



HAL
open science

Un Modèle Unifié de l'Interaction en Environnement 3D

Julien Casarin, Dominique Bechmann, Marilyn Keller

► **To cite this version:**

Julien Casarin, Dominique Bechmann, Marilyn Keller. Un Modèle Unifié de l'Interaction en Environnement 3D. journées de Réalité Virtuelle (j•RV), 2017, Rennes, France. hal-03191204

HAL Id: hal-03191204

<https://hal.science/hal-03191204>

Submitted on 13 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Un Modèle Unifié de l'Interaction en Environnement 3D

Casarin Julien*

Université de Strasbourg
Gfi Informatique

Dominique Bechmann**

Université de Strasbourg

Marilyn Keller***

Gfi Informatique



Figure 1: Nous proposons un nouveau modèle pour concevoir des applications de RV, RA & RM indépendamment de tout dispositif. Notre modèle permet à l'utilisateur de visualiser un monde virtuel et d'interagir avec les objets qui le composent via tout type de dispositif. Ses photos montrent un utilisateur interagissant avec le même objet virtuel en utilisant un ordinateur de bureau, un écran tactile et un casque de réalité virtuelle.

ABSTRACT

Les dispositifs de Réalité Virtuelle (RV), Augmentée (RA) et Mixte (RM) évoluent actuellement de manière fulgurante. Cette évolution rapide impacte fortement la maintenabilité et la portabilité des applications. Dans cet article, nous présentons un modèle permettant de concevoir des applications de RV, RA & RM indépendamment de tout dispositif. Pour ce faire, nous utilisons les degrés de liberté pour définir une couche d'abstraction entre les tâches à réaliser et le dispositif d'interaction.

Mots clés : Réalité Virtuelle, Réalité Augmentée, Réalité Mixte, Interaction, Degrés de liberté.

Index Terms: • Human-centered computing → Interaction design; • Computing methodologies → Graphics input devices; Mixed / augmented reality; Virtual reality;

1 INTRODUCTION

L'évolution rapide des dispositifs de Réalité Virtuelle (RV), Augmentée (RA) et Mixte (RM) réduit considérablement la durée de vie des applications de RV, RA & RM. D'autre part, les différences entre dispositifs sont très importantes (dimensions de l'affichage, nombre de degrés de liberté des périphériques d'interaction). En conséquence, la portabilité d'une application d'un dispositif vers un autre est bien souvent longue et compliquée.

L'approche existante consiste à utiliser des moteurs de jeu (Unity, Unreal Engine, ...) permettant le développement d'une même application pour plusieurs types de dispositifs. La combinaison de cette approche avec des outils tels que Middle VR

[5] permet de simplifier le développement d'application de RV, RA & RM en fournissant des outils de développement de haut niveau. Toutefois, l'interaction devant être adaptée aux propriétés propres à chaque type de dispositif, il est nécessaire d'apporter des modifications à l'application pour permettre le support d'une nouvelle technologie. Ainsi, l'apparition sur le marché d'un nouveau dispositif entraîne la nécessité de mettre à jour l'ensemble des applications.

Nous proposons une nouvelle approche visant à augmenter de manière significative la portabilité des applications de RV, RA & RM. Celle-ci consiste à considérer les dispositifs comme des outils pouvant interagir avec tout environnement 3D. Cette approche nécessite la création d'un standard permettant la description de tout environnement avec un dispositif quelconque. S'il existe des standards pour décrire les objets composants une scène 3D (OBJ, DAE, VRML, ...), il n'existe pas de modèle unifié permettant de décrire les interactions du ou des utilisateurs.

Pour réaliser ce modèle, nous nous basons sur l'utilisation de degrés de liberté pour définir une couche d'abstraction entre les tâches à réaliser et le dispositif. Les manipulations, le contrôle d'application et le pointage sont définis à l'aide d'un ensemble réduit de briques d'interaction. Ces briques sont transmises via un réseau au dispositif. Le dispositif les interprète alors et les active en fonction des actions de l'utilisateur.

Après avoir présenté les différentes briques d'interaction qui le composent (section 3.3), nous présentons notre implémentation du modèle (section 4), nous le testons alors sur deux environnements 3D avec les dispositifs présentés *Figure 1* (section 5). Enfin, nous montrons l'apport de cette méthode en termes de portabilité des applications par rapport aux solutions existantes (section 6).

2 ÉTAT DE L'ART

2.1 Environnement 3D

Dans la suite de cet article, nous appelons environnements 3D, l'ensemble des environnements virtuels, augmentés et mixtes qui comportent des objets virtuels positionnés dans un espace à 3 dimensions.

