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
MOUTOT Nequi
NGUYEN Viet Hung
le *05 11 2022*

encadré par
GALLINA Ygor

Nantes Université

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. Voici l'ensemble des classes, interfaces et méthodes utilisant le polymorphisme de la programmation orientée objet nécessaires à notre jeu :

-Position La classe Position nous permet de situer dans le plan l'ensemble de nos corps celestes ainsi que notre astronaute. Elle possède une méthode additionnerPos pour incrémenter celle-ci et afficherPos pour afficher la position de la classe en question.

-Déplacement L'interface Deplacment contient une méthode seDeplacer qu'utilise nos astéroïde et notre astronaute pour se mouvoir dans l'univers, le plan en 2D.

-Univers :La classe Univers sera représentée par la liste de tous les Corps Céleste mais aussi par l'astronaute présent lors de la simulation. Cette classe sera munie d'une méthode intersectionCorpsCeleste permettant de vérifier si notre astronaute est placé dans le plan sur un corps céleste en parcourant la liste de ceux-ci et en vérifiant à l'aide de leur centre et de leur diamètre si l'astronaute est dessus. La méthode statique universTest nous permet de créer un univers de test pour l'exécution du programme. La méthode moveAste effectue le déplacement de tous les astéroïdes présents dans l'univers grâce à la méthode seDeplacer. La dernière méthode est celle d'affichage, qui ne permet d'afficher l'entièreté des corps célestes de notre univers. On présentera ensuite les classes qui découlent de l'Univers :

-CorpsCeleste : La classe CorpsCeleste sera la classe mère de toutes les masses et astres qui seront présents dans notre espace. Elle sera une classe abstraite et ses méthodes seront codées dans les classes qui l'implémentent. Elle aura comme attributs sa température, son diamètre, sa gravité et sa position. Tous ces attributs seront donc présents dans ses sous-classes. Elle implémentera trois méthode abstraite : estVivable, pour savoir si le corps céleste peut accueillir la vie humaine. estDangeureux nous permettra de savoir si celui-ci est mortel ou non pour notre astronaute. Enfin, la méthode abstraite afficherCorpsCeleste affichera toutes les données du corps en question.

-Astéroïde : Les Astéroïdes serviront d'obstacles dans notre simulation. En se déplaçant dans l'espace grâce à une méthode, il 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 tuera l'astronaute.

-Étoile : Les étoiles seront les obstacles statiques mais mortelles pour le joueur. En effet, ceux-ci possèdent un rayon de chaleur faisant carboniser toute personne s'en approchant de trop près, comme notre astronaute. Ils seront comme dans la réalité les corps célestes les plus gros de notre univers.

-Planète : 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 oxygène et la nocivité de son air (sous forme de booléen) lui permettront de savoir si en cas de planète tellurique, celle-ci est mortelle, vivable ou habitable.

-Tellurique : Les planètes telluriques sont les seules planètes où notre astronaute peut atterrir. Si celles-ci ont des conditions de vie mortelles, alors notre astronaute mourra. Si les conditions ne sont pas mortelles mais ne permettent pas d'y habiter au long terme, alors l'astronaute rechargera sa bouteille d'oxygène si il y en a de présent sur la planète.

-Gazeuse : Les planètes gazeuses sont des corps célestes présents dans l'univers mais non-utiles à notre astronaute. Il ne sont pas dangereux car l'astronaute ne pourra pas atterrir dessus, et ne sont donc pas des planètes sur lesquels il peut vivre.

-Astronaute : La classe Astronaute sera notre classe principale concernant l'interaction avec les Corps Céleste. Notre joueur possédera une réserve d'oxygène contenu dans une bouteille d'oxygène qui lorsqu' elle tombe à zéro provoquera sa mort 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 Céleste est trop élevée ou que l'atmosphère de celui-ci est toxique, menant à la mort de notre astronaute). Il aura aussi une position, qui sera changée au fur et à mesure des déplacements. La méthode seDeplacer de notre astronaute demandera à l'utilisateur une direction selon les quatre points cardinaux valides (N,S,E,W) ainsi qu'une puissance nécessaire à l'avancer de son vaisseau. Cette méthode sera notre unité de temps de la simulation. Chaque fois qu'elle sera utilisé, une unité de temps s'écoulera donc notre astronaute perdra une certaine quantité d'oxygène avec la méthode reduc02 et les astéroïdes se déplaceront d'une fois par le vecteur avec leur méthode seDeplacer. Notre astronaute pourra aussi recharger son oxygène s'il se situe sur une planète qui le lui permet à l'aide d'une méthode recharger02. Si la méthode recharger02 est appelée, notre astronaute repart et peut refaire un déplacement vers un autre Corps Céleste.

