# 1 Introduction

## 1.1 Description du projet

Cette application sert à afficher et animer des rotations de points dans l’espace autour d’un axe en utilisant deux méthodes différentes, les matrices de rotation en utilisant les angles d’Euler et les Quaternions (généralisation des nombres complexes).

## 1.2 Objectif

Avoir une interface graphique qui affiche en direct les mouvements à effectuer pour faire une rotation d’un objet dans l’espace ainsi que les résultats sous formes de matrices de rotation et de calculs matriciels.

## 1.3 Planning

# 2 Spécification des besoins

## 2.1 Besoins Utilisateurs

### 2.1.1 List des acteurs

* L’Utilisateur

### 2.1.2 Cas d’utilisation

|  |  |
| --- | --- |
| * UC1 : Ajouter un objet * UC2 : Ajouter un axe de rotation * UC3 : Ajouter une rotation * UC4 : Sauvegarder * UC5 : Ouvrir * UC6 : Gestion de la vue | C:\Users\karim.luy\He-ARC\INF2dlm-b\Projet P2 - Java\TurboSpin\Documentation\Spec\UseCase.png |

## 2.2 Scenarios

### 2.2.1 UC1 : ajouter un objet

#### 2.2.1.1 Description textuelle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | UC1 | **Nom** | Ajouter un objet | | |
| **Date** | 13 avril 2015 | **Auteur** |  | **Priorité** | High |

|  |  |
| --- | --- |
| **Description et objectifs** | Permettre à l’utilisateur de rajouter un objet dans la vue 3D. |
| **Acteurs** | * Utilisateur |
| **Déclencheurs** | * Bouton « Add » du menu de gestion d’objets |
| **Scénario nominal** | 1. L’utilisateur est sur l’interface de base 2. Il clique sur « Add » pour rajouter un objet |
| **Scénarios alternatifs** |  |
| **Préconditions** | * L’application doit être exécutée * Les paramètres (position, longueur, etc.) de l’objet doivent être entré |
| **Post-Conditions** | * L’objet apparait dans la vue 3D |

#### 2.2.1.2 Diagramme de séquence système

### C:\Users\karim.luy\He-ARC\INF2dlm-b\Projet P2 - Java\TurboSpin\Documentation\Spec\DSS_AO.png

### 2.2.2 UC2 : Ajouter un axe de rotation

#### 2.2.2.1 Description textuelle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | UC2 | **Nom** | Ajouter un axe de rotation | | |
| **Date** | 13 avril 2015 | **Auteur** |  | **Priorité** | High |

|  |  |
| --- | --- |
| **Description et objectifs** | Permettre à l’utilisateur de rajouter un axe de rotation sur la vue 3D. |
| **Acteurs** | * Utilisateur |
| **Déclencheurs** | * Bouton « Add » du menu de gestion des axes. |
| **Scénario nominal** | 1. L’utilisateur est sur l’interface de base. 2. Il clique sur « Add » pour rajouter un axe. |
| **Scénarios alternatifs** | * L’utilisateur clique sur « Draw » du menu de gestion des axes puis fais un « click and drag » sur la vue 3D |
| **Préconditions** | * L’application doit être exécutée * Les paramètres (points, couleur, etc.) de l’axe doivent être entrés |
| **Post-Conditions** | * L’axe apparait dans la vue 3D |

#### 2.2.2.2 Diagramme de séquence système

### C:\Users\karim.luy\He-ARC\INF2dlm-b\Projet P2 - Java\TurboSpin\Documentation\Spec\DSS_AA.png

### 2.2.3 UC3 : Ajouter une rotation

#### 2.2.3.1 Description textuelle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | UC3 | **Nom** | Ajouter une rotation | | |
| **Date** | 13 avril 2015 | **Auteur** |  | **Priorité** | High |

|  |  |
| --- | --- |
| **Description et objectifs** | Permettre à l’utilisateur d’insérer une rotation. |
| **Acteurs** | * Utilisateur |
| **Déclencheurs** | * Bouton « new rotation » du menu de gestion des rotations |
| **Scénario nominal** | 1. L’utilisateur est sur l’interface de base 2. Il clique sur « new rotation» pour rajouter un objet |
| **Scénarios alternatifs** | * Il clique sur « new rotation » puis sélectionne l’objet à tourner, l’axe sur le quel tourner, et introduit l’angle de rotation * Il sélectionne l’objet et l’axe depuis les menus de gestion puis insère une valeur en dégrées |
| **Préconditions** | * L’application doit être exécutée * Un objet doit exister * Un axe doit exister (axe par default (1,1,1)) * Une quantité de rotation doit être fournit |
| **Post-Conditions** | * La rotation s’anime sur l’interface * Le calcul matriciel s’affiche dans la zone de calculs * Le quaternion s’affiche dans la zone de calculs |

