**Sommaire**

1. Contexte et Motivation
   * Applications en informatique graphique, modélisation et robotique
2. Objectifs du Travail
3. Partie 1 : Rotation en 2D
   * Chapitre 1 : Fondements Mathématiques de la Rotation en 2D
     + Notions de Base
     + Matrice de Rotation en 2D
     + Coordonnées Homogènes en 2D
   * Chapitre 2 : Implémentation et Visualisation en 2D
     + Méthodes Numériques
     + Programmation et Simulation
     + Applications et Exemples
4. Partie 2 : Rotation en 3D
   * Chapitre 3 : Fondements Mathématiques de la Rotation en 3D
     + Matrice de Rotation en 3D
     + Quaternions et Rotation
     + Démonstration de toutes choses avancées
   * Chapitre 4 : Implémentation et Visualisation en 3D
     + Programmation de la Rotation 3D
5. Conclusion et Perspectives
   * Résumé des Résultats Obtenus
   * Perspectives et Travaux Futurs

**Contexte et Motivation**

Ce qui m’a motivé à choisir ce thème était mon affinité avec les mathématiques. Mais je ne voulais pas faire un travail de maturité à 100 % en mathématiques, je cherchais une partie pratique dans laquelle je pourrais appliquer des math. Durant la période à laquelle je cherchais un TM, j’ai reçu une mauvaise note en math appliqué sur des matrices, par curiosité et désire de vengeances, j’ai choisi de m’orienter de ce côté-là. Par chance, les matrices cochaient parfaitement mes désirs de mixité et il semblait très intéressent à étudier les applications de celle-ci dans la rotation d’objet. De plus, ce sujet m’offrait une partie de codage challengeant pour moi qui ne suis pas très doué en informatique.

**Objectifs du Travail**

L’objectif de mon travail est de comprendre et d’expliquer les rotations de forme géométrique que ça soit dans l’espace ou sur le plan. Ma démarche sera de guider le lecteur à travers des explications des sujets, de lui démontrer par voie calculatoire les sujets qui lui seront expliqués et pour finir, le convaincre par des simulations informatiques qui seront bien sûr aussi expliquées. Mon objectif ne sera pas de vulgariser ou d’écrire un texte trop léger, je souhaite être le plus rigoureux possible.

Durant la lecture de cette feuille, il est conseillé que le concept de matrice vous soit familier

* Étudier et implémenter la rotation en 2D et en 3D
* Explorer les méthodes mathématiques et algorithmiques associées

**Partie 1 : Rotation en 2D**

**Chapitre 1 : Fondements Mathématiques de la Rotation en** **2D**

**Notions de Base**

Avant de commencer dans le développement le sujet de la rotation en 2 dimensions il est important de savoir ce qu’est un est un espace vectoriel et ce qu’est un vecteur.

**Définition d’un espace vectoriel et vecteur :**

En algèbre linéaire un espace vectoriel est un ensemble d’objets, appelées des vecteurs. Un vecteur est un élément d’un espace vectoriel. Il est représenté comme un segment en forme de flèche ayant un point de départ et un point d’arrivé. Sa direction et sa taille sont définis à l’aide d’une matrice de dimension 1×2 () que l’on appelle ses composantes, a et b représente le nombre d’unités nécessaire pour aller d’un point de départ à un point d’arrivé en se déplacent de parallèlement à l’axe des abscisses puis celle des ordonnées.

Les vecteurs peuvent être additionner entre eux pour créer de nouveaux vecteurs et peuvent être multiplier par un scalaire (généralement un nombre réel) pour s’allonger, se rétrécir ou changer de direction. L’addition des deux propriétés dites s’appelle une combinaison linéaire

On les représente comme ça dans un plan :

Une image contenant ligne, Parallèle, diagramme, Tracé

Le contenu généré par l’IA peut être incorrect.

Ici nous avons fait deux combinaisons linéaires pour représenter les vecteur à l’aide des vecteurs et le scalaire 1 et -2

**Définition du système de coordonnées cartésien et polaire :**

Un plan est une surface plate illimité, munie de notion de d’alignement, d’angle et de distance. Il y a un point nommé origine au centre du plan et deux axes passant par ce point (axe : X et Y) parallèlement par rapport a eu. On peut y définir des points de façon cartésienne en notent ses coordonnées par rapport à l’axe X et Y. Ou encore de façon polaire ou le point va être défini par la distance qui le sépare de l’origine et l’angle que forme le segment entre l’origine et le point par rapport à l’axe X.

On le représente :

Une image contenant diagramme, ligne, cercle, Tracé

Le contenu généré par l’IA peut être incorrect.

Le point B est donné comme avec des coordonnées cartésiennes.

Le point B est donné comme (distance ; angle)

* Espaces vectoriels et matrice de transformations
* Systèmes de coordonnées cartésien et polaire

**Matrice de Rotation en 2D**

* Définition et propriétés
* Transformation d’un point dans le plan
* Invariants de la rotation

**Coordonnées Homogènes en 2D**

* Introduction aux matrices 3×3
* Avantages pour les transformations combinées

**Chapitre 2 : Implémentation et Visualisation en 2D**

**1. Méthodes Numériques**

* Rotation par matrice de transformation et de rotation

**2. Programmation et Simulation**

* Implémentation en Python avec Matplotlib

**3. Applications et Exemples**

* Rotation d’objets géométriques (polygones, cercles)

**Partie 2 : Rotation en 3D**

**Chapitre 3 : Fondements Mathématiques de la Rotation en 3D**

**1. Matrice de Rotation en 3D**

* Rotation autour des axes X, Y, Z
* Composition de rotations (angles d'Euler) et effet de Blocage de quadrant
* Partie calculatoire

**2. Quaternions et Rotation**

* Définition et propriétés
* Multiplication et interpolation de quaternions
* Partie calculatoire

**3. Démonstration de toutes choses avancées**

**Chapitre 4 : Implémentation et Visualisation en 3D**

**1. Programmation de la Rotation 3D**

* Implémentation avec matrices 4×4 (translation)
* Gestion des quaternions en Python
* Animation d’objets en 3D

**Conclusion et Perspectives**

**1. Résumé des Résultats Obtenus**

* Comparaison des méthodes utilisées en 2D et 3D
* Avantages et inconvénients des approches étudiées

**2. Perspectives et Travaux Futurs**

* Applications avancées (cinématique inverse, rendu photoréaliste)