-Équipement : Pour aider l'astronaute dans sa quête spatiale, deux types d'équipements seront nécessaires : les bouteilles d'O_2 et le Vaisseau. Nous les catégorisons comme ceci mais nous avons décidé de créer ces deux classes indépendamment sans les lier.

-Bouteille d'O_2 : Pour permettre à l'astronaute de respirer lors de son voyage, notre astronaute aura besoind e bouteilles d'oxygène.. Chaque bouteille baissera en quantité d'oxygène avec chaque unité de temps passée et pourra être remplie uniquement sur les planètes avec un air respirable, ce qui obligera l'astronaute à chercher des planètes habitables s'il ne veut pas tomber à cours d'oxygène (ce qui mettrait fin à la partie).

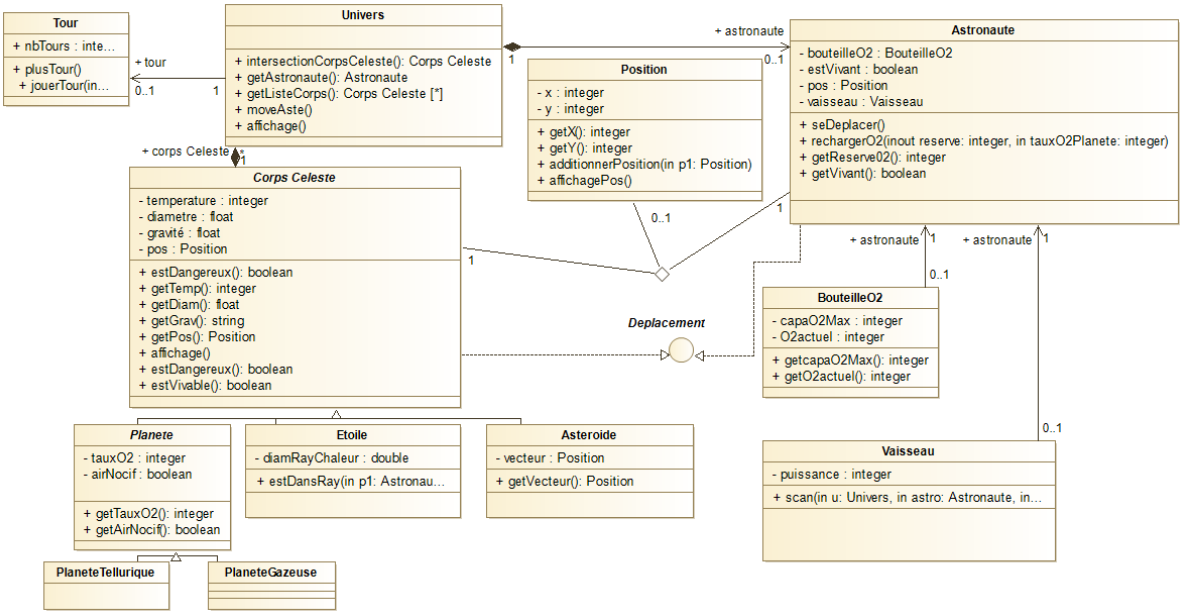
-Vaisseau : Le Vaisseau sera le seul moyen de l'astronaute pour se déplacer dans la simulation. Il pourra changer sa puissance pour se déplacer plus ou moins loin selon les besoins de l'utilisateur (demandé dans la méthode seDeplacer de l'astronaute). Il est équipé d'un Scanner sous forme de méthode pour obtenir les spécificités des Corps Céleste aux alentours, pour savoir s'il est mortel de s'en approcher ou de s'y poser.

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 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

Un des aspects spécifiques à notre projet est la gestion (augmentation/réduction) de la réserve d'oxygène au gré de l'exploration et des environnements rencontrés. Il y aura également la gestion des tentatives d'atterrissage de notre astronaute sur une planète : sa distance du centre de celle-ci devra être égale à son rayon (sinon : échec de l'atterrissage). Les unités de temps seront liées au déplacement de l'astronaute, et non déterminées automatiquement.

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). Ou bien implémenter d'autres équipements aidant l'astronaute ; enfin, il pourrait y avoir d'autres dangers

présents sur les corps célestes (obstacles, aliens, etc). 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.