

Premier document

PROJET TUTORÉ – APPLICATION VR EN 4D

KOMODZINSKI – RYSAK – TROHA

S5 RA-IL 1

Dimanche 10 Novembre 2024

Description du Sujet

L'objectif est de développer une application en VR permettant de naviguer dans un espace en quatre dimensions, qui est une réalité que nous ne pouvons percevoir. Le projet comporte deux grandes étapes :

1. La première étape consiste à afficher des objets 4D tels que des hypercubes ou des hypersphères. Ils seront affichés en 3D via un jeu de perspective, afin que nous puissions les visualiser et les manipuler dans un environnement VR. Cela nécessite de comprendre les transformations mathématiques en rapport à la 4D.
2. La deuxième étape consiste à mettre en place une interface permettant de se déplacer dans l'environnement 4D. En s'inspirant du jeu Fez, on pourrait tourner, explorer et interagir avec les objets et l'espace d'une manière unique.

Ainsi, pour mener à bien ce projet, nous avons plusieurs missions à accomplir :

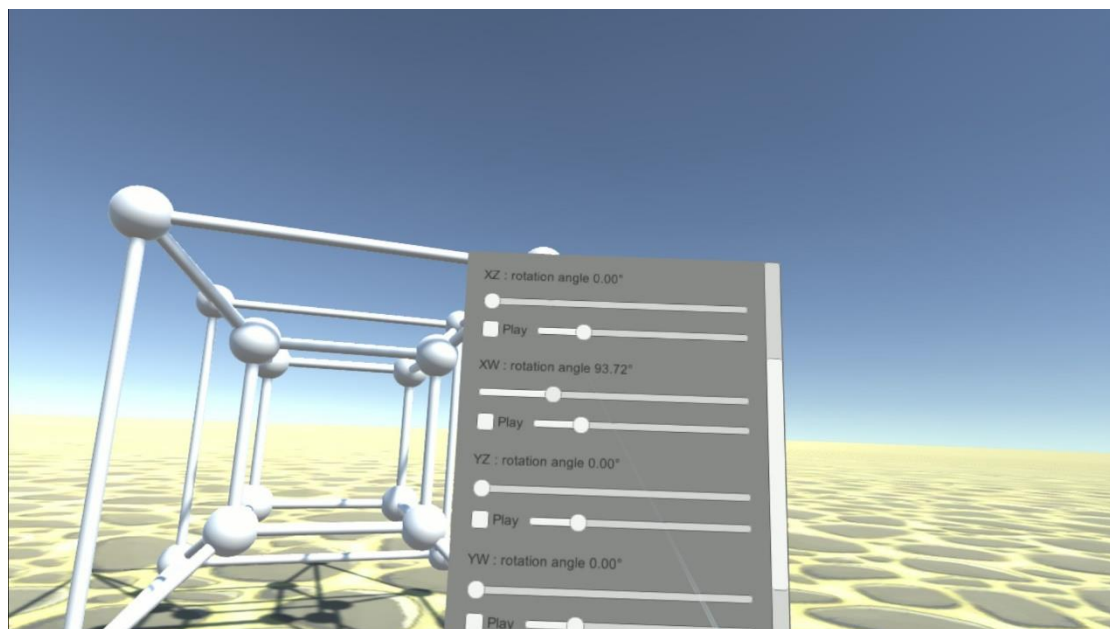
1. Nous devons commencer par étudier la quatrième dimension, ses concepts mathématiques, notamment les projections d'objets 4D dans un sous-espace 3D. Il est essentiel de comprendre comment ces objets réagissent lors d'une transformation sur 1, 2 ou 3 axes pour que le tout reste visuellement compréhensible lors des interactions en VR.
2. Ensuite, nous devons nous familiariser avec le moteur de jeu Godot. Etant donné que nous l'utiliserons tout le long du projet, nous devons maîtriser l'outil afin que nous puissions faire face à n'importe quelle situation.
3. Après ces 2 étapes, nous commencerons par créer une bibliothèque d'objets géométriques 4D (qui possède 4 coordonnées), et nous y appliquerons des transformations pour les projeter en 3D.
4. Puis, nous débuterons la conception d'un système de déplacement inspiré par Fez, que nous adapterons pour manipuler des perspectives 4D.
5. Lorsque la conception sera à un stade avancé, nous implémenterons les transformations mathématiques pour permettre des mouvements fluides entre les perspectives 3D d'un espace 4D.
6. Enfin, nous implémenterons l'application en VR.