Un **environnement virtuel** est un environnement entièrement généré par ordinateur en trois dimensions [10]. La Réalité

* julien.casarin@gfi.fr

** bechmann@unistra.fr

*** marilyn.keller@gfi.fr

Augmentée permet de superposer des environnements virtuels au monde réel. On appelle **environnement augmenté** une zone de l'espace réel à laquelle se superpose un environnement virtuel. Dans le cadre de la Réalité Augmentée, le niveau d'interaction entre les objets réels et virtuels est très variable. Il peut aller du positionnement graphique des objets virtuels sur une image réelle [8][9] jusqu'à une inclusion réaliste des objets virtuels dans le monde réel [11].

Dans [6], Milgram définit la réalité mixte comme étant l'ensemble d'un continuum représenté *Figure 2*. Tout environnement incluant des éléments réels et des éléments virtuels peut donc être considéré comme étant un **environnement mixte**.

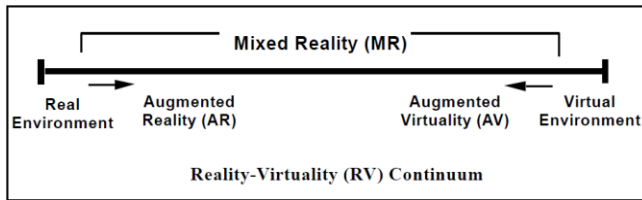


Figure 2 : Le continuum de Milgram 1994 [6]

En pratique, on parle de Réalité Mixte lorsque les objets réels et les objets virtuels interagissent (presque) aussi librement que s'ils appartenaient tous à un même espace physique (respect des contraintes physiques telles que la gravité et les collisions, occlusion, échanges de données bidirectionnels, ...).

2.2 Manipulations en environnement 3D

En 1994, les travaux de R.J.K. Jacob & al. [1] ont montré que le dispositif dont la structure de contrôle est la plus proche de la structure perceptuelle d'une tâche est le plus adapté pour exécuter cette tâche. Il semble donc que les degrés de liberté composant la structure perceptuelle d'une tâche puissent être utilisés pour la décrire de manière générique. Dans des travaux plus récents, M. Veit & al ont montré que les utilisateurs décomposent instinctivement certaines tâches complexes dans le but de les simplifier [2]. De plus, cette décomposition peut varier au cours de la tâche en fonction de la vitesse d'exécution et/ou de la précision souhaitées [3]. Enfin, il apparaît qu'une séparation de certains degrés de liberté lors de la conception de l'interaction conduit à de meilleures performances de l'utilisateur si cette décomposition est conforme à la manière dont l'utilisateur simplifierait de lui-même la tâche [2][4]. Il paraît donc nécessaire d'intégrer à notre description générique des tâches, des informations sur la manière dont les degrés de libertés de la tâche sont préférentiellement intégrés ou séparés par les utilisateurs.

3 NOTRE MODELE

Le but de notre travail est de permettre de limiter le portage d'un ensemble d'environnements 3D sur un nouveau dispositif au développement d'une unique application dédiée.

Pour cela, nous séparons totalement le développement de l'environnement 3D du dispositif de l'utilisateur. Les environnements 3D développés selon notre modèle peuvent donc s'exécuter sur un ordinateur séparé ou encore sur un serveur. Comme ces environnements sont décrit de manière générique, il est possible, pour un dispositif donné, de développer une application dédiée capable d'afficher tout environnement 3D. La portabilité des environnements 3D est nettement améliorée par le modèle car le nombre total d'applications à développer pour permettre le déploiement de m environnement sur n dispositifs est de $n*m$ avec l'approche classique contre $n+m$ pour notre approche. Ainsi, il

devient suffisant de développer une unique application (ou client) pour permettre l'utilisation d'un nouveau dispositif par tous les environnements existants. De plus, tout nouvel env. peut être déployé sur tous les dispositifs disposant déjà d'un client dédié.

Pour réaliser notre modèle, nous proposons l'utilisation d'une couche d'abstraction afin de concevoir l'interaction indépendamment de la nature du dispositif. Cette couche d'abstraction regroupe un ensemble de briques d'interaction de base communes à tous les dispositifs. L'instanciation et la composition de ces briques permet de scénariser l'interaction du ou des utilisateurs avec l'environnement.

Du fait de la séparation entre les environnements 3D et les dispositifs, les clients doivent instancier les briques d'interaction transmises par l'environnement afin de construire dynamiquement l'interface permettant à l'utilisateur d'interagir avec lui.

3.1 Fonctionnement général

Une Scène 3D est distribuée en temps réel vers un ou plusieurs dispositifs grâce à un réseau de communication. Une copie de la scène est alors créée et synchronisée par le dispositif. Le même réseau permet la transmission au dispositif d'un ensemble de briques d'interaction de base. Ces briques constituent un ensemble d'objets standards permettant la description des actions que le dispositif peut avoir sur l'environnement. Chaque dispositif est capable de réaliser une projection de ces briques sur lui-même. Cette opération consiste à générer dynamiquement l'interface utilisateur permettant de réaliser, avec le dispositif, les actions décrites. Des menus et des moyens de paramétrage sont créés à la volée pour permettre le paramétrage de l'environnement et la sélection de tâches géométriques. Les tâches décrites sont projetées sur les degrés de liberté du dispositif. Enfin, les briques d'interaction indiquent également les objets pouvant être pointés et/ou sélectionnés par l'utilisateur.

