

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 classe.

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 sous-projet 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 tout en montrant la gestion d'erreurs lié aux Matrices.

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 lies aux Toupies.

testIntegrateur : Ce sous-projet permet à l'utilisateur de tester tous nos Intégrateurs (et leur performance) en fonction d'un ConeSimple et aussi de vérifier nos invariants du mouvement textuellement.

testSysteme : Ce sous-projet permet à l'utilisateur d'appréhender les systèmes de toupies ainsi que de simuler textuellement une comparaison entre un Cone Simple et un Solide de Revolution représentant notre cadre d'une toupie générale.

Exercice_P9 : Ce sous-projet est externe à 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 :

1-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. On offre à l'utilisateur 8 simulations différentes qu'il pourrait sélectionner dans le main (main_qt_gl.cc).

2-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 : réinitialise 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 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.

3- 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.cc. À partir de là vous trouverez à la fin du document des blocs de Toupies ajoutées (masquées par commentaire ou non). Pour exécuter la simulation voulu, démasquer l'ajout des Toupies utiliser dans la simulation et masquer les autres simulations. Ceci est précise en plus de détails dans le main (main_qt_gl.cc).

Il vous sera aussi possible d'ajouter des Toupies manuellement grâce à la méthode ajouter_toupie du widget. Cependant privilégiez l'affichage d'une Toupie à la fois pour obtenir une meilleure performance graphique.

testInvariants :

1-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 un widget nous affichons les invariants choisis par l'utilisateur (à l'aide de boutons dédiés) en fonction de nos trois Intégrateurs.

2-Explication des boutons :

- scale : étends les graphs au maximum
- clear : nettoie les graphs
- autres : permet de choisir les invariants du mouvement de notre Toupie

testComparaison :

1-But : Ce sous-projet nous permet de comparer de manière graphique les paramètres et dérivées des paramètres d'un ConeSimple à un Solide de Revolution approximant ce même ConeSimple (dans le cadre d'une toupie générale).

2-Explication des boutons :

- psi, theta, phi : donne les coordonnées psi, theta, 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 de dérivée des paramètres de la Toupie
- clear: nettoie les graphes