

Projet ENSemenC - Programmation avancée



I. Présentation de l'univers et déroulement d'une partie

L'objectif de ce projet de programmation avancée était de créer un simulateur de potager avec un certain nombre de fonctionnalités obligatoires à implémenter.

Pour notre jeu, l'utilisateur a le choix entre 7 pays : Égypte, Bangladesh, France, Maroc, Mexique, Chine et "ChezBéa", une terre imaginaire idyllique.

Il existe 3 types de terrains, sable, argile et terre, chaque pays peut disposer de 1 à 2 de ces types de terrains excepté ChezBéa qui comporte les 3 types de terrains.

De plus, une séquence de jeu se déroule sur 1 semaine et l'année s'écoule comme une année réelle. Ainsi chaque pays possède également une météo qui varie en fonction des saisons mais qui est la même pour tous les terrains (voir prochain paragraphe) et ne peut accéder qu'aux plantes correspondantes aux types de terrains disponibles sur ce dernier.

Chaque plante possède ses propres caractéristiques, notamment la place qu'elle prend, l'écart donc elle a besoin pour pousser (par exemple pour 1, il devra y avoir une case vide tout autour de la plante pour qu'elle pousse correctement, pas implémenté), la vitesse de croissance et d'autres que vous pourrez découvrir en jouant au jeu. De façon générale, chaque plante a un certain besoin en eau et luminosité qui devra être respectée pour qu'elle pousse au maximum de sa vitesse de croissance, si certaines conditions ne sont pas respectées, la plante pousse plus lentement, voire pas du tout, voire elle mourra.

Au démarrage du jeu, l'utilisateur choisit son pays ainsi que le nombre de terrains sur lesquels il souhaite jouer (entre 1 et 5 parmi les terrains disponibles dans son pays). Il peut également sélectionner la surface du terrain en m²(entre 1 et 5) et reçoit alors initialement une somme d'argent égale à la longueur d'un côté (exemple : s'il choisit un terrain de 5x5, il recevra 5 unités monétaire au départ). Il pourra ensuite investir cet argent dans un magasin pour acheter des semis, qu'il stockera dans un inventaire, dont le prix varie de 1 unité pour les moins rares à 20 pour les plus rares. Chaque type de terrain possède donc 5 plantes différentes ayant un niveau de rareté allant de 1 à 5. Il pourra également plus tard revendre ses semis s'il en a en trop, ou encore vendre les plantes qu'il aura fait pousser.

De manière générale, le joueur peut effectuer 5 actions par tour parmi la liste suivante :

- Désherber
- Pailler : ajoute 1 semaine de pousse à la plante
- Arroser: ajoute de l'eau à la parcelle
- Traiter : guérit une plante malade
- Semer : plante un semi disponible dans le stock
- Récolter : récolte une plante "arrivée à maturation"
- Aller au magasin : pour acheter/vendre des semis/plantes
- Installer une serre : protège une parcelle des intempéries (nous n'avons pas eu le temps de l'implémenter)

- Installer une barrière : protège la parcelle des obstacles (nous n'avons pas eu le temps de l'implémenter)
- Installer pare-soleil : limite la luminosité d'une parcelle (nous n'avons pas eu le temps de l'implémenter)
- Finir le tour (ne compte pas comme une action)

Comment aborder rapidement précédemment, le joueur peut faire face à des obstacles pouvant être nocifs pour son potager. Ces plantes peuvent notamment être touchées par une maladie, attaquées par des pucerons, escargots, rongeurs ou oiseaux et autres qui auront chacun une certaine probabilité de tuer une plante. Il peut aussi parfois être aidé par des coccinelles qui ont 50 % de chance de tuer un puceron.

Lorsqu'il est attaqué ou qu'un intempérie est présente, le mode urgence est déclenché. Le joueur est dirigé vers l'endroit où se trouve l'urgence et il a la possibilité d'agir en conséquence pour la contrer.

II. Modélisation objet

Program.cs

Pour ce qui est de la modélisation de notre jeu. Nous avons commencé par la création de notre jardin, que l'on place dans le pays souhaité. Le pays constitue l'élément qui décide du type des terrains de notre jardin. Une fois le pays choisi, le joueur choisit le nombre de terrain(s) qu'il souhaite dans son jardin (entre 1 et 5). Si il se trouve dans un pays qui possède plusieurs types de sols, il doit choisir le type de sol de chaque terrain. Ensuite, il choisit la dimension de son/ses terrain(s), de 1x1 à 5x5. Une fois tout cela fait, le joueur choisit la saison à laquelle il commence à jouer (par défaut hiver).

Ensuite, la simulation peut commencer avec un appel contenant toutes les données nécessaires :

```
Simulation simulation = new Simulation(nbTerrains, typesDeTerrains,
dimension, saison!, paysChoisi);
```

Meteo.cs

Toutes les semaines, une météo aléatoire dépendant de la saison est générée. Elle donne les précipitations, la température, et la luminosité qui vont être appliquées au jardin, donc au(x) terrain(s).

Elle diffère selon les saisons afin d'avoir un rendu assez réaliste.

```
switch (saison)
{
    case "Hiver":
```

```

        Precipitations = rng.Next(20, 100);
        Temperature = (float)(rng.NextDouble() * 5.0 - 2.0);
// -2 à 3°C
        Luminosite = rng.Next(1, 4); // 1 à 3
        break;

    case "Printemps":
        Precipitations = rng.Next(30, 80);
        Temperature = (float)(rng.NextDouble() * 10.0 + 5.0);
// 5 à 15°C
        Luminosite = rng.Next(3, 7); // 3 à 6
        break;

    case "Été":
        Precipitations = rng.Next(0, 60);
        Temperature = (float)(rng.NextDouble() * 10.0 + 20.0);
// 20 à 30°C
        Luminosite = rng.Next(6, 10); // 6 à 9
        break;

    case "Automne":
        Precipitations = rng.Next(40, 90);
        Temperature = (float)(rng.NextDouble() * 8.0 + 5.0); //
5 à 13°C
        Luminosite = rng.Next(2, 6); // 2 à 5
        break;
    }
}

```

Simulation.cs

Lorsque la simulation est lancée, on récupère la semaine de jeu en fonction de la saison choisie (`ObtenirSemaineDeDepart(string saison)`).

Ensuite la météo est générée aléatoirement avec la classe présentée précédemment `Meteo.cs`. Elle est présentée au joueur sous forme de bulletin météo (`AfficherBulletinMeteo()`). Lorsque la météo a un comportement inhabituel (fortes précipitations, forte chaleur ...) des alertes sont présentées au joueur dans la bulletin météo de la semaine :

```

AfficherAlerteCanicule()
AfficherAlerteGel()
AfficherAlertePluie()
AfficherAlerteLumiFaible()
AfficherAlerteSoleil()

```

AfficherAlerteSecheresse()

Nous avons également une fonction (`RappelMeteo()`) qui rappelle rapidement la météo quand le tour est lancé.

Ensuite, les terrains sont affichés 1 à 1, et nous avons 5 actions par semaine par terrain à réaliser.

Actions restantes : 5	Actions disponibles :
<ul style="list-style-type: none">1. Désherber3. Arroser5. Semer7. #Installer serre9. #Installer pare-soleil11. Voir l'inventaire13. Voir détails essentiels plantes # = méthode pas encore développée	<ul style="list-style-type: none">2. Pailler4. #Traiter6. Récolter8. #Installer barrière10. Aller au magasin12. Voir détails plantes0. Finir le tour de jeu

La méthode correspondante est appelée selon l'action, et le compteur d'actions restantes diminue lorsqu'une action est effectuée.

Terrain.cs

C'est la classe la plus dense de notre projet. Elle va contenir une grille de parcelles (voir `Parcelle.cs`), qui contiennent les informations les plus importantes. Les terrains sont initialement vides, et on y applique ensuite la météo hebdomadaire pour actualiser les données de toutes les parcelles. Une méthode `TrouverOrigine(int x, int y)` est créée afin de gérer les plantes 2x2. Les plantes 2x2 sont gérées par les caractéristiques de la parcelle en haut à gauche. Donc pour les 3 autres parcelles, on utilise la fonction citée ci-dessus pour se référer à la case d'origine de la plante.

Ensuite, nous avons les méthodes correspondants aux actions disponibles (voir code et tests).

Nous avons ensuite le magasin, qui propose d'acheter des semis, d'en vendre, ou de vendre les récoltes. Chacune de ces propositions correspond à une méthode appropriée.

Pour finir, nous avons `Afficher()` qui permet d'afficher correctement le terrain, avec un affichage agréable.

Parcelle.cs

Ces parcelles contiennent les plantes, les informations de luminosité et d'eau qui seront plus tard appliquées aux plantes pour effectuer les tests de croissance.

La ligne du haut d'une parcelle contient l'initiale de la plante plantée, et un émoji correspondant au stade de croissance de la plante.

La ligne du bas présente la luminosité en bas à gauche (de 0 à 9) et à droite nous avons la quantité d'eau de la parcelle (de 0 à 9).

C'est dans cette classe que nous avons la méthode `AppliquerCroissance` (`Meteo meteo`) qui est responsable de la gestion de la croissance des plantes, en fonction des données de la météo et de la parcelle. Sur les 4 conditions (eau, lumière, température et espacement qui n'est pas implémenté) :

- Si aucune n'est respectée : la plante meurt
- Si une ou deux sont respectées : la plante ne pousse pas
- Si trois sont respectées : la plante pousse à la moitié de sa vitesse de croissance
- Si quatre sont respectées : la plante pousse à pleine vitesse.

Plante.cs

Cette classe permet de créer chaque instance de plantes. Elle contient toutes leurs informations, ainsi que les émoji qui traduisent leur stade de croissance (ou leur mort).

```
0 => "🌱",
1 => "🌿",
2 => "🌳",
3 => "🌼",
4 => "💀",
```

CataloguePlantes.cs

Le catalogue de plantes permet de créer nos 15 plantes (5 par type de terrain), avec toutes leurs données (nom, taille, rareté, prix d'achat, prix de vente, besoin en eau, température, saison de semis ...).

Inventaire.cs

L'inventaire est essentiel, et répertorie :

- l'argent du joueur
- les semis qu'il possède
- les plantes qu'il a récoltées.

Nous avons donc les méthodes qui permettent de gérer les stocks de l'inventaire du joueur, pour y ajouter ou y enlever des éléments.

III. Tests réalisés

Dans cette partie, nous allons vous proposer quelques captures d'écran pour donner des exemples d'une partie.

 France : BULLETIN MÉTÉO DE LA SEMAINE 

Semaine : 013
Saison : Printemps
Température : 14,1°C
Précipitations : 050 mm
Luminosité : 5/9

Pas d'alerte météo cette semaine. Tout va bien dans le jardin !

Voici un exemple de bulletin météo, il n'y a pas d'alerte ici, rien à signaler.

	0	1	2
0	5 	5 	5 
1	5 	5 	5 
2	5 	5 	5 

Voici un exemple de terrain 3x3 qui est vide. L'ensoleillement est de 5/9 et chaque parcelle possède 3 L/m² d'eau.

	0	1	2
0	T  5 	T 5 	5 
1	T 5 	T 5 	5 
2	5  5 	5  5 	5  5 

Actions restantes : 5
Actions disponibles :

- 1. Désherber
- 3. Arroser
- 5. Semer
- 7. #Installer serre
- 9. #Installer pare-soleil
- 11. Voir l'inventaire
- 13. Voir détails essentiels plantes

= méthode pas encore développée

- 2. Pailler
- 4. #Traiter
- 6. Récolter
- 8. #Installer barrière
- 10. Aller au magasin
- 12. Voir détails plantes
- 0. Finir le tour de jeu

►Choix : 6
 Vous souhaitez récolter une parcelle.
 x : 0
 y : 0

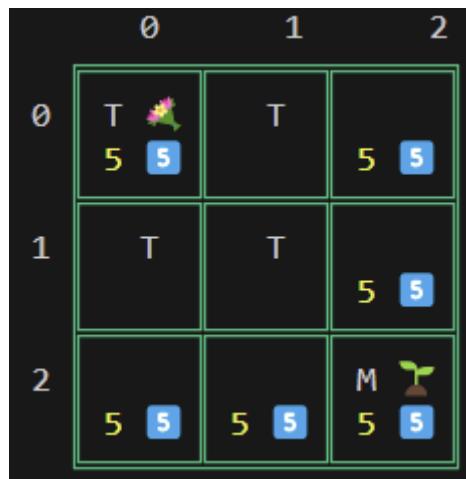
Ci-dessus, un exemple de récolte. Le plant de tomate 2x2 est prêt à être récolté.



Voici le résultat une fois la récolte effectuée.



Voici la présentation du magasin, avec les options proposées.



Voici un exemple de terrain, on a un plant de tomate 2x2 qui est poussé, et un plant de menthe tout jeune.

☀️ France : BULLETIN MÉTÉO DE LA SEMAINE ☀️

Semaine : 019
Saison : Printemps
Température : 11,6°C
Précipitations : 078 mm
Luminosité : 5/9

🌧️ PLUIE ABONDANTE 🌧️

Attendez-vous à de fortes précipitations cette semaine !
Vos plantes auront largement assez d'eau. ☀️☀️

Voici un bulletin météo avec une alerte.

🌿 Détails de la plante : Menthe

Nom: Menthe
 Rareté: 1
 Terrain Préféré: Terre
 Dimensions (lxL): 1x1
 Ecart Nécessaire: 0
 Saisons de Semis: Printemps, Automne
 Besoin en Eau: 3 L/m²/semaine
 Besoin en Luminosité: 100%
 Température Minimum: 10°C
 Température Maximum: 25°C

Appuyez sur une touche pour revenir au catalogue...

Voici ci-dessus un exemple de détails de la plante menthe pour réussir à la faire pousser correctement.

Lien du Github pour voir le code :

<https://github.com/PROGAV-PRJ25/projet-ensemenc-chambron-monier.git>

IV. Gestion de projet

Dans un premier temps, nous avons pris le temps de découvrir et comprendre correctement le sujet. Nous avons ensuite réfléchi ensemble à toutes les fonctionnalités et détails que nous voulions implémenter.

Nous avons donc choisi de prendre les premières séances de travail à créer un document récapitulatif de toutes les fonctionnalités que nous voulions intégrer à notre jeu.

Ensuite, nous avons commencé par bien implémenter les bases du monde (la sélection du pays, la création des terrains, le système de météo..), puis les différentes plantes disponibles. Nous avons continué avec le système de plantation ainsi que les actions de base (pailler, arroser..) et en parallèle, le magasin et l'inventaire pour gérer l'achat et la vente de semis/plantes. Enfin, nous avons terminé par les malus, les actions pour contrer les malus et le mode urgence. Voici ci-dessous une matrice d'implication qui résume les tâches de chacun pour ce projet.

MATRICE D'IMPLICATION : PROJET PROGRAMMATION AVANCÉE			
	Bienvenue chez Béa	date:	
MEMBRES DE L'ÉQUIPE: <u>MONIER Rémi</u> , <u>CHAMBRON Jérémie</u>			
n°	Tâche/Fonctionnalité/Fonction	Nom du/des Codeurs	Pourcentage de participation si tâche partagée
1	Conception de l'univers du projet	<u>MONIER/CHAMBRON</u>	50/50
2	Création du monde	<u>MONIER/CHAMBRON</u>	50/50
3	Implémentation des plantes	<u>CHAMBRON</u>	
4	Système de plantation et actions de base	<u>MONIER/CHAMBRON</u>	25/75
5	Création et liaison du magasin et de l'inventaire	<u>MONIER</u>	
6	Implémentation des malus et protection	<u>MONIER/CHAMBRON</u>	75/25
7	Implémentation du mode urgence	<u>MONIER/CHAMBRON</u>	50/50
8	Rédaction du rapport	<u>MONIER/CHAMBRON</u>	50/50
9			

V. Bilan du projet

Ce projet a été plutôt enrichissant et amusant. Il est toujours plus intéressant de travailler sur un projet ludique que l'on peut voir fonctionner et évoluer au fil du temps. Ayant déjà effectué un projet dans le même style au semestre 5, nous avons essayé de ne pas reproduire au maximum les erreurs que nous avions commises, chacun dans nos groupes respectifs, notamment au niveau de l'organisation et de la planification du projet. Aussi, bien que nous

soyons partis sur une base claire pour notre jeu, nous avons su nous adapter et apporter les modifications nécessaires au fil du projet, notamment sur des fonctionnalités trop ambitieuses que nous souhaitons implémenter au début. De façon générale, ce projet nous a permis d'utiliser les compétences acquises au cours de l'année et même de les développer puisque ce projet a été mené en semi-liberté (bien que nous avions des consignes précises à respecter).

Pour améliorer le rendu et aller plus loin dans ce projet nous aurions voulu ajouter plusieurs fonctionnalités. Nous y avons réfléchi au début du projet, et n'avons pas eu le temps de toutes les implémenter, plus de temps nous aurait permis d'en ajouter certaines. Tout d'abord nous voulions ajouter un système de maladies, qui auraient pu être soignées en traitant les plantes. Ces maladies auraient dû se propager. Nous voulions aussi gérer l'espacement entre les plantes qui freinerait la croissance des plantes. Chaque case du terrain aurait une probabilité de 5% de posséder une espèce envahissante, espèces qui ont aussi une certaine probabilité d'apparaître toutes les semaines. Les espèces envahissantes sont des plantes qui ne sont pas cultivables, on ne peut donc les acheter. Elles ne peuvent apparaître que sur des cases vides et le joueur est notifié lors de leur apparition, il a alors la possibilité de l'enlever, s'il ne le fait pas l'espèce envahissante évolue de semaine en semaine. Aussi, nous voulions ajouter un système de malus qui attaquent nos plantes, comme des taupes, des pucerons, ou des rongeurs. Nous aurions également aimé ajouter les "bonnes fées" qui nous auraient aidé à sauver nos plantes (coccinelles pour tuer les pucerons). Nous voulions également ajouter le mode "urgence" qui nous aurait permis de lancer une sorte de "mini-jeu" ou nous aurions pu chasser une taupe en temps réel. Pour continuer, nous voulions introduire des événements, comme un système de "flemme" qui serait aléatoire et qui limite le nombre d'actions hebdomadaires, ou encore une probabilité de tomber dans notre potager ce qui aurait pu casser des plantes, ou même encore de la faim, qui nous force à manger nos plantes et qui nous empêche par conséquent de les récolter. Et pour finir, nous avions envisagé un système de compétences pour le joueur comme dans un RPG qui lui aurait permis par exemple de faire pousser les plantes plus vite, ou de les vendre plus chères.