

## **TP AP4B**

Automne 2020

# **CONCEPTION ET RÉALISATION DU JEU « *TERRAGENESIS* » EN JAVA**

**Réalisé par :**

**AUDRAN Hugo**

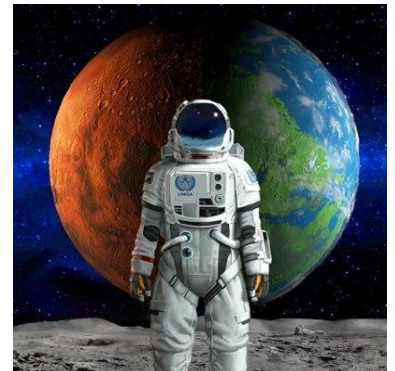
**WANDJI Daniele**

**MERCY Valentin**

**BOUNNIT Zakaria**

**Encadré par :**

**Mr. GECHTER Franck**



**Année Universitaire : 2020-2021**



# **RESUME**

Le présent document est le rapport de conception du projet effectué dans le cadre du TP d'AP4B.

L'objectif de ce projet est d'exploiter et de mettre en pratique toutes les connaissances acquises lors des cours et des TD.

Le présent travail consiste à la conception et à la réalisation du jeu « TerraGenesis ».

Pour la concrétisation de notre projet, nous avons fait appel au langage de programmation orientée objet : (JAVA), et au langage de modélisation unifié : (UML).

# **ABSTRACT**

This document is the report of the mini project carried out within the framework of AP4B course.

The objective of this project is to use and put into practice all the knowledge acquired during the course and the tutorials.

The present work consists in the conception and the realization of the game «TerraGenesis».

For the realization of our project, we used the object-oriented programming language: (JAVA), and the unified modeling language: (UML).

# **LISTE DES FIGURES**

Tableau 1 : Méthode QQQQCP .....	3
Tableau 2 : Matrice des risques .....	5
Figure 1 : Diagramme de Gantt (partie 1/2).....	7
Figure 2 : Diagramme de Gantt (partie 2/2).....	7
Figure 3 : Diagramme de PERT.....	8
Figure 4 : Diagramme de classes .....	12
Figure 5 : Diagramme de cas d'utilisation.....	13

# **T**

## **ABLE DES MATIERES**

RESUME

ABSTRACT

LISTE DES FIGURES

INTRODUCTION GENERALE..... 1

**Chapitre I : Contexte général du projet..... 2**

1. Contexte du projet et problématique.....3

2. Définition des acteurs du projet... .. 4

3. Objectifs du projet... ..4

4. Matrice des risques... ..4

5. Gestion de projet... ..6

a. Diagramme de GANTT .....6

b. Diagramme de PERT ..... 8

**Chapitre II : Analyse et conception..... 9**

1. Analyse ..... 10

2. Conception..... 11

a. Diagramme de classes ..... 12

b. Diagramme de cas d'utilisation..... 13

**Chapitre III : Réalisation de la plateforme..... 16**

1. Langage de développement utilisé ..... 17

2. Logiciels utilisés ..... 17

WEBOGRAPHIE / BIBLIOGRAPHIE

## INTRODUCTION GÉNÉRALE

Dans le cadre du TP d'AP4B, nous avons été amenés à réaliser notre version du jeu vidéo mobile « TerraGenesis ».

Pour y arriver, il a fallu d'abord suivre une méthodologie de recherche basée sur deux parties, la première est une partie théorique qui consiste à trouver des sources afin de s'informer sur notre sujet tandis que la deuxième concerne la conception et réalisation du projet.

Ce rapport synthétise l'essentiel du travail que nous avons effectué et il s'organise de la manière suivante :

Le premier chapitre présente le contexte général du projet, son but est d'étaler la problématique imposée ainsi que présenté la partie planification du projet.

Le deuxième chapitre expose l'analyse et la conception de notre projet qui doivent être initialement réalisées avant d'entamer le travail technique, puis mentionner la méthode utilisée.

Le dernier chapitre est consacré à la partie réalisation du projet, où nous présentons l'environnement du travail à savoir les outils et langages utilisés.

# Premier chapitre

---

## Contexte générale du projet

✎ Le but de ce chapitre est de comprendre la problématique imposée ainsi que d'établir la planification du projet afin de mener à terme notre jeu.