Etude de l'existant

1. APPLICATIONS DE VISUALISATION EN 4D EXISTANTES

1.1 Tesseract Vr GO

Tesseract Explorer est un « visualiseur » ligne d'hypercube/tesseract (disponible sur Steam), offrant aux utilisateurs la possibilité de manipuler cet objet 4D à travers des rotations dans divers plans (XW, YW, ZW). Grâce à des projections 4D vers 3D, l'application donne une représentation visuelle simplifiée des objets en quatre dimensions, sans immersion en réalité virtuelle. Ce projet montre comment des transformations bien pensées permettent de rendre la 4D accessible dans un espace plus familier, ou du moins compréhensible pour nous humains, évoluant dans un environnement 3D.



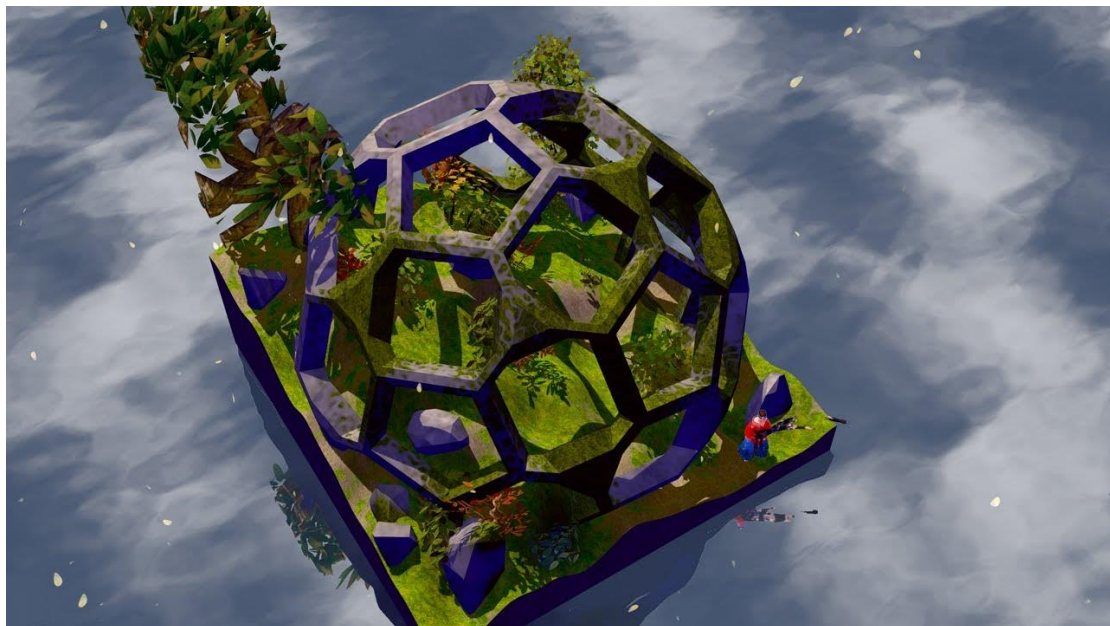
1.2 4D Toys

4D Toys est une application ludique développée par Marc ten Bosch, permettant aux utilisateurs de manipuler et de découvrir des objets en quatre dimensions à travers des interactions directes. Les utilisateurs peuvent observer comment des objets 4D réagissent dans un espace 3D tout en découvrant leurs propriétés géométriques uniques. Bien que cette application ne soit pas en VR, elle met en avant une interface simple et ludique qui rend accessible la compréhension de la quatrième dimension.