Lors de l'implémentation d'un client pour un dispositif donné, la projection des briques d'interaction est optimisée en fonction des propriétés propres au dispositif (nature de l'affichage, nombre de degrés de liberté du dispositif d'interaction, ...).

Une fois la projection sur le dispositif achevée, l'interaction de l'utilisateur avec les périphériques du dispositif entraîne l'activation de certaines briques. Cette activation est demandée à l'environnement grâce au réseau de communication déjà utilisé pour la transmission de la scène et des briques d'interaction. L'environnement réagit alors par une modification de la scène et/ou des interactions données aux utilisateurs.

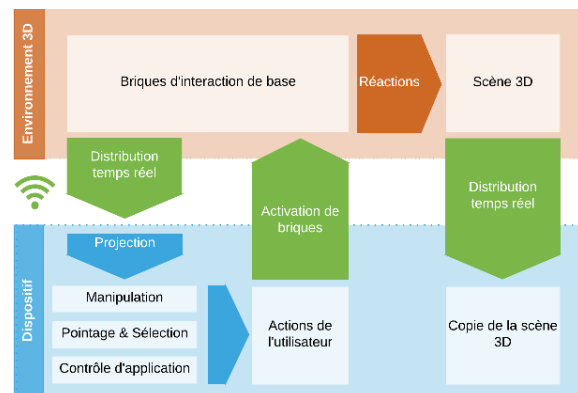


Figure 3 : Modèle général

3.2 Briques d'interaction de base

Nous définissons les briques d'interaction comme étant un ensemble fini d'objets instanciables et paramétrables (Figure 5). Ces objets décrivent de manière générique l'interaction avec un environnement 3D. Chaque dispositif de RV, RA ou RM doit être capable d'interpréter et d'activer l'ensemble de ces briques. L'évolution en temps réel des briques attribuées aux utilisateurs, en fonction de leurs actions, définit le scénario interactif associé à l'environnement 3D.

Nous avons choisi nos briques d'interactions en fonction des critères suivants. Tout d'abord, les briques doivent être interprétables par tout type de dispositif. Ensuite, la composition et le paramétrage des briques doivent permettre de concevoir tous les types d'interaction avec un environnement 3D. Il doit également être possible de donner des informations sur la structure perceptuelle des tâches de sorte à orienter la projection sur le dispositif. Enfin, le nombre de briques doit rester réduit pour faciliter le développement de l'interpréteur pour un dispositif donné.

3.2.1 Manipulation

L'approche existante consiste à relier directement les degrés de liberté d'un objet 3D aux degrés de liberté du dispositif d'interaction. Notre approche consiste à associer à une tâche un mouvement générique dans l'espace 3D. Ce mouvement, décrit à l'aide des briques d'interaction **Dof** et **Dof Integration**, est projeté par le dispositif sur ses propres degrés de liberté. La Figure 4 montre comment la translation d'un cube sur une table peut être réalisée en combinant une brique Dof Integration avec deux briques Dof.

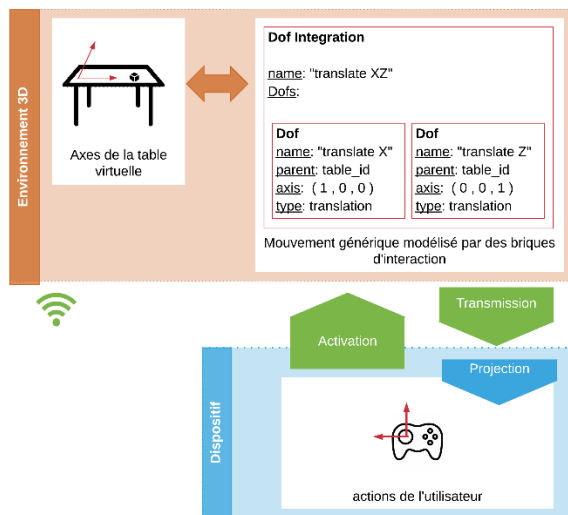


Figure 4 : Exemple d'utilisation des briques d'interaction pour la manipulation d'un objet virtuel. Deux briques Dof et une brique Dof Integration sont composées pour associer, à la translation d'un cube virtuel dans un plan, un mouvement générique à deux degrés de liberté. Ces briques sont transmises à un dispositif doté d'une manette de jeu. Celui-ci les interprète et les projette sur la manette en associant au d'un stick à deux degrés de liberté, la production du mouvement générique décrit. Cette association permet, lors d'un mouvement du stick, d'activer la brique Dof Interaction et ainsi de déplacer le cube.

