Evan Elbaz Boyer Simon

Projet de Programmation avancée :
Sujet : ENSemenC

Le Projet Potager Kenya

Sommaire :

I/ Univers du jeu et ses spécificités

I/ a) Choix du pays

Nous avons d'abord cherché à trouver un pays qui changerait de nos codes habituels qui correspondent à notre vision du potager, car celà paraissait plus intéressant et nous permettait de découvrir d'autre manière de fonctionner. Nous sommes rapidement partis sur l'Afrique qui, contrairement à ce que l'on pourrait penser, dispose d'une très grande diversité territoriale. Un pays qui symbolise cet aspect est le Kenya, disposant de plus de 4 biomes très importants, changeant considérablement les possibilités de culture pour chacun d'eux. Il se trouve aussi que nous savions grâce à nos connaissances personnelles que le Kenya fait partie des premiers exportateurs au monde de roses (1er en 2022). Notre potager allait donc se baser au Kenya, il ne restait plus qu'à préciser les biomes, et le plus important, trouver quels produits locaux y cultiver.

I/ b) Choix et description des biomes et des plantes

Notre choix s'est orienté vers les 4 biomes prépondérants au sein du pays : Une zone montagneuse, une savane, une zone de forêt tropicale humide et une zone côtière.

Voici les produits les plus cultivés dans ces terres :

Savane (zones arides et semi-arides) : Haricots, Pois chiches, Mil, Sorgho, Patates douces, Igname, Manioc, Melons, Mangues, Papayes, Bananes, Chou, Kale, Carottes, Poivrons, Oignons, Acacia, Baobab.

Forêt tropicale humide : Haricots, Soja, Maïs, Riz, Patates douces, Igname, Manioc, Mangues, Avocats, Oranges, Papayes, Ananas, Fruit de la passion, Bananes, Chou, Kale, Brocoli, Tomates, Poivrons, Épinards, Haricots verts, Safou, Cocotiers.

Zones montagneuses : Haricots, Lentilles, Pois, Maïs, Blé, Orge, Pommes de terre, Pommes, Poires, Avocats, Oranges, Papayes, Bananes, Chou, Brocoli, Carottes, Tomates, Poivrons, Oignons, Épinards, Haricots verts, Eucalyptus, Pin de montagne.

Zone côtière : Haricots, Maïs, Riz, Manioc, Papayes, Mangues, Ananas, Fruit de la passion, Bananes, Tomates, Poivrons, Oignons, Chou, Kale, Safou, Cocotiers, Palmiers à huile, Mangroves.

On peut constater que quelques produits se retrouvent dans plusieurs biomes et sont tout de même universel bien que les conditions d'épanouissement sont très diverses : les haricots (rouges), le chou kale. On a donc trouvé intéressant de choisir les plus uniques aux biomes plutôt que des produits qui auraient pu se trouver dans n'importe lequel. On a donc commencé à structurer notre potager, pour savoir quels produits allait être choisis. Nous avons opté pour l'élaboration de quatre terrains, chacun représentant un biome et comprenant une météo propre :

Zone côtière : altitude moyenne très basse (10 m), type de sol sableux et alluvial, température moyenne de 28°C constante le long de l'année, humidité moyenne de 85% (proximité de l'Océan Indien) et une luminosité moyenne de 90% (ensoleillement intense excepté durant la mousson).

Le parc national de Kora

Zone montagneuse : Altitude moyenne très élevée (2500m), avec des andosols volcaniques comme type de sols (riche en minéraux), une température moyenne de 12°C, qui devient de plus en plus fraîche au fur et à mesure que l'altitude augmente, une humidité moyenne de 75% grâce à l'altitude et enfin une luminosité moyenne de 70%, qui peut varier selon la couverture nuageuse (et donc, de l'altitude)

Le parc national du Mont Kenya

Forêt tropicale : altitude moyenne plutôt haute (1500m) avec pour sol un sol ferrallitiques, une température moyenne de 20°C, une humidité moyenne de 67% et enfin une luminosité moyenne de 80% environ, mais le couvert les arbres couvrent globalement la plupart de la lumière.

La réserve nationale de la forêt de Kakamega

Savane : altitude moyenne haute (1500m) avec des sols ferrallitiques, , une température moyenne de 27°C, une humidité plutôt basse de 55% et enfin une luminosité moyenne de 85%.

La réserve nationale du Masai Mara

Après avoir bien défini ces biomes, place aux plantes que nous avons sélectionnées pour notre potager. Nous avons commencé par trier les plantes cultivables selon le biome par catégories : légumineuses, fruits, céréales, et autres (bois, cactus, etc…).

Pour le biome **savane**, c'est le Sorgho que nous avons sélectionné comme céréale, représenté par : 🌿, la mangue comme fruit : 🥭
et enfin le baobab en "autre" : 🌳

Pour le biome **forêt tropicale humide,** nous avons choisi l'avocat comme fruit : 🥑, le cocotier (🥥) et le safou (🍆) comme "autres".