1.3 Miegakure

Miegakure est un jeu de plateforme unique qui permet aux joueurs de naviguer dans un espace en quatre dimensions. Dans ce jeu, les joueurs peuvent traverser des obstacles en "changeant de dimension", révélant une 4e dimension invisible dans la perspective 3D. Inspiré par la manière dont "Fez" joue avec des perspectives 2D et 3D, Miegakure utilise un mécanisme de "coupes" entre dimensions qui pourrait inspirer votre projet pour permettre le déplacement dans un espace en 4D en VR. Ce jeu propose une excellente illustration de la manipulation de l'espace multidimensionnel pour résoudre des puzzles, dont on pourrait s'inspirer pour la 2^{ème} partie du projet, lors des déplacements dans la quatrième dimension à la « Fez ».



2.1 COMPARATIF DE L'EXISTANT

Afin de mieux comprendre les enjeux et fonctionnement des diverses applications existantes, nous avons créé un tableau les comparant :

Projet	Technologie utilisée	Objectif principal	Points forts	Limites
Tesseract Explorer	WebGL.	Visualisation de tesseracts/hypercubes.	Interface de manipulation simple et visuelle. Rotation dans différents plans.	Pas de VR implémentée, ni de déplacements.
4D Toys	Moteur de jeu propre.	Expérience « ludique » en 4D.	Interface intuitive, interaction immersive avec objets 4D.	Limité aux manipulations simples. (Pas de VR non plus)
Miegakure	Moteur de jeu propre.	Exploration de plateformes	Navigation entre les dimensions, interface ludique et graphiques intéressants.	Pas de VR implémentée, focalisé sur des puzzles.

Avec ces technologies, bien qu'elles ne soient pas implémentées en VR, peuvent nous aider sur deux grands axes :

- Techniques de projection et d'interaction : Les projets comme Miegakure montrent que des concepts multidimensionnels peuvent être utilisés pour manipuler des perspectives et dimensions. Ainsi, nous pouvons nous baser sur ces techniques pour notre projet.
- Accessibilité et compréhension intuitive : Sur un aspect moins technique, 4D Toys est exemplaire pour sa simplicité et son approche plutôt « intuitive » de la 4D. Nous pourrions également nous baser sur cette approche pour notre projet, afin qu'il représente toute la complexité de la 4D, mais sous une forme compréhensible et interactive pour les novices.

En conclusion, en tirant parti de ces 3 projets, nous pourrions utiliser des pratiques de projection, navigation et manipulation similaires en 4D pour notre projet.

Analyse technique

1 GODOT

Godot est un moteur de jeu qui a la particularité d'être open source et complètement gratuit, utilisé pour créer des jeux 2D et 3D. Il est particulièrement facile d'utilisation avec une interface facile à comprendre, voici le fonctionnement de Godot :

1.1 Structure de Godot

Scènes : Une scène peut représenter un objet simple (comme un personnage ou objet interactif comme un coffre) comme complexe (un niveau, une interface utilisateur). Les scènes peuvent être imbriquées les unes dans les autres, par exemple un niveau peut contenir un personnage ainsi qu'un coffre

Nœuds : Chaque scène est constituée de nœuds. Les nœuds sont la structure principale de Godot. Chaque Nœuds représente un élément spécifique (un sprite, la hitbox d'un objet ou d'un personnage, une lumière ect...). Chaque Nœuds peut être enfant d'un autre ce qui permet d'établir une hiérarchie, un personnage par exemple aura un nœud pour définir le type, un nœuds enfant pour la zone de collision et un autre nœuds enfant pour le sprite de ce personnage

2.2 Langage de programmation :

Godot utilise comme langage de programmation GDscript, qui est un langage semblable à python, facile à apprendre qui s'aligne dans le fait que godot est un moteur de jeu relativement simple à utiliser. Cependant il est possible de coder en C#.

Chaque Nœuds peut avoir un script attaché à lui pour définir son comportement, un personnage va par exemple avoir un script pour définir ces déplacements etc.

