LES TROPHÉES NSI

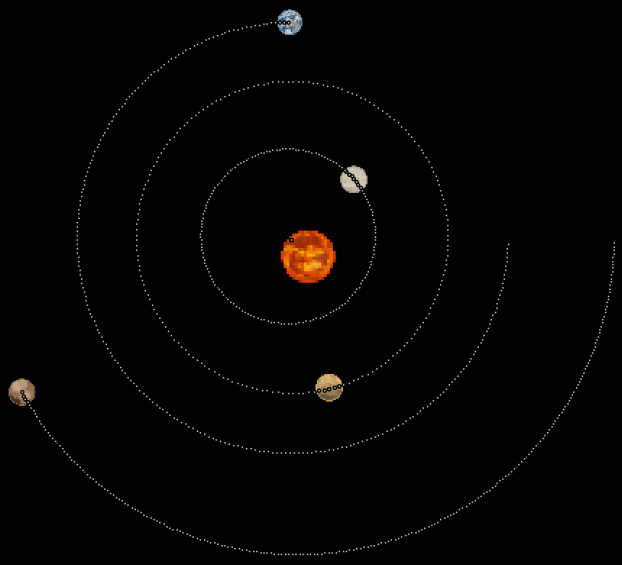
Édition 2022

DOSSIER DE CANDIDATURE PRÉSENTATION DU

Trophées NSI

Simulateur Orbital

Arthur, Léa et Noé



Ce document est l'un des livrables à fournir lors du dépôt de votre projet : 4 pages maximum (hors documentation).

Pour accéder à la liste complète des éléments à fournir, consultez la page **Préparer votre participation**.

Vous avez des questions sur le concours ? Vous souhaitez des informations complémentaires pour déposer un projet ? Contactez-nous à **info@trophees-nsi.fr**.

NOM DU PROJET : Simulateur Orbital

> PRÉSENTATION GÉNÉRALE :

- *Idée et objectifs*
- *Origines et intérêts du projet*
- *(…)*

En créant ce projet, nous souhaitions créer un moyen de déterminer par la simulation l'évolution de système planétaire, en particulier ceux à trois corps pour observer comment ceux-ci finissent souvent par se simplifier en système à deux corps en éjectant le troisième. Le principal objectif que nous-nous sommes fixés pour y parvenir fut de créer une simulation graphique du système solaire à l'échelle

> ORGANISATION DU TRAVAIL :

• Présentation de l'équipe (prénom de chaque membre et rôle dans le projet)
• Répartition des tâches
• Organisation du travail (répartition par petits groupes, fréquence de réunions, travail en dehors de l'établissement scolaire, outils/logiciels utilisés pour la communication et le partage du code, etc.)

L'équipe est composée de Arthur(programmation python du projet), Noé(gestion des calculs et tests du simulateur), et Léa(interface graphique). Nous avons utilisé Nextcloud pour partager le code, afin que chacun puisse le relire et y apporter des modifications puis poster le nouveau code en indiquant les changements en commentaires. Nous communiquions souvent par mail pour communiquer les bugs et l'avancée du projet, et faisant parti du même groupe NSI, nous pouvions parfois venir le midi travailler ensemble sur le projet en salle informatique.

LES ÉTAPES DU PROJET :

• Présenter les différentes étapes du projet (de l'idée jusqu'à la finalisation du projet)

L'on a tout-d'abord essayé de travailler sur un modèle utilisant les lois de Kepler en les généralisant à 3 dimensions et superposant les résultats pour des problèmes à n corps, mais lors des simulations, on s'est aperçu que les bugs, le temps d'execution et les imprécisions étaient plus conséquents qu'en utilisant des vecteurs et en les mettant à jours après une certaine intervalle de temps. Nous avons ensuite créé le code chargé de calculer les orbites, et qui fut testé par l'interface graphique créée peux-après. En voyant les défauts du programme, qui tend à augmenter l'énergie du système, nous avons commencé à coder pour que l'énergie de chaque planète reste constante durant la simulation.

> FONCTIONNEMENT ET OPÉRATIONNALITÉ :

• Avancement du projet (ce qui est terminé, en cours de réalisation, reste à faire)
• Approches mises en œuvre pour vérifier l'absence de bugs et s'assurer de la facilité d'utilisation du projet
• Difficultés rencontrées et solutions apportées

Notre modèle est actuellement restreint aux trajectoires d'excentricité faible, comme dans le système solaire. Dans le cas où la distance entre le centre de masse et les objets orbitant autours varie, comme pour les comètes ou les système avec plus d'un corps massif, l'énergie augmente, résultant en des planètes partant à des vitesses proche de celle de la lumière au confins de l'univers. Nous travaillons actuellement sur résoudre ce problème en utilisant les coordonnées cycliques dans le code pour adapter la vitesse en fonction de l'énergie cinétique du corps tout en conservant l'orientation du vecteur.

Pour vérifier que notre représentation est à l'échelle, nous entrons les données facilement comparables du système solaire dans le terminal, et nous testons les fonctions pour s'assurer qu'elles ne présentent pas de bugs. Nous avions au départ souhaité passer par des librairies comme numpy pour faciliter les calculs des vecteurs, mais certains de notre groupe n'arrivant pas à y accéder, l'idée fut abandonnée pour faire place à du 'code brut'.

> OUVERTURE :

• Idées d'améliorations (nouvelles fonctionnalités)
• Stratégie de diffusion pour toucher un large public (faites preuve d'originalité !)
• Analyse critique du résultat (si c'était à refaire, que changeriez-vous dans votre organisation, les fonctionnalités du projet et les choix techniques ?)

Le projet se tournera ensuite vers l'optimisation avec notamment la méthode de Runge-Kutta, et la possibilité d'entrer les coordonnées directement depuis l'écran, et de manipuler la simulation en cours de route, en sélectionnant une planète et en modifiant ses attributs. De telles améliorations permettraient à ceux qui ne savent pas utiliser un terminal, de pleinement profiter du programme, que ce soit pour améliorer assister ses recherches en astrophysique, ou simplement par plaisir de simuler. Des modèles de planètes et systèmes solaires pourraient être inclus dans le code pour accélérer la création et épargner de rentrer l'entièreté des données à chaque nouvelle simulation.
Si on n'a eut autant de retard sur ces améliorations, je pense que c'est en partie car de nombreux éléments ont été tenté pour le projet, comme Kepler et Numpy, et se sont révélés chronophage.

DOCUMENTATION

• Spécifications fonctionnelles (guide d'utilisation, déroulé des étapes d'exécution, description des fonctionnalités et des paramètres)
• Spécifications techniques (architecture, langages et bibliothèques utilisés, matériel, choix techniques, format de stockage des données, etc)
• Illustrations, captures d'écran, etc

Terminal :

On entre dans le terminal :

python3 Simulation_orbitale.py dt r plns

où dt est le changement de temps par ms dans la simulation(Si on l'augmente, la simulation devient plus rapide mais moins précise, et inversement), r le rayon des planetes sur le canvass, et plns sont les coordonnées de n planetes, qu'on entre dans le terminal au format :

$masse_1\ v\_x_1\ v\_y_1\ v\_z_1\ p\_x_1\ p\_y_1\ p\_z_1\ nbr\_image_1\ masse_2 \ldots nbr\_image_n$

Bibliothèques python :

- traitement d'images : os et pillow
- affichage : tkinter
- arguments du terminal : argparse
- les éléments de trigonométrie importés avec le module math sont utilisés pour que le programme se conforme à la première loi de la thermodynamique.

Images :

stockées dans img_NSI, elles sont bien dessinées et numérisées au format png