## **1. Contexte du projet et problématique**

Pour mieux définir notre projet, nous avons choisi la méthode QQQQCP qui représente une carte d'identité du projet permettant une bonne compréhension des différents aspects de notre travail.


Le tableau ci-dessous résume notre projet :

<b><i>Méthode de QQQQCP</i></b>	
<b><i>Quoi</i></b>	<i>Jeu « TerraGenesis »</i>
<b><i>Qui</i></b>	<i>TP AP4B</i>
<b><i>Où</i></b>	<i>UTBM</i>
<b><i>Quand</i></b>	<i>13/11/2020 au 07/01/2021</i>
<b><i>Comment</i></b>	<ul style="list-style-type: none"><li>• <i>Choix d'outils de travail</i></li><li>• <i>Conception</i></li><li>• <i>Réalisation et mise en œuvre</i></li></ul>
<b><i>Pourquoi</i></b>	<i>Réaliser une nouvelle version du jeu « TerraGenesis »</i>


Tableau 1 : méthode QQQQCP

Le présent travail consiste à développer le jeu « TerraGenesis – Les Colons de l'Espace ». Pour développer le jeu, nous allons utiliser le langage de programmation Java. Afin de représenter la planète et que l'utilisateur puisse suivre son évolution dans le jeu, nous allons implémenter une interface graphique représentant la planète sur laquelle l'utilisateur agit. Également, nous allons nous servir des notions d'héritages en Java pour implémenter des classes d'Animaux afin de les séparer en différentes catégories (mammifères, aquatiques, ...). Il en va de même pour les végétaux (arbre, plantes).

## **2. Définition des acteurs du projet**

 Côté Maître d'ouvrage :

- Utilisateur
  - Rôles dans le projet : Terraformer la planète puis la rendre habitable

 Côté maître d'œuvre :

- Nom : Hugo AUDRAN
- Nom : Daniel WANDJI
- Nom : Valentin MERCY
- Nom : Zakaria BOUNNIT
  - Organisme : UTBM
  - Fonction : Etudiants
  - Rôle dans le projet : Gestion, conception et réalisation du projet

## **3. Objectifs du projet**

Le projet a pour objectif principal de réaliser une adaptation du jeu « *TerraGenesis – Les Colons de l'Espace* » c'est-à dire de réaliser un jeu où l'utilisateur peut interagir avec une planète (terraforming, constructions, implantation d'espèces) afin de pouvoir y faire vivre d'abord des colons puis à long terme une population complète.

## **4. Matrice des risques**

La démarche d'identification des risques s'inscrit dans une volonté d'anticipation pour réagir au plus tôt. Cette démarche passe par l'identification des facteurs de risque associés à chaque tâche et de leurs classifications en fonction de leur criticité : ceux qui pourraient entrainer de légers retards dans le planning ou ceux qui bloquent la continuation du projet.

Type	Description du risque	Impact	Actions
<b>Fonctionnel</b>	Besoins incompris	Echec du projet	Présentation détaillée des fonctionnalités et des besoins
	Perfectionnisme dans l'expression des besoins	Ralentissement des travaux	Examen critique de l'expression des besoins
<b>Technique</b>	Blocage sur les limites technologiques du jeu	Ralentissement des travaux	Analyse technique. Enquête sur les configurations. Vérification des fonctionnalités techniques
	Pannes et bugs	Ralentissement des travaux	Suivi des règles de codage et de tests
<b>Projet</b>	Mauvaise gestion du temps	Ralentissement des tâches	Suivre la planification faite au départ
	Des modifications fréquentes pendant le développement	Ralentissement de la réalisation	Définition d'un seuil d'acceptation des changements
	Perte des données	Echec du projet	Stratégies de sauvegarde

Tableau 2 : Matrice des risques

## **5. Gestion de projet**

### **a. Diagramme de GANTT**

Cet outil est un calendrier de réalisation d'un projet qui permet de visualiser l'ensemble du projet dans le temps avec tous ses lots de travail, ainsi que tous ses jalons. Cet outil est efficace car il permet de positionner le projet dans le temps en prenant compte des jours fériés, des week-ends...

