

Rapport de projet d'ingénierie informatique

Aménagement d'intérieur en réalité virtuelle

Auteurs :
Sacha Malray, Antoine Patoux, Alex Thayse

Cours :
Projet d'ingénierie informatique

Année académique :
2024-2025

Professeur :
Mohammed Benjelloun

Promoteur :
Mathis Delehouzée

Table des matières

1 Objectif du projet	5
2 Cahier des charges	5
3 Etat de l'existant	5
4 Besoins du projet	8
4.1 Besoins Hardware	8
4.2 Besoins Software	8
4.2.1 Choix du moteur de jeu	8
4.2.2 Outils et bibliothèques logiciels	9
4.2.3 Estimation du prix	11
5 Planification et Organisation	11
6 Conception du projet	11
6.1 Choix de la programmation	11
6.1.1 Blueprint	11
6.1.2 C++	12
6.2 Fonctionnalités et composants de l'application	12
6.2.1 Déplacements	12
6.2.2 Interactions VR	13
6.2.3 Levels	15
6.2.4 Menus	15
6.2.5 Objets	25
6.2.6 Textures	25
6.2.7 Eclairage naturel	25
6.2.8 Eclairage artificiel	26
6.2.9 Sauvegarde de la session	26
6.2.10 Import des modèles et textures	28
6.2.11 Export des sauvegardes	28
7 Retour d'expérience utilisateur et évaluation de l'immersion	28
8 Résultats et évaluations	28
9 Perspectives d'amélioration	28
10 Conclusion	29

Table des figures

1	Logo de Homestyler	5
2	Logo de Ikea Place	6
3	Logo de SketchUp	6
4	Logo de Twinmotion	6
5	Logo de Home Design 3D VR	6
6	Meta Quest 2	8
7	Logo de OpenXR	9
8	Logo de GitHub	9
9	Logo de C++	10
10	Logo de Visual Studio Code	10
11	Logo de Fab	10
12	Logo de Visual Paradigm	10
13	Méthode Agile	11
14	Exemple simple de code en Blueprint	12
15	Diagramme d'utilisation - Déplacements	12
16	Diagramme d'utilisation - Interactions	13
17	Objet devenant rouge pendant la prévisualisation de la suppression	14
18	Prévisualisation du placement d'objet	14
19	Menu principal	15
20	Emplacement de sauvegarde	15
21	Menu de sauvegarde	16
22	Affichage de confirmation de la suppression	16
23	Menu interactif	17
24	Paramètres	17
25	Menu de sélection du type d'objet à placer	18
26	Menu de sélection du meuble	18
27	Menu de sélection de la texture à appliquer	19
28	Menu de modification de l'éclairage naturel	19
29	Menu de sélection de l'éclairage artificiel à appliquer	20
30	Confirmation de la réussite de la sauvegarde	20
31	Menu principal - Diagramme de cas d'utilisation	21
32	Menu interactif - Diagramme de cas d'utilisation	22
33	Menu principal - Diagramme d'activités	23
34	Menu interactif - Diagramme d'activités	24
35	Objets - Diagramme de classes	25
36	Struct_PlayerSaveData	27
37	Struct_SavedObject	27
38	Struct_SavedLamp	27
39	Struct_SavedDoor	27
40	Variables de BP_GameSave	27
41	Fonctions de BP_GameInstance	27

Liste des tableaux

1	Comparatif des solutions existantes avec notre projet	7
2	Comparaison entre Unreal Engine 5 et Unity	9

Résumé

Ce projet consiste en la conception d'une application d'aménagement d'intérieur en réalité virtuelle, offrant une expérience immersive et interactive. L'utilisateur a la possibilité de se déplacer librement dans l'espace et peut imaginer l'aménagement de ses rêves.

L'application permet de modifier en temps réel l'aménagement d'une pièce : ajouter, supprimer, déplacer, faire pivoter et redimensionner des objets. L'utilisateur peut également personnaliser les matériaux et textures des murs, sols, plafonds et meubles.

La modification de l'éclairage est intégrée, ce qui permet de visualiser l'influence de la lumière naturelle ainsi que des sources d'éclairage artificiel. Il est également possible d'importer des modèles 3D aux formats OBJ et FBX pour permettre à l'utilisateur d'ajouter ses propres créations.

Un système de sauvegarde et d'exportation permet à l'utilisateur de conserver ses projets et de les consulter sur d'autres appareils. Cette solution s'adresse aux professionnels de l'aménagement comme aux particuliers en leur permettant de mettre en lumière l'aménagement qu'ils ont imaginé.

Enfin, le projet est conçu sur Unreal Engine 5 reposant sur une implémentation en Blueprint et C++ et a été développé en utilisant le casque Oculus Quest 2 comme support principal.

Abstract

This project involves the design of a virtual reality interior design application, offering an immersive and interactive experience. The user has the ability to move freely within the space and can envision their dream layout.

The application allows real-time modifications of the room layout : adding, removing, moving, rotating, and resizing objects. The user can also customize the materials and textures of walls, floors, ceilings, and furniture.

Lighting modification is integrated, allowing the user to visualize the impact of natural light as well as artificial light sources. It is also possible to import 3D models in OBJ and FBX formats, enabling the user to use their own creations.

A saving and exporting system allows the user to preserve their projects and view them on other devices. This solution is aimed at both professionals in interior design and individuals, allowing them to bring their imagined layout to life.

Finally, the project is developed using Unreal Engine 5, incorporating both Blueprint and C++ implementations and was developed using the Oculus Quest 2 headset as the primary platform.

1 Objectif du projet

L'objectif de ce projet est de donner vie aux idées d'aménagement intérieur grâce à une application en réalité virtuelle innovante et interactive. Cette solution permet d'importer facilement des modèles 3D de meubles, murs et objets de décoration, offrant ainsi une personnalisation totale des espaces. Grâce à une simulation de l'éclairage naturel et artificiel, les utilisateurs peuvent tester différentes ambiances et ajuster leur environnement en temps réel. L'expérience immersive, optimisée pour un casque de réalité virtuelle, plonge l'utilisateur au cœur de son futur intérieur, lui permettant d'explorer chaque détail. De plus, les projets peuvent être sauvegardés et exportés pour une consultation sur d'autres supports, facilitant la collaboration et la prise de décision. Cette application s'adresse aux architectes, décorateurs, promoteurs immobiliers et particuliers désireux d'optimiser leur espace avant d'entamer des travaux en leur offrant un outil puissant, intuitif et inspirant.

2 Cahier des charges

Les fonctionnalités principales sont :

- Exploration en VR d'un espace 3D. Déplacements libres et téléportations.
- Modification en temps réel de l'aménagement. Ajout, suppression et déplacement d'objets. Rotation et redimensionnement des éléments. Choix des matériaux et textures pour les murs, sols plafonds et meubles.
- Simulation d'éclairage avec une gestion réaliste des ombres et reflets. Gestion de l'éclairage naturel. Simulation d'éclairage artificiel (lumières personnalisables).
- Importation et placement de modèles 3D (fichiers OBJ, FBX).

Ces fonctionnalités sont implémentées sur l'Oculus Quest fourni.

3 Etat de l'existant

La réalité virtuelle est un domaine en pleine expansion. Il est d'ores et déjà possible de développer des espaces avec un niveau d'immersion très élevé. Voici les différentes solutions existantes qui se rapprochent de ce que nous souhaitons créer :

Homestyler :



Homestyler est une application qui permet de créer et aménager des espaces en 3D. Les fonctionnalités principales sont la création de plans 2D et 3D, l'ajout de meubles et objets, et la possibilité d'utiliser la réalité augmentée.

Pour plus d'informations, consultez : <https://www.homestyler.com>.

FIGURE 1 – Logo de Homestyler

Ikea Place :



FIGURE 2 – Logo de Ikea Place

Ikea Place est une application qui permet de visualiser des meubles en réalité augmentée. Elle permet aux utilisateurs de placer virtuellement des meubles Ikea dans leurs maisons d'évaluer l'adéquation de l'espace et de faire des choix d'aménagement. Pour plus d'informations, consultez : <https://www.ikea.com/us/en/ikea-place/>.

SketchUp :



FIGURE 3 – Logo de SketchUp

SketchUp est un logiciel de modélisation 3D largement utilisé, notamment dans les domaines de l'architecture, de l'aménagement intérieur et du design. À la base, il permet de créer des modèles 3D précis et détaillés. Cependant, il se distingue également par sa compatibilité avec la réalité virtuelle permettant aux utilisateurs d'importer leurs projets et de les visualiser en immersion totale. Pour plus d'informations, consultez : <https://www.sketchup.com>.

Twinmotion :



FIGURE 4 – Logo de Twinmotion

Twinmotion, développé par Epic Games (les créateurs d'Unreal Engine), permet de réaliser des rendus ultraréalistes en temps réel, principalement pour les architectes et les designers. Ce logiciel intègre la réalité virtuelle offrant ainsi la possibilité d'explorer les projets de manière immersive. Pour plus d'informations, consultez : <https://www.twinmotion.com>.

Home Design 3D VR :



FIGURE 5 – Logo de Home Design 3D VR

L'application Home Design 3D VR se rapproche le plus de notre projet actuel. Elle permet la personnalisation d'intérieurs, la modification et l'importation de modèles 3D. Cependant, la simulation d'éclairage proposée par cette solution reste assez restreinte et les graphismes ne sont pas très réalistes. Une amélioration de cette fonctionnalité rendrait l'immersion bien meilleure.

Pour plus d'informations, consultez : <https://www.homedesign3d.net>.

