1. Nous avons pu finir l’entièreté du projet en incluant toutes les extensions (excepté les points de reprise).

2. Nous avons une version graphique faite sur Qt (15.14.2)

3. Nous avons passé en moyenne 10 heures par semaine (par personne) à travailler sur ce projet.

Le fichier *details conception* vous donnera plus d’information sur la conception de chaque classes.

**Ouverture du projet :**

Pour entrer dans le projet, lancez le fichier “projet.pro”, se situant dans le dossier projet, à l’aide de la bibliothèque Qt. À partir de là vous aurez accès à plusieurs sous-projets de tests et un d’affichage graphique nommé “Qt\_GL”.

**Lancement d’un sous-projet:**

- Build le projet

- placez-vous dans le dossier du sous projet désiré

- Run le sous-projet

**testVecteur:** Ce sous projet permet à l’utilisateur d’appréhender toutes opérations et autres méthodes que lui offre notre classe Vecteur tout en montrant la gestion d’erreurs lié aux Vecteurs.

**testMatrice:** Ce sous projet permet à l’utilisateur d’appréhender toutes opérations et autres méthodes que lui offre notre classe Matrice.

**testToupie:** Ce sous projet donne à l’utilisateur toutes les caractéristiques physiques importantes d’une toupies et montre les fonctionnement de nos différentes toupies. De plus, il montre aussi à l’utilisateur la gestion de nos erreurs.

**testIntegrateur:** Ce sous projet permet à l’utilisateur de tester tous nos intégrateur (et leur divergence) en fonction d’une toupie ConeSimple et ToupieChinoise.

**testSysteme:** Ce sous projet permet l’utilisateur d’appréhender les systèmes de toupies mit en place ainsi que leurs méthodes offerte et ce qu’elles retournent.

**Exercice\_P9 :** Ce sous est projet externe de Qt. Il vous faut donc retourner dans le dossier projet et ouvrir le dossier Exercice\_P9 dans votre projet. Lancez la commande make et ensuite la commande ‘’./Exercice\_P9’’. Cela vous permettra d’ouvrir un équivalent de testSysteme mais avec notre fichier makefile et non celui de Qt.

**Qt\_GL:**

I-But : Ce sous projet permet tout l’affichage graphique de nos Toupies. En effet, grâce au GLWidget il est possible d’afficher nos toupies et leur trace et interagir avec notre point de vue dans cette fenêtre. Il est aussi possible d’afficher plusieurs toupies à la fois.

II-Commandes Importantes:

- flèches et Q, E : vous permettra de faire une rotation de votre point de vue - WASD: Déplacement de votre point de vue

- I : reinitialise la position

- R et F: déplacement vers le haut (resp. vers le bas)

- V: enclenchera la vue tangentielle

- espace : met en pause la simulation

- il est possible de se déplacer avec la souris en cliquant sur la fenêtre. Cependant attention cela demande plus de mémoire et peut donc générer du lag et des problèmes de modélisation. Utilisez cela que si vous avez une machine supportant facilement ce genre de condition.

III-Changement de toupies et recommandations:

Pour changer de toupie rentrez dans le sous projet Qt\_GL, puis Sources et enfin ouvrez le document main\_qt\_gl. À partir de la vous trouverez à la fin du document des blocs de Toupies ajoutées (commentées ou non) il vous suffira donc de suivre les étapes marqué ci-contre pour ajouter ou supprimer des blocs de Toupies.

Il vous sera aussi possible d’ajouter des fonctions manuellement grâce à la fonction ajouter\_toupie() .Cependant privilégie l’affichage d’une Toupie à la fois pour obtenir un meilleur rendu.

**testInvariants:**

I-But: Ce sous projet à comme utilité principale de pouvoir facilement montrer les invariants du mouvement et la stabilité de nos intégrateurs. En effet, grâce à une widget nous affichons les invariants choisit par l’utilisateurs (à l’aide de buttons dédiés) en fonction de nos trois intégrateurs.

II-Explication buttons : - scale: étends les graphs au maximum

- clear: nettoie les graphs

- autres: permet de choisir les invariants du mouvement de notre toupie

**testComparaison:**

I-But: Ce sous-projet nous permet de comparer de manière graphique les paramètres et dérivées des paramètres d’un ConeSimple à un solide de révolution approximant ce même ConeSimple.

II-Explication buttons :

- psi, thêta, phi : donne les coordonnées psi, thêta, phi du Vecteur des paramètres de la toupie

- psi\_point, theta\_point, phi\_point : donne les coordonnées psi\_point, theta\_point, phi\_point du Vecteur dérivée des paramètres de la toupie

- clear: nettoie les graphs