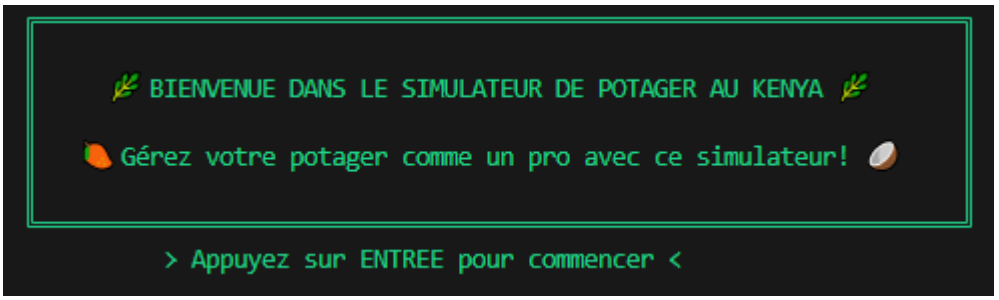
Pour le biome **zone montagneuse**, nous avons choisi la rose(🌹) comme "autres", la lentille (🟢) en légumineuse et le blé en céréale (🌾).

Pour le biome de la **zone côtière**, nous avons sélectionné l'ananas (🍍) et la tomate (🍅) comme fruits et le palmier à huile (🌴) comme "autres".

II/ Les possibilités des joueurs et le déroulement d'une partie

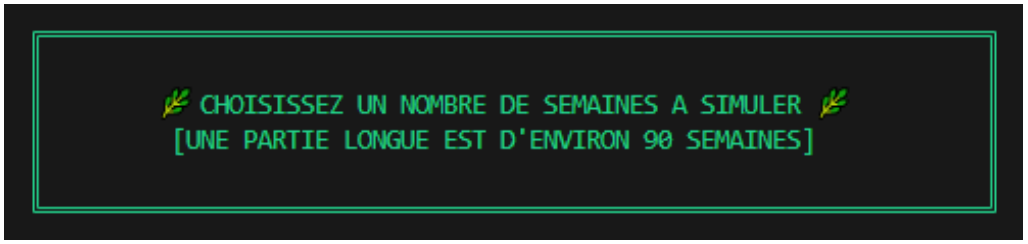
II/ a) La mise en contexte

Pour que le joueur soit réellement plongé dans une simulation de potager du Kenya, il est d'abord introduit avec un écran titre clair et parlant. Il se lance alors dans l'aventure peut-être pas si reposante que l'on ne pourrait le penser à première vue. Il appuie alors sur le bouton entrée et commence sa partie.



Il est alors confronté aux règles du jeu décrites ci-dessous :

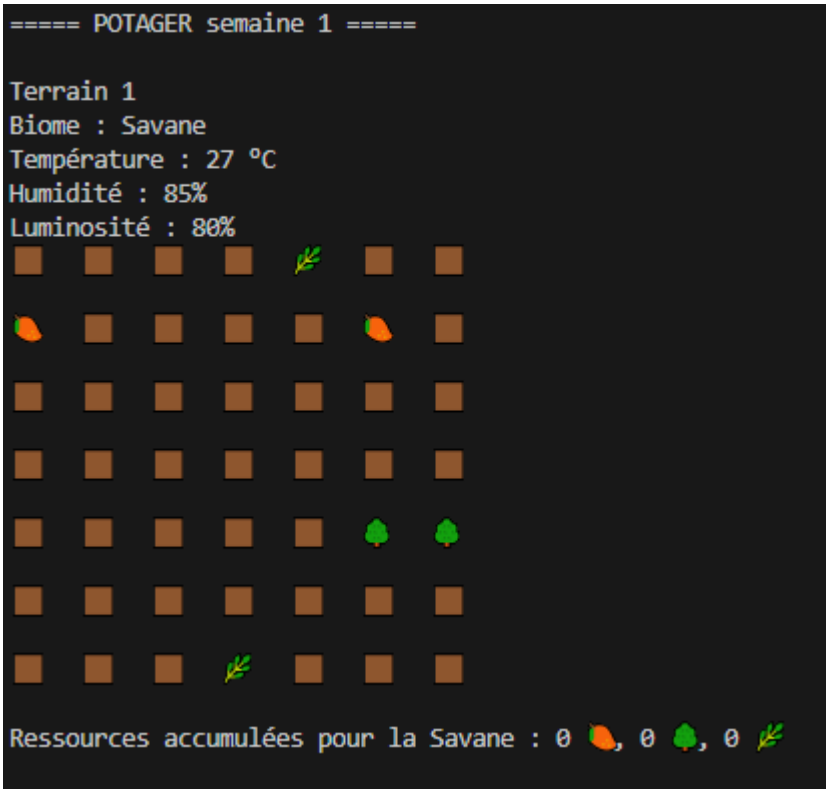
Après avoir une nouvelle fois appuyé sur le bouton Entrée, il choisi son nombre de semaine de simulation, via un nouvel écran affiché :