Voici un tableau comparatif qui permet d'identifier les points dans lesquels notre projet se démarque :

Critère / Outil	Homestyler	Ikea Place	SketchUp	Twinmotion	Home Design 3D VR	Notre projet
Modélisation 3D libre	Oui (plans 2D/3D simples)	Non	Oui (modélisation avancée)	Non (rendu uniquement)	Partielle	Oui, modélisation complète avec interaction directe
Réalité virtuelle (VR)	Non (RA uniquement)	Non (RA uniquement)	Oui (via plugins)	Oui	Oui	Oui, VR native avec interaction immersive
Éclairage réaliste	Basique	Non	Possible via extensions	Très réaliste (temps réel)	Limité	Oui, gestion avancée de l'éclairage naturel et artificiel
Interaction avec objets	Glisser-déposer	Placement statique	Manipulation souris/clavier	Non	Ajout simple	Ajout, suppression, déplacement, rotation, redimensionnement en temps réel
Import de modèles personnalisés	Limité	Non	Oui (formats variés)	Oui	Oui (limité)	Oui, import en runtime de fichiers OBJ et FBX
Personnalisation des matériaux	Basique	Non	Oui	Oui	Moyenne	Oui, personnalisation murs, sols, plafonds, meubles
Rendu graphique	Moyen	Réaliste (objets Ikea)	Variable	Très réaliste	Moyen à faible	Réaliste grâce à Unreal Engine 5
Sauvegarde et exportation	Oui	Non	Oui (fichiers projet)	Oui (exports visuels)	Oui (local)	Oui, sauvegarde intégrée + export multiplateforme

TABLE 1 – Comparatif des solutions existantes avec notre projet

4 Besoins du projet

4.1 Besoins Hardware

L'unique matériel nécessaire pour mener à bien ce projet est un casque de réalité virtuelle et un ordinateur permettant d'utiliser le moteur Unreal Engine 5 et l'application en elle-même. En effet, la casque est connecté au PC via un câble USB-C. Cela permet de tirer parti de la puissance de calcul de l'ordinateur pour le rendu 3D.

Pour ce projet, la configuration de l'ordinateur est :

- **Carte graphique** : NVIDIA GeForce RTX 4060 (Laptop)
- **Processeur** : Intel Core i7-14650HX
- **Mémoire vive (RAM)** : 16 Go

Pour ce projet, le casque utilisé est le Meta Quest 2 :



FIGURE 6 – Meta Quest 2

Pour plus d'informations sur le Meta Quest 2, consultez : https://www.meta.com/be/fr/quest/products/quest-2/?srsltid=AfmB0orXx7LdwrGCmTnyn97_0RJXjNQj4uHf-Y8LrYqcVxC9ABX4L-

4.2 Besoins Software

4.2.1 Choix du moteur de jeu

Les deux principaux moteurs utilisés pour la VR sont Unity et Unreal Engine 5. Nous présentons un tableau de comparaison pour pouvoir justifier notre choix :

Critères	Unreal Engine 5	Unity
Langage de programmation	Blueprints et C++	C#
Facilité d'apprentissage	Complexé	Plus simple
Qualité graphique	Ultra-réaliste	Moins réaliste
Performance et optimisation	Gourmand en ressources	Plus léger
Mode Collaboratif en VR	Possible avec Unreal Multi-User Editing	A développer avec des outils externes
Outils d'éclairage	Lumen, Ray Tracing	Baking, URP pour l'optimisation

TABLE 2 – Comparaison entre Unreal Engine 5 et Unity

Nous avons choisi de créer notre application VR avec Unreal Engine 5 car celui-ci est ultraréaliste et permet une meilleure gestion de l'éclairage. Bien qu'il soit plus complexe, la modélisation en Blueprint est assez intuitive et la difficulté résulte plus dans l'utilisation de C++. De plus, ce moteur permettra un développement plus poussé dans le futur.

Pour plus d'informations sur Unreal Engine 5, consultez : <https://www.unrealengine.com/en-US/unreal-engine-5>

Pour plus d'informations sur Unity, consultez : <https://unity.com/fr>

4.2.2 Outils et bibliothèques logiciels

OpenXR :



FIGURE 7 – Logo de OpenXR

OpenXR est une API standardisée, développée par Khronos Group, permettant de créer des applications de réalité virtuelle et réalité augmentée. Cette interface de programmation permet de ne pas devoir reprogrammer le code lorsque l'on change de constructeur de casque ou de plateforme. Sur Unreal Engine 5, OpenXR est intégré nativement et est recommandé par Epic Games pour le développement de projet VR.

Pour plus d'informations sur OpenXR, consultez : <https://www.khronos.org/openxr/>

GitHub :



FIGURE 8 – Logo de GitHub

GitHub est une plateforme de gestion de versions et de collaboration que nous avons utilisé pour gérer le travail en équipe permettant le partage de nos modifications du projet rapidement et efficacement. Pour plus d'informations sur GitHub, consultez : <https://github.com/>

C++ :



FIGURE 9 – Logo de C++

C++ est un langage de programmation qui nous a été utile pour implémenter les fonctionnalités plus complexes de l'application qui ne sont pas faisables en Blueprint.

Pour plus d'informations sur C++, consultez : <https://isocpp.org/>

Visual Studio Code :

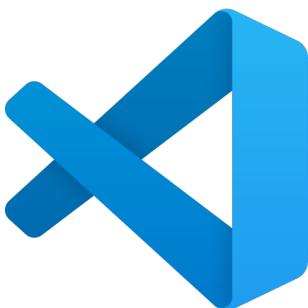


FIGURE 10 – Logo de Visual Studio Code

Visual Studio Code est un environnement de développement que nous avons utilisé pour implémenter nos scripts C++.

Pour plus d'informations sur Visual Studio Code, consultez : <https://code.visualstudio.com/>

Fab :



FIGURE 11 – Logo de Fab

Fab est un marché numérique qui offre aux créateurs un endroit pour découvrir, partager, acheter et vendre des assets de jeu de haute qualité..

Pour plus d'informations sur Fab, consultez : <https://www.fab.com/>

Visual Paradigm :



FIGURE 12 – Logo de Visual Paradigm

Visual Paradigm est un outil de modélisation visuelle utilisé pour la conception de logiciels, la modélisation UML, la gestion de projets, ainsi que la documentation technique. Nous utilisons ce logiciel afin de modéliser plusieurs diagrammes permettant de comprendre plus facilement certaines interactions que permettent ce projet.

Pour plus d'informations sur Visual Paradigm, consultez : <https://www.visual-paradigm.com/>

4.2.3 Estimation du prix

Le casque étant prêté par l'UMONS, nous n'avons pas besoin de budget pour le matériel. Les seules dépenses étaient un abonnement durant 3 mois à 5€/mois pour GitHub Large File Storage car les packs de modèles étaient trop volumineux pour la version gratuite de GitHub.

5 Planification et Organisation

Lors de ce projet, nous allons utiliser la méthode Agile afin de fournir un produit fini à chaque point de contrôle. Nous avons fait ce choix afin d'avoir un processus itératif, cela nous permettra en temps réel de mieux évaluer les différents temps que nécessitent l'implémentation de chaque fonctionnalité afin d'améliorer l'organisation du projet. L'utilisation de la méthode Agile nous permettra également d'avoir plus de contrôle sur le rendu du projet en ayant régulièrement une version fonctionnelle.

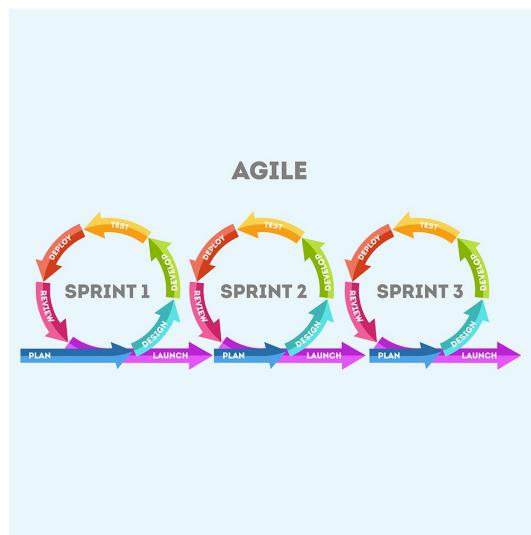


FIGURE 13 – Méthode Agile

6 Conception du projet

6.1 Choix de la programmation

6.1.1 Blueprint

Les fonctionnalités internes de l'application ont en majorité été développées en Blueprint. Le Blueprint est un langage de programmation visuelle qui permet de créer la logique d'un jeu ou d'une application sans écrire de code, en connectant des noeuds visuels représentant des fonctions, des événements ou des variables. Grâce à ce langage, les fonctionnalités principales nécessaires au projet ont pu être implémentées sans recourir à du code en C++.