2.3 Gestion des dimensions 2D 3d et VR

Godot gère très facilement les jeux en 2D et 3D avec une interface dédiée pour chaque dimension. Godot possède des nœuds aussi spécialement pour la 2D et la 3D.

Godot n'a pas d'interface directement pour la VR comme la 2D et la 3D mais il y a un support VR et AR avec des plugins pour les casques comme le MetaQuest qui sera celui qu'on utilisera au long de ce projet.

2.4 Fonctionnement du système de physique

Godot dispose d'un moteur pour la physique qui gère automatiquement les collisions, les forces et les interactions entre les objets. Voici les 3 éléments de base :

Corps rigides : Objets physiques avec qui il peut y avoir des collisions, qui peut être déplacé par une force extérieure, ils suivent les lois de la physique directement contrôlé par le moteur de jeu mais on peut influencer le comportement de ces objets en changeant par exemple la masse ou autre propriété

Corps statique : Objets utilisés principalement comme obstacle, il y a des collisions mais ils ne peuvent être bougés par une quelconque force extérieure

Corps Kynématique : ce sont les corps directement contrôlés par le script par exemple avec un personnage

Pour conclure, Godot est donc un moteur de jeu polyvalent, facile à utiliser mais néanmoins puissant et complet qui s'adapte bien aux projets que nous cherchons à réaliser. Le seul bémol est que Godot ne possède pas directement d'objet en 4 dimensions et ce sera notre rôle de les créer.

2. CASQUE META QUEST 3

Le casque Meta Quest 3 est un casque de réalité virtuelle de Meta, conçu spécialement pour offrir des expériences immersives et complètes aux utilisateurs.

2.1 Caractéristique technique du Casque Meta Quest 3

Le casque Meta Quest 3 possède plusieurs caractéristiques qui font de lui un casque de choix pour tout projet lié à la réalité virtuelle

Affichage Haute résolution et clarté visuelle : Le casque propose une clarté et une fidélité des couleurs améliorées comparées à son prédécesseur le casque Meta Quest 2. Ce niveau de résolution est important pour visualiser des objets détaillés comme des projections 4D dans un espace 3D

Processeur avancé : Le casque utilise le processeur Snapdragon XR2 Gen2, qui est en fait un processeur spécialement conçu pour le VR et l'AR et qui possède des capacités graphiques et de calcul améliorées, optimisées pour des environnements complexes et dynamiques. Donc ce processeur permet de rendre les interactions et les animations avec des modèles géométriques complexes beaucoup plus fluides

Capteurs et Suivi de l'Espace : Le casque est équipé de capteurs de suivi de l'intérieur vers l'extérieur, cela veut dire qu'il n'a pas besoin de capteurs externes pour cartographier l'espace. Il détecte en temps réel la position dans la pièce et les mouvements effectués dans celle-ci. Cette fonction est donc très importante pour permettre des déplacements précis dans un espace simulé.

2.2 Programmation et Développement avec Godot

Pour développer une application VR sur le Meta Quest 3, Godot est bon choix car :

Plugins de VR : Godot offre des plugins VR compatibles avec le Quest, ce qui permet l'intégration directe de plusieurs fonctionnalités comme le suivi de mouvement et les interactions avec des objets.

Moteur Graphique Performant : Avec Godot, il est possible de créer des environnements et objets 3D optimisés pour la VR, ce qui permet donc de ne pas surcharger les performances du casque.

Gestion des interactions VR : Godot gère les interactions en VR avec des outils intégrés, ce qui simplifie la création d'une interface pour des déplacements dans l'espace virtuel.

