Rapport Projet

ENSemenC

Desfarges Yann

Bourguignon Louis-Ferdinand

Desfarges Yann

Bourguignon Louis-Ferdinand

Sommaire:

Desfarges Yann

Bourguignon Louis-Ferdinand

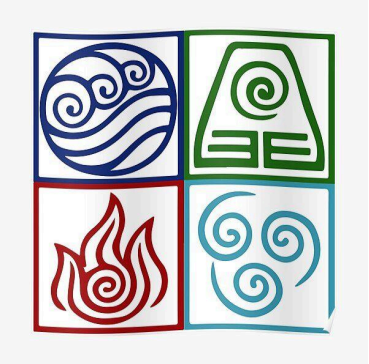
L’univers du jeu et ses spécificités

Ce simulateur de potager est inspiré de la série animée Avatar le Dernier Maître de l’Air. Dans ce dernier, vous allez jouer 15 semaines durant lesquelles vous pourrez faire pousser différentes plantes mais attention ! vous pouvez être attaqué par la nation du feu ! Si vous répondez mal à leurs questions, ces derniers brûleront une partie de votre potager ! Lors de ce jeu, vous pourrez choisir votre nation, le type de terrain (sable,terre,argile) à favoriser pour votre potager, ainsi que la taille de ce dernier ! Vous pourrez ensuite planter des cactus, des choux, des tomates ou encore des tulipes ! Le but du jeu est d’avoir le plus beau potager possible.

Le choix de la nation influencera la météo, si vous êtes de la nation du feu la température sera beaucoup plus élevée (à l’inverse pour la tribu de l’eau). Les cactus seront parfaitement adaptés aux conditions climatiques de cette météo par exemple… Les choux à l’inverse sont faits pour la météo de la nation de la terre !

Le mode urgence (attaque de la nation du feu) se déclenche quelle que soit votre nation (ils sont méchants sans réfléchir donc c’est complètement cohérent).

Les 3 nations disponibles sont : La tribu de l’eau, La nation du feu ou le royaume de la terre. (La tribu de l’air ayant été dévastée).



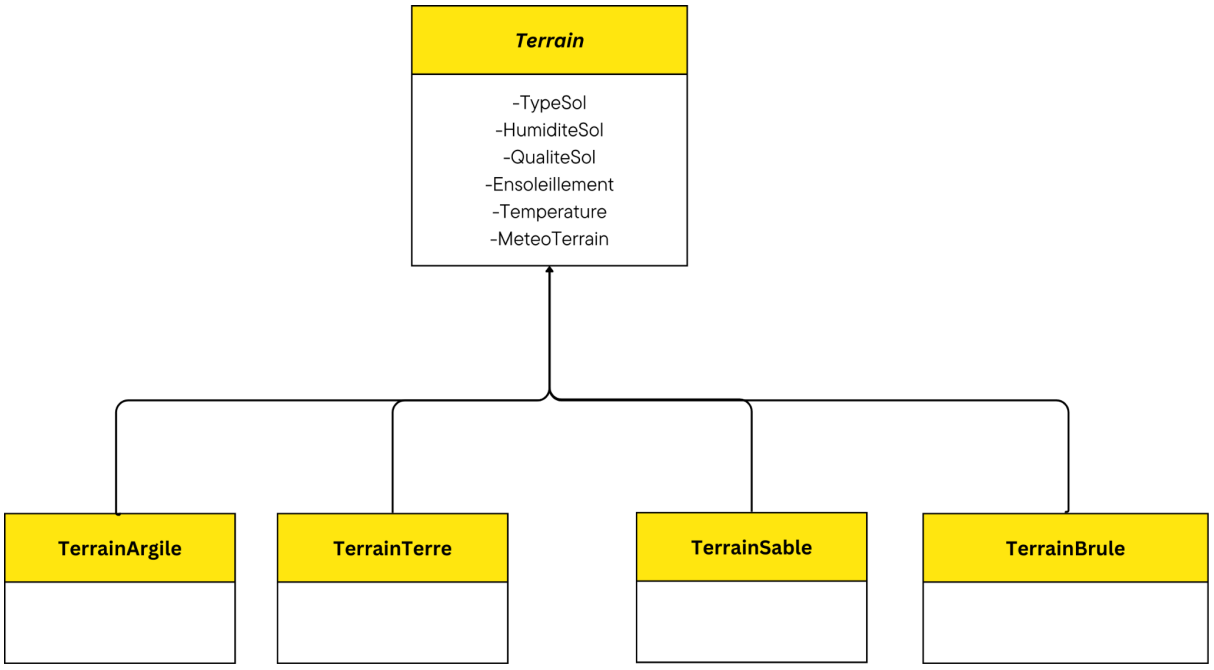
Desfarges Yann

Bourguignon Louis-Ferdinand

Déroulement d'une partie :

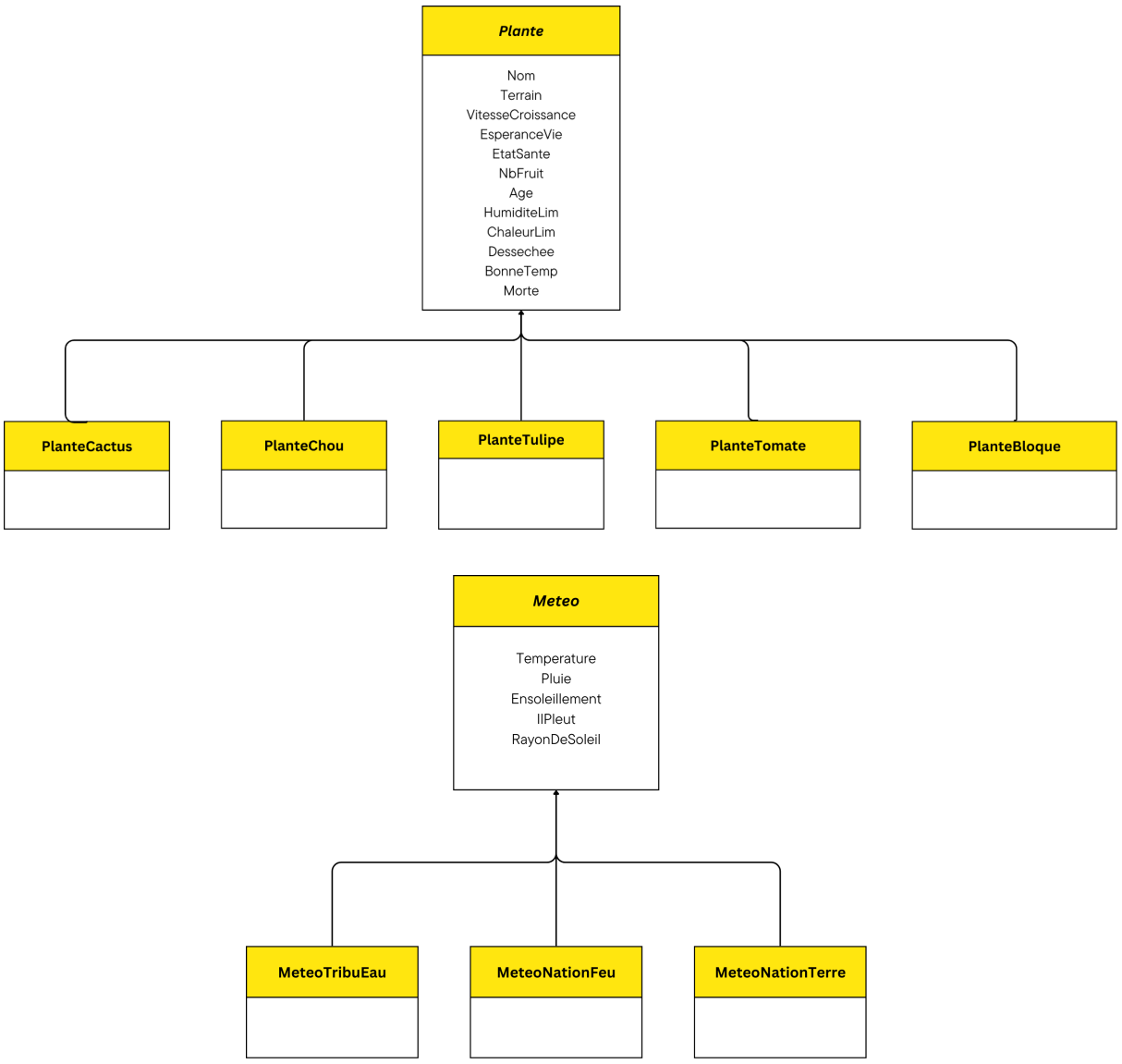
1. Mise en contexte et affichage des règles du jeu.
2. Choix de la nation
3. Taille du potager (limite min 5x5 max 12x12)
4. Choix du type de terrain à favoriser
5. Affichage du potager vide et du numéro de la première semaine
6. Choix pour l'utilisateur de planter une plante sur une case du potager (ce choix se répète maximum 5 fois due au 5 actions max par tour)
7. Mode urgence qui s'active avec une proba de 1/10
8. Après les 15 tours, le jeu se termine et affiche le potager final

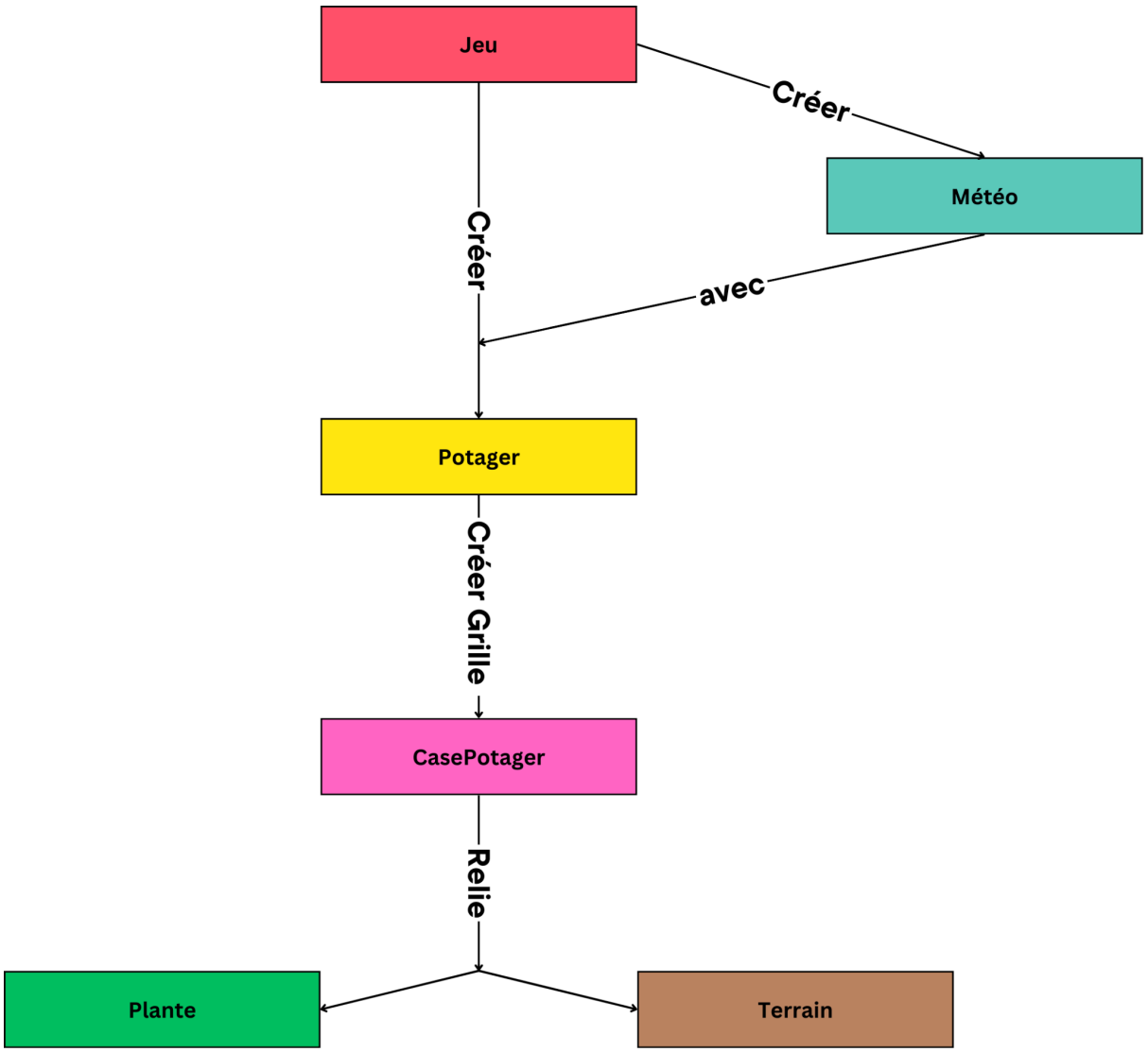
Modélisation simple des classes



Desfarges Yann

Bourguignon Louis-Ferdinand





organisation générale

Tests

```
 MeteoNationFeu m = new MeteoNationFeu();

Console.WriteLine("Température de la semaine " + m.Temperature);

Console.WriteLine("Chance sur 10 qu'ils pleuvent : " + m.Pluie);

m.AppliquerMeteoTour();
```

```
Console.WriteLine("Nouvelle température " + m.Temperature);
if (m.IlPleut)
{
    Console.WriteLine("Il pleut aujourd'hui");
}
else
{
    Console.WriteLine("Il ne pleut pas");
}



Potager po = new Potager(m);


po.AfficherPotager();


po.ChoisirPlanter();


po.AfficherPotager();


po.JouerModeUrgence();


po.AfficherPotager();
```

```
TerrainArgile ter = new TerrainArgile();

Console.WriteLine(ter.HumiditeSol);



TerrainTerre test = new TerrainTerre();

PlanteCactus p = new PlanteCactus();

CasePotager ca = new CasePotager(test, m, p);

Console.WriteLine(p.Morte);

p.TourPlante();

Console.WriteLine(p.Morte);

Console.WriteLine(p.Age);

Console.WriteLine(p.EtatSante);

p.TourPlante();

Console.WriteLine("morte" +p.Morte);

Console.WriteLine(p.Age);

Console.WriteLine("sante" +p.EtatSante);

Console.WriteLine("dessechee" +p.Dessechee);

Console.WriteLine("temp" + p.BonneTemp);

Console.WriteLine(p.Age);
```

Gestion de projet

Organisation de l'équipe :

- Bourguignon Louis-Ferdinand
- Desfarges Yann

Nous nous sommes répartis les tâches de manière assez aléatoire, nous avons toujours plus ou moins travaillé sûr tout en fonction de ce qui nous inspirait le plus. Il aurait peut-être fallu mieux se répartir les tâches, mais la plupart des classes étant reliées, nous avions aussi peur qu'en changer une trop d'un coup sans trop se concerter nous apporterait des problèmes. Si notre structure de classe avait été bien faite dès le début cela aurait pu énormément nous aider mais ce n'était pas le cas.

Planning :

Premières semaines : Durant les premières semaines, nous avons créé les premières classes de bases, Terrain, Plante, Potager, CasePotager etc.. Nous avons réfléchi à comment nous voulions structurer nos classes à l'écrit (même si nous l'avons fait de la mauvaise manière au départ, on a tout de même passer du temps dessus durant les premières semaines en plus de coder les premières classes (semaine 1 + semaine 2)

Durant les deux semaines suivantes, nous avons été assez pris et avons donc coder sur le code uniquement lors des séances dédiées. Nous avons continué à coder les classes principales mais sommes restés concentrés sur la classe Jeu qui regroupait à ce moment énormément de choses. C'est à la fin de ces deux semaines que nous nous sommes rendu compte que nous avions un problème avec notre code et de reprendre les choses à la racine.

Semaines finales : Réflexion nouvelle sur l'organisation, réalisation du mode urgence, de l'affichage et des fonctionnalités principales du jeu.

Bilan critique / avis

Difficultés rencontrées :

L'un des principaux obstacles que nous avons rencontrés au cours de ce projet a été lié à une mauvaise compréhension des concepts fondamentaux de la programmation orientée objet, en particulier la gestion des classes et de l'héritage. Au départ, nous avons centralisé une grande partie du code dans la classe principale **Jeu**, pensant à tort qu'une classe devait regrouper un maximum de fonctionnalités, comme une sorte de « grosse variable ». Cette approche s'est rapidement révélée problématique : bien que certaines fonctionnalités de base (comme planter des plantes ou afficher le potager) fonctionnaient, nous n'arrivions pas à faire évoluer le projet de façon cohérente.

C'est en reprenant des exercices de travaux dirigés que nous avons pris conscience de l'ampleur du problème : notre structure n'était pas viable, et nous avons donc dû, à un moment donné, reprendre presque tout le projet depuis la base. Nous avions cependant toujours les différentes fonctions qui ne demandaient qu'une légère retouche pour s'implémenter des le programme.

Cette reprise n'a pas été simple. Il ne s'agissait pas simplement de corriger quelques éléments, mais de repenser entièrement la hiérarchie des classes et les interactions entre les différents objets du jeu. Nous ne voulions pas pour autant jeter tout le travail déjà accompli. Nous avons donc pris le temps de redessiner sur papier une nouvelle architecture avant de commencer à coder à nouveau. Cette étape a été longue et complexe, d'autant plus qu'elle impliquait d'articuler une quinzaine de classes déjà partiellement développées.

Enfin, la gestion du temps a été un autre défi. Au début, nous pensions pouvoir finaliser les fonctionnalités principales assez rapidement. Mais la découverte de nos erreurs de conception a bouleversé notre planning, rendant la suite du développement bien plus chaotique que prévu.

Conclusion

Cependant malgré les fortes et nombreuses difficultés que nous avons rencontrées, nous avons réussi à les surmonter et à réaliser cette simulation de potager de la meilleure manière possible selon nous, et nous sommes satisfaits du travail accompli, le jeu est jouable, l'affichage est propre (sans être trop complexe) et permet tout de même une utilisation agréable.

Nous restons tout de même sur une note de frustration étant donné ou nous en sommes arrivés. Nous sommes persuadés qu'avec plus de temps nous aurions pu créer de nombreuses autres fonctionnalités et améliorer celles déjà existantes. (Par exemple, le mode urgence qui ne pose qu'une seule question, ou ajouter un inventaire qui donne un certain nombre de graines à planter et qui permet de récolter les fruits puis les afficher etc…). On a aussi supprimé certaines classes comme les classes maladies… (qui n'apportaient rien car elles n'étaient pas fonctionnelles mais cela reste assez frustrant.)

Desfarges Yann

Bourguignon Louis-Ferdinand

ANNEXE

Classes principales :

Plante (abstraite) : Modélise les caractéristiques communes à toutes les plantes comme son nom ou ses besoins en eau.

Plantes spécialisées : Il nous a paru plus facile de créer pour chaque plante (Tomate, Chou, Tulipe, Cactus) une classe héritière de **Plante** en y redéfinissant ses comportements (quelle est sa température préférée, son besoin en eau etc…). De plus, nous avons aussi créé un classe **PlanteBloque** elle aussi héritière de **Plante** qui intervient lorsqu'une plante est brûlée par les maîtres du feu pour empêcher le joueur de planter une autre plante.

Terrain (abstraite) : Cette classe permet la définition des terrains avec des caractéristiques comme l'humidité de son sol, sa température. De plus, la météo (lorsqu'il pleut par exemple) change les caractéristiques du terrain, autrement dit la météo agit sur le terrain et non directement sur les plantes.

Terrain (abstraite) : Nous avons créé trois classes dérivées de **Terrain** : Sable, Terre, Argile. Chaque type a des propriétés influant sur les plantes, chaque terrain a des caractéristiques spécifiques (humidité du sol, qualité du sol (qui ne sert pas ici car certaines idées que nous avions n'ont pas pu être implémentées)).

Meteo : Classe représentant les conditions climatiques d'un tour. Elle modifie la température,le taux d'humidité des terrains.

Meteo spécialisée : Nous avons créé ici des météos suivant les différentes nations représentées (Feu, Eau, et Terre), chacune ayant une météo spécifique avec par exemple plus de chaleur pour la nation du Feu.

CasePotager permet de réunir les classes terrains et plantes, en effet, chaque case du potager peut être associée à un terrain et une plante. Cette classe permet l'affichage de tous ces éléments

Potager : Cette classe permet de générer automatiquement le potager (en générant une grille de CasePotager). Cela permet de gérer facilement tous les éléments du potager. Sa méthode pour jouer un tour (jouer un tour du potager et non pas un tour du jeu) regroupe tous les éléments principaux pour planter une plante dans le potager (potentiellement avec

plus de temps nous aurions aimé pouvoir récolter les plantes, ces options auraient été gérées ici aussi).

Jeu : Classe essentielle pour le lancement de la partie (génération de la météo), la gestion des tours (semaines, actualisation de la météo, mode urgence) et la fin de la partie. C'est la classe qui vient rassembler toutes les autres classes.

Question : classe qui permet de créer une nouvelle question, avec ses différentes réponses et un index pour sa bonne réponse.

Questionnaire : classe qui regroupe les 15 questions et qui permet de les poser aléatoirement.