Il existe plusieurs logiciels de diagramme de chaque tâche est matérialisée par une barre horizontale, dont la position et la longueur représentent la date de début, la durée et la date de fin. Généralement, le projet se découpe en différentes phases, représentées par des tâches récapitulatives.

Chaque tâche récapitulative englobe la liste des tâches individuelles à effectuer pour accomplir cette phase du projet. Au fur et à mesure de l'avancement du travail.

Les figures ci-dessous exposent la durée consacrée pour chaque étape du projet :

GANTT project		
Nom	Date de début	Date de fin
• TerraGenesis	13/11/20	07/01/21
• Phase de conception	13/11/20	30/11/20
• Diagramme de classes	13/11/20	22/11/20
• Diagramme de cas d'utilisation	21/11/20	30/11/20
• Rédaction du rapport	01/12/20	14/12/20
• Phase de développement	04/12/20	29/12/20
• Choix des outils de travail	04/12/20	04/12/20
• Réalisation du jeu	05/12/20	29/12/20
• Phase de clôture	30/12/20	07/01/21
• Tests et corrections des erreurs	30/12/20	04/01/21
• Réalisation de la vidéo	05/01/21	07/01/21

Figure 1 : Diagramme de Gantt (partie 1/2)

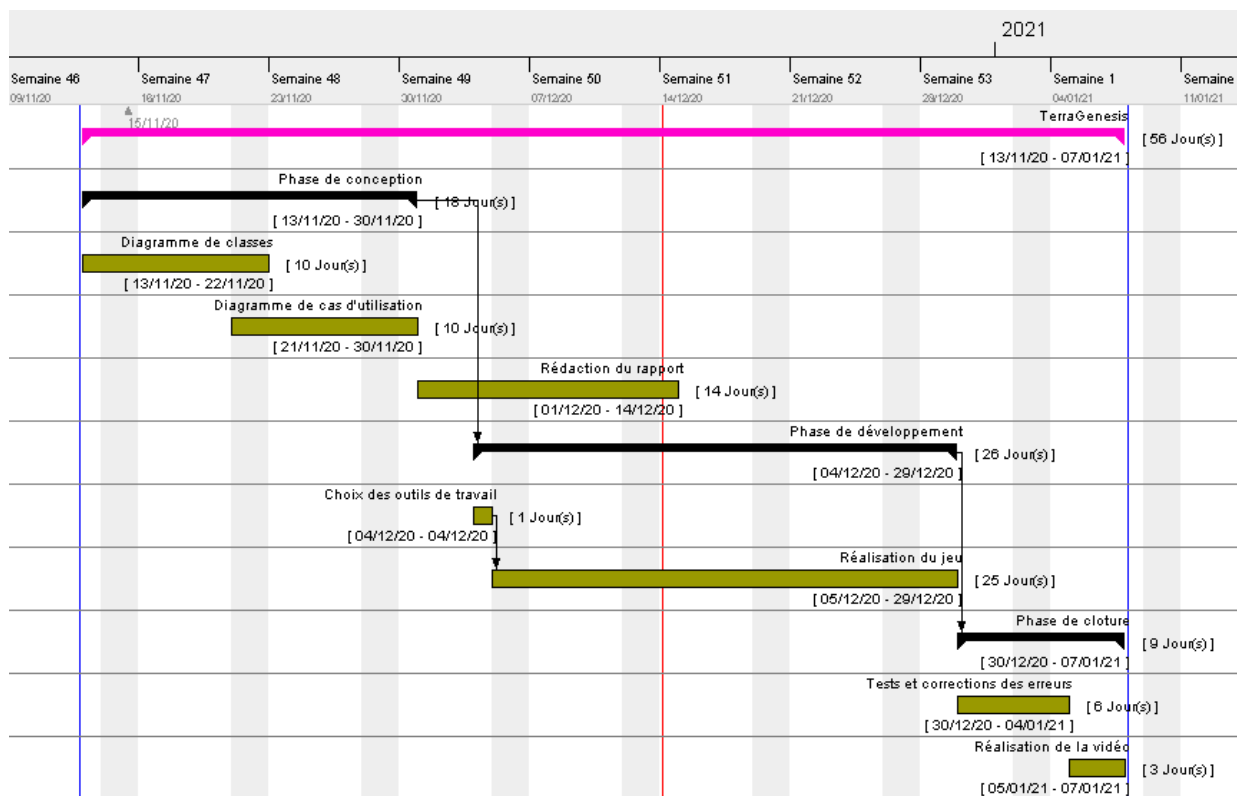


Figure 2 : Diagramme de Gantt (partie 2/2)