L'objet Dof définit un mouvement générique selon un unique degré de liberté. Ce degré de liberté peut être soit une translation, soit une rotation autour d'un axe. Le mouvement décrit est soit

continu soit discret. Ce degré de liberté est décrit relativement à un repère de l'espace 3D qui peut être simplement un objet 3D de la scène. Si aucun repère n'est défini, par défaut l'espace utilisateur donne le repère.

L'objet Dof Integration permet de coupler des degrés de liberté, par exemple pour définir une translation non pas le long d'un axe (1 Dof) mais dans un plan (2 Dofs). Ainsi, la brique d'interaction Dof Intégration permet d'obtenir un mouvement à plusieurs degrés de liberté. Le dispositif sur lequel ces briques d'interaction sera instancié doit alors intégrer ces degrés de liberté si sa structure le permet. Si le dispositif n'est pas capable d'intégrer l'ensemble des degrés de liberté comme défini par la brique Dof Integration alors ils restent séparés. Dans ce cas, un menu est généré pour permettre à l'utilisateur de sélectionner tour à tour les degrés de liberté séparés.

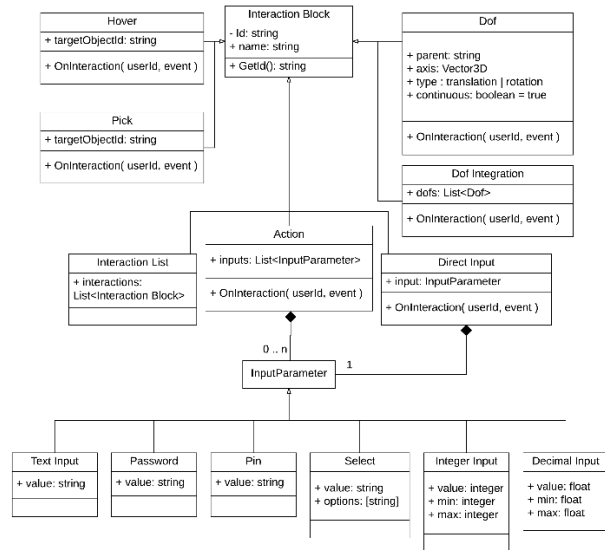


Figure 5 : Briques d'interaction (UML)

3.2.2 Sélection & pointage

La manière de désigner ou de sélectionner un objet 3D diffère fortement d'un dispositif à l'autre. En effet, même si le lancer de rayon (raycasting) est la technique la plus répandue, d'autres techniques existent. D'autre part, la manière de manipuler le rayon lancé dépend totalement de la structure du dispositif. Il nous paraît donc inapproprié de contraindre la manière dont sont réalisés le pointage et la sélection par le dispositif. Nous introduisons deux interactions de base permettant d'indiquer qu'un objet 3D peut être respectivement pointé ou sélectionné.

L'objet **Hover** indique au dispositif qu'un objet virtuel peut être pointé par l'utilisateur. Cette brique est activée pour indiquer le début et la fin du pointage et permet de créer un retour visuel indiquant à l'utilisateur qu'il lui est possible d'interagir avec l'objet.

L'objet **Pick** permet d'associer un événement à la sélection d'un objet virtuel par l'utilisateur.

3.2.3 Contrôle d'application

Le contrôle d'application regroupe tous les moyens mis à disposition de l'utilisateur pour paramétrer l'application et/ou

sélectionner des outils : menus, boutons, saisie de texte, ... La création de l'environnement 3D se faisant sans connaissance préalable du dispositif, il n'est pas possible de prévoir la nature de l'interface utilisateur. En effet, l'interface la plus adaptée peut varier en fonction des caractéristiques du dispositif (dimensions de l'affichage, taille de l'écran, ...). Nous proposons de décrire, de manière générique, les moyens de paramétrage par deux briques d'interaction :

L'objet **Action** permet le déclenchement d'un évènement. Cet objet peut avoir pour argument une liste de paramètres. Une Action sans paramètres correspond typiquement à un bouton dans l'interface finale. Une action avec des paramètres correspond à un formulaire de saisie.

L'objet **Direct Input** permet de modifier en temps réel un paramètre de l'application ou d'un objet 3D (valeur numérique, sélection d'options, champ de texte, ...).

Si les deux briques précédemment décrites sont suffisantes pour décrire tous les éléments nécessaires au contrôle d'application, elles ne permettent pas toujours de générer une interface utilisateur optimale. En effet, la disposition et le regroupement des éléments similaires de l'interface sont essentiels pour l'ergonomie. Nous introduisons donc une dernière brique d'interaction appelée **Interaction List** qui permet de guider la création de menus et sous-menus lors de la génération de l'interface utilisateur. La brique Interaction List est utilisée pour regrouper un ensemble d'autres briques d'interaction qui ont une valeur sémantique ou des usages proches. Il s'agit d'indiquer quels éléments doivent éventuellement être regroupés dans l'interface utilisateur.