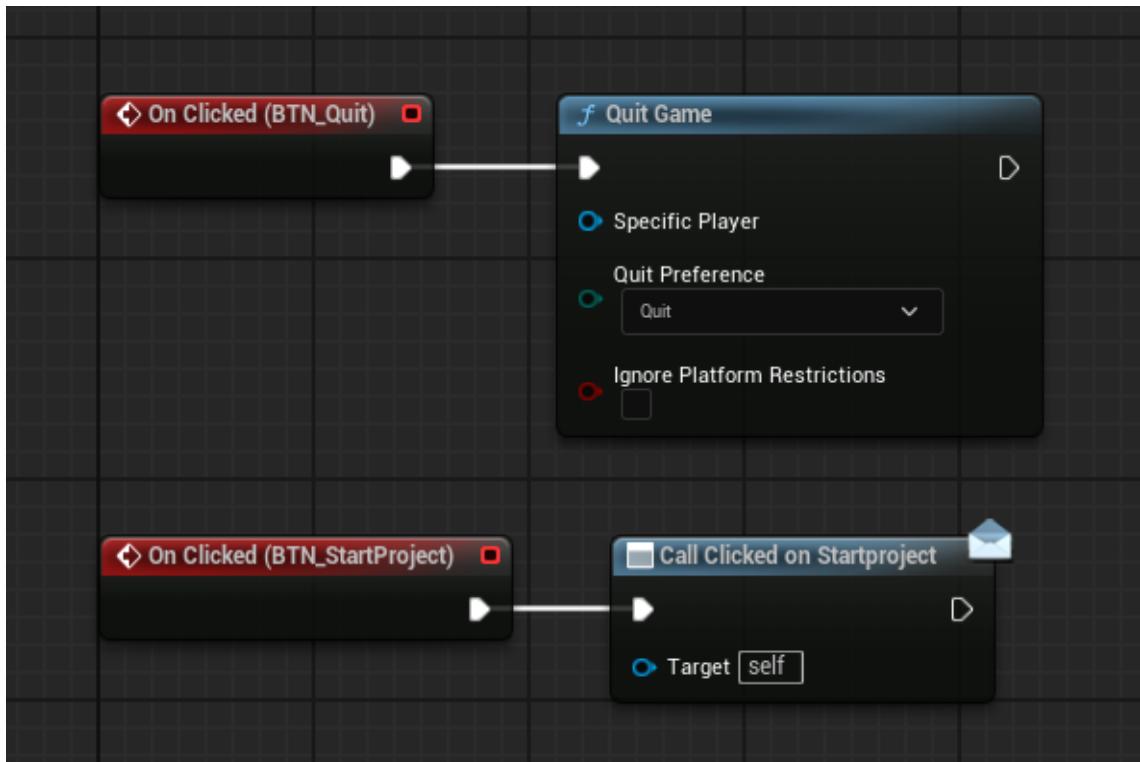


FIGURE 14 – Exemple simple de code en Blueprint

6.1.2 C++

Nous avons implémenté certaines parties des fonctionnalités non réalisables en Blueprint en C++. Nous avons donc lié Unreal Engine à VS Code afin de pouvoir utiliser les codes sources dans notre projet.

6.2 Fonctionnalités et composants de l'application

6.2.1 Déplacements

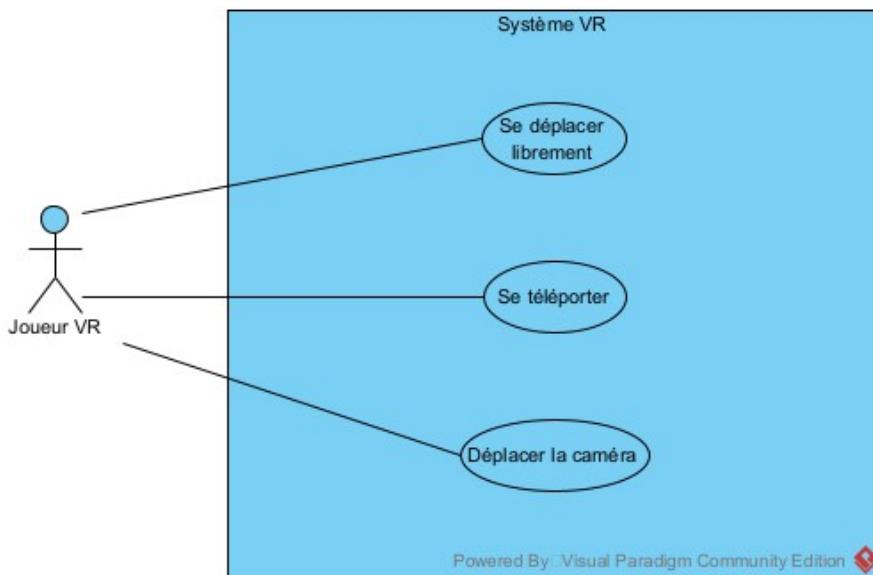


FIGURE 15 – Diagramme d'utilisation - Déplacements

Déplacements libres :

En utilisant le joystick gauche, le joueur peut se déplacer librement sur la carte. La vitesse de déplacement varie en fonction de l'inclinaison du joystick.

Téléportations :

En poussant le joystick droit vers le haut, le joueur peut se téléporter à l'endroit visé par sa manette droite. Une prévisualisation de sa position après téléportation est affichée.

6.2.2 Interactions VR

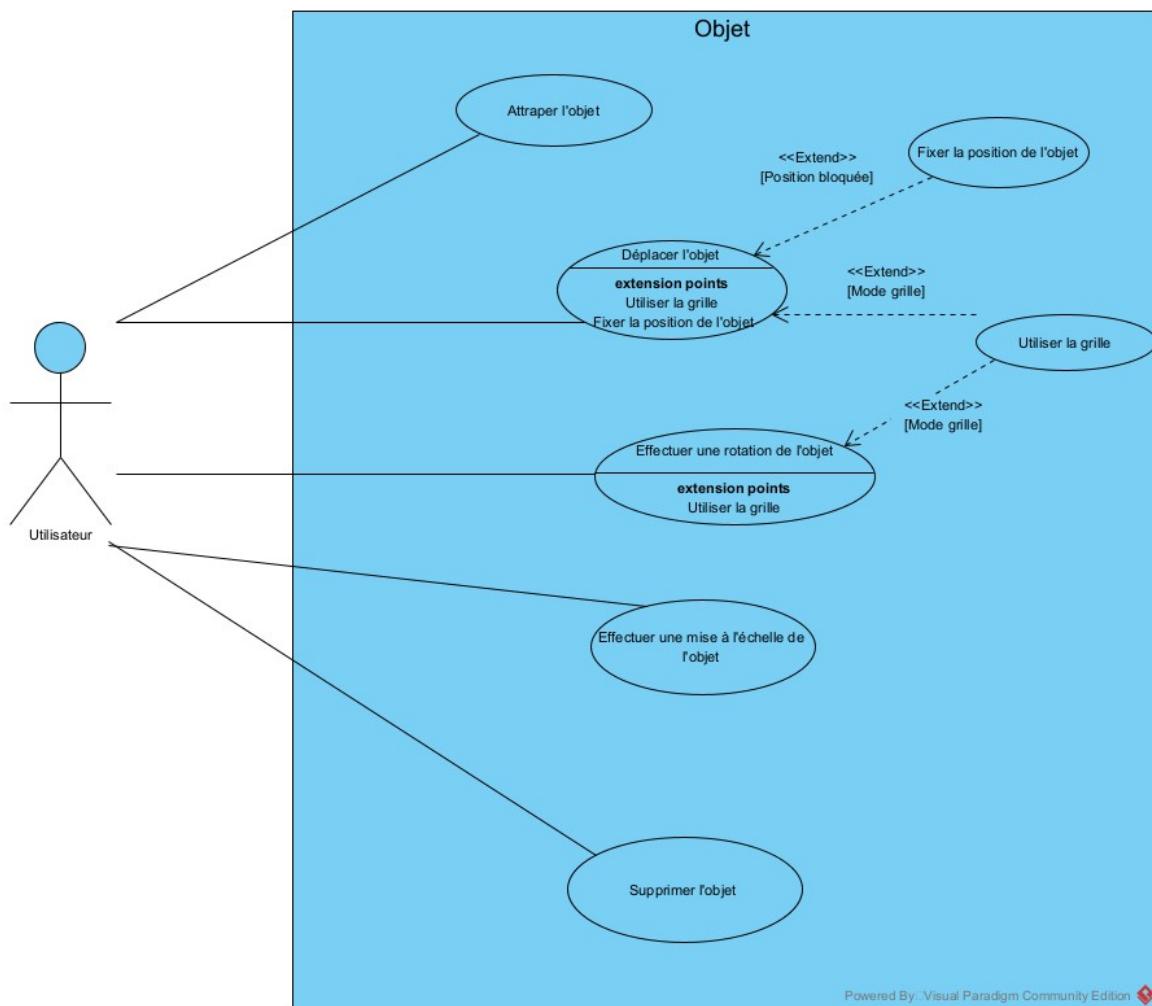


FIGURE 16 – Diagramme d'utilisation - Interactions

Nos interactions reposent pour la plupart sur la mécanique d'attrapage de l'objet. Cette mécanique fait que lorsqu'on pointe vers un objet implementant l'interface `InteractionsInterface`, l'utilisateur peut l'attraper pour pouvoir le déplacer. Il est à prendre en compte qu'on ne peut attraper qu'un objet simultanément.

1. Mode Sélection : permet d'interagir avec certains objets implémentant une fonctionnalité spécifique comme les portes.
2. Mode Déplacement : permet de déplacer les objets attrapés soit selon l'axe reliant le joueur à l'objet, soit par rapport à une grille. L'utilisateur peut fixer la position de l'objet et se déplacer avant de le placer.
3. Mode Rotation : permet d'effectuer la rotation de l'objet attrapé.

4. Mode Scaling : permet d'appliquer une mise à l'échelle sur un objet, qu'il soit attrapé ou posé au sol.
5. Mode Suppression : permet de supprimer des objets. Les objets supprimables deviendront rouges lorsque l'utilisateur passe le pointeur de la manette dessus.

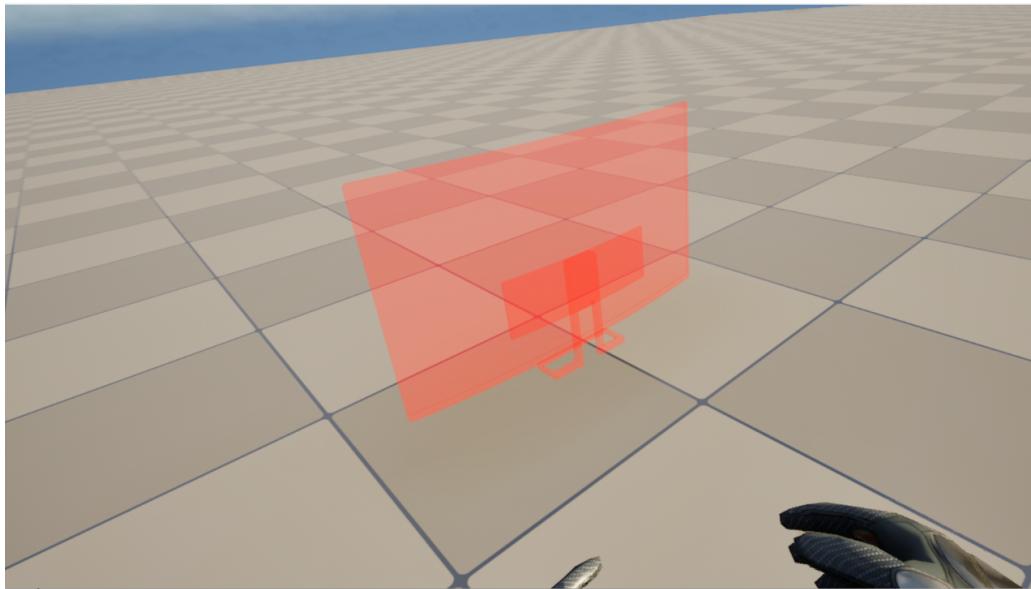


FIGURE 17 – Objet devenant rouge pendant la prévisualisation de la suppression

Il est à noter que l'on peut attraper des objets dans tous les modes sauf le mode suppression.

