

PROJET DE POO DE L2 INFORMATIQUE

CaptainPlanet

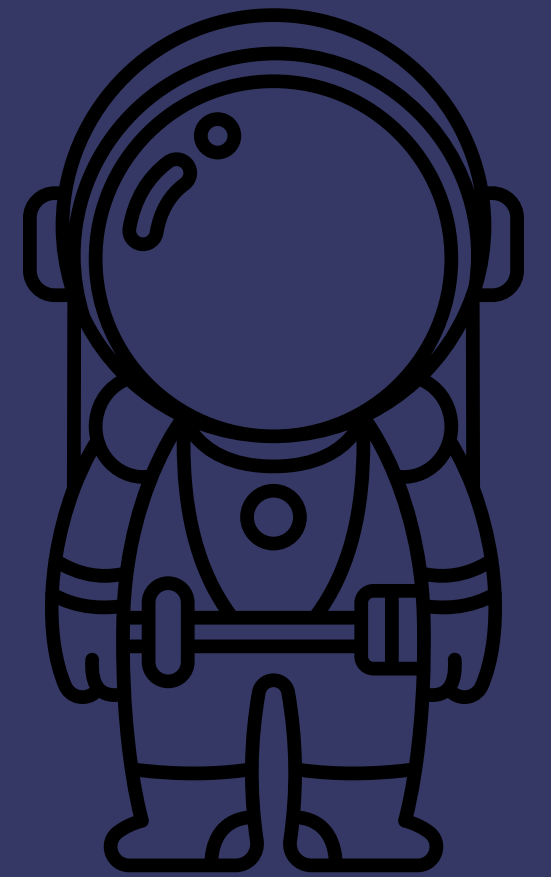


présenté par Théotime & Lucas

Groupe POO22_385K_C

Contexte: cahier des charges

- Quoi?: plan en deux dimensions représentant l'Univers
- Qui?: astronaute se déplaçant dans celui-ci
- Objectif: visiter les corps célestes présents, à la recherche d'un environnement (planète) propice à sa survie au long terme
- Comment?: sous forme d'un jeu interactif; peut susciter un intérêt pour l'exploration spatiale chez le joueur



Contexte: cahier des charges

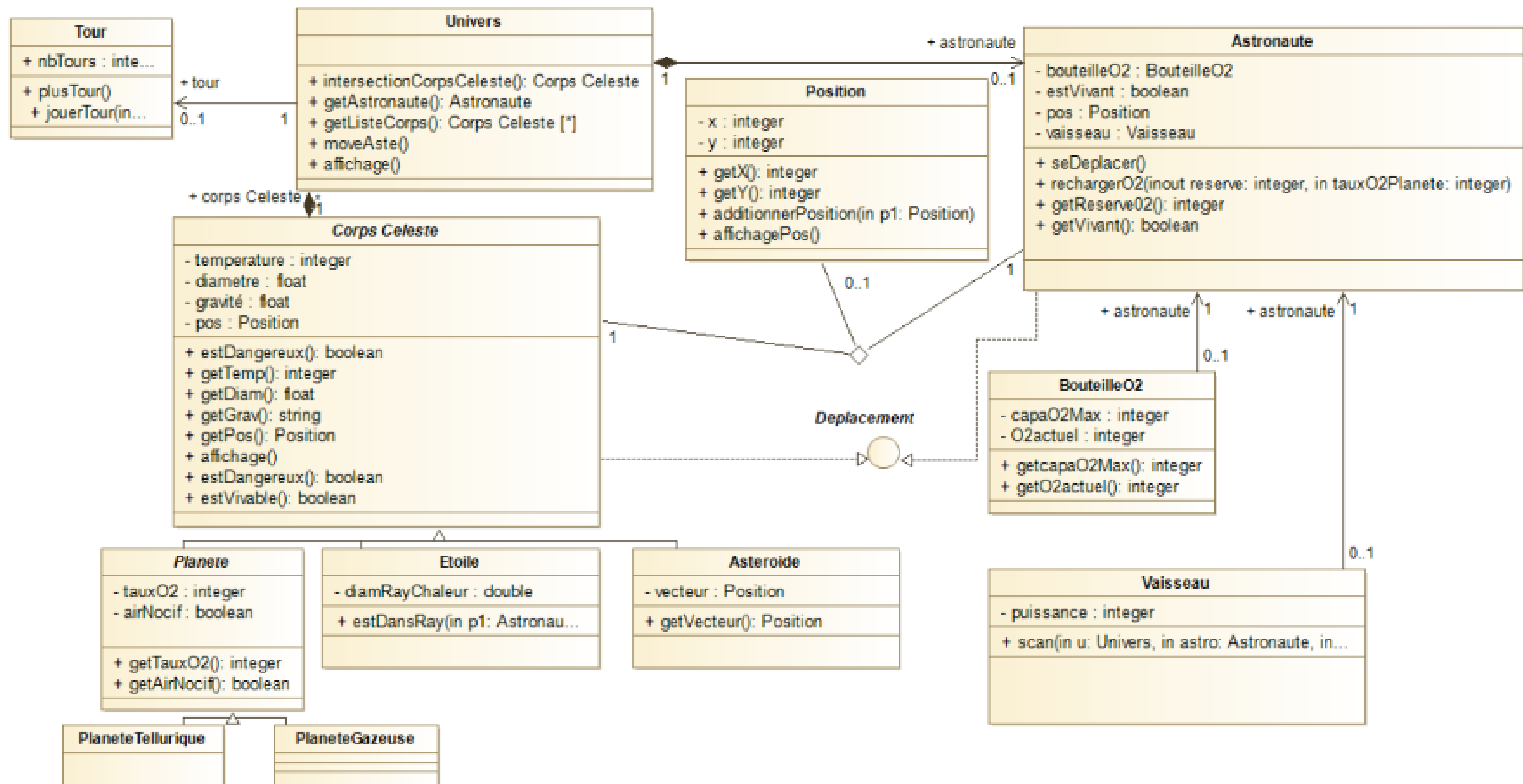
La classe **Univers** sera représentée par la liste de tous les Corps Celeste mais aussi par l'astronaute présent lors de la simulation.

CorpsCéleste: classe mère de tous les astres/toutes les masses présent(e)s dans l'Univers; possède plusieurs caractéristiques déterminant sa dangerosité et sa viabilité pour la survie de notre astronaute (O2, air nocif, température...)

- **Astéroïde:** obstacle se déplaçant dans l'espace en même temps que l'astronaute; pourra détruire le vaisseau donc mettre fin à la simulation
- **Étoile:** possède un rayon de chaleur qui carbonise le pilote en cas de distance trop petite entre les deux
- **Planète:** peut être tellurique ou gazeuse; certaines d'entre elles entraînent la fin de la partie car elles sont propices à la survie de l'astronaute

Astronaute: interagit avec les corps célestes; a une réserve d'O2 qui diminue en fonction de ses déplacements; se déplace avec la Position de son vaisseau (points cardinaux N-S-E-W et puissance de déplacement)

Equipement: composé de BouteilleO2 et Vaisseau, deux classes importantes pour **Astronaute**



Réalisation technique



Interfaces utilisées: graphique (via JavaFX); l'utilisateur prend le contrôle de l'astronaute et décide de ses mouvements grâce à une interaction clavier.

A l'intérieur du programme, les interactions se font principalement entre l'astronaute (son vaisseau) et les corps célestes autour de lui: il scanne les caractéristiques d'un CorpsCéleste via le scanner de son vaisseau, et peut se déplacer pour s'en rapprocher, ou s'en éloigner si il détermine qu'il s'agit d'un objet dangereux (cf étoile qui brûle la coque de son vaisseau)

Réalisation technique



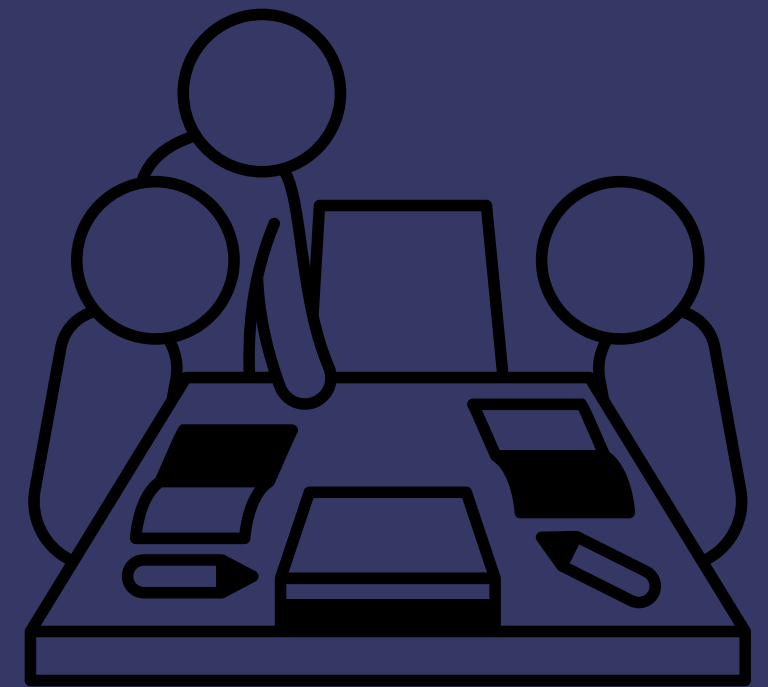
Un des aspects spécifiques de ce projet est la gestion de la réserve d'O₂, qui est importante pour la survie de l'astronaute. Elle pourra être rechargée (augmentation) grâce à certaines planètes, et diminuera en fonction du temps.

Le temps, ou plutôt son unité, sera découlé en fonction des déplacements de l'utilisateur (et non automatiquement).

De plus, il faut atterrir! Ce qui entraîne des calculs précis (si la distance entre le centre d'un C_{Ce} et l'astronaute = le rayon de cette dernière, l'atterrissage est réussi; sinon, c'est un échec).

Gestion de projet

L'équipe: Lucas, Théotime, Enzo, Viet Hung et Nequi



Réunions hebdomadaires, soit pendant les cours, soit sur un serveur Discord privé, créé pour l'occasion

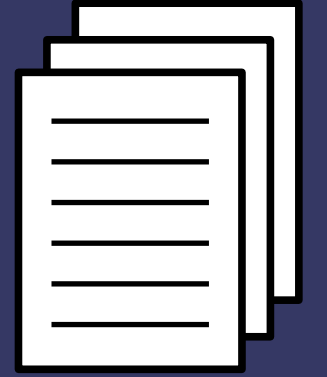
Utilisation de plusieurs outils collaboratifs:

Overleaf (LATEX collaboratif) et Canva (présentations)

Pour coder: Visual Studio Code, en autonomie chez soi & pendant les TPs guidés

Livrables: notre proposition de sujet, le rapport de conception, le code fourni sur Gitlab

Bilan



Objectifs réussis:

Création d'un projet de groupe en Java, interactions et participation de chacun

Technologies apprises:

LATEX, apprentissage de JavaFX et de ses fonctionnalités

Compétences acquises? Succès obtenus? Difficultés rencontrées?

JavaFX est très complexe à manipuler, mais une fois maîtrisé peut être presque ludique. Pour certains, les interactions ont été difficiles; mais ce projet a aidé à mieux comprendre comment les faire.

Critère / Évaluation	D : en-deçà des attendus	C : proches des attendus	B : niveau attendu	A : au-delà des attendus
Vue d'ensemble				
Cahier des charges	Sujet peu original, cahier des charges mal rédigé	Sujet assez adapté, cahier des charges partiel	Sujet bien adapté, cahier des charges clair et détaillé	Sujet d'exception, cahier des charges parfait
Interfaces	Interfaces peu soignées, difficile de tester l'application	Interfaces correctes, mais ne donnant pas accès à toutes les fonctionnalités	Interfaces (graphique ou texte) conviviales, facile de tester l'application	Interface graphique complète permettant une utilisation complète du programme
Quantité de travail	Peu de travail fourni, manque de sérieux, interaction en groupe quasi inexistante	Travail mal découpé ou sujet trop ambitieux, organisation à parfaire	Travail découpé en parties avec une juste quantité de classes à développer	Sujet ample adapté à un groupe très performant et bien organisé
Conception et Rapport				
Architecture	Architecture en opposition au cahier des charges	Architecture correcte avec des faiblesses (couplage fort, classes inadéquates)	Diagramme de classes solide, solution extensible	Belle architecture utilisant des patrons de conception (MVC...)
Regard critique	Absence de discours ou mauvaise argumentation	Discours clair sur quelques points, argumentation correcte	Discours clair sur tous les points, bonne argumentation	Un grand recul sur tous les points
Expression écrite	Trame du rapport non respectée, rapport très léger	Trame du rapport respectée, langage à amender, des fautes d'orthographe	Trame du rapport respectée, langage clair, peu de fautes	Rapport excellent à tous niveaux
Programmation				
Code Java	Code obscur, pas de style, algorithmes compliqués	Style en partie respecté, découpage du code et algorithmes perfectibles	Respect du style, méthodes élémentaires, algorithmes corrects	Code professionnel, excellent découpage du code, algorithmes efficaces
Réutilisation	Absence d'utilisation de la librairie standard	Sous-utilisation la librairie standard et/ou redondance de code	Bonne utilisation de la librairie standard (collections, fichiers, ...)	Bonne utilisation de la librairie standard et d'autres librairies
Documentation	Pas de documentation	Présence de trous dans la documentation (Javadoc et/ou algorithmes)	Classes et méthodes décrites au format Javadoc et algorithmes commentés	Documentation au niveau de celle du JDK et algorithmes commentés
Gestion de projet				
Organisation équipe	Groupe en conflit, interactions faibles, peu de motivation	Motivation présente mais groupe en sous régime, activité irrégulière	Tâches bien réparties, réunions fréquentes, communication efficace	Groupe très autonome montrant une activité soutenue
Contenu du GitLab	Code source partiel, pas de <i>wiki</i> , mises à jour rares	Mises à jour épisodiques du <i>wiki</i> et/ou absence de code ou de documents	Code source et <i>wiki</i> mis à jour régulièrement, <i>wiki</i> structuré (livrables et CR)	Utilisation extensive du GitLab, rubriques additionnelles dans le <i>wiki</i>
Respect du planning	Nombreux retards et absences	Retards légers pour des livrables ou quelques absences en TD/TP	Fourniture stricte des livrables, présence des groupes en TD/TP	En plus du niveau attendu, CR de toutes les réunions prévues
Soutenance				
Maîtrise du sujet	Le message n'est pas passé	Le message est passé, quelques faiblesses ou imprécisions dans le discours	Exposé clair et adapté montrant une maîtrise des concepts et des techniques	Exposé percutant montrant une grande maîtrise du sujet
Qualité des supports	Trame non respectée, informations partielles, diapositives peu lisibles	Trame respectée, bon niveau d'information, des diapositives à améliorer	Trame respectée, bon niveau d'information, diapositives correctes et lisibles	En plus du niveau attendu, belles diapositives captant l'attention
Expression orale	Exposé difficile montrant un manque de préparation	Ton dynamique, bon niveau de langage mais des hésitations	Ton dynamique, bon niveau de langage, discours relié au support	En plus du niveau attendu, communication très persuasive et argumentée

B
B
A
B
B
A
B
A
A
A
A
A
A
B
A
B

Autoévaluation du livret

Merci pour votre écoute!

Des questions?