taille.

5.2.12 Tests du changement de page d'un fichier texte

- **Données :** un casque HoloLens et l'application développée déployée sur ce casque.
- **Résultats attendus :**
 - positif : une fois le bouton "Next" pressé, associé à un certain fichier texte, la page suivante du texte est affichée. Si le bouton "Previous" est pressé, alors la page précédente de ce texte est affichée.
 - négatif : ni le bouton "Previous" ou "Next" a été pressé, et donc la page actuelle du texte reste affichée.
- **Scénarios positifs :**
 - **Sénario 1**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.

- L'utilisateur sélectionne un ou plusieurs fichiers textes dans la liste via la gestuelle "Pinch".
 - L'utilisateur place le curseur avec son regard sur le bouton visualize et valide avec la gestuelle "Pinch".
 - Les fichiers sélectionnés apparaissent devant les yeux de l'utilisateur.
 - L'utilisateur décide d'appliquer des transformations sur un des textes présents devant lui.
 - L'utilisateur place le pointeur sur le bouton "Next" avec son regard.
 - L'utilisateur appuie sur ce bouton avec la gestuelle "Pinch".
 - La page suivante du fichier texte concerné est affichée.
- **Sénario 2**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un ou plusieurs fichiers textes dans la liste via la gestuelle "Pinch".
 - L'utilisateur place le curseur avec son regard sur le bouton visualize et valide avec la gestuelle "Pinch".
 - Les fichiers sélectionnés apparaissent devant les yeux de l'utilisateur.
 - L'utilisateur place le pointeur sur le bouton "Next" avec son regard.
 - L'utilisateur appuie sur ce bouton avec la gestuelle "Pinch".
 - L'utilisateur place le pointeur sur le bouton "Previous" avec son regard.
 - L'utilisateur appuie sur ce bouton avec la gestuelle "Pinch".
 - La première page du fichier texte concerné est affichée.
- **Sénario 3**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un ou plusieurs fichiers textes dans la liste via la gestuelle "Pinch".
 - L'utilisateur place le curseur avec son regard sur le bouton visualize et valide avec la gestuelle "Pinch".
 - Les fichiers sélectionnés apparaissent devant les yeux de l'utilisateur.
 - L'utilisateur place le pointeur sur le bouton "Next" avec son regard.
 - L'utilisateur appuie sur ce bouton avec la gestuelle "Pinch" jusqu'à atteindre la dernière page.
 - L'utilisateur appuie de nouveau sur le bouton "Next".
 - La première page du fichier texte concerné est affichée.
- **Sénario 4**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un ou plusieurs fichiers textes dans la liste via la gestuelle "Pinch".
 - L'utilisateur place le curseur avec son regard sur le bouton visualize et valide avec la gestuelle "Pinch".
 - Les fichiers sélectionnés apparaissent devant les yeux de l'utilisateur.
 - L'utilisateur place le pointeur sur le bouton "Previous" avec son regard.
 - L'utilisateur appuie sur ce bouton avec la gestuelle "Pinch".
 - La dernière page du fichier texte concerné est affichée.
- **Scénario négatif :**
 - **Sénario 1**
 - L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
 - L'application se lance et le menu principal apparaît devant les yeux de l'utilisateur.
 - L'utilisateur sélectionne un ou plusieurs fichiers textes dans la liste via la gestuelle "Pinch".
 - L'utilisateur place le curseur avec son regard sur le bouton visualize et valide avec la gestuelle "Pinch".
 - Les fichiers sélectionnés apparaissent devant les yeux de l'utilisateur.
 - L'utilisateur place le pointeur sur le bouton "Next" avec son regard.
 - L'utilisateur place le pointeur sur le bouton "Close" avec son regard.
 - La même page du fichier texte est toujours affichée, elle n'a pas changée.

5.2.13 Test du mode de visualisation n°2 avec le système de forces

- **Données :** un casque HoloLens et l'application développée déployée sur ce casque.
- **Résultats attendus :**

- positif : les fichiers sélectionnés sont réparties équitablement selon les trois zones les plus regardées, et sont à une certaine distance du centre de leur zone associée, et restent écartés l'un de l'autre.
- négatif : les fichiers sélectionnés ne sont pas répartis sur les zones et s'éloignent sans arrêt du centre de la zone ou sont en totale collision l'un avec l'autre.