4 IMPLEMENTATION

4.1 Plateforme Environnements Virtuels Partagés (PEVP)

Pour valider notre modèle, nous avons implémenté une plateforme Web permettant d'héberger simultanément plusieurs environnements 3D. Cette plateforme est développée dans la technologie Node.js ce qui nous permet notamment de créer rapidement des scènes 3D avec la librairie Three.js. Un environnement 3D hébergé sur la plateforme est composé d'un script, définissant conjointement une scène 3D et des briques d'interaction, et de fichiers ressources correspondant aux modèles 3D et textures. Enfin, Socket.io a été utilisé pour créer un WebSocket permettant la communication temps réel entre les environnements 3D et les dispositifs. L'ensemble des choix technologiques pour la plateforme ont été fait dans le but d'une validation rapide du modèle. Une utilisation future de moteurs de jeu tels que Unity ou Unreal Engine est envisagée dans le but d'enrichir les environnements 3D développés et d'accroître les performances. L'architecture globale de PEVP est présentée par la Figure 6.

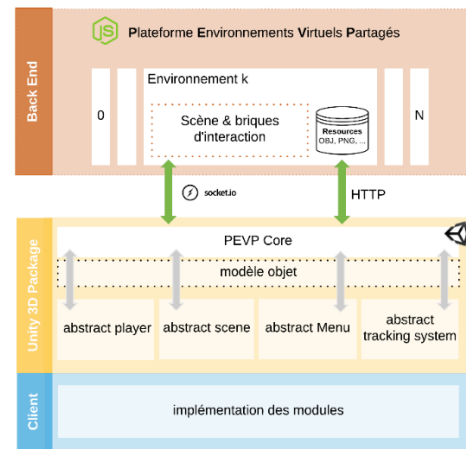


Figure 6 : PEVP

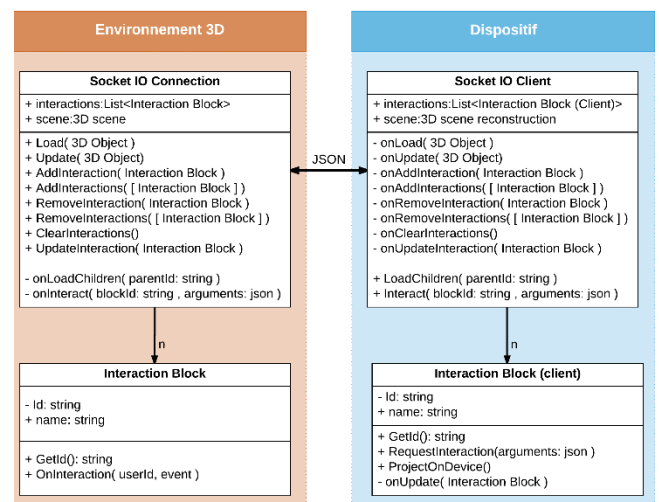


Figure 7 :

4.2 PEVP : client Unity 3D

Dans le but d'accélérer l'implémentation d'un client pour un dispositif donné, nous avons développé un package pour le moteur de jeu Unity 3D. Ce package permet de faire abstraction de la connexion réseau avec la plateforme PEVP lors du développement d'un client. De plus, il permet de mutualiser l'architecture de tous les clients utilisant Unity (Figure 6). Un client PEVP est composé des 5 modules suivants :

4.2.1 PEVP Core

Le module PEVP Client est en charge de la communication entre les clients et les environnements 3D hébergé par la plateforme (Figure 7). Il converti par ailleurs les messages envoyés par PEVP vers un modèle objet. Enfin, il pilote les autres modules.

4.2.2 Scene

Le module Scene permet de charger dynamiquement les objets virtuels de l'environnement 3D et de les synchroniser en temps réel. Le module Scene doit étendre la classe Abstract Scene du package Unity. L'utilisation d'une classe abstraite permet de limiter les développements pour un client donné à l'implémentation d'une stratégie de chargement dynamique adaptée aux capacités du dispositif. Plusieurs dispositifs peuvent donc utiliser la même implémentation de ce module s'ils ont des performances

semblables (mémoire vive, mémoire graphique, qualité de la connexion réseau).

4.2.3 Player

Le module Player est chargé de la projection dynamique des briques d'interactions sur le dispositif. Si une classe abstraite du package Unity permet de mutualiser les interfaces avec le module PEVP Client, il est nécessaire d'implémenter spécifiquement la projection des différentes briques d'interaction sur le dispositif.

4.2.4 Menu

Le module Menu permet à l'utilisateur de naviguer entre les différents environnements 3D mis à sa disposition par la plateforme PEVP.