Une autre fonctionnalité disponible est la possibilité de placer des objets. Pour cela, le joueur peut cliquer sur le bouton "Placer un objet" dans le menu situé sur sa main. Il peut ensuite choisir l'objet souhaité et le positionner grâce au head tracking, qui permet de placer l'objet dans la direction de son regard. Une prévisualisation du placement est également affichée pour aider le joueur.

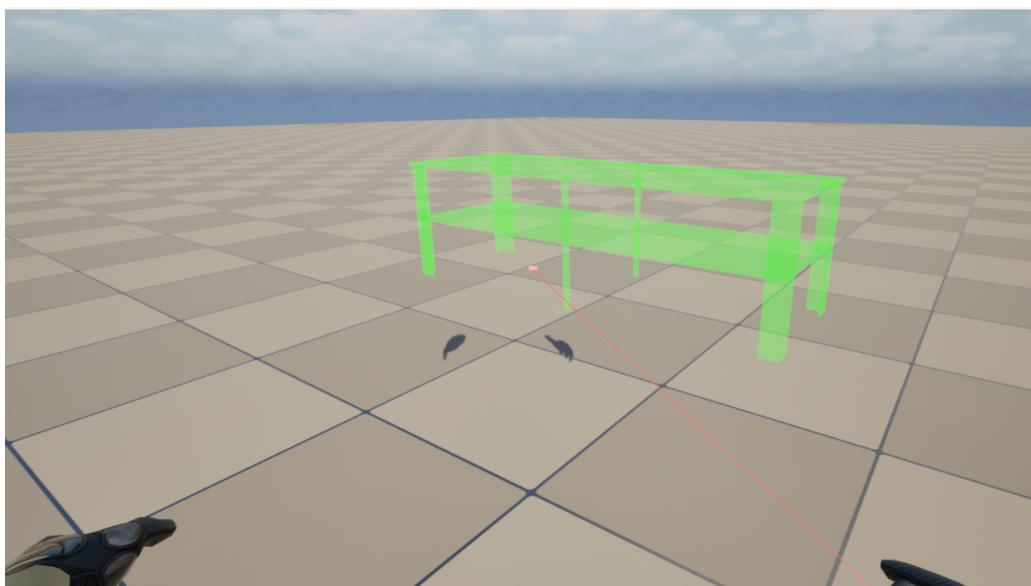


FIGURE 18 – Prévisualisation du placement d'objet

6.2.3 Levels

L'application contient deux levels différents :

- Lobby est le level dans lequel le joueur apparaît lors du lancement du jeu. Il est composé d'une pièce dans laquelle est posée le menu principal.
- BaseLevel est le level qui est lancé à chaque fois qu'une partie est lancée. Lors du lancement de la sauvegarde, le joueur apparaît dans ce level qui n'est composé que d'un sol.

6.2.4 Menus

Menu principal :

Le menu principal se trouve dans le level Lobby. Ainsi, chaque joueur qui lance le jeu ou souhaite le quitter doit passer par ce menu, que ce soit pour accéder aux parties sauvegardées ou pour fermer le jeu.



FIGURE 19 – Menu principal

Emplacement de sauvegarde :

L'emplacement de sauvegarde est composé de deux boutons. Un bouton sert pour lancer la sauvegarde correspondant à cet emplacement. L'autre est nécessaire pour supprimer la sauvegarde de l'emplacement. Un texte interactif permet d'afficher la date de la dernière sauvegarde. Si il n'y a pas encore de partie sauvegardée à cet emplacement, le texte affiché est : 1/1/1 00 : 00.



FIGURE 20 – Emplacement de sauvegarde

Menu de sauvegarde :

Le menu de sauvegarde est accessible depuis le menu principal. Il est composé de 3 emplacements de sauvegarde et d'un bouton "Retour" afin de revenir au menu principal.



FIGURE 21 – Menu de sauvegarde

Confirmation de la suppression d'une sauvegarde :

Cet affichage permet de demander à l'utilisateur s'il souhaite réellement confirmer la suppression lorsqu'il appuie sur le bouton de suppression de l'emplacement de sauvegarde. Il peut également appuyer sur le bouton "Annuler" pour, tout simplement, revenir sur le menu de sauvegarde. Cela permet d'éviter les mauvaises manipulations et des suppressions non-souhaitées.



FIGURE 22 – Affichage de confirmation de la suppression

Menu interactif :

Le menu interactif est un menu que le joueur peut faire apparaître à tout moment de sa partie. Ce menu est attaché à la main de l'utilisateur. Le joueur a accès aux différentes fonctionnalités principales : placer un objet, changer une texture, modifier l'éclairage, sauvegarder la partie, retourner au Lobby. L'utilisateur peut également choisir le mode d'interaction qu'il veut avec les objets. Les modes d'interactions sont disposés en bas du Widget et représentés par des pictogrammes qui correspondent respectivement de gauche à droite : le mode normal, le mode de déplacement des objets, le mode de rotation des objets, le mode de mise à l'échelle des objets et le mode de suppression des objets. Le menu dispose également d'un pictogramme correspondant aux paramètres, permettant au joueur d'ajuster certaines fonctionnalités.



FIGURE 23 – Menu interactif

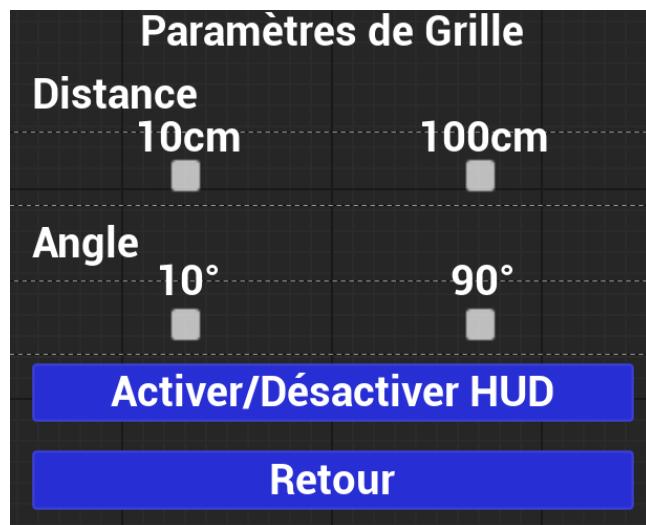


FIGURE 24 – Paramètres

Menu de sélection du type d'objet à placer :

Après avoir choisi de placer un objet dans le menu interactif, le joueur voit apparaître un menu qui lui permet de sélectionner le type d'objet qu'il souhaite. Les deux types sont les meubles et les constructions. Dans la catégorie "Meubles et décos", on retrouve tout ce qui n'est pas un élément de construction, c'est-à-dire les meubles et les lampes. Dans "Constructions", on retrouve tout ce qui est sols, murs, toits, portes, fenêtres, etc.



FIGURE 25 – Menu de sélection du type d'objet à placer

Menu de sélection de l'objet à placer :

Après avoir choisi une catégorie d'objet, l'utilisateur voit un menu qui correspond aux objets de la base de données qu'il peut placer dans le level. Il dispose d'une barre défilement sur la droite permettant ainsi la navigation parmi les différents objets.

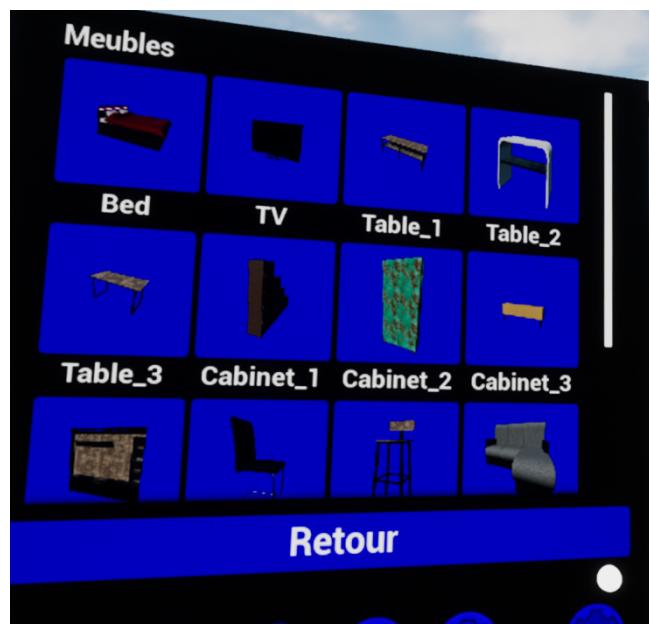


FIGURE 26 – Menu de sélection du meuble

Menu de sélection de la texture à appliquer :

Lorsque le joueur clique sur le bouton "Appliquer une texture" dans le menu interactif, il peut ensuite choisir le type de texture qu'il souhaite utiliser comme, par exemple, du bois, de la pierre, etc. Nous ne remettons pas l'affichage de ce menu car il est semblable à celui pour les objets à placer.