La simulation est alors lancée, pas de retour en arrière possible…

II/ b) Le début de l'aventure

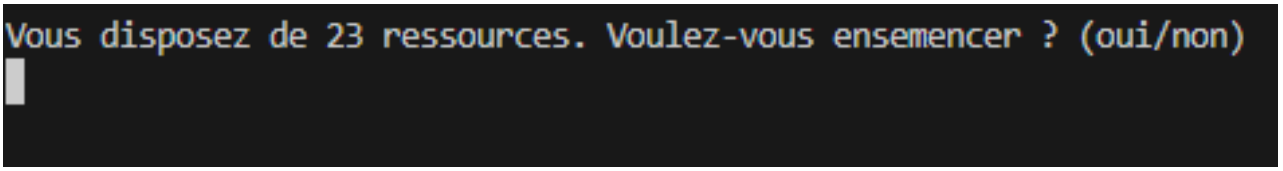
Le joueur n'est alors pas préparé aux difficultés qui l'attendent…
Il voit alors apparaître ses terrains dans lesquels des plantes sont déjà en train de pousser. Ces terrains représentent chacun un des quatres biomes décrits précédemment, et disposent de toutes leurs caractéristiques propres (luminosité, humidité, température). Il a accès à ses terrains en scrollant dans la console, et peut remarquer que chaque plante diffère d'un terrain à l'autre.



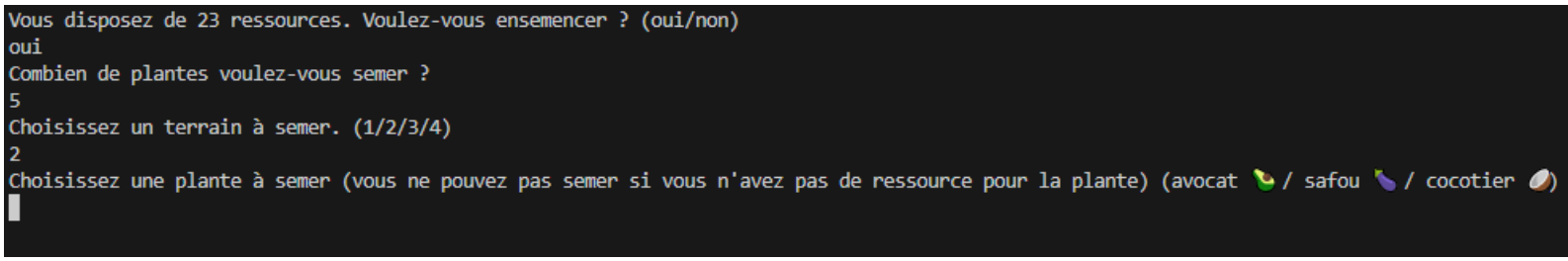
Le nombre de semaines est comptabilisé, c'est alors la première semaine de son aventure, et il ne dispose d'aucune ressource si ce n'est les quelques plantes éparpillées dans le jardin.
La première semaine se finit rapidement étant donné qu'aucun produit n'est récupérable, pas assez de temps ne s'est écoulé. Patience, ça pousse...

II/ c) Agrandir son empire !

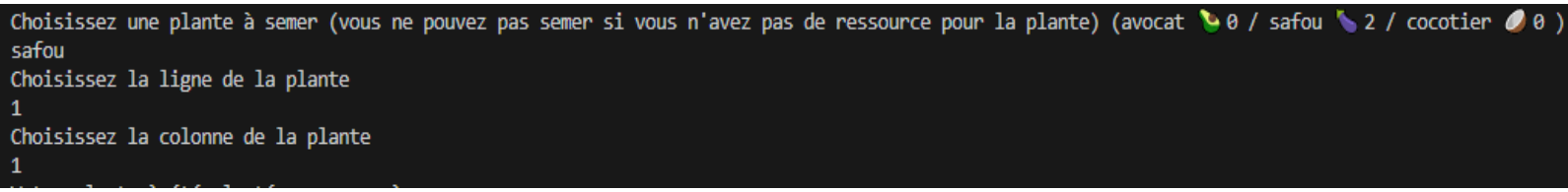
Les semaines passent, et les premières ressources apparaissent. La simulation propose alors d'ensemencer, seule manière d'agrandir son potager dans chaque biome :



Le Joueur peut alors choisir le nombre de ressources qu'il veut utiliser pour ensemencer, sachant que chaque ressource permet de semer la plante correspondante. Il choisit alors dans quel terrain il souhaite semer et la plante qu'il veut semer (uniquement celles disponibles dans le terrain choisi).