#### 2.2.3.2 Diagramme de séquence système

### C:\Users\karim.luy\He-ARC\INF2dlm-b\Projet P2 - Java\TurboSpin\Documentation\Spec\DSS_AR.png

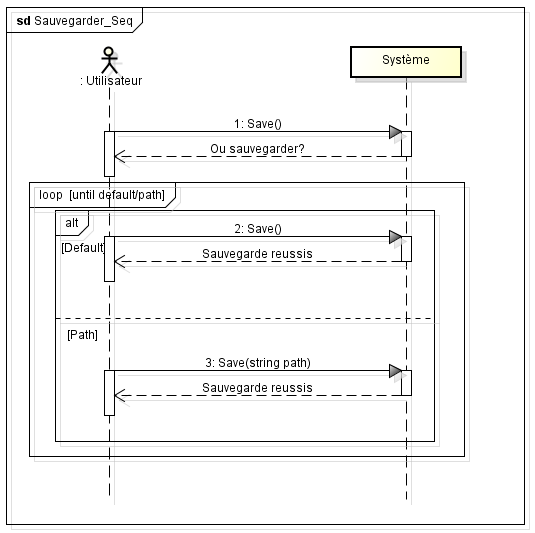
### 2.2.4 UC4 : Sauvegarder

#### 2.2.4.1 Description textuelle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | UC4 | **Nom** | Sauvegarder | | |
| **Date** | 13 avril 2015 | **Auteur** |  | **Priorité** | High |

|  |  |
| --- | --- |
| **Description et objectifs** | Permettre à l’utilisateur de sauvegarder la vue 3D et les objets/axes dans un fichier. |
| **Acteurs** | * Utilisateur |
| **Déclencheurs** | * Bouton « Save » du menu « Edit » de la barre de menus |
| **Scénario nominal** | 1. L’utilisateur est sur l’interface de base 2. Il clique sur « Save » pour sauvegarder l’état du programme |
| **Scénarios alternatifs** | * L’utilisateur clique sur « Save as » dans le menu « Edit » * Quand l’utilisateur ferme un fichier ou le programme il sera proposer de sauvegarder avant de quitter. |
| **Préconditions** | * L’application doit être exécutée * Il doit y avoir eu des modifications depuis l’état de base du programme |
| **Post-Conditions** | * Un fichier sera créé là où il décide de le sauvegarder ou dans un dossier par default qui se situe dans le dossier d’installation du programme contenant l’état du programme. |

#### 2.2.4.2 Diagramme de séquence système



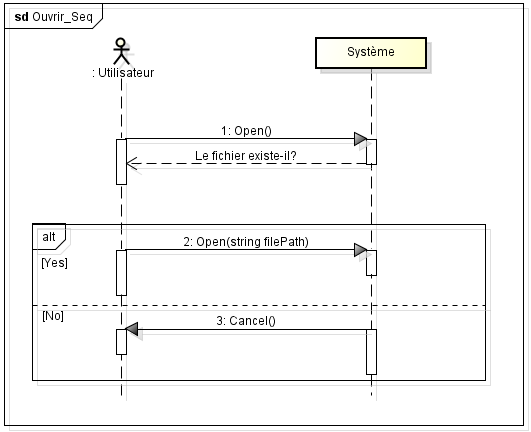
### 2.2.5 UC5 : Ouvrir

#### 2.2.5.1 Description textuelle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | UC5 | **Nom** | Ouvrir | | |
| **Date** | 13 avril 2015 | **Auteur** |  | **Priorité** | High |

