



HAL
open science

MIREEDGE : un outil communautaire pour l'implémentation d'interactions immersives

Guillaume Loup, Audrey Serna, Sébastien George

► **To cite this version:**

Guillaume Loup, Audrey Serna, Sébastien George. MIREEDGE : un outil communautaire pour l'implémentation d'interactions immersives. 29ème conférence francophone sur l'Interaction Homme-Machine, AFIHM, Aug 2017, Poitiers, France. 3 p. hal-01577688

HAL Id: hal-01577688

<https://hal.science/hal-01577688>

Submitted on 27 Aug 2017

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.

Miredge : un outil communautaire pour l'implémentation d'interactions immersives

A Community-Based Development Tool for immersive interactions

G. Loup

Université Bretagne Loire, Université
du Maine, EA 4023, LIUM
72085 Le Mans
France
guillaume.loup@univ-lemans.fr

A. Serna

Université de Lyon, CNRS, INSA-
Lyon, LIRIS, UMR 5205
F-69621 Villeurbanne
France
audrey.serna@insa-lyon.fr

S. George

Université Bretagne Loire, Université
du Maine, EA 4023, LIUM
72085 Le Mans
France
sebastien.george@univ-lemans.fr

RÉSUMÉ

Les périphériques de réalité virtuelle et mixte (RVM) sont de plus en plus accessibles ainsi que les moteurs de jeux. Même si le nombre de développeurs d'applications 3D temps réel est aujourd'hui important, rares sont ceux qui parviennent à intégrer des interactions immersives. Cependant la communauté des experts RVM possède aussi un besoin qui est celui de la réutilisation de leurs scripts. Notre concept est répondre au besoin des experts tout en faisant bénéficier les novices. Pour cela, nous avons conçu un nouvel outil indépendant nommée MIREEDGE (Mixed and virtual Reality Development toolkit pour Game Engine). Cet outil est basé sur un méta-modèle permettant de capitaliser les algorithmes d'interaction et les associer à des représentations simplifiées. Un méta-modèle garantit une réutilisation d'algorithmes d'interactions en prenant en compte de la compatibilité logicielle et matérielle. L'architecture spécifique de MIREEDGE donne la possibilité aux communautés des développeurs réguliers et occasionnels d'écrire de façon efficace et efficiente un algorithme d'interaction RVM.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics • **Networks** → *Network reliability*

MOTS-CLEFS

Réalité Mixte, réalité virtuelle, interactions, réutilisation générateur de code, partage d'expertise, moteur de jeu

1 INTRODUCTION

Au cours des derniers mois, le grand public a été particulièrement intéressé par la réalité virtuelle et même par la réalité augmentée. Les plus grands industriels de l'informatique proposent chacun à leur tour de nouveaux casques immersifs équipés d'une grande variété de fonctionnalités interactives. Le matériel semble aujourd'hui accessible, mais qu'en est-il des applications ? Selon le cycle de Gartner [1] sur les technologies émergentes, les mondes virtuels seront bientôt adoptés définitivement. Cependant, malgré le potentiel déjà prouvé dans des domaines tels que l'éducation ou la santé, le choix des applications de réalité virtuelle ou mixte (RVM) est encore très limité.

Ce manque de diversité pourrait être expliqué par la nécessité de chaque développeur de maîtriser l'ensemble des contraintes des matériels immersifs pour implémenter une nouvelle interaction. Les développeurs spécialisés en RVM ont, quant à eux, la nécessité de réécrire l'intégralité des algorithmes d'interaction d'un projet à l'autre.

2 L'outil MIREEDGE

2.1 Concept

A l'heure où la communauté des développeurs d'applications 3D temps réel n'a jamais été aussi importante et active, il reste à étudier si un outil pourrait leur permettre d'ajouter facilement de l'immersion à leurs produits. Pour cela, nous avons décidé de classer les multiples outils de développement [2] de RVM en 4 catégories : les plates-formes de développement dédiées à la réalité virtuelle, les kits fournis par les constructeurs, les middleware ainsi les extensions de moteurs de jeu. Notre enquête s'est concentrée plus particulièrement sur leurs atouts pour réaliser des algorithmes d'interaction. On en déduit que certains offrent la facilité d'utilisation à l'aide de la programmation par blocs, d'autres une meilleure personnalisation de l'algorithme ou encore la réutilisation de certaines parties du code. La quasi-totalité de ces outils sont dédiés à une utilisation régulière alors que le besoin auquel nous nous intéressons est une utilisation occasionnelle pour des développeurs d'applications 3D temps réel.

En s'appuyant sur ce constat et des solutions existantes dans des domaines connexes tels que la robotique, nous souhaitons nous appuyer sur le concept réutilisabilité. Ainsi pour assurer un niveau d'efficacité, d'efficience et de satisfaction important pour l'environnement de programmation, nous proposons une première interface de programmation par blocs. Les limites connues pour ce type d'interface sont liées à l'expertise requise pour alimenter la librairie des blocs. Pour répondre à cela, nous avons mis en place plusieurs outils afin que les experts RVM trouvent un intérêt direct à partager leurs algorithmes. Pour cela, nous avons intégré une conversion de la programmation par blocs en scripts réécrivables dans le moteur de jeu. En effet, pour assurer une banque de blocs compatibles avec un maximum d'environnement et de périphériques, notre outil MIREEDGE (Mixed or virtual REality Development tool for Game Engine) est un logiciel gratuit et totalement indépendant.