Cette première version ne nous convenait cependant pas étant donné que le nombre de ressources disponibles ne s'affiche que bien plus haut, avec les terrains, nous avons donc opté pour une option permettant d'afficher les ressources directement dans la question :

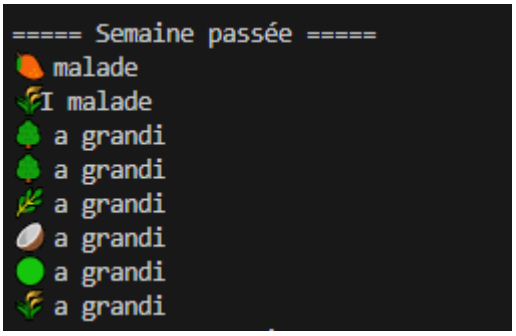


La plante est alors plantée avec succès ! (! Attention, vérifiez bien que l'emplacement est disponible sur le terrain correspondant)

Une nouvelle semaine passe alors…

II/ d) Les éléments perturbateurs et autres intéractions

Les premiers événements émouvants arrivent, autant positifs que négatifs d'ailleurs, vu qu'il est alors renseigné les récents événements : Les plantes grandissent, arrivent à maturité, ce qui est réjouissant ! Mais ces heureux événements sont vite gâchés par des maladies qui se propagent dans les biomes, et que rien n'est faisable pour sauver les plantes malades, qui dépérissent et ne grandissent plus. Elles n'attendent plus qu'une météo désastreuse pour passer l'arme à gauche.
En effet, les terrains sont très sensibles aux changements météorologiques trop violents, et des dégâts pourraient être observés.

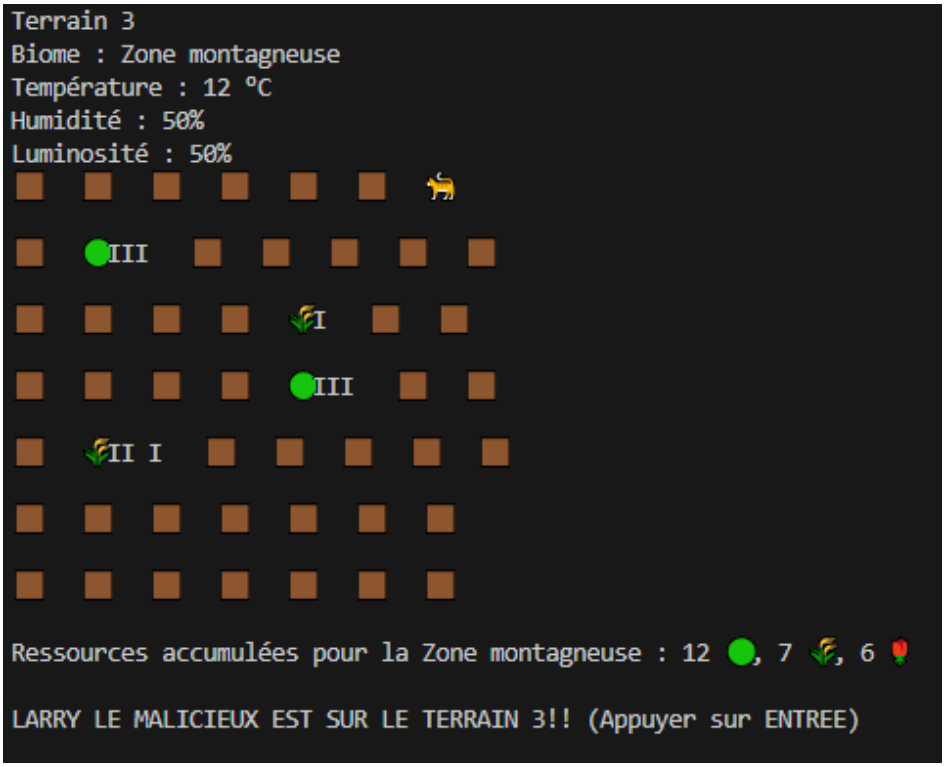


C'est bien la météo qui a rendu malade les plantes, chaque semaine, celle-ci varie, et les plantes n'ont pas forcément la force d'adaptation requise pour tenir bon…
Cependant, la météo n'est pas le seul danger qui terrorise votre potager. En effet, il apparaît de temps en temps une terrible petite bête nommée Chunk.

Chunk apparaît sur un terrain et mange une plante si vous n'arrivez pas à le faire fuir. Pour cela, vous pouvez choisir entre essayer de l'impressionner ou lui donner une ressource. Ces deux solutions n'ont pas 100% de réussite mais peuvent suffire à le faire fuir !

Si Chunk n'avait pas pris peur, il aurait sans doute dévoré le palmier ...

Un autre adversaire auquel le joueur doit faire face est Larry le Malicieux. Bien qu'il puisse paraître un simple chat ne représentant aucun danger, il ne faut surtout pas sous-estimer sa malice car celà pourrait handicaper grandement le joueur dans son potager...