4.2.5 Tracking System

Le module Tracking System (ou TS) est implémenté uniquement pour les dispositifs de Réalité Augmentée et Mixte. Il est en charge du positionnement dynamique des objets virtuels en fonction des objets réels de l'environnement (marqueurs, position Gps, ...).

5 RESULTAT

5.1 Dispositifs testés

Dans le but d'évaluer la portabilité des environnements 3D développés selon notre modèle, nous avons implémenté trois clients pour la plateforme PEVP (Figure 1). Nous avons ainsi pu vérifier que le développement d'une seule application permet bien le portage de l'ensemble des environnements 3D sur un nouveau dispositif. Les points notables de l'implémentation sont résumés ci-dessous ainsi que dans le Tableau 1.

Dispositif	1	2	3
Affichage	Smart TV	Pc Écran tactile	Oculus rift HMD
Navigation	Souris & Clavier	ARToolKit	Oculus rift IR tracking
Tracking System	.	ARToolKit & Caméra	.
Système d'interaction	Souris & Clavier	Écran tactile	Oculus rift remote
Max Dof Intégrés	2	2	1 + 1
UI	2D overlay	2D overlay	2D world position

Tableau 1 : dispositifs évalués

5.1.1 Scene

La puissance de calculs et les capacités graphiques de nos dispositifs étant semblables, ils partagent une même implémentation du module Scene.

5.1.2 Tracking System

Seul le dispositif 2 est capable d'afficher un environnement 3D en Réalité Augmentée. Il est donc le seul dispositif implémentant le module Tracking System. Le module permet comme dans [7] d'effectuer, en temps réel, le recalage avec des marqueurs à l'aide de l'ARToolKit.

5.1.3 Menu & contrôle d'application

Pour les dispositifs 1 et 2, l'interface utilisateur regroupant le module Menu ainsi que le contrôle d'application généré par le module Player, est une interface 2D traditionnelle superposée à la scène 3D. Toutefois, l'Oculus Rift du dispositif 3 ne permet pas ce type d'interface. Une interface en trois dimensions a donc été conçue pour ce dispositif.

5.2 Environnement 1 : Manipulation d'objets virtuels et retours visuels

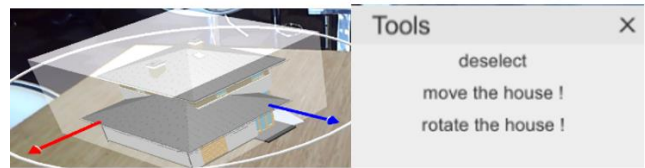


Figure 8 : Intégration de degrés de liberté

Dans le but d'évaluer notre modèle, nous avons développé, et publié sur la plateforme PEVP, un environnement 3D permettant à un utilisateur de manipuler un objet virtuel (Figure 8). Notre objectif était de vérifier que le modèle permet la manipulation d'un objet virtuel par un dispositif quelconque, qu'une réaction aux manipulations de l'utilisateur peut être générée dynamiquement, et enfin, qu'il est bien possible de mettre à jour en temps réel les interactions dont dispose l'utilisateur de sorte à scénariser ses échanges avec l'environnement.

5.2.1 Scénario d'interaction



Figure 9 : Séparation de degrés de liberté

L'utilisateur dispose de deux modes d'interaction proposés dans ce scénario. Le mode Manipulation permet la manipulation des degrés de liberté de l'objet qui peut ainsi être positionné par translation dans le plan XY et orienté par une rotation d'axe Z. Une action permet de plus de passer dans le mode Configuration. Ce deuxième mode donne à l'utilisateur un ensemble d'interactions lui permettant de configurer les degrés de liberté du mode manipulation. La Figure 10 montre le menu généré par le dispositif 1 (voir Tableau 1) pour le mode de configuration. La configuration permet de choisir si les manipulations se font de manière continue ou discrète. Dans le cas d'une rotation discrète, le pas de rotation

est configurable. Les translations selon les axes X et Y peuvent être intégrées (Figure 8) ou séparées (Figure 9). Enfin, une sélection de l'objet entraîne un retour au mode de manipulation.

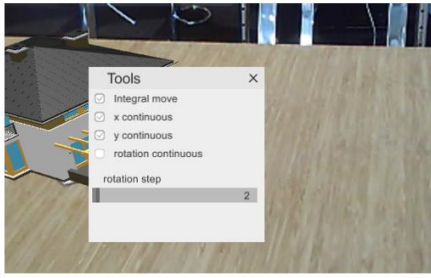


Figure 10 : Configuration de la manipulation

5.2.2 Retours visuels

Le dispositif n'ayant connaissance ni de la nature sémantique des manipulations, ni de l'objet manipulé, il ne lui est pas possible de générer lui-même un retour visuel durant l'interaction pour indiquer à l'utilisateur la prise en compte de son interaction et les mouvements possibles. Ce retour doit donc être généré directement par l'environnement lors de l'activation des briques d'interaction. La Figure 11 résume les différents retours générés par l'environnement 3D. Lors du pointage, la boîte de sélection est affichée en transparence autour de l'objet. Si l'objet est sélectionné, des flèches ou des cercles sont affichés pour indiquer les translations ou les rotations possibles. Enfin les flèches et cercles affichés lors de la sélection changent de couleurs lors de l'activation des briques d'interaction qu'ils représentent.

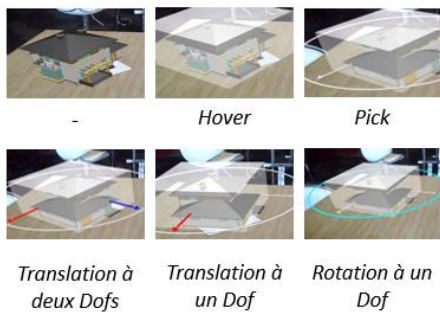


Figure 11 : Retours visuels

5.3 Environnement 2 : Manipulation simultanée de plusieurs objets

Si l'implémentation de l'environnement présenté Section 5.2 a permis de valider les principes de base de notre modèle, ce test ne permet pas à lui seul d'évaluer la pertinence du modèle pour la scénarisation d'interactions complexes. Nous présentons ci-dessous un environnement complémentaire montrant comment la définition d'un mouvement générique par des briques d'interaction permet de scénariser la manipulation simultanée de plusieurs objets. Avec ce test, nous voulions nous assurer que les briques d'interaction Dof et Dof Interaction ne permettent pas uniquement modéliser des translations et des rotations d'objets virtuels, mais également des interactions complexes tel que le grossissement, la sélection multiple ou encore le déclenchement d'évènements. L'environnement présenté ci-dessous montre qu'il est possible de modéliser de telles manipulations par un mouvement générique composé de briques d'interaction Dof et Dof Integration.



Figure 12 : Le Crépier

5.3.1 Scénario d'interaction

L'environnement est composé d'une pile de plateaux. Le but de l'utilisateur est de trier les plateaux en fonction de leur taille. La seule opération autorisée consiste à retourner le haut de la pile (Figure 12).

5.3.2 Sélection à un degré de liberté

Une brique à un degré de liberté est utilisée en association avec un mouvement vertical. Un mouvement de haut en bas ou de bas en haut permet d'ajouter ou d'enlever un plateau à la sélection (Figure 13).

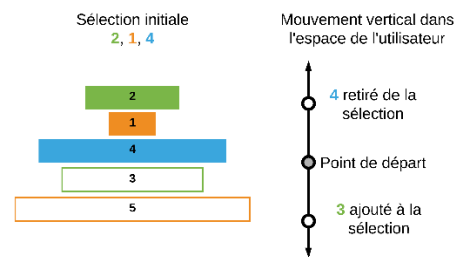


Figure 13 : Sélection d'objets à 1 Dof

L'utilisation d'un Dof discret permet de définir l'amplitude du mouvement utilisé pour incrémenter/décémenter la sélection d'un plateau. La possibilité de sélectionner les plateaux en plusieurs fois permet à l'utilisateur d'éventuellement corriger sa sélection.

5.3.3 Tâche complète

Un deuxième degré de liberté discret est utilisé pour déclencher le retournement de la sélection en associant avec un mouvement horizontal. L'intégration de ces deux Dof permet à l'utilisateur de sélectionner et retourner le haut de la pile d'un seul mouvement (Figure 14).

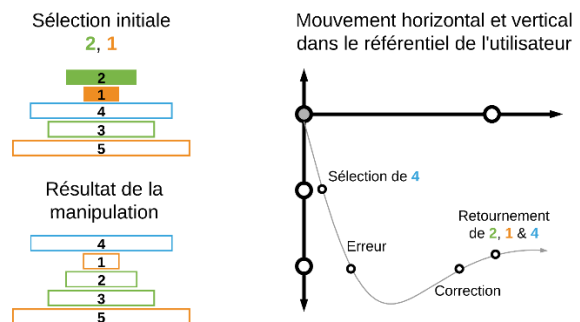


Figure 14 : Intégration de la sélection et du retournement

6 DISCUSSION

Par construction, le portage sur un nouveau dispositif de l'ensemble des environnements développés selon notre modèle ne dépend pas du nombre d'environnements mais dépend de la difficulté à implémenter la projection des briques d'interaction sur

le dispositif. L'utilisation du modèle permet donc un gain de portabilité croissant lorsque le nombre d'environnements augmente. Au vu des tests présentés *Section 5*, il semble que la composition et l'évolution dynamique des briques d'interaction (*Section 3.2*) permettent de scénariser et de décrire de manière générique les interactions avec un environnement 3D.

Par ailleurs, il apparaît que ce modèle permet naturellement la collaboration entre des dispositifs de natures différentes. Toutefois, des travaux complémentaires seront nécessaires pour établir les bonnes pratiques de scénarisation en environnement partagé. En effet, il est probable que les retours visuels, tel que ceux proposés pour le pointage *Tableau 2*, génèrent, dans certains cas, des surcharges cognitives pour les autres utilisateurs. Ces retours visuels pourraient à l'inverse être profitable pour la résolution de problèmes connus tels que celui de l'attention jointe présenté dans [12].

7 CONCLUSIONS

L'objectif premier de cet article était de présenter un modèle générique de l'interaction en environnement 3D permettant de faire totalement abstraction de la nature du dispositif de l'utilisateur lors du développement d'un environnement 3D. Pour ce faire, nous avons montré que les degrés de liberté peuvent être utilisés comme couche d'abstraction commune à tous les dispositifs pour définir la manipulation des objets 3D. Enfin, après avoir présenté notre implémentation du modèle, nous avons montré comment il peut être utilisé concrètement pour la scénarisation en environnement 3D.

Par ailleurs, ce modèle semble adapté pour la création d'environnements collaboratifs. Les possibilités offertes par une collaboration facilitée entre dispositifs complémentaires ouvrent des perspectives nouvelles, notamment dans le domaine de la conception collaborative. Toutefois, des travaux complémentaires établissant les bonnes pratiques de scénarisation avec notre modèle sont à prévoir avant toute évaluation rigoureuse de son apport dans le domaine.

REFERENCES

- [1] Robert J. K. Jacob, Linda E. Sibert, Daniel C. McFarlane, and M. Preston Mullen, Jr.. 1994. Integrality and separability of input devices. *ACM Trans. Comput.-Hum. Interact.* 1, 1 (March 1994), 3-26. DOI=<http://dx.doi.org/10.1145/174630.174631>
- [2] Manuel Veit, Antonio Capobianco, and Dominique Bechmann. 2009. Influence of degrees of freedom's manipulation on performances during orientation tasks in virtual reality environments. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology (VRST '09)*, Steven N. Spencer (Ed.). ACM, New York, NY, USA, 51-58. DOI=<http://dx.doi.org/10.1145/1643928.1643942>
- [3] Manuel Veit, Antonio Capobianco, and Dominique Bechmann. 2010. Dynamic decomposition and integration of degrees of freedom for 3-D positioning. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology (VRST '10)*. ACM, New York, NY, USA, 131-134. DOI=<http://dx.doi.org/10.1145/1889863.1889891>
- [4] M Veit, A Capobianco, and D Bechmann. 2011. An experimental analysis of the impact of Touch Screen Interaction techniques for 3-D positioning tasks. In *Proceedings of the 2011 IEEE Virtual Reality Conference (VR '11)*. IEEE Computer Society, Washington, DC, USA, 75-82. DOI=<http://dx.doi.org/10.1109/VR.2011.5759440>
- [5] S. Kuntz, "MiddleVR a generic VR toolbox", *2015 IEEE Virtual Reality (VR)*, pp. 391-392, 2015.
- [6] Augmented reality: A class of displays on the reality-virtuality continuum. P Milgram, H Takemura, A Utsumi, F Kishino. *Telem manipulator and Telepresence Technologies (1994) – SPIE Vol. 2351*, 282-292
- [7] H. Kato, K. Tachibana, M. Billinghurst, M. Grafe, "A registration method based on texture tracking using ARToolKit", *The Second IEEE Int. Augmented Reality Toolkit Workshop*, 7th October 2003.
- [8] Azuma, R., Y. Bailiot, R. Behringer, S. Feiner, S. Julier, and Blair MacIntyre. "Recent Advances in Augmented Reality." *IEEE Computer Graphics and Applications* 21, no. 6 (November 2001) : 34 – 47. *A Survey of Augmented Reality*.
- [9] Tumler, J., R. Mecke, M. Schenk, A. Huckauf, F. Doil, G. Paul, E.A. Pfister, I. Bockelmann, and A. Roggentin. "Mobile Augmented Reality in Industrial Applications: Approaches for Solution of User Related Issues." In *7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2008. ISMAR 2008, 87 – 90, 2008. doi:10.1109/ISMAR.2008.4637330.
- [10] BRYSON, S. 1996. Virtual reality in scientific visualization. *Commun. ACM* 39, 5, 62–71
- [11] Koleva, Boriana, Holger Schnädelbach, Steve Benford, and Chris Greenhalgh. "Traversable Interfaces Between Real and Virtual Worlds." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 233–40. CHI '00. New York, NY, USA: ACM, 2000.
- [12] Schnier C., Pitsch K., Dierker A., Hermann T. (2011) Collaboration in Augmented Reality: How to establish coordination and joint attention?. In: Bødker S., Bouvin N., Wulf V., Cioffi L., Lutters W. (eds) *ECSCW 2011: Proceedings of the 12th European Conference on Computer Supported Cooperative Work*, 24-28 September 2011, Aarhus Denmark. Springer, London