

# Rapport de conception du projet de programmation orientée objet

Licence d'informatique – 2ème année  
Faculté des sciences et techniques de Nantes

*CaptainPlanet*

présenté par

*BOCQUENE Lucas*

*BOTANS Enzo*

*GODEFROY Theotime*

*MOUOT Nequi*

*NGUYEN Viet Hung*

le *05 11 2022*

encadré par

*Ygor GALLINA*

# 1 Cahier des charges

Dans ce projet, nous nous situons dans un plan en deux dimensions représentant l'Univers. Pour exploiter cet espace, nous y placerons un astronaute, qui pourra se déplacer et explorer ses alentours. L'objectif de cet astronaute sera de visiter les corps célestes l'entourant à bord de son vaisseau pour déterminer si l'environnement est propice au développement de la vie.

Ce programme dans sa version finale prendra la forme d'un jeu interactif, l'utilisateur contrôlera l'astronaute dans la simulation et pourra le déplacer comme il le souhaite à travers le plan en deux dimensions. Cette simulation peut prendre une forme ludique et éducative et développer un goût pour l'exploration spatiale chez le joueur.

L'Univers sera représenté avec plusieurs classes qui représenteront chacune un corps céleste différent, avec chacun leurs propriétés uniques et particularités :

- La classe `CorpsCeleste` sera la classe mère de toutes les masses et astres qui seront présents dans notre espace. Une température/gravité trop élevée ou trop basse rendra la planète inhabitable voire dangereuse pour l'astronaute. Le diamètre, la position et `permetAtterissage` seront nécessaires à la méthode de calcul et les interactions avec l'astronaute, pour savoir quand et où il est possible d'atterrir. Ces données seront également utilisées pour savoir si un astéroïde rentre en contact avec le vaisseau, et en général comment un corps céleste interagit avec ses alentours.
- Les astéroïdes serviront d'obstacles dans notre simulation. En se déplaçant dans l'espace grâce à une méthode, ils pourront potentiellement entrer en collision avec le vaisseau. Ils auront un vecteur qui à chaque déplacement de l'astronaute, sera additionné avec sa position pour le faire bouger dans le temps. En cas de rencontre avec le vaisseau, la collision pourra abîmer ou détruire celui-ci.
- Une étoile possédera un rayon de chaleur à ne pas traverser sous risque d'endommagement voire fusion totale du vaisseau.
- Les planètes seront l'objectif principal de notre astronaute et le but poursuivi durant toute la simulation. Les propriétés d'une planète comme son taux en  $O_2$  et la nocivité de son air (sous forme de booléen) décideront si la planète est habitable et intéressante pour l'astronaute. Une fois sur une planète l'astronaute aura plusieurs possibilités.
- La classe `Astronaute` sera notre classe principale concernant l'interaction avec les `Corps Celeste`. Il aura une réserve d'oxygène qui lorsque elle tombe à zéro provoquera la mort de notre astronaute en mettant à `false` le booléen `estVivant`, stoppant alors la simulation. Ce même booléen, sera utile pour de nombreuses méthodes permettant de mettre fin à la simulation par exemple si la température d'un `Corps Celeste` est trop élevée ou que l'atmosphère de celui-ci est toxique. Il aura aussi une position, qui sera amené à changer au fur et à mesure des déplacements. Il y aura d'ailleurs une méthode de déplacement de notre astronaute. Cette méthode sera un peu notre unité de temps de la simulation. Chaque fois qu'elle sera utilisée, une unité de temps s'écoulera donc notre astronaute perdra une certaine quantité d'oxygène, les astéroïdes se déplaceront d'une fois par le vecteur (d'autres méthodes possible au fur et à mesure de la conception). Notre astronaute pourra aussi recharger son oxygène si il se situe sur une planète qui le lui permet à l'aide de la méthode `rechargerO2`. Il aura aussi la méthode qui lui permettra de gagner le droit de s'installer sur une planète, une méthode `peutSinstaller` qui, en fonction de la planète sur laquelle il se situe, retournera un

booléen en fonction du taux d'O2, de la température, la gravité du Corps Celeste (d'autres paramètres seront ajoutés au fur et à mesure de la conception). Si la méthode retourne vrai, la simulation est fini mais surtout reussi, si elle retourne faux, notre astronaute repart et peut refaire un déplacement vers un autre corps celeste.