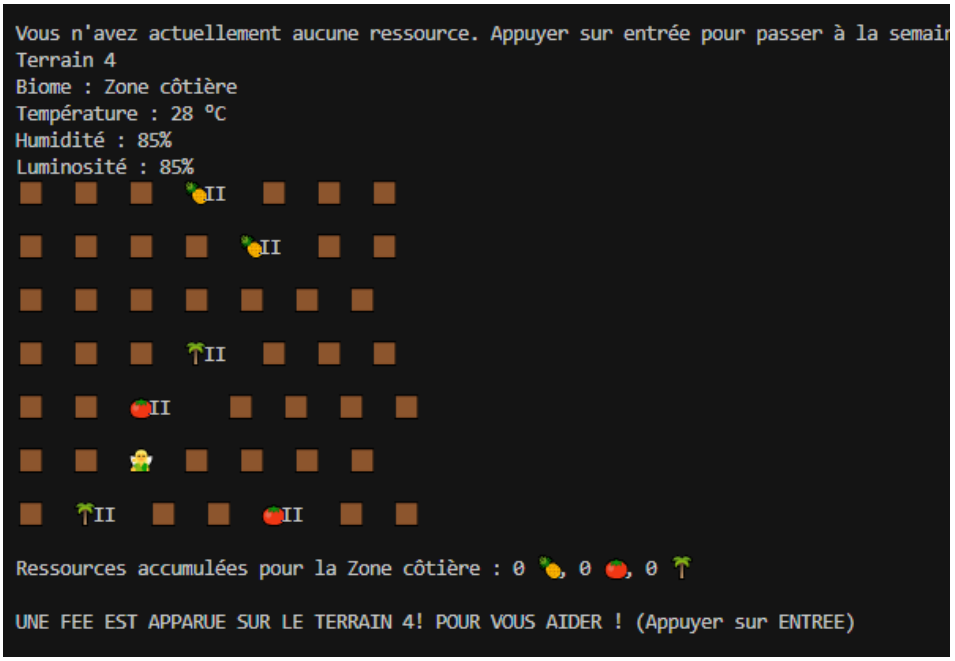


Rare représentation de Larry le Malicieux

Ici Larry le Malicieux vient d'apparaître, la simulation propose alors les même choix que pour Chunk :

Cependant, ici, il semblerait que Larry ne s'est pas laissé démonter par notre tentative de lui faire peur, et sa malice s'est manifestée en un piège qu'il a su déposer dans le potager, sur lequel aucune plante ne peut pousser. Jamais personne n'a su comment fonctionnaient ses pièges tant ils sont emplis de malice.

Cependant, il semble qu'il n'ait pas plus envie de propager sa malice dans notre potager, et est reparti perpétuer sa malice ailleurs.
Enfin, il est fréquent qu'une charmante rencontre se produise entre notre joueur et une fée, sur laquelle il pourra toujours compter pour apporter sa pierre à l'édifice.



En effet, celle-ci apprécie beaucoup nos potagers et se réjouit de nous aider ! Lorsqu'elle arrive, elle nous plante gratuitement une graine !!!

Enfin, elle disparaît pour errer dans d'autres potagers.

Soyez bien attentifs, il se peut qu'un des visiteurs vous réserve une petite surprise visuelle, mais il paraît que c'est très rare de les croiser sous ces formes…

En effet, il est conté que la reine des fées rend visite aux cultivateurs les plus acharnés. Si le destin vous sourit, vous pourriez l'apercevoir un jour…

La reine des fées n'a pas un effet différent des autres fées, si ce n'est qu'il paraît qu'elle porte chance...

III/ Gestion de projet

III/ a) Communication

Nous avons pu communiquer nos avancées et nos rendez-vous grâce aux outils de communications tels que whatsapp ou message

De plus, pour partager nos documents au fur et à mesure, un drive a été créé, et pour ce qui est du code, nous verrons dans la prochaine partie.

III/ b) Organisation de travail et répartition des tâches

Nous avons chacun participé à chaque étape du projet, mais notre implication a été variable selon nos compétences.
Pour ce qui est de notre organisation pour effectuer le code, nous nous sommes basés sur une stratégie liée à notre niveau. En effet, ayant commencé le code cette année, nous nous sommes arrangés pour coder la plupart du temps ensemble physiquement. Un de nous (Evan) avait plus de facilités et de détermination, ce qui a permis à celui qui était moins à l'aise sur le code (Simon) d'apprendre plus vite et de s'améliorer. Notre organisation a bien fonctionné car nous avons respecté nos prévisions et avons pu rendre notre travail fini sans trop de regrets.

III/ c) Gestion de Github

Malgré nos multiples tentatives, nous n'avons jamais vraiment réussi à travailler sur Github avant la dernière semaine. Cela est dû principalement à un manque d'expérience, nous étions impressionnés par la charge de travail que Github pouvait rajouter, alors qu'il s'est révélé être très simple de s'y retrouver une fois la prise en main effectuée.