Menu de sélection du type d'éclairage à modifier :

Après avoir cliqué sur le bouton "Modifier l'éclairage" du menu interactif, le joueur voit un menu qui lui permet de choisir entre l'éclairage naturel ou artificiel.



FIGURE 27 – Menu de sélection de la texture à appliquer

Menu de modification de l'éclairage naturel :

Lors de la modification de l'éclairage naturel, le joueur a le choix entre différents éclairages prédéfinis : ensoleillé, pluvieux, neigeux, coucher de soleil et la nuit. Ce choix est fait en remplissant le bouton faisant dérouler les choix (combobox). Le choix initial est "ensoleillé". Après avoir cliqué sur le bouton de confirmation, l'éclairage naturel change selon le choix effectué.

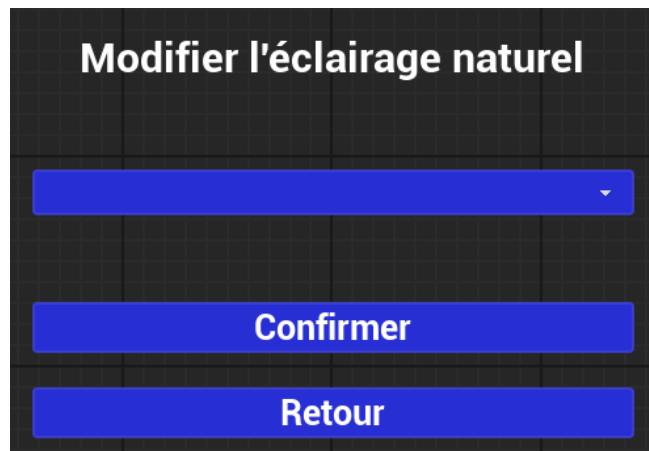


FIGURE 28 – Menu de modification de l'éclairage naturel

Menu de sélection de l'éclairage artificiel à appliquer :

Lors de la modification de l'éclairage artificiel, le joueur peut cliquer sur le bouton "Changer l'intensité". Le joueur peut donc, en cliquant sur deux boutons différents en pointant une lampe, augmenter ou diminuer l'intensité de l'éclairage. L'utilisateur peut aussi créer sa couleur avec les valeurs RGB modifiables. Ensuite, l'utilisateur peut, en cliquant sur le bouton "Changer la couleur", changer la couleur des lampes qu'il pointe.

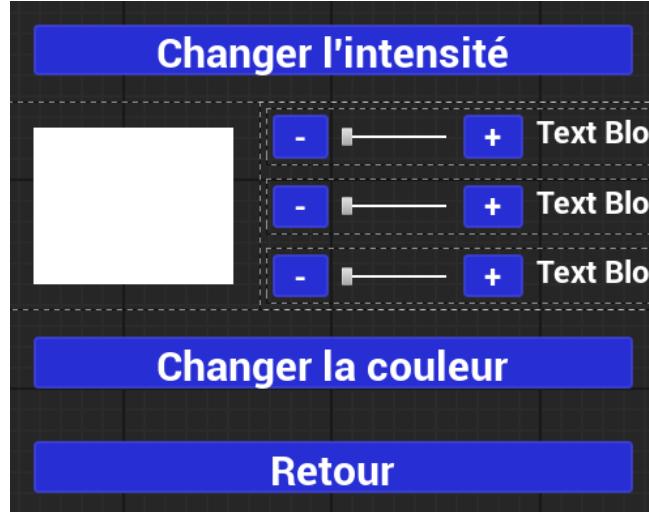


FIGURE 29 – Menu de sélection de l'éclairage artificiel à appliquer

Confirmation de la réussite de la sauvegarde :

Un affichage apparaît pendant 1 seconde sur lequel il est écrit "Sauvegarde réussie" dans le cas où la sauvegarde s'est bien passé. Sinon , il ne se passe rien et l'utilisateur sait que la sauvegarde ne s'est pas effectué ou qu'il n'a pas bien cliqué sur le bouton de sauvegarde.



FIGURE 30 – Confirmation de la réussite de la sauvegarde

Diagrammes UML :

Les architectures du menu principal et du menu interactif sont représentées à partir de diagrammes de cas d'utilisation et d'activités.