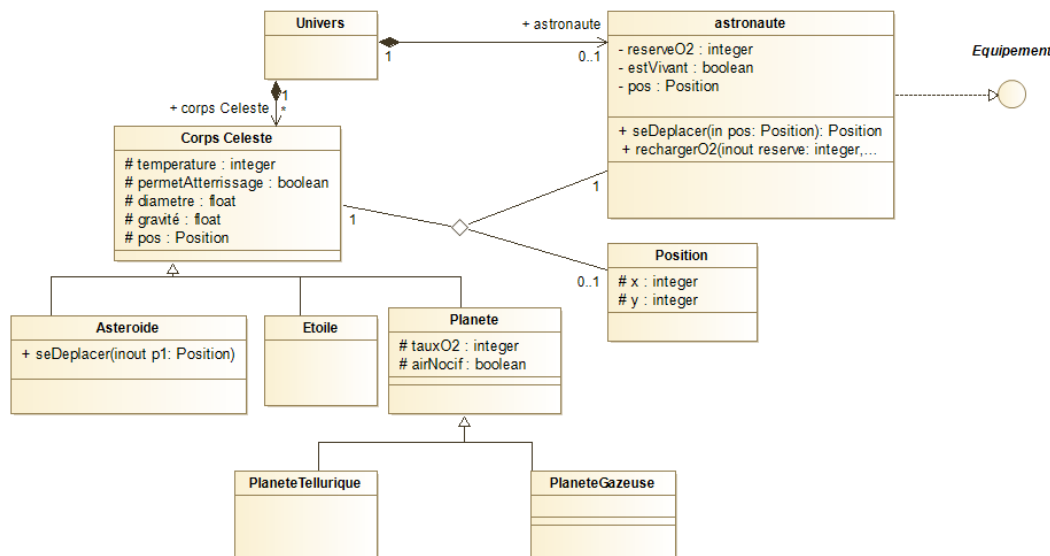
- Pour aider l'astronaute, l'interface équipement lui permettra de choisir entre plusieurs lors d'une simulation comme une bouteille d'O2 contenant plus d'O2 que ce qui est prévu au depart, ou un meilleur reacteur lui permettant des déplacement plus grands à chaque appel de la methode de déplacement de l'astronaute.

## 2 Architecture

### 2.1 Description générale

Dans l'Univers, il existe une quasi infinité de types de corps célestes : les planètes, les étoiles, les astéroïdes, ETC... Toutes ces entités peuvent être liées au sein d'un même système 'solaire' et interagir entre elles à travers les nombreuses méthodes implémentées.

### 2.2 Diagramme de classes



### 2.3 Interfaces

Le personnage principal se déplacera sur un plan en deux dimensions. Le tout sera généré par le logiciel JavaFX, qui nous permettra de gérer une interface graphique ainsi que d'avoir une vraie représentation de l'exécution de notre programme. L'utilisateur prendra le contrôle de l'astronaute, et contrôlera ses mouvements ainsi que toutes les actions réalisables par celui-ci (se soigner, se déplacer avec son vaisseau, analyser, etc..).

## 2.4 Aspects spécifiques

Algorithme basé sur les positions du personnage et de la planète sur laquelle il veut atterrir (collision exacte entre points sinon : échec de l'atterrissage). Algorithmes de gestion (augmentation/réduction) des réserves d'eau et d'oxygène au gré de l'exploration et des environnements rencontrés.

## 3 Regard critique

Ce projet nous a permis de faire appel à la création d'interfaces ; nous avons choisi un sujet riche, qui nous permet de bien exploiter toutes les compétences et concepts qui nous ont été enseignés au cours de ce module.

Au départ, il a été perturbant d'avoir une telle liberté sur le sujet à créer, rendant le calibrage complexe. Mais de nombreuses discussions au sein de l'équipe ont permis de trouver des idées précises et réalisables dans le cadre d'un projet de L2. Les réunions hebdomadaires, qu'elles soient en cours ou via le serveur Discord créé, aident à se concentrer et à trouver de nouvelles solutions aux problèmes que l'équipe peut rencontrer tout au long de la création.

Les qualités de chaque membre de l'équipe permettent de s'entraider et de compléter les idées de chacun au besoin. Comme précédemment dit dans notre proposition de sujet, "C'est un gain de temps et d'énergie non négligeables". Ce document a également permis d'établir les dites qualités.

Sur les compétences de l'équipe au début du projet et le développement de nouvelles compétences. Amélioration de l'investissement pour effectuer un travail de groupe. Développement des compétences de codage en Java, découverte de JavaFX. L'équipe est très motivée pour se soutenir et débattre de nouvelles idées.

Ce projet pourrait être réadapté en 3 dimensions, ce qui complexifierait les méthodes utilisées (3ème coordonnée à prendre en compte). Nous pourrions également implémenter un système de physiques dans le plan à travers notre code et programmer les orbites de tous les astres et les forces qui interagissent entre ces corps (force gravitationnelle, rotation sur elle même). Cependant étant donnée l'échéance précisée pour ce projet ces démarches semblent trop ambitieuses.

La définition d'un chef de projet permet d'avoir un membre référent pour effectuer la communication entre les membres du groupe et l'encadrant du projet. Grâce au projet GitLab, chaque membre du groupe se tient informé des différentes modifications, ajouts et changements durant l'intégralité du projet et permet une cohésion certaine entre tous les membres du groupe pour tout le déroulement du projet.