Pour conclure, le Meta Quest 3 avec ses fonctionnalités avancées représente un outil puissant, particulièrement adapté à notre projet pour ce qui est de manipuler des formes géométriques. Couplé avec Godot, nous possédons une plateforme de développement plus que conséquente pour réaliser des simulations des déplacements d'objets 4D dans un espace 3D

3 MODELISATION 4D ET VISUALISATION 3D

La modélisation de la 4d et sa visualisation dans un espace 3d représente un des enjeux majeurs de ce projet, car cela implique entre autres de représenter des objets qui se trouvent au-delà de nos perceptions habituelles et qu'on ne peut habituellement pas représenter entièrement dans un espace en 3 dimensions. La 4d n'est pas directement visible dans un espace à 3 dimensions cependant, il existe plusieurs méthodes pour essayer de se représenter ces objets dans un espace à 3 dimensions.

3.1 Projection de la 4d en 3d

Pour visualiser des objets 4d comme des hypercube ou des hypersphères en 3d il faut procéder par ce qu'on appelle la projection, il existe plusieurs méthodes de projection qui peuvent être envisagées :

- Projection par Perspective : Dans cette approche, les objets 4D sont projetés vers un espace 3D cela marche de manière similaire lorsque nous visualisons des objets 3D dans un espace 2D par exemple un cube que l'on dessine sur une feuille n'est pas réellement 3D mais la perspective nous donne l'impression de voir un cube. Il est possible comme cela de représenter des objets 4d dans la 3ème dimension, même s'il sera toujours difficile pour nous de les appréhender

- Projection Stéréographique : La projection stéréographique est une technique souvent utilisée pour projeter une sphère en 3D sur un espace 2D mais elle peut aussi s'appliquer pour représenter des objets 4D dans un espace 3D. La propriété de cette projection est qu'elle préserve les angles mais elle ne préserve pas les distances
- Découpage par Plans 3D : C'est une autre approche pour se représenter les objets en 4d, on "tranche l'objet" en multitude d'objet 3D et on les affiche les uns à la suite des autres dans le plan 3D. Pour prendre un exemple de la 3D à la 2D, c'est comme couper un cube, en multitude de carré que l'on ferait passer les uns à la suite des autres dans un plan 2D

4. DEPLACEMENT

Pour pouvoir déplacer des objets en 4 dimensions, il ne suffit pas de les représenter, il faut aussi faire des rotations et des translations :

Rotation en 4D

La rotation en 4D est beaucoup plus complexe que la rotation en 3D, car elle introduit des axes de rotation supplémentaires. En 3D, on peut tourner un objet autour de trois axes principaux :

- Autour de l'axe X (comme tourner la tête pour dire "non"),
- Autour de l'axe Y (comme incliner la tête d'un côté),
- Autour de l'axe Z (comme hocher la tête pour dire "oui").

Mais en 4D, il y a six axes de rotation possibles, car chaque rotation peut se faire entre deux des quatre axes (par exemple, X-W, Y-W, etc.). Donc on peut faire des rotations entre des « paires de dimensions » :

1. X-Y, X-Z, Y-Z : les rotations que nous connaissons en 3D.
2. X-W, Y-W, Z-W : les rotations spécifiques à la 4D.

Comment cela fonctionne-t-il ?

- Rotation 3D (par exemple, X-Y) : Imaginez tourner un objet dans le plan X-Y en le faisant pivoter autour de l'axe Z. L'objet change d'orientation, mais il reste dans notre espace 3D.

- Rotation 4D (par exemple, X-W) : En 4D, une rotation X-W signifie que l'objet tourne dans un plan qui mêle l'axe X de notre 3D et l'axe W de la 4D. C'est une sorte de « basculement » vers la quatrième dimension. C'est assez difficile de se le représenter.

Maintenant, parlons de la structure d'une matrice de rotation en 4D.

En 3D, une rotation est décrite par une matrice de 3x3. En 4D, on utilise une matrice de 4x4, car on doit inclure le nouvel axe W. Une matrice de rotation en 4D permet donc de transformer les coordonnées (X, Y, Z, W) d'un point dans cet espace.

Chaque rotation implique uniquement une paire d'axes vu plus haut et laisse le reste inchangé. Par exemple, une rotation dans le plan X-W n'affectera que les axes X et W, et laisse les axes Y et Z constants.

Matrice de rotation

Prenons l'exemple de la rotation dans le plan X-W avec un angle θ . La matrice de rotation dans le plan X-W serait la suivante :

$$R_{XW}(\theta) = \begin{bmatrix} \cos \theta & 0 & 0 & -\sin \theta \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sin \theta & 0 & 0 & \cos \theta \end{bmatrix}$$

Donc, X et W sont transformés selon les cosinus et sinus de θ , tandis que Y et Z restent inchangés. On suit la même logique pour chaque paire d'axes en modifiant les termes cos et sin pour les deux axes impliqués et en laissant les autres fixes.

Autre point intéressant, les rotations composées. Il est possible de combiner plusieurs rotations en différents plans, par exemple X-Y et Z-W, en multipliant les matrices correspondantes. Et contrairement aux rotations en 3D, qui finissent par interférer entre elles, en 4D, les rotations dans des plans indépendants peuvent coexister sans conflit.

En conclusion, les rotations en 4D sont fascinantes car elles peuvent permettre à des objets à changer de forme et de position d'une manière qui semble impossible en 3D. Par exemple, un objet 4D peut sembler "inversé" dans sa projection 3D après une rotation 4D, similaire à ce qu'on verrait avec un miroir.

Ces concepts sont difficiles à visualiser, mais en VR, on peut les représenter par des effets de distorsion et de changement de perspective.

Translation

En 4D la translation est similaire à celle que dans les espaces de dimension inférieure, elle ajoute cependant un axe de mouvement supplémentaire, typiquement dans un espace 3D une translation consiste à déplacer un objet selon les coordonnées x,y,z en 4D on va ajouter la quatrième coordonnées w.

La formule de translation sera donc :

$$(x',y',z',w')=(x+dx,y+dy,z+dz,w+dw)$$

Traduction en 3D

Lorsque l'on va déplacer un objet 4D selon l'axe w cela equivaut à translater ses points dans la dimension qui n'est pas visible en 3D. Cependant cela va tout de même modifier la projection 3D de l'objet ce qui va donner à l'objet l'impression de se transformer ou même changer de forme. A noter que lorsqu'on déplace un objet 3D sur l'axe w, il va tout simplement disparaître de l'espace 3D, comme quand on déplace un carré vers l'avant, celui-ci disparaît du plan 2D.

5. INTERFACE DE NAVIGATION INSPIREE DE « FEZ »

Dans Fez, l'univers est un univers 3D mais représenté comme une série de vue 2D qui peuvent être "pivoté" pour changer de perspective et par exemple accéder à des endroits qui jusque-là n'était pas accessible.

En résumé dans Fez l'univers est en 3D mais le joueur ne voit qu'une projection 2D. Il peut cependant faire tourner le monde à 90 degrés pour changer la perspective et montrer une autre vue 2D de l'univers 3D

Extension de l'interface FEZ de la 3D vers la 4D

Pour appliquer une logique similaire de la 4D vers la 3D il nous faudrait adapter les concepts suivants :

- Projection en 3D des Objets 4D : En utilisant les techniques de projection de la 4D vers la 3D, cité plus haut, il nous faudrait représenter les objets 4D dans un espace 3D
- Rotation dans l'Espace 4D pour Changer de Perspective 3D : Plutôt que de tourner en quart de tour dans l'espace 3D, il serait possible de faire pivoter l'espace 4D autour de deux plans de rotation en utilisant les méthodes de rotation citées plus haut. A l'instar de Fez, cette rotation nous permettrait d'afficher des vue 3D différentes de l'espace 4D.
- Exploration de Vues Différentes avec une Interface de Sélection de Plan : Notre interface pourrait permettre à l'utilisateur de visualiser différentes projections

ou combinaisons de dimensions 4D. Par exemple, l'utilisateur pourrait voir les plans xyz ou xyw, cela permettrait de révéler des aspects cachés de l'espace 4D, un peu comme la façon dont on pivote l'univers pour révéler de nouvelles parties de l'univers.