Travail à faire	date de début	durée	date de fin	15/04/2025	17/04/2025	22/04/2025	27/04/2025	29/04/2025	02/05/2025	06/05/2025	11/05/2025	16/05/2025	23/05/2025
idée de l'univers	mardi 15 avril 2025	3 Jour(s)	vendredi 18 avril 2025										
Connaissances sur l'univers	jeudi 17 avril 2025	10 Jour(s)	dimanche 27 avril 2025										
Structure des Classes	lundi 21 avril 2025	25 Jour(s)	vendredi 16 mai 2025										
Code C#	mardi 22 avril 2025	31 Jour(s)	vendredi 23 mai 2025										
Prise en main de github	vendredi 2 mai 2025	6 Jour(s)	jeudi 8 mai 2025										
Constitution du Rapport	vendredi 16 mai 2025	7 Jour(s)	vendredi 23 mai 2025										

En vert, nous avons le travail qui a été réalisé dans le temps imparti qu'on s'était fixé, en rouge, les moments de travail imprévus. Chaque jour représente des sessions de 2 à 5 heures de travail, la plupart du temps en présentiel côte à côte. On peut voir que pour ce qui est de la création de l'univers nous avons respecté ce que nous avions prévu. Cependant la prise en main de Github a pris plus de temps que prévu ce qui nous a par exemple empêcher de travailler ensemble correctement pendant les vacances de Avril-Mai. En plus de celà, notre travail sur la structure des Classes a légèrement changé vers la fin car nous avons eu l'idée d'implémenter de nouvelles fonctionnalités interactives et originales. Globalement nous sommes satisfaits du respect du planning bien que Github aurait dû être un élément sur lequel nous aurions dû nous concentrer plus tôt.

IV/ Structure et modélisation des classes

Maintenant, intéressons-nous aux différentes classes composant le jeu :

Classe Terrain :

Un terrain est défini comme une grille de taille 7*7. Le terrain a aussi un ID : son numéro. Cette grille est un double tableau composé de string afin de pouvoir mettre des émojis représentant les plantes, la terre et les animaux. Chaque terrain contient une liste d'objets Plante. Un terrain possède aussi des ressources, un tableau de taille 3 contenant 3 nombres entiers, chacun modélisant le nombre d'une des trois ressources du terrain. La grille du terrain est initialisée à des cases de terre : "🟫".

Sur l'aspect physique du terrain, il est défini par les conditions climatiques dans lesquelles il évolue que cela soit son altitude, son humidité moyenne, sa température moyenne ou sa luminosité moyenne.

Pour ses méthodes : la méthode DebutJeu() permet de créer 6 plantes par terrain réparties aléatoirement. Chaque terrain ne pouvant posséder que 3 plantes différentes, les 6 plantes par terrain sont en réalité pas toutes différentes. *Ex : le terrain commence avec 2 mangues, 2 baobabs, et deux sorghos.* Le do while de cette méthode assure que les plantes soient créées sur un espace vierge. Le switch case permet de différencier les différentes plantes présentes sur le terrain selon son numéro. La méthode Afficher() permet d'afficher la grille du terrain avec un string "enclos" qui compile toutes ses cases. Cette méthode permet d'être appelée dans la classe Monde pour afficher le monde. Le override string ToString() fait la même chose et sera appelé dans le cas où le terrain est affiché seul : i.e lorsqu'un animal est dessus

Classe Plante:

Une plante est d'abord définie par différents attributs tels que son numéro de série en int, sa variété en string, son visuel (la manière dont la plante va apparaître sur le terrain : c'est donc un string). Elle a aussi des coordonnées en int (les attributs x et y). Ensuite, chaque plante a des conditions climatiques et des qualités telles que si elle est malade, comestible, sa température maximale et minimale qu'elle peut endurer, ainsi que son humidité minimale et maximale, de même pour sa luminosité. Dans son constructeur, on rentre simplement son numéro de série afin de définir sa variété et son aspect visuel.

Classe Monde:

La classe Monde reprend en quelque sorte les classes que l'on a abordées précédemment et les fait marcher ensemble. Un monde possède en attributs une liste d'objets Terrain. Les quatres terrains différents sont initialisés dans son constructeur et ajoutés à la liste de terrains "terrainsDuMonde". De même un monde possède un chaque animal en attribut : un seul Larry le malicieux existe dans le jeu, un seul Chunk existe existe aussi de même que la fée. Le monde possède aussi une temporalité et a donc un attribut semaine en int.