2.2 Description de l'outil

2.2.1 Méta modèle d'algorithme d'interaction

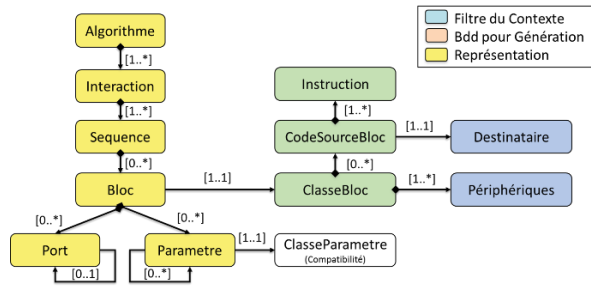


Figure 1: Méta modèle MIREEDGE

Le premier pilier de notre solution est une structuration de données permettant de modéliser chaque interaction. Comme illustré dans la Fig. 1, une partie est destinée à définir l'organisation de la représentation graphique et une seconde partie est dédiée à la structure des lignes de programmation associées. Enfin il est à noter que ce méta modèle prend en compte la notion de compatibilité les moteurs de jeu et les périphériques. L'ensemble de ce méta modèle a été conçu pour assurer un maximum de généricité et ainsi son potentiel de réutilisabilité.

2.2.2 Interface de programmation par blocs

En se basant sur l'efficacité d'interfaces de programmation telle que celle Virtools, MIREEDGE propose une palette latérale répertoriant la bibliothèque des blocs [3]. Chacun de ces blocs peut être instancié en étant déposé dans l'espace central. Cet espace propose des conteneurs encapsulés représentant les interactions et des séquences telle que celle de l'initialisation, la séquence continue ou encore l'événementielle. Après avoir placé le bloc, il est possible d'établir des connexions avec d'autres. Ces connexions sont vouées à définir l'ordre de séquençement ainsi que l'échange de données.

Il existe 3 catégories de blocs. Une première catégorie regroupe les méthodes de consultations et modifications de propriétés des périphériques. Une seconde catégorie contient des éléments de logique permettant d'ajouter des conditions et des répétitions. La troisième catégorie permet d'intégrer des liens vers d'autres éléments déjà présents dans l'application en cours de développement.

2.2.2 Outil communautaire

L'intérêt de cet outil repose sur la capacité de la communauté à produire rapidement toujours plus de blocs d'interactions sur les périphériques RVM pour différents moteurs de jeu. Pour que cette communauté souhaite participer pleinement, MIREEDGE prend en compte le besoin de réutilisation d'algorithmes des experts. Pour cela, MIREEDGE propose de stocker rapidement et récupérer avec flexibilité tout ou partie d'algorithmes d'experts tel la Fig. 2. Ainsi, la première étape propose via une interface web de convertir un script d'interaction en bloc de programmation. Ces informations sont ensuite stockées dans

une base de données. Cette proposition est soumise et évaluée par l'ensemble de la communauté. Lorsque les experts souhaitent réutiliser ce script, ils peuvent rapidement déposer et relier quelques blocs afin d'exporter vers leur projet dans le moteur de jeu. Cet export se traduit par une génération automatique du script correspondant à la représentation graphique modélisée. Ce script correctement commenté, et pleinement accessible, peut ensuite être modifié selon les besoins. Les experts peuvent ainsi en quelques clics réutiliser des centaines de lignes de code validées par la communauté.

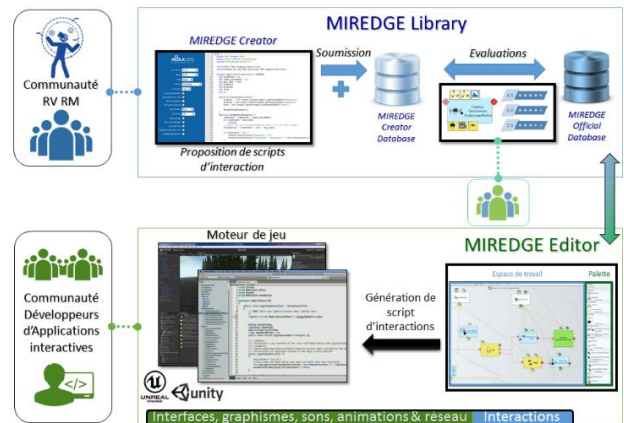


Figure 2: Processus de développement communautaire d'interaction immersive

4 CONCLUSIONS

L'objectif principal de MIREEDGE est de donner l'opportunité aux développeurs sans expertise VMR d'ajouter des interactions immersives à leurs applications. Pour cela, les novices ont la possibilité de programmer graphiquement une interaction personnalisée, en utilisant un méta modèle d'interaction et une architecture logicielle innovante. Pour assurer un enrichissement constant des éléments à disposition, une base de données permet aux experts de capitaliser et réutiliser leurs algorithmes d'interaction. L'utilisabilité de cet outil a déjà été évaluée par des expérimentations auprès de novices et d'experts sur une courte durée. D'autres évaluations seront nécessaires pour mesurer l'usage de la communauté sur le long terme.

REMERCIEMENTS

Nous remercions l'ANR qui finance notre travail de recherche au travers du projet JEN.lab.

REFERENCES

- [1] Fenn, J. and LeHong, H. 2011. Hype Cycle for Emerging Technologies, 2011. (Jul. 2011).
- [2] Kreylos, O. 2008. Environment-Independent VR Development. *Advances in Visual Computing* (Dec. 2008), 901–912.
- [3] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. and Kafai, Y. 2009. Scratch: Programming for All. *Commun. ACM*. 52, 11 (Nov. 2009), 60–67.