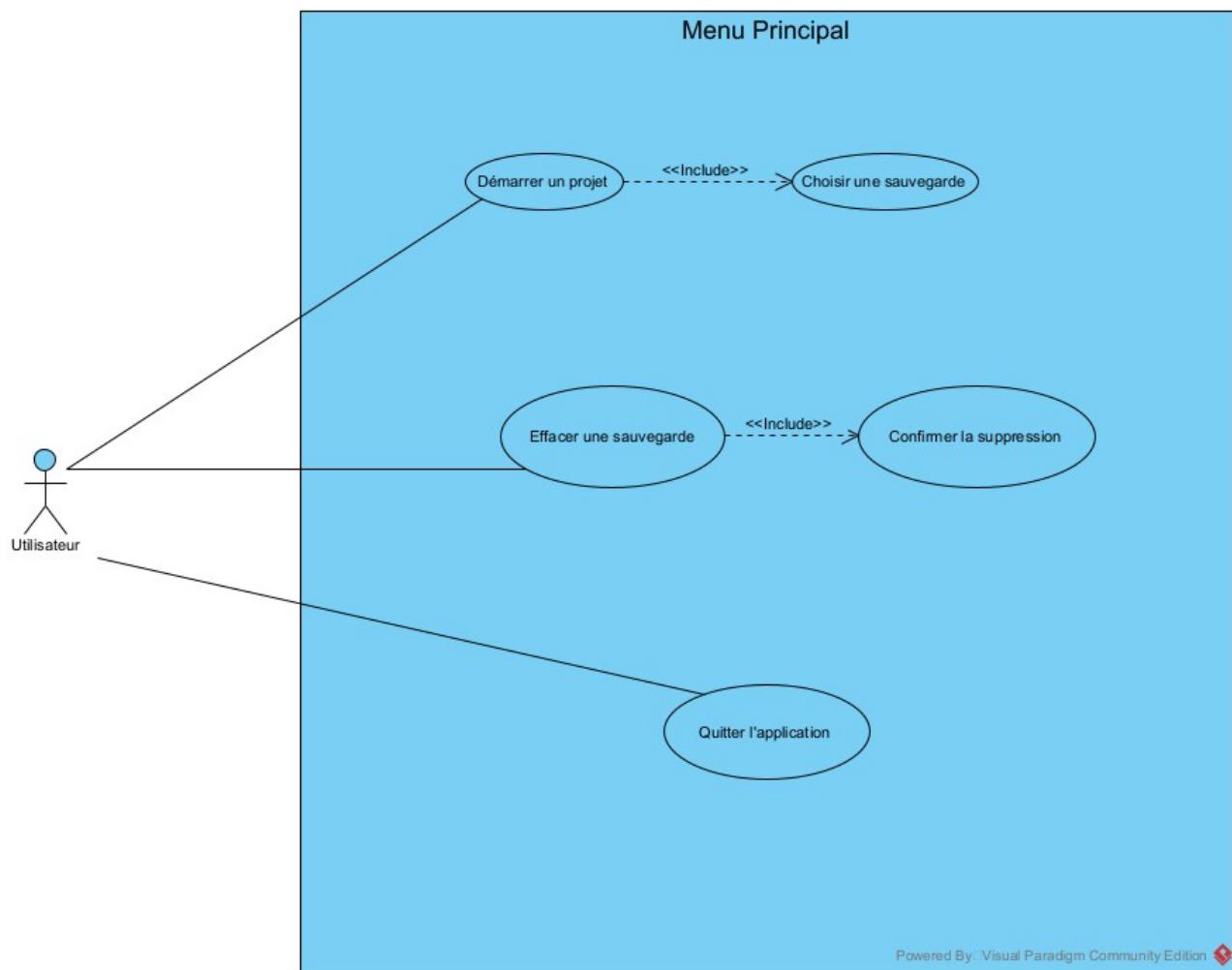


FIGURE 31 – Menu principal - Diagramme de cas d'utilisation

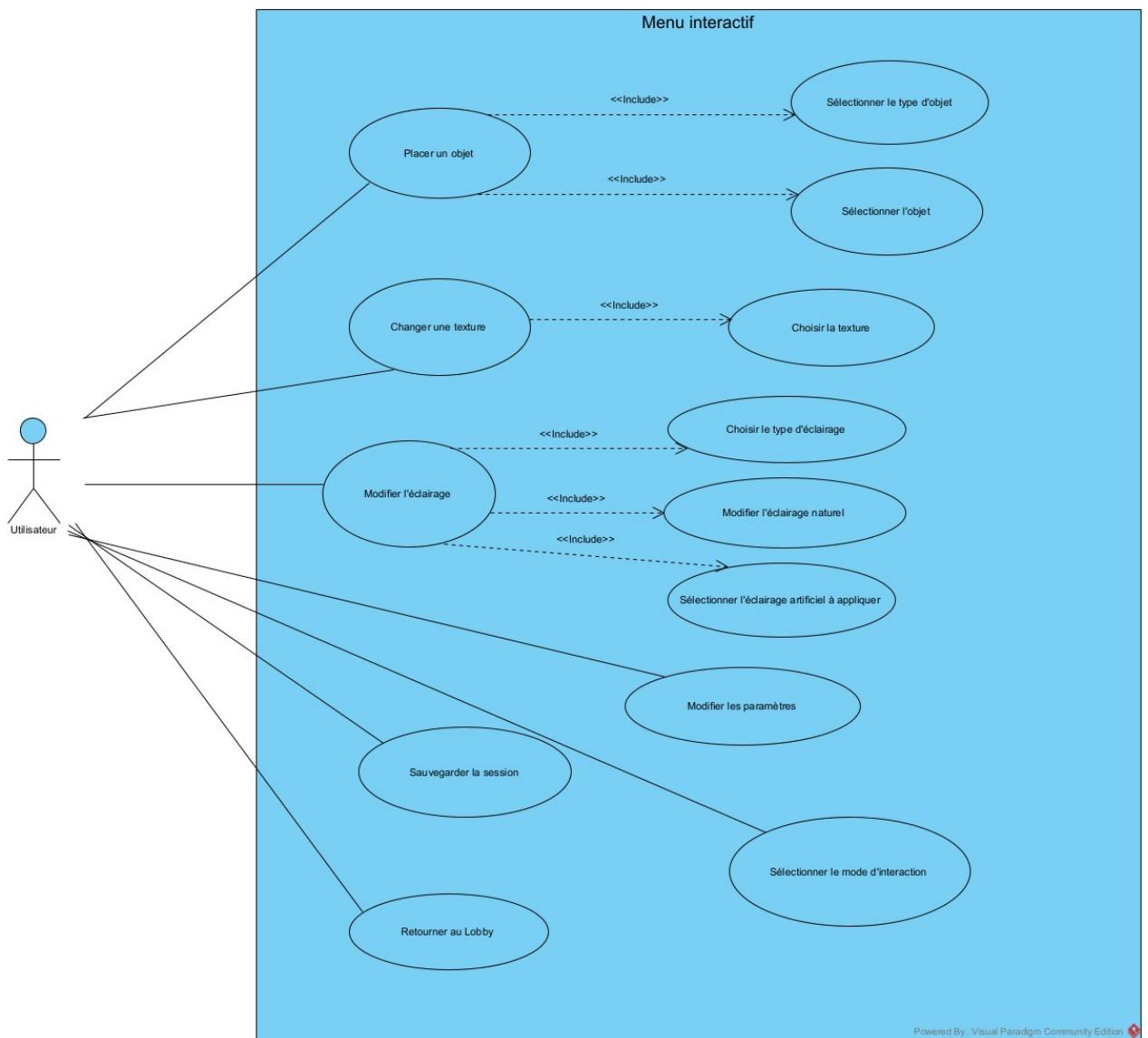


FIGURE 32 – Menu interactif - Diagramme de cas d'utilisation

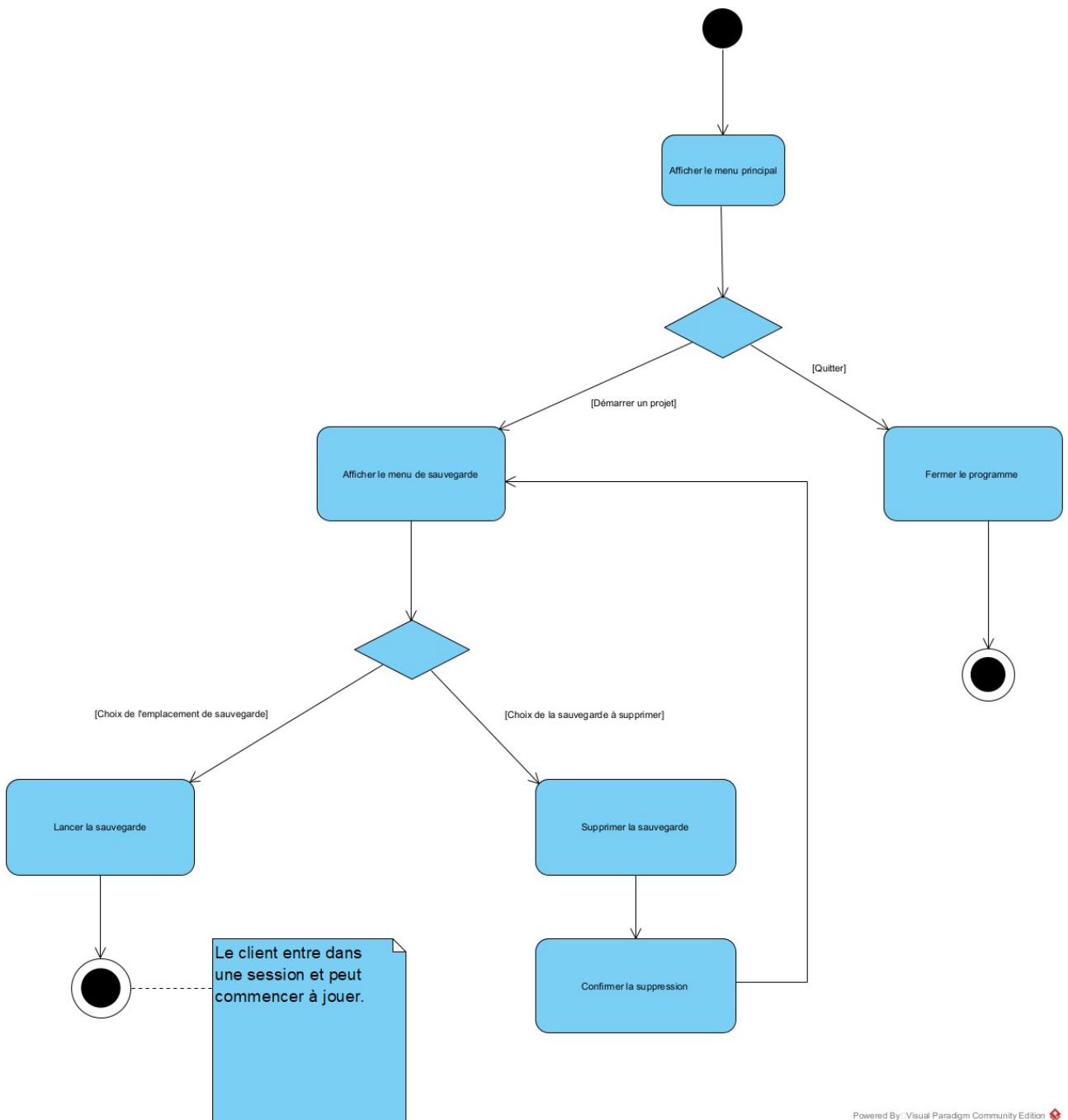


FIGURE 33 – Menu principal - Diagramme d’activités

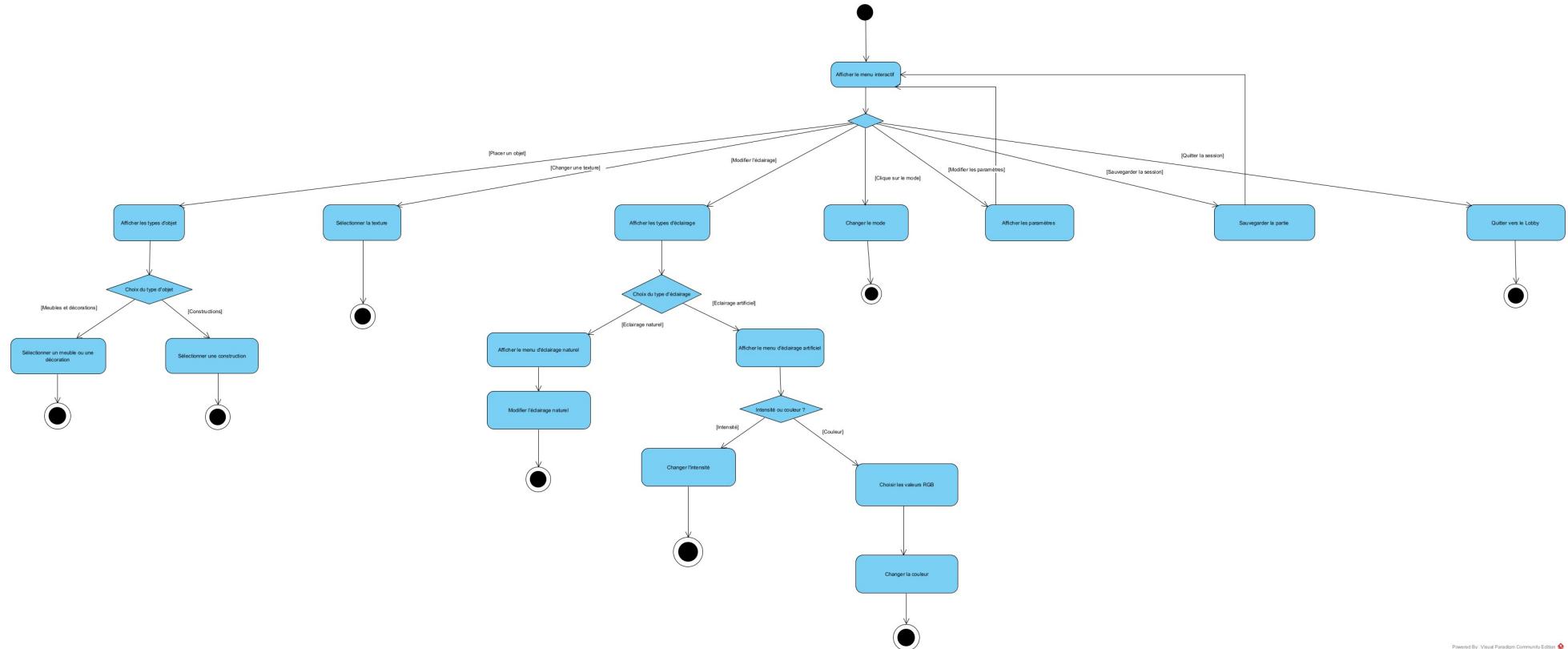


FIGURE 34 – Menu interactif - Diagramme d’activités

6.2.5 Objets

Les fonctionnalités de translation, rotation, mise à l'échelle et suppression sont applicables sur les objets de la classe BuildActor. Cette classe hérite de la classe Actor qui est native de Unreal Engine et implémente l'interface InteractionsInterface qui permet d'implémenter la fonctionnalité pour pouvoir attraper les objets. Le BuildActor est simplement composé d'un staticmeshcomponent, le modèle 3D de l'objet.

Objets responsables de l'éclairage artificiel :

Ces objets proviennent de la classe LightActor qui hérite de BuildActor et implémente l'interface Light Interface. Ils ont donc toutes les fonctionnalités d'un objet quelconque car ils implémentent l'interface InteractionsInterface par héritage. Cet Actor, en plus d'un staticmesh-component, possède un pointlight component. Les fonctionnalités propres à un objet lumineux sont ajoutées à cette classe c'est-à-dire le changement d'intensité et de couleur via l'interface Light Interface.

Portes et fenêtres :

Ces objets proviennent de la classe DoorActor qui implémente l'interface Door Interface et hérite également de la classe BuildActor. Ils héritent des fonctionnalités d'un objet quelconque car ils héritent de InteractionsInterface. L'ouverture et fermeture de la porte sont implémentées via l'interface Door Interface.

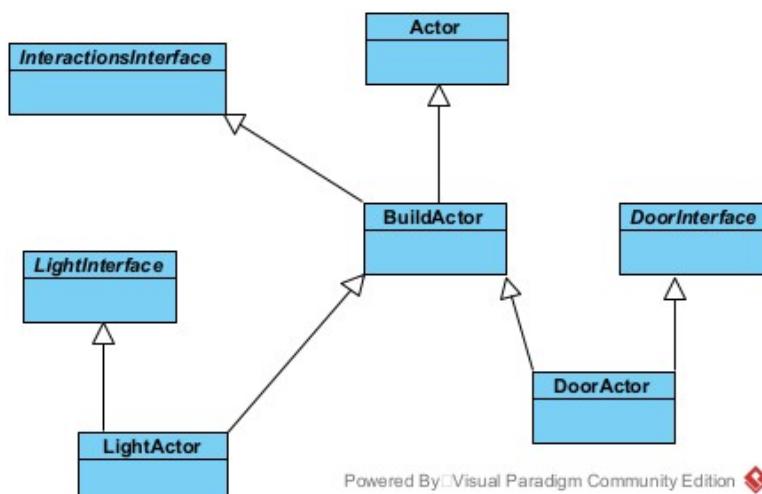


FIGURE 35 – Objets - Diagramme de classes

6.2.6 Textures

Nous avons mis en place un système permettant de changer les textures. Lorsque le joueur accède au menu situé sur sa main, il peut sélectionner l'option "Changer les textures", puis choisir la texture souhaitée. Il est alors automatiquement placé en mode "changement de texture" et peut modifier la texture de l'objet vers lequel il pointe en appuyant sur la gâchette de la manette droite.

Pour les objets ayant plusieurs

6.2.7 Eclairage naturel

L'éclairage naturel permet de complètement changer l'atmosphère du level. Un Actor, "BP_WeatherManager" permet de changer l'éclairage en fonction du choix fait dans la combobox. Il permet d'appliquer l'Actor dans la scène correspondant à l'éclairage choisi. Ces Ac-

tors sont : BP_Weather_Sunny, BP_Weather_Rainy, BP_Weather_Snow, BP_Weather_Sunset et BP_Weather_Night correspondant respectivement à ensoleillé, pluvieux, neigeux, coucher de soleil et nuit. Ces différentes ambiances sont paramétrées à l'aide de différents composants : DirectionalLight, SkyLight, SkyAtmosphere, VolumetricCloud et SkyAtmosphere. Pour la nuit, nous avons ajouté une SkySphere qui permet d'avoir un ciel étoilé. Pour la pluie et la neige, les particules ont été ajoutées en créant des Niagara Particle System Components. Les aperçus des différents éclairages naturels sont disponibles en Annexe A.

6.2.8 Eclairage artificiel

L'éclairage artificiel permet de modifier deux aspects des objets lumineux. Le premier aspect est leur intensité. Si l'utilisateur choisi de changer celui ci, il est amené à appuyer sur les boutons A et B pour respectivement diminuer et augmenter l'intensité de l'objet lumineux pointé par le pointeur laser. Le deuxième aspect est leur couleur. Le joueur doit tout d'abord choisir la couleur qu'il souhaite appliquer. Une fois satisfait, il peut appliquer cette couleur à l'objet lumineux pointé.

6.2.9 Sauvegarde de la session

Un menu de sauvegarde contient 3 emplacements de sauvegarde. Lorsque l'utilisateur clique sur un emplacement vide, le level BaseLevel est chargé et l'utilisateur apparaît à la position du PlayerStart. Lorsque l'utilisateur clique sur un emplacement contenant une sauvegarde, il apparaît avec sa position sauvegardée dans le level BaseLevel et les objets sauvegardés sont placés également dans le level. Ensuite, lorsque l'utilisateur sauvegarde via le menu interactif, la position du joueur et les objets sont sauvegardés dans le stockage de l'emplacement choisi.

La classe Blueprint "BP_GameInstance" est créée pour gérer des données persistantes durant une session de jeu. Contrairement aux autres classes comme le Player Controller ou le Player Character qui sont recréées à chaque fois que l'on change de level, cette classe reste active en mémoire durant toute l'exécution du jeu. Elle permet de stocker les sauvegardes dans des objets de la classe "BP_GameSave". BP_GameInstance comporte les fonctions SavePlayerData, LoadPlayerData, ClearPlayerData, Autosave et RetrieveSaveData. SavePlayerData permet de sauvegarder les données, LoadPlayerData permet de les charger au lancement de la session, ClearPlayerData permet de supprimer une sauvegarde, Autosave qui permet une sauvegarde automatique toutes les 10 minutes et RetrieveSaveData qui permet d'afficher la date de la sauvegarder sur son emplacement dans le menu de sauvegarde.

Nous avons créé les structures Struct_PlayerSaveData, Struct_SavedLamp, Struct_SavedDoor et Struct_SavedObject. Struct_PlayerSaveData contient la variable Transform de type Transform qui permet de sauvegarder la position du joueur. Struct_SavedObject permet de sauvegarder un BuildActor. Elle contient les variables ObjectTransform et ObjectClass qui permettent respectivement de sauvegarder la position du BuildActor et la classe et de garder une référence de la classe BuildActor. Struct_SavedLamp permet de sauvegarder un LightActor. Elle contient les variables LampTransform, LampClass, LampIntensity et LampLinearColor. Les variables qui diffèrent avec Struct_SavedObject sont LampIntensity et LampLinearColor qui permettent respectivement de sauvegarder l'intensité et la couleur du LightActor. Struct_SavedDoor permet de sauvegarder un DoorActor. Elle contient les variables DoorTransform, DoorClass et DoorState. La variable qui diffère avec Struct_SavedObject est DoorState qui permet de sauvegarder l'état de la porte, c'est-à-dire, si elle est ouverte ou non.

Le BP_GameSave présente comme variable PlayerData de type Struct_PlayerSaveDate, SaveDate qui permet de sauvegarder la date de sauvegarde afin de l'afficher sur l'emplacement

de sauvegarde, ObjectsData qui est un vecteur de Struct_SavedObject permettant de sauvegarder tous les BuildActors de la scène, LampsData qui est un vecteur de Struct_SavedLamp permettant de sauvegarder tous les LampActors de la scène et DoorsData qui est un vecteur de Struct_SavedDoor permettant de sauvegarder tous les DoorActors de la scène.

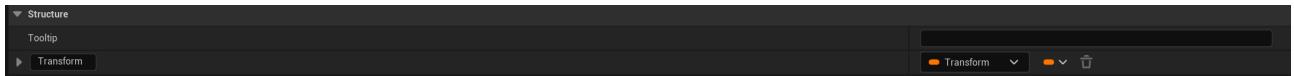


FIGURE 36 – Struct_PlayerSaveData

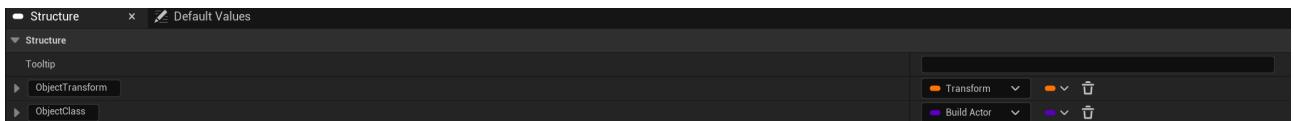


FIGURE 37 – Struct_SavedObject



FIGURE 38 – Struct_SavedLamp



FIGURE 39 – Struct_SavedDoor

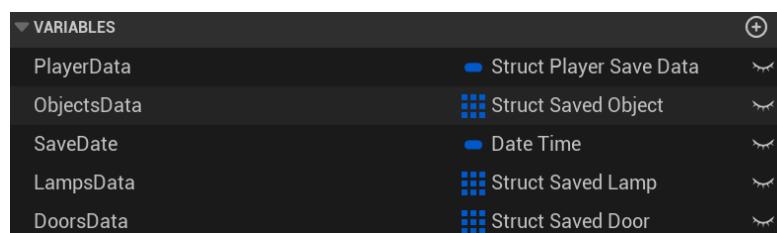


FIGURE 40 – Variables de BP_GameSave

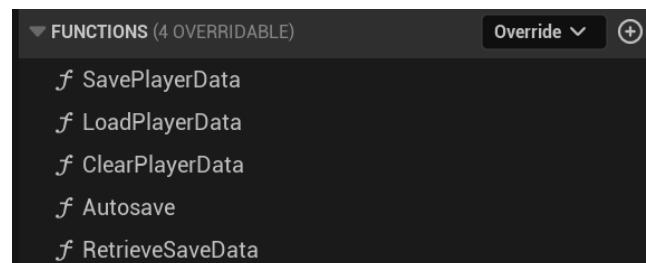


FIGURE 41 – Fonctions de BP_GameInstance

6.2.10 Import des modèles et textures

L'importation des modèles se base sur le projet github opensource UE4_Assimp. En utilisant la librairie de Assimp qui gère divers format de fichiers 3D, adapté pour une utilisation sous Unreal Engine. Un bouton est disponible sur le menu principal. Celui ci permet d'ouvrir l'explorateur de fichiers avec un chemin spécifique, permettant de montrer à l'utilisateur à quel endroit il doit insérer ses fichiers 3D.

6.2.11 Export des sauvegardes

L'export des sauvegarder se fait en transférant uniquement les objets importés et le BP_GameSave qui détient toutes les informations nécessaires pour pouvoir transférer sa sauvegarde sur un autre appareil.

7 Retour d'expérience utilisateur et évaluation de l'immersion

Afin d'évaluer l'ergonomie et la qualité immersive de l'application, nous l'avons fait tester à une dizaine de personnes. Chaque test a duré environ 10 minutes au cours desquelles les utilisateurs ont pu explorer librement l'environnement et utiliser les différentes fonctionnalités. Globalement, les retours ont été très positifs. La majorité ont souligné un fort degré d'immersion, un rendu réaliste et des commandes intuitives. Ces dernières sont affichées dans le Lobby ce qui permet de faciliter la prise en main. L'utilisation du casque a été jugée confortable et les tutoriels ont été considérés utiles. Cependant, plusieurs utilisateurs ont mentionné une sensation de malaise après une utilisation prolongée. De plus, l'installation et le lancement de l'application ont été jugés difficile pour les personnes qui n'ont pas l'habitude d'utiliser un casque VR. Nous avons pris en compte les remarques des utilisateurs au fur et à mesure des tests afin d'améliorer progressivement l'expérience et corriger certains points d'ergonomie.

8 Résultats et évaluations

Les résultats obtenus à la fin de la conception de ce projet permettent à l'utilisateur d'utiliser toutes les fonctionnalités de l'application et de réellement concevoir les environnements imaginés de manière réaliste. Les objectifs attendus et le cahier des charges ont été remplis. La performance de l'application est également satisfaisante car, malgré une carte graphique qui n'est pas à la pointe de la technologie, notre ordinateur permet d'atteindre un niveau de réalisme très élevé couplé avec une excellente fluidité. Démontrer le résultat de notre travail est difficile avec le rapport comme support, c'est pourquoi nous vous invitons à visionner la vidéo de démonstration du projet afin de pouvoir évaluer la qualité de ce projet.

9 Perspectives d'amélioration

Les perspectives d'amélioration que nous avons identifiés sont :

- Amélioration de la beauté visuelle.
- Développement d'un mode collaboratif.
- Amélioration de l'ergonomie.
- Amélioration de la personnalisation de l'aménagement.
- Régler le bug qui intervient pour certains modèles importés. Lorsque l'on pose l'un de ces objets spécifiques, une plaque apparaît en plus devant celui-ci ce qui n'est pas souhaité.

- Développement d'un mode fantôme pour traverser les objets.
- Pouvoir modifier les textures (couleur, rendu, ...).

10 Conclusion

En conclusion, nous sommes très enthousiastes au fait de pouvoir vous proposer cette application complètement utilisable. La réalisation de ce projet fut un réel plaisir malgré les difficultés rencontrées. Les difficultés majeures étaient évidemment l'apprentissage de l'utilisation d'Unreal Engine et de Blueprint et la longue durée pour implémenter cette application. De plus, certaines fonctionnalités, comme l'import de modèles, étaient assez compliqués à coder et nous a pris beaucoup de temps de réflexion. La seule limitation de cette application est la nécessité d'utiliser un ordinateur avec une configuration correcte pour pouvoir avoir un rendu réaliste. Nous espérons que l'expérience offerte par cette application répond pleinement à vos attentes.

Références

- [1] Ryan Laley. *Unreal Engine 4 Tutorial - Saving & Loading Pt.1 Basics*. YouTube, 9 mai 2019.
<https://www.youtube.com/watch?v=hf98GRZA2d8>
- [2] Ryan Laley. *Unreal Engine 4 Tutorial - Saving & Loading Pt.2 Level States*. YouTube, 23 mai 2019.
<https://www.youtube.com/watch?v=i8enbadjRF4>
- [3] Ryan Laley. *Unreal Engine 4 Tutorial - Saving & Loading Pt.3 Multiple Save Slots*. YouTube, 6 juin 2019.
<https://www.youtube.com/watch?v=hhm7UP7cUyk>
- [4] Seredias. *Unreal Engine Save Game Tutorial*. YouTube, 23 juillet 2023.
<https://www.youtube.com/watch?v=KTMZ18VJzcI>
- [5] CodeLikeMe. *Unreal Engine 5 Snow Particle Tutorial*. YouTube, 7 août 2022.
<https://www.youtube.com/watch?v=sUW3or-YCrk>
- [6] CodeLikeMe. *How to Make a Night Sky and a Moon in Unreal Engine 5*. YouTube, 28 juil. 2022.
<https://www.youtube.com/watch?v=4b33FJ6yjuE>
- [7] Torkuma Shija (Teekday). *Lighting a RAINY Forest in Unreal Engine 5 (Tutorial)*. YouTube, 18 déc. 2023.
<https://www.youtube.com/watch?v=3F7V6SbSshQ>
- [8] CodeLikeMe. *UE5 Open World #15 - Rainy Weather*. YouTube, 14 janv. 2023.
<https://www.youtube.com/watch?v=QDq1xMjNlPM>
- [9] CodeLikeMe. *Unreal Engine Rain Particle with Splashes using Niagara FX System*. YouTube, 1 nov. 2021.
<https://www.youtube.com/watch?v=qmZCW7eQ6rc>
- [10] Gamedev Teacher. *Apprendre Unreal Engine 5 de A à Z (Formation Débutants)*. YouTube, 23 avr. 2023.
<https://www.youtube.com/watch?v=hn98tbztoBg>
- [11] VR Playground. *Create and Interact with UI for Virtual Reality - Unreal Engine Virtual Reality Tutorial*. YouTube, 25 juin 2021.
https://www.youtube.com/watch?v=H5nVjSwM_Uk
- [12] Yaro. *HOW TO MAKE A MULTI-LEVEL WIDGET WITH CUSTOM POINTER IN UE5 VR — TUTORIAL*. Youtube, 9 mars 2024.
<https://www.youtube.com/watch?v=x9fKx7hjq9g>
- [13] Gediminas Kirdeikis. *VR with Unreal Engine 5 - Full Beginner Course*. Youtube, 26 mai 2023.
<https://www.youtube.com/watch?v=wqjJU4V6bGM>
- [14] Gediminas Kirdeikis. *VR with Unreal Engine 5 - Full Intermediate Course*. Youtube, 14 juin 2023.
<https://www.youtube.com/watch?v=u3yk75fUR-U>
- [15] Tyler Serino. *Casting Explained — Unreal Engine 5 Tutorial*. Youtube, 29 mai 2023.
<https://www.youtube.com/watch?v=DBI1sxj5qua>
- [16] Druid Mechanics. *Change Your Mesh's Pivot Point in the Engine! #unrealengine5 #modelingmode #pivot #ue5 #blender*. Youtube, 22 avril 2024.
<https://www.youtube.com/shorts/f50yoTWThnw>

- [17] Spacemarine658. *UE 5.4 : Making Common Action Buttons*. Youtube, 19 août 2024.
<https://www.youtube.com/watch?v=Rye3j11-6CQ>
- [18] SoftStoneGames. *UE5 Tutorial - Add Video to a Widget*. Youtube, 3 juil. 2022.
<https://www.youtube.com/watch?v=UwJ1Ty2crxM>
- [19] Toot Hzf. *Dynamic import mesh into UE5 with Assimp library*. Youtube, 6 mars 2025.
<https://www.youtube.com/watch?v=50nb2axCcA8>
- [20] irajsb. *UE4_Assimp, Assimp Mesh Importer for Unreal Engine*. GitHub https://github.com/irajsb/UE4_Assimp
- [21] Qworco. *Unreal Engine - Grab - Move - Throw and rotate objects tutorial*. YouTube, 19 nov. 2024.
<https://www.youtube.com/watch?v=3H83UJ80Vqg>
- [22] Fcb Dev. *How To Fix Stretched Textures — Unreal Engine 5 Tutorial*. Youtube, 11 janv. 2024.
<https://www.youtube.com/watch?v=0WXdgaHElvQ>
- [23] Unreal Sensei. *Unreal Engine 5 Beginner Tutorial - UE5 Starter Course*. Youtube, 30 juin 2022.
<https://www.youtube.com/watch?v=k-zMkzmduqI&t=3303s>
- [24] Gorka Games. *How to Grab Objects and Move Them Around in Unreal Engine 5*. Youtube, 4 déc. 2022.
<https://www.youtube.com/watch?v=k6oKBVRaLLo&t=137s>
- [25] Gorka Games. *How to Import FBX file in Unreal Engine 5 Easily*. Youtube, 29 oct. 2022.
<https://www.youtube.com/watch?v=FsV3ZUxmWLo>
- [26] Procedural Mind. *Dynamically Scale and Randomize Textures — Unreal Engine 5*. YouTube, 17 sept. 2023.
<https://www.youtube.com/watch?v=WlwfA4wd338>
- [27] ImpalerVR. *How to make Smooth Locomotion movement in VR — Unreal Engine 5*. YouTube, 4 déc. 2023.
<https://www.youtube.com/watch?v=aUBoYVXlMh8>
- [28] DrakeMakes. *UE5 VR SMOOTH LOCOMOTION — Unreal Engine 5.2 Tutorial + Environment Showcase*. Youtube, 6 juin 2023.
<https://www.youtube.com/watch?v=mLF1Jhg7WDY&t=245s>
- [29] Tinkr Academy. *Build User Interfaces In VIRTUAL REALITY For Unreal Engine 5*. YouTube, 23 juil. 2022.
<https://www.youtube.com/watch?v=kM27HYbpvc0&t=545s>
- [30] Werewolven (*Improved*). *Twin-Stick VR Player Movement with Speed Control and Independent Head Tracking in UE 5.1*. Youtube, 3 mars 2023.
<https://www.youtube.com/watch?v=6GKBXl7Jul4&t=82s>