|  |  |
| --- | --- |
| **Description et objectifs** | Permettre à l’utilisateur d’ouvrir un fichier contenant déjà des objets/axes. |
| **Acteurs** | * Utilisateur |
| **Déclencheurs** | * Bouton « Open » du menu « Edit » de la barre de menus |
| **Scénario nominal** | 1. L’utilisateur est sur l’interface de base 2. Il clique sur « Open » depuis le menu « Edit » |
| **Scénarios alternatifs** | * Au démarrage de l’application l’utilisateur sera suggérer l’ouverture d’un fichier |
| **Préconditions** | * L’application doit être exécutée * Il ne doit pas y avoir de fichier déjà ouvert |
| **Post-Conditions** | * Le programme charge les objets et les affiche sur la vue 3D |

#### 2.2.5.2 Diagramme de séquence système

****

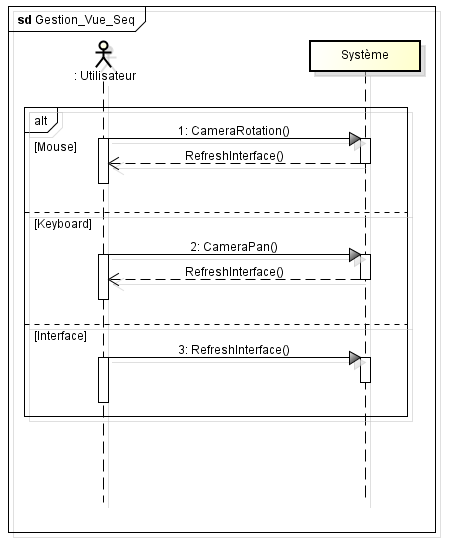
### 2.2.6 UC6 : Gestion de la vue

#### 2.2.6.1 Description textuelle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | UC6 | **Nom** | Gestion de la vue | | |
| **Date** | 13 avril 2015 | **Auteur** |  | **Priorité** | High |

|  |  |
| --- | --- |
| **Description et objectifs** | Permettre à l’utilisateur de changer son point de vue sur la vue 3D. |
| **Acteurs** | * Utilisateur |
| **Déclencheurs** | * Clique gauche * Touches directionnel * Touche de reset * Boutton « Reset » du menu « View » |
| **Scénario nominal** | 1. L’utilisateur est sur l’interface de base 2. Il manipule la vue 3D avec une des méthodes cité si dessus |
| **Scénarios alternatifs** |  |
| **Préconditions** | * L’application doit être exécutée * Une vue 3D doit être ouverte |
| **Post-Conditions** | * Rafraichissement de la vue 3D en fonction de l’action effectué |

#### 2.2.6.2 Diagramme de séquence système

****

## 2.3 Exigences fonctionnelles et non-fonctionnelle

|  |  |  |
| --- | --- | --- |
| Requirement ID | Type | Description |
| FR1 | Fonctionnelle | Contrôle caméra. |
| FR2 | Fonctionnelle | Visualisation d’une rotation en 3D avec ses composantes affichables ou non affichables |
| NFR1 | Performance | Le *framerate* doit être stable, autour de ~30 ips. |

# 3 Spécification Technique

## 3.1 Analyse du domaine métier

*Les classe entités et leurs descriptions*

Matrix et Quaternion : ces classes sont les outils mathématiques permettant d'effectuer des rotations.

QuaternionTools : cette classe comporte des méthodes utilitaires permettant de générer et d'utiliser simplement les Quaternions de rotation.

RotationMatrix : cette classe comporte des méthodes utilitaires permettant de générer et d'utiliser simplement les Quaternions de rotation.

Vector3D : Cette classe représente un vecteur dans ℝ3.

Point3D : Cette classe représente un point dans ℝ3.

Line3D : Cette classe représente une ligne dans ℝ3.

Main3DWindow : Cette classe contient l'affichage tridimensionnel de l'espace où se trouve les objets 3D.

Secondary2DWindow : Cette classe contient un affichage bidimensionnel d'une coupe de l'espace tridimensionnel.

MainInterface : Cette classe contient les affichages de l'espace tridimensionnel et des trois coupes bidimensionnels. Elle contient également les listes d'objets.

## 3.2 Diagramme de classe complet