## b. Diagramme de PERT

Le diagramme PERT représente le planning des travaux par un graphe de dépendances. Son formalisme en réseau se focalise sur l'interconnexion des tâches à effectuer et sur le calcul des chemins critiques. Une différence importante avec le diagramme de Gantt est l'échelle de temps conventionnelle du diagramme PERT qui représente un enchaînement de tâches et non des durées ou un calendrier.

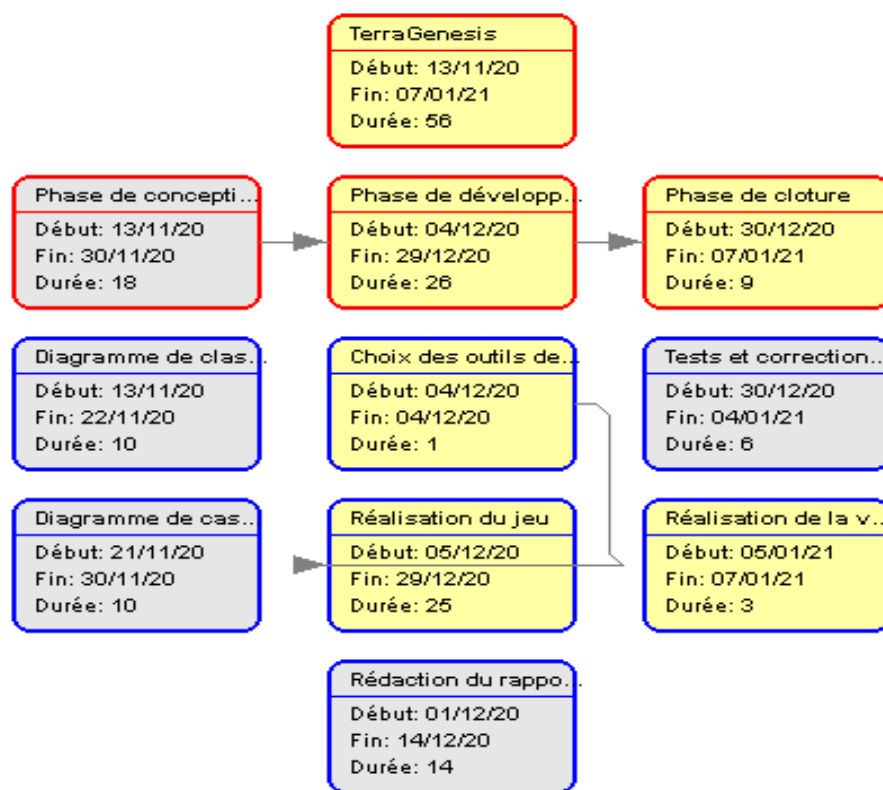


Figure 3 : Diagramme de PERT

## Deuxième chapitre

---

### Analyse et Conception

✎ L'intérêt de ce chapitre est d'étudier l'analyse et la conception du projet qui doivent être initialement réalisées avant d'entamer le travail technique, puis mentionner la méthode utilisée.

## 1. Analyse

Afin de rendre une planète habitable à partir de rien, nous avons réfléchi à différents éléments à intégrer à notre jeu que nous allons présenter dans cette partie.






Dans un premier temps, nous voyions l'utilisateur comme le maître de la planète qui pourrait influencer dessus à sa guise. Ainsi, il aurait été capable de régler les différents paramètres manuellement sans avoir à passer par l'instauration de nouvelles espèces, le développement d'une ville.

Cependant, cette approche aurait rendu la partie moins intéressante pour le joueur et n'aurait pas été fidèle au TerraGenesis initial qui se veut « réaliste ». Nous avons donc décidé que nous allons instaurer un système de ressources nécessaires à l'utilisateur pour progresser dans le jeu. Ces ressources seront acquises en complétant des missions évolutives au fur et à mesure de l'avancée du jeu ou en faisant fonctionner des usines productives de ressources sur la planète hôte.

Ensuite, l'utilisateur pourra utiliser ces ressources pour agrandir la ville (achat de matériel nécessaire à la construction d'usines, de laboratoires, de dortoirs pour installer des colons, ...) ou encore pour faire des expériences qui mèneront à la création de nouvelles espèces et à leur instauration dans la biodiversité locale.

Maintenant que nous avons vu comment l'utilisateur va pouvoir avancer dans le jeu, parlons de ses objectifs et de la manière dont il va pouvoir les atteindre.

Premièrement, l'objectif premier de l'utilisateur sera de rendre la planète habitable, c'est-à-dire de réunir les conditions propices à la vie sur cette planète :

-  Température
-  Pression
-  Biomasse
-  Proportion d'eau
-  Proportion des gaz composant l'atmosphère



Pour réunir ces conditions, l'utilisateur devra en amont construire une ville pour que les premiers colons puissent y vivre, développer des laboratoires des jardins, créer de nouvelles espèces qui constitueront la faune et la flore de la planète et qui permettront finalement l'obtention de ces paramètres propices à la vie.

## **2. Conception**

Pour atteindre les résultats prévus, il est recommandé avant de commencer la réalisation de faire une conception qui détermine les différents éléments qui participent à notre projet.

Donc il est indispensable d'utiliser une méthode de conception et de modélisation, telle que le langage de modélisation unifié UML.

Nous avons décidé lors de la conception de nos diagrammes UML que nous ferions un diagramme de classes et un diagramme de cas d'utilisation. En effet, à la suite de la réalisation de ces deux diagrammes, faire un diagramme de séquence ne nous a pas paru nécessaire puisque nous avons déjà une idée claire de l'organisation de notre programme, de son implémentation et de comment il allait fonctionner par la suite.

## a) Le diagramme de classes

Le diagramme de classes est un schéma utilisé pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci.

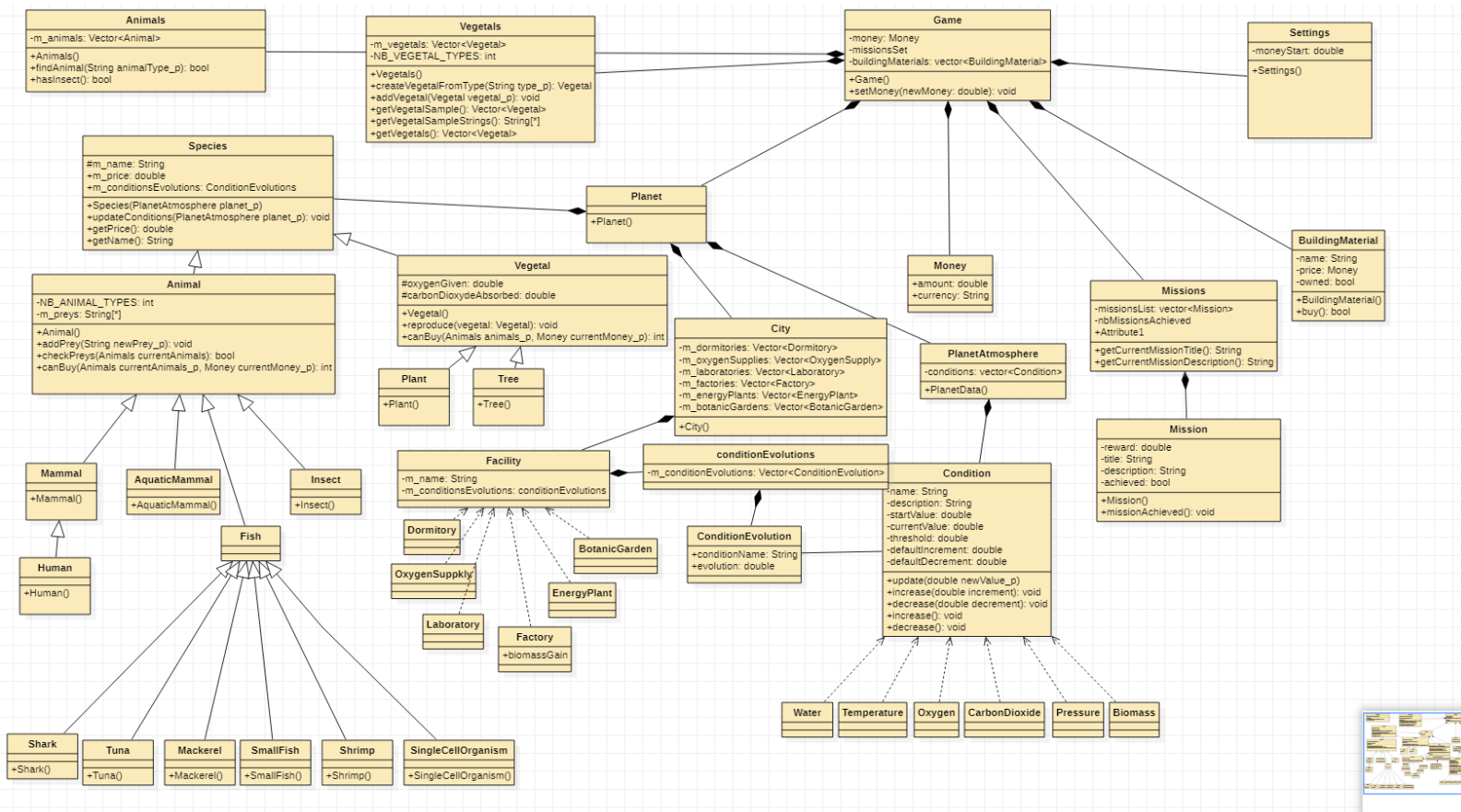


Figure 4 : Diagramme des classes

## b) Le diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est un schéma utilisé pour décrire les fonctions du système selon le point de vue de l'utilisateur.

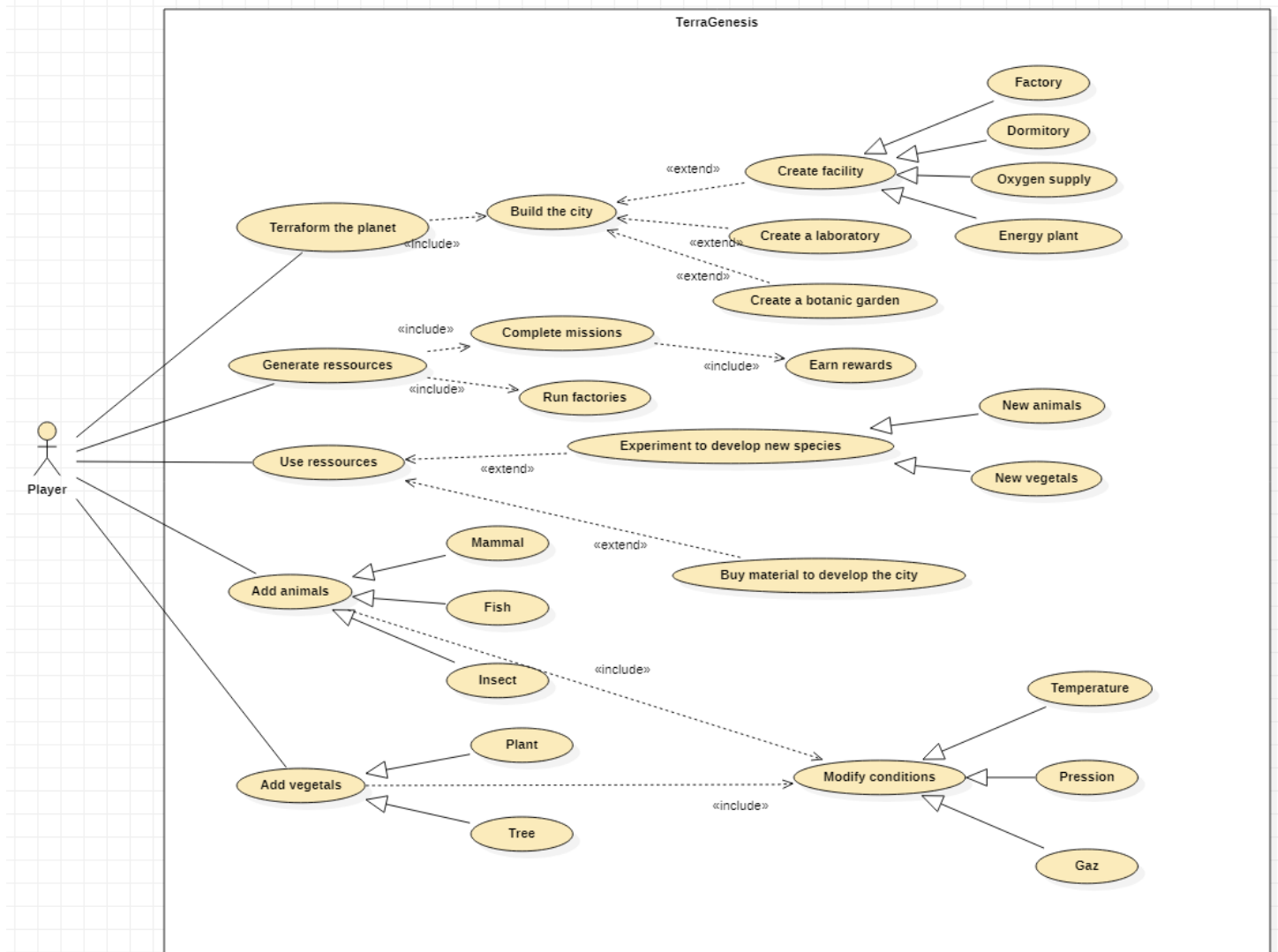


Figure 5 : Diagramme des cas d'utilisation

## Informations utiles pour jouer :

### Proies des animaux :

- Insect
  - Single Cell Organism
- Aquatic Mammal
  - Small Fish
  - Shrimp
- Shark
  - Aquatic Mammal
  - Tuna
- Tuna
  - Mackerel
  - Small Fish
- Mackerel
  - Small Fish
  - Shrimp
- Small Fish
  - Shrimp
  - Sing Cell Organism
- Shrimp
  - Sing Cell Organism
- Single Cell Organism

### Prix des animaux :

- Insect : 20\$
- Aquatic Mammal : 50\$
- Shark : 500\$
- Tuna : 100\$
- Mackerel : 120\$
- Small Fish : 50\$
- Shrimp : 30\$
- SingleCellOrganism : 2\$

### Prix des végétaux :

- Plant : 50\$
- Tree : 150\$

### Liste des missions :

- Add animal (Ajouter un animal à la planète)
- Buy a material (Achetez des matériaux de construction pour construire des bâtiments)
- Build a Dormitory (Ajouter un dortoir à la ville pour héberger les colons)
- Build a Energy Plant (Ajouter une usine d'énergie à la ville pour abriter les colons)
- Build a Factory (Ajouter une usine dans la ville)
- Build a Laboratory (Ajouter un laboratoire dans la ville pour débloquent l'achat d'animaux)
- Build a Botanical Garden (Ajoutez un jardin botanique dans la ville pour débloquent l'achat de plantes)
- Build a Oxygen Supply (Ajoutez un approvisionnement en oxygène à la ville pour héberger les colons)
- Buy a Tree (Achetez un arbre pour permettre la photosynthèse)
- Buy a Plant (Achetez une plante pour permettre la photosynthèse)
- Reach 25% on all atmospheric conditions (Atteindre 25% des conditions)
- Reach 50% on all atmospheric conditions (atteindre 50% des conditions)
- Reach 75% on all atmospheric conditions (Atteindre 75% des conditions)
- Achieve 100% atmospheric conditions (Atteindre 100% des conditions)

Prix des installations (facilities):

- Dormitory : 300\$
- Oxygen Supply : 1500\$
- Laboratory : 1000\$
- Factory : 600\$
- Energy Plant : 500\$
- Botanic Garden : 1200\$

Matériaux nécessaires pour les constructions :

Facility\Material	Wood	Metal	Concrete	Zinc	Copper	Iron	Gold
Dormitory	1	0	1	0	0	0	0
OxygenSupply	0	1	1	0	0	1	0
Laboratory	0	0	1	0	0	0	1
Factory	0	1	1	1	0	1	0
EnergyPlant	1	1	0	1	1	1	1
BotanicGarden	1	1	0	0	0	0	0

Evolution finales des conditions (en %) en fonction des installations dans la ville :

Facility\Condition	Water	Temperature	Oxygen	CarbonDioxide	Pressure	Biomass
Dormitory	0.0	+0.5	0.0	0.0	0.0	0.0
OxygenSupply	+1.0	+0.4	+1.5	0.0	+1.0	0.0
Laboratory	+0.5	0.0	+0.4	0.0	0.0	+0.8
Factory	0.0	0.0	0.0	+1.2	0.0	0.0
EnergyPlant	+0.5	+1.1	+0.8	+0.8	0.0	0.0
BotanicGarden	+0.4	0.0	+1.2	0.0	+1.0	+1.0

## Troisième chapitre

---

### Réalisation

Après avoir achevé la partie de la définition des besoins et la partie de conception, ce chapitre sera dédié à :

🌀 *La présentation de  
l'environnement de travail*

## **1. Environnement de travail**

Dans cette partie, nous présentons les différents outils, langages et technologies utilisés pour réaliser cette application.

### **a) Langage de développement utilisé**



- ❖ **Java**<sup>1</sup> : est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy, présenté officiellement le 23 mai 1995 au *SunWorld*. Il reprend en grande partie la syntaxe du langage C++. Néanmoins, Java a été épuré des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, ou l'héritage multiple contourné par l'implémentation des interfaces.

### **b) Logiciels utilisés**





- ❖ **Eclipse**<sup>2</sup> : est un environnement de développement intégré libre (le terme Eclipse désigne également le projet correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.


---

<sup>1</sup> (Wikipédia, 2020)

<sup>2</sup> (Techno Sciences)

- ❖  **GanttProject**<sup>3</sup> : GanttProject est un logiciel libre de gestion de projet écrit en Java, ce qui permet de l'utiliser sur de nombreux OS tel que Windows, Linux, MacOS. Il permet la planification d'un projet à travers la réalisation d'un diagramme de Gantt. L'outil permet de créer des diagrammes de Gantt, des diagrammes de ressources et des réseaux PERT.

- ❖  **StarUML**<sup>4</sup> : est un logiciel de modélisation UML qui est entré récemment dans le monde de l'OpenSource. Ecrit en Delphi, il est modulaire et propose plusieurs générateurs de code.  
StarUML peut remplacer dans bien des situations des logiciels commerciaux et coûteux comme Rational Rose 1 ou Together 2.  
Ce logiciel constitue une excellente option pour une familiarisation à la modélisation. Cependant, seule une version Windows est disponible.

- ❖  **git**<sup>5</sup> : Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. En 2016, il s'agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes.

---

<sup>3</sup>\_(Wikipédia, 2020)

<sup>4</sup>\_(Wikipédia, 2020)

<sup>5</sup> (Wikipédia, 2020)



# W EBOGRAPHIE / B IBLIOGRAPHIE

**Krol. 2018.** Terra Genesis sur iPhone par Krol. *senscritique.com*. [En ligne] 13 juillet 2018. [Citation : 12 décembre 2020.] [https://www.senscritique.com/jeuvideo/Terra\\_Genesis/critique/172628815](https://www.senscritique.com/jeuvideo/Terra_Genesis/critique/172628815).

**Techno Sciences.** Eclipse (logiciel) - Définition et Explications. *techno-science.net*. [En ligne] [Citation : 2 décembre 2020.] <https://www.techno-science.net/definition/517.html>.

**Wikipédia. 2020.** GanttProject. *wikipedia.fr*. [En ligne] 15 juillet 2020. [Citation : 2 décembre 2020.] <https://fr.wikipedia.org/wiki/GanttProject>.

**Wikipédia. 2020.** Git. *wikipedia.fr*. [En ligne] 7 septembre 2020. [Citation : 2 décembre 2020.] <https://fr.wikipedia.org/wiki/Git>.

**Wikipédia. 2020.** Java (langage). *wikipedia.fr*. [En ligne] 9 décembre 2020. [Citation : 02 décembre 2020.] [https://fr.wikipedia.org/wiki/Java\\_\(langage\)](https://fr.wikipedia.org/wiki/Java_(langage)).

**Wikipédia. 2020.** StarUML. *wikipedia.fr*. [En ligne] 8 septembre 2020. [Citation : 2 décembre 2020.] <https://fr.wikipedia.org/wiki/StarUML>.