- **Scénarios positifs :**

- **Sénario 1**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance avec un bouton Start qui apparaît.
- L'utilisateur clique sur ce bouton via la gestuelle "Air Tap".
- L'utilisateur dispose de 20 secondes pour regarder des zones où il souhaite disposer les fichiers.
- L'utilisateur crée des sphères aux positions qu'il souhaite.
- L'utilisateur voit le menu apparaître après la fin du compteur.
- L'utilisateur choisit des fichiers et clique sur visualize.
- Les fichiers sont répartis équitablement autour des trois point les plus observés en restant à une certaine distance du centre de leur zone respective et en étant espacés entre eux.

- **Sénario 2**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance avec un bouton Start qui apparaît.
- L'utilisateur clique sur ce bouton via la gestuelle "Air Tap".
- L'utilisateur dispose de 20 secondes pour regarder des zones où il souhaite disposer les fichiers.
- L'utilisateur ne bouge pas le regard aucune sphère n'apparaît.
- L'utilisateur voit le menu apparaître après la fin du compteur.
- L'utilisateur choisit des fichiers et clique sur visualize.
- Les fichiers sont répartis équitablement autour des zones par défaut situées respectivement en (-2.0, 0.3, 1.5), (0.0, 0.3, 1.5) et (2.0, 0.3, 1.5) en restant à une certaine distance du centre de ces zones et en étant espacés entre eux.

- **Scénarios négatifs :**

- **Sénario 1**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance avec un bouton Start qui apparaît.
- L'utilisateur clique sur ce bouton via la gestuelle "Air Tap".
- L'utilisateur dispose de 20 secondes pour regarder des zones où il souhaite disposer les fichiers.
- L'utilisateur crée des sphères aux positions qu'il souhaite.
- L'utilisateur voit le menu apparaître après la fin du compteur.
- L'utilisateur choisit des fichiers et clique sur visualize.
- Les fichiers sont répartis tous sur la même zone et se confondent.

- **Sénario 2**

- L'utilisateur lance l'application depuis le menu principal du HoloLens via la gestuelle "Air tap".
- L'application se lance avec un bouton Start qui apparaît.
- L'utilisateur clique sur ce bouton via la gestuelle "Air Tap".
- L'utilisateur dispose de 20 secondes pour regarder des zones où il souhaite disposer les fichiers.
- L'utilisateur ne parvient pas à créer des sphères aux positions qu'il souhaite.
- L'utilisateur voit le menu apparaître après la fin du compteur.
- L'utilisateur choisit des fichiers et clique sur visualize.
- Les fichiers sont répartis tous sur la même zone et se confondent.

6 Résultats

Afin d'avoir un meilleur retour sur les avantages et les limites de notre application, nous avons expérimenté nos solutions auprès de plusieurs personnes. Nous avons pu faire tester à 8 participants, 3 femmes et 5 hommes entre 18 et 60 ans. La majorité des utilisateurs n'ayant pas eu d'expériences en RA, nous avons décidé, pour que nos tests soient le moins biaisés possible par l'utilisation classique de l'appareil, d'indiquer au préalable les commandes possibles et les gestuelles (en faisant le tutoriel inclus à HoloLens quand le temps le permettait). Néanmoins, l'utilisation du casque n'a pas été intuitive pour tous les individus. Ainsi, nous pensons que cela a pu biaiser leurs avis.

Dès lors, nos tests se définissent simplement par l'utilisation de nos deux solutions, l'une après l'autre, assistée par l'un des développeurs, afin de fluidifier le test et d'indiquer les étapes à suivre en cas de problèmes. Chaque utilisateur a pu user de chaque solution dans son ensemble en utilisant chaque fonctionnalité. Néanmoins, aucun scénario n'était prédéfini, nous avons préféré laisser aux participants le choix d'expérimenter nos solutions pour ne pas influencer sur leurs perceptions.

Ainsi, d'après les retours des participants, nous pouvons tout d'abord remarquer que le mode de visualisation n°2, c'est-à-dire la carte du regard, a été préférée même si ce n'est pas une majorité absolue. Cette visualisation semble d'après eux plus facile, malgré quelques propositions de la part des individus pour rendre cette solution plus intuitive. Nous estimons que ce choix peut aussi être influencé par l'utilisation de l'appareil, le premier mode de visualisation demandant plus d'interactions de la part de l'utilisateur, qui doit placer les zones d'étiquettes au préalable. Dans la même idée, nous pensons que l'utilisation du casque à tout de même influencé leur appréciation de l'expérience. Parmi les participants, 5 n'avait jamais expérimenté la RV ou la RA, une personne uniquement la RV, et une personne les deux. Nous avons ainsi facilement remarqué que la majorité a pris un certain temps pour être à l'aise et réellement comprendre comment bien utiliser les commandes du casque. Cette remarque nous montre aussi qu'il y a un effort à faire sur la clarté et l'efficacité de l'interface pour que l'utilisateur comprenne rapidement et efficacement. Toutefois, la manipulation des éléments représentant les fichiers texte ou image été apprécié. De plus, le système de forces a aussi particulièrement intéressé les utilisateurs, car les mouvements et le placement automatique des éléments à l'ouverture était impressionnants et inattendus d'après eux.

Les utilisateurs ont aussi pu remarquer quelques problèmes d'affichage, lié à l'appareil en lui-même, comme le champ de vision restreint. Mais aussi à notre implémentation, par exemple avec des affichages trop proches de l'utilisateur, à la création et l'ouverture de certains éléments. Nous pouvons cependant signaler que nous avons déjà remarqué le problème du champ de vision 3.

Enfin, on peut ajouter que tous les utilisateurs étaient curieux à propos de la réalité augmentée et posaient des questions au sujet des utilisations possibles et actuelles de cette technologie. Selon nous, cette curiosité est intéressante à noter, car elle vient renforcer la pertinence de notre problématique.

7 Conclusion

En quelques mots, rappelons que les interfaces de RA permettent d'organiser librement des éléments virtuels dans l'espace qui nous entoure ayant ainsi le potentiel de faciliter à la fois la productivité et les interactions. Notre but durant ce projet est de déterminer des stratégies intéressantes pour proposer un placement adéquat des fichiers dans un espace de travail en RA. Ainsi notre solution propose principalement deux méthodes de placement automatique. La première utilise des étiquettes et des zones correspondantes autour desquelles les fichiers associés s'ouvriront. La deuxième se base sur la zone la plus regardée par l'utilisateur comme position pour l'ouverture et l'affichage des fichiers. Au travers ces deux solutions, nous avons essayé d'implémenter deux stratégies variées, l'une se basant majoritairement sur les interactions utilisateurs et l'autre se basant sur une utilisation plus passive. Après avoir fait tester ces solutions à un groupe d'individus, il en ressort une préférence plus marquée pour l'algorithme de la carte du regard (et l'utilisation des forces physiques pour améliorer le placement). Cela peut s'expliquer par le peu d'interaction que cette stratégie requiert et le niveau d'expérience en RA des participants. Cependant, l'algorithme d'étiquetage est tout de même considéré comme étant intéressant. On peut donc imaginer que la pertinence des algorithmes dépend principalement du public visé, mais aussi du cadre d'utilisation.

Il pourrait alors être intéressant d'optimiser ces modes de visualisation en les rendant plus accessibles. Par exemple en améliorant l'affichage des éléments, avec un système de symboles pour indiquer la position de ceux en dehors du champ de vision. Mais aussi, en développant un mode de visualisation exploitant l'ancrage des éléments à la vue. Malgré tout, nos algorithmes de visualisation ne sont pas exhaustifs et d'autres solutions pourraient apporter des nuances supplémentaires à nos résultats. Une méthode se basant sur la topologie de l'environnement pourrait par exemple permettre d'éliminer les problèmes d'occultation tout en gardant un côté passif en ne nécessitant qu'un seul scan de l'espace autour de l'utilisateur.

Références

- [1] Ruiz Alfredo, Resnick Ben, Griffin James, Peng Rose, and Murphy Rosstin. Ibm immersive insights. <https://awards.design/now/project.asp?ID=16128>, 2017.
- [2] Blaine Bell, Steven Feiner, and Tobias Höllerer. View management for virtual and augmented reality. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 101–110, 2001.
- [3] Wolfgang Büschel, Stefan Vogt, and Raimund Dachzelt. Augmented reality graph visualizations. *IEEE computer graphics and applications*, 39(3) :29–40, 2019.
- [4] Yifei Cheng, Yukang Yan, Xin Yi, Yuanchun Shi, and David Lindlbauer. Semanticadapt : Optimization-based adaptation of mixed reality layouts leveraging virtual-physical semantic connections. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 282–297, 2021.
- [5] Andreas Fender, Philipp Herholz, Marc Alexa, and Jörg Müller. Optispace : Automated placement of interactive 3d projection mapping content. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–11, 2018.
- [6] Andreas Fender, David Lindlbauer, Philipp Herholz, Marc Alexa, and Jörg Müller. Heatspace : Automatic placement of displays by empirical analysis of user behavior. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 611–621, 2017.
- [7] Ran Gal, Lior Shapira, Eyal Ofek, and Pushmeet Kohli. Flare : Fast layout for augmented reality applications. In *2014 IEEE international symposium on mixed and augmented reality (ISMAR)*, pages 207–212. IEEE, 2014.
- [8] Knut Hartmann, Kamran Ali, and Thomas Strothotte. Floating labels : Applying dynamic potential fields for label layout. In *Smart Graphics : 4th International Symposium, SG 2004, Banff, Canada, May 23-25, 2004. Proceedings 4*, pages 101–113. Springer, 2004.
- [9] Denis Kalkofen, Erick Mendez, and Dieter Schmalstieg. Interactive focus and context visualization for augmented reality. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 191–201. IEEE, 2007.
- [10] David Lindlbauer, Anna Maria Feit, and Otmar Hilliges. Context-aware online adaptation of mixed reality interfaces. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*, pages 147–160, 2019.
- [11] MaxWang-MS. Microsoft mixed reality toolkit v2.8.3. <https://github.com/microsoft/MixedRealityToolkit-Unity/releases/tag/v2.8.3>, dec 2022.
- [12] Leonel Merino, Mario Hess, Alexandre Bergel, Oscar Nierstrasz, and Daniel Weiskopf. Perfvis : Pervasive visualization in immersive augmented reality for performance awareness. In *Companion of the 2019 ACM/SPEC International Conference on Performance Engineering*, pages 13–16, 2019.
- [13] Microsoft. Unity development for hololens. <https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unity/unity-development-overview>, 2022.
- [14] Arnaud Prouzeau, Antoine Lhuillier, Barrett Ens, Daniel Weiskopf, and Tim Dwyer. Visual link routing in immersive visualisations. In *Proceedings of the 2019 ACM international conference on interactive surfaces and spaces*, pages 241–253, 2019.
- [15] Patrick Reipschlagel, Tamara Flemisch, and Raimund Dachzelt. Personal augmented reality for information visualization on large interactive displays. *IEEE Transactions on Visualization and Computer Graphics*, 27(2) :1182–1192, 2020.
- [16] Markus Tatzgern, Valeria Orso, Denis Kalkofen, Giulio Jacucci, Luciano Gamberini, and Dieter Schmalstieg. Adaptive information density for augmented reality displays. In *2016 IEEE Virtual Reality (VR)*, pages 83–92. IEEE, 2016.
- [17] Unity Technologies. Unity scripting api. <https://docs.unity3d.com/ScriptReference/>, 2023.
- [18] Dan Richards Yoon Parks. COVID19 Data Visualization with HoloLens2 + MRTK, 12 2020.
- [19] Alan Zucconi. Arrays shaders : heatmaps in unity. <https://www.alanzucconi.com/2016/01/27/arrays-shaders-heatmaps-in-unity3d/>, jan 2016.