Pour les méthodes de la classe Monde, la méthode SemaineNiveauxRessources() parcourt les terrains du monde grâce à sa liste et les plantes du monde grâce à la liste de chaque terrain. Ensuite si la plante est en vie et n'est pas malade, elle monte de niveau (on peut voir son niveau avec la longueur de l'attribut "visuelPlante" de la plante) si elle a niveau inférieur à 3, sinon elle ajoute une ressource correspondant à son espèce au terrain sur lequel elle est. Les plantes ayant grandi sont annoncées. Cette méthode incrémente la semaine. Ensuite la méthode MaladieOuMort() permet de définir si une plante est malade (une chance sur sept et ne peut pas retomber malade si elle l'a été 3 semaines d'affilée). Ensuite la plante meurt et est retirée de la liste si ses conditions de vie ne sont pas respectées au minimum à 50%. Pour la retirer de la liste, on a créé une liste transitive "plantesMortes" pour les stocker avant de parcourir cette liste pour les retirer de la partie. Enfin la méthode PlanterNouvellePlante() est une méthode très longue mais peu technique. Elle demande au joueur s' il veut planter, combien de plantes il veut planter et sur quels terrains. Si le joueur n'a pas de ressources, on le lui indique et ne peut rien faire. Sinon à chaque question les réponses sont stockées dans des variables et pour être sûr d'avoir une réponse cohérente, nous avons fait des do while. Ensuite le joueur choisit la ligne et la colonne et une nouvelle plante est créée, ajoutée à la liste et la ressource associée du tableau "ressources" est décrémentée. Le code est inutilement long car nous n'avons pas réussi à faire le cas général c'est pour cela que nous l'avons décomposé en switch case et en if et else if. La mort des plantes est annoncée.

Classe Animal :

Un animal est défini par son espèce en string. Ses coordonnées et son visuel n'interviennent que dans sa méthode ApparitionAnimal().

Pour ce qui est de la méthode ApparitionAnimal() elle est définie en abstract donc il faut se pencher sur les classes dérivées pour mieux la comprendre.

Classe Chunk

Un int est défini de manière aléatoire pour savoir quand est-ce que Chunk apparaît. Une fois sur quatre, Chunk va apparaître et interagir avec le joueur. Trois fois sur quatre,

l'appel de la méthode ne fera strictement rien dans le jeu. Dans le cas sur quatre, Chunk apparaît sur un terrain aléatoire, sur une case avec une plante : grâce au do while. Ensuite le joueur choisit entre lui faire peur (1) et lui donner une ressource pour qu'il parte (2). Si il tape 1, Chunk a une chance sur 3 de partir, et deux chances sur 3 de rester, manger la plante et se déplacer. Si il tape 2, il choisit la plante, Chunk une chance sur 2 cette fois de partir, sinon il mange et se déplace. Il ira toujours vers le bas, à part s'il est au bord. Tous ces cas sont gérés avec des if et else if et des switch case lorsque l'on choisit les plantes associées au numéro du terrain à lui donner. Cette boucle se répète indéfiniment, jusqu'à ce que Chunk choisisse de partir.

Classe LarryLeMalicieux

Sa méthode marche exactement comme celle de Chunk, sauf qu'elle a une chance sur cinq de s'activer et que Larry le malicieux ne reste qu'un seul tour, il part qu'il ait réussi à poser son piège ou non. Il n'y a donc pas de boucle do while. Aussi, Larry le malicieux n'apparaît que sur des cases vierges

Classe Fee

Sur le même principe, la méthode de la fée n'apparaît qu'une fois sur trois. Elle apparaît sur une case vierge d'un terrain au hasard. Elle plante une fait apparaître une plante au hasard que ce terrain pourrait contenir. On utilise donc encore un foreach pour parcourir les terrains, et un simple switch case pour sélectionner une plante au hasard qui correspond au terrain ou elle est apparue. Il y a une chance sur 10 que son apparence change et qu'elle soit une reine des fées : cas if et else if.

Classe Simulation:

Une simulation possède en attributs : un monde, une météo et un écran titre à simuler.

Sa seule méthode est Simuler(). Elle permet de simuler une partie de jeu en faisant appel à la suite à chacune des méthodes que nous avons définies dans les autres classes. D'abord, elle affiche l'écran titre avec "AfficherEcranTitre()" puis les règles du jeu avec ReglesDuJeu(). Ensuite elle demande le nombre de semaines à simuler au joueur. Une boucle for répète le nombre de semaines choisies par le joueur un tour de jeu. Avant le DebutJeu() est appelé pour chaque terrain afin de positionner les plantes sur les terrains, et NouvelleMeteo(mondeSimule) pour créer la première météo du monde. Chaque tour de jeu commence par l'affichage du monde, puis l'annonce de la semaine, puis le récapitulatif des plantes mortes (MaladieOuMort()) et ayant grandi (SemaineNiveauxRessources()), puis le monde est affiché et on propose au joueur de planter des plantes (PlanterNouvellePlante()). Puis la fée a une chance d'apparaître (ApparitionAnimal()) et si on est à la semaine 3 ou plus, Chuck et Larry le malicieux ont une chance d'apparaître. Une fois les tours finis, la simulation s'arrête et un message de remerciement apparaît.

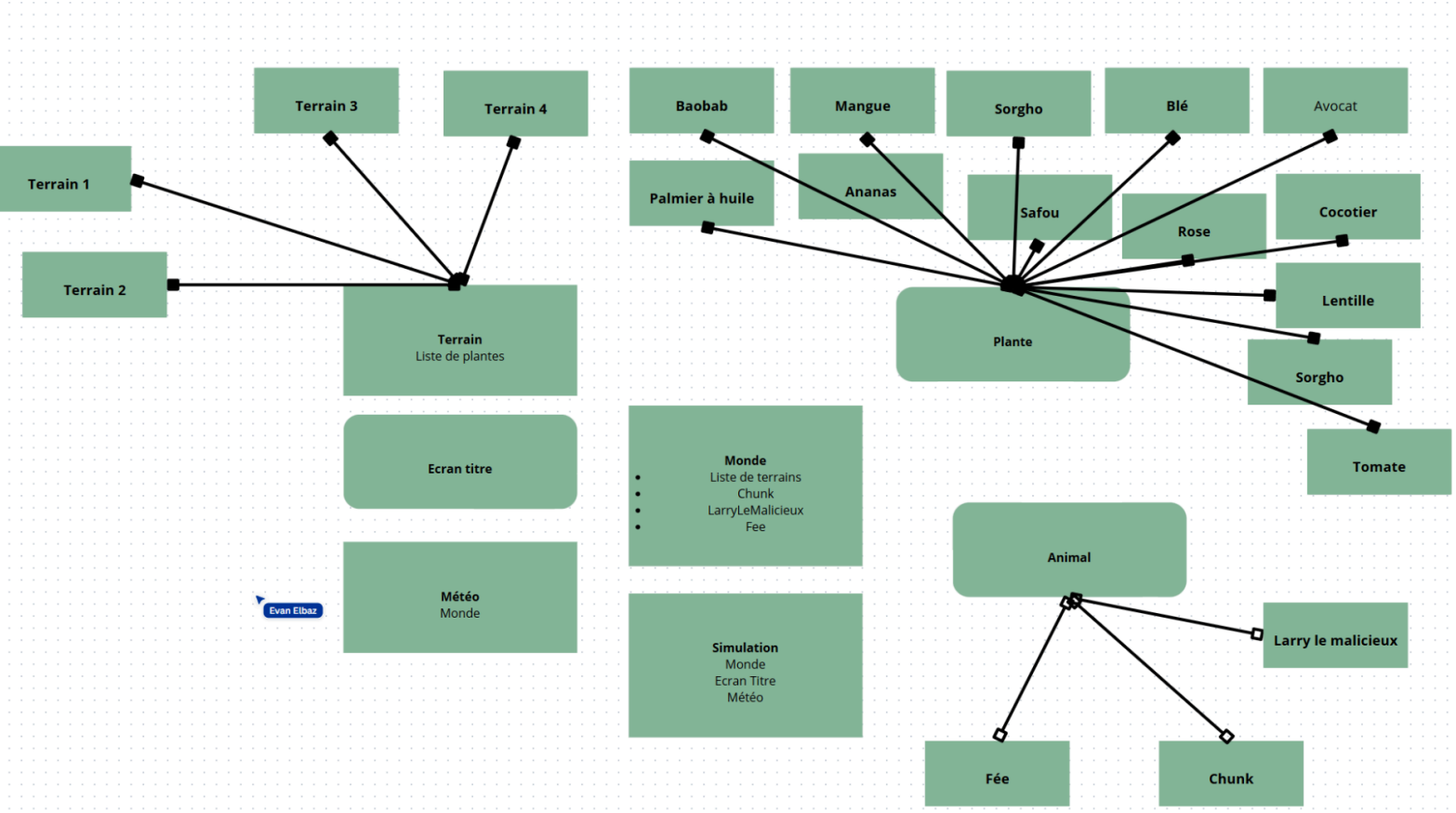
Classe Météo

Une météo a un objet de type Monde en attribut, car une météo apparaît dans un monde.

Sa seule méthode, NouvelleMétéo(), permet avec des nombres générés aléatoirement de faire varier les attributs tempActuelleEnDeg de chaque terrain. Et en fonction de si les indicateurs sont à des valeurs extrêmes, alors la température, l'humidité, ou la luminosité varient plus. Ces trois paramètres sont cependant indépendants entre eux. On parcourt les terrains avec un foreach et on applique les conditions météo des nombres aléatoires avec des if else if. Les météos sont indépendantes entre elles car les paramètres

de température, luminosité ou humidité actuelles sont réinitialisés à chaque appel de la méthode.

Légende : en gras le nom des classes, en non gras les attributs étant des objets ou une liste d'objets et les traits pour symboliser les classes dérