
Rapport Programmation Avancée

Projet ENSemenC



Sommaire

Sommaire	2
Introduction	3
Contexte	3
Règles et spécificités	3
Classes	4
Joueur	4
Jeu	4
Plantes	4
Terrain	5
Potager	6
Visiteur	6
Ressources	6
Gestion de projet	7
Planning	7
Conclusion/Bilan	8

Introduction

Contexte

Ce projet est un simulateur de potager dans lequel le joueur cultive des plantes en fonction du terrain, réel ou imaginaire. Ce choix influence le climat, les types de semis disponibles et les conditions de culture.

Dans le cadre de notre projet, nous avons choisi de travailler sur un monde imaginaire afin d'obtenir plus de liberté sur le choix des plantes, ressources, personnages et événements.

Règles et spécificités

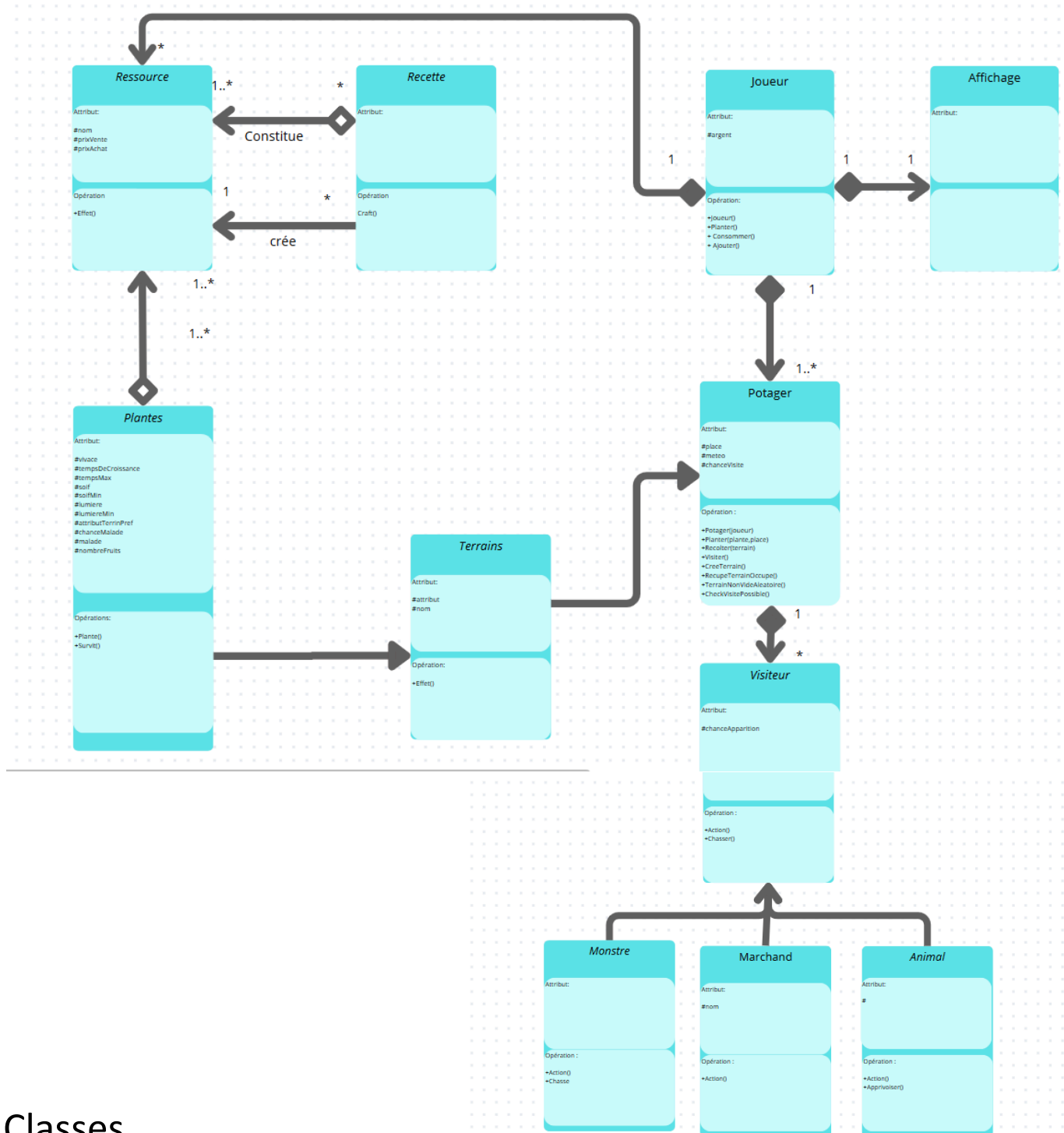
Pour le joueur, le but du jeu est de récolter le plus de plantes possible, dont il devra gérer la croissance, afin de les revendre et obtenir de l'argent, tout cela en survivant aux événements aléatoires qui peuvent survenir durant une partie.

Chaque semaine, le joueur peut réaliser différentes actions:

- passer la semaine
- arroser des plantes
- planter
- voir les informations du joueur: argent et tous les objets qu'il possède
- récolter une plante
- voir son potager

Lorsqu'une semaine passe, les différents attribut des plantes évoluent selon l'environnement.

Les événements apparaissent entre deux semaines et affectent la progression du joueur. Le joueur a alors le choix de réaliser différentes actions selon l'évènement.



Classes

La programmation orientée objet a été utilisée pour coder l'ensemble du projet, ainsi, nous avons commencé par réaliser un diagramme UML des différentes classes qui sont utilisées.

Le diagramme UML réalisés initialement permet la bonne compréhension des différentes classes utilisés ainsi que leur relations.

Joueur

La classe Joueur est la classe centrale du code.

Elle regroupe toutes les informations de la partie: l'argent possédé par le joueur, son inventaire (l'ensemble des ressources qu'il possède (objets, graines, ...)) et les potagers qu'il possède.

Le constructeur de la classe permet d'initialiser les attributs cités précédemment, notamment en donnant quelques ressources de base au joueur pour qu'il puisse commencer le jeu (eau, argent, graines).

Les méthodes associées à cette classe permettent la gestion de l'inventaire ainsi que la réalisation d'action (arroser, récolter, planter) par le joueur durant la partie.

Jeu

Il s'agit de la classe globale du projet, celle qui fait office de simulation. Elle possède un accès au joueur ainsi qu'à l'affichage

Plantes

La classe Plante est une classe abstraite qui sert à la gestion des plantes.

Les attributs sont nombreux et contiennent toutes les informations associées aux plantes (lumière, eau, croissance, ...).

```
public abstract class Plantes
{
    6 references
    public string nom;
    2 references
    public bool vivace;
    17 references
    public double temps;
    5 references
    public int[] tempsDeCroissance;
    2 references
    protected int tempsMax;
    2 references
    protected int vitesseBoire; //Les plantes vont boires l'eau à une certaine vitesse
    7 references
    protected int eau;
    2 references
    protected int eauMin;
    7 references
    protected int lumiere;
    2 references
    protected int lumiereMin;
    2 references
    protected string[] attributPref;
    1 reference
    protected bool malade;
```

```
2 references
public int Eau
{
    get
    {
        return eau;
    }
    set
    {
        if (value < 100)
        {
            eau = value;
        }
        else
        {
            eau = 100;
        }
    }
}
```

Terrain

La classe Terrain gère les différents types de terrains qui peuvent être utilisés ainsi que leur spécificités (eau, lumière).

Les différents types de terrains présents dans le jeu sont donc les sous classes de Terrain.

Potager

Un potager est constitué de plusieurs terrains, et c'est ici que se déroule les événements durant la partie, affectant ainsi tous les terrains/plantes associés à un potager.

La présence de visiteur est générée aléatoirement et les méthodes de cette classe permettent la gestion des terrains utilisés.

Visiteur

La classe visiteur contient les 3 types de visiteurs (=événements) qui peuvent survenir lors d'une partie: les animaux, les monstres et les marchands.

Ressources

La classe abstraite ressource contient toutes les ressources qui sont présentes dans le jeu, associées à leur valeur marchande. Les graines qui peuvent être plantées sont incluses dans cette classe.

Affichage

La classe Affichage contient toutes les méthodes qui permettent d'afficher le contenu du jeu dans la console. Elle a pour vocation de séparer cette partie du reste du code.

Gestion de projet

Concernant la répartition du travail et des tâches, nous nous sommes réunis et avons réfléchi ensemble pour que l'on parte sur la même idée de jeu. Ainsi, nous avons réfléchi à la structure des classes ensemble avant de commencer le code, ce qui nous a permis d'avoir une base claire.

Pour la partie codage, nous avons fait attention à ne pas travailler sur la même partie du jeu pour éviter les conflits, ce qui nous a fait gagner beaucoup de temps.

Planning

	S1	S2	S3	S4	S5	S6	S7	S8
Brainstorming et partage d'idées								
Diagramme UML								
Création des classes								
Code général								
Interface graphique du jeu								
Rédaction livrables								

Conclusion/Bilan

Ce projet à permis d'appliquer la programmation orientée objet dans un cadre concret et donc d'améliorer nos compétences dans ce domaine de la programmation.

Même si la gestion du code est parfois complexe, sa structure a permis de créer des fonctionnalités avancées afin de garantir une meilleure expérience à l'utilisateur.

Le point principal qui aurait pu être amélioré est l'équilibrage global du jeu. En effet, dû à la gestion du temps dans le jeu, les plantes grandissent trop vite par rapport à leur besoin en eau. Il devient donc rapidement inutile d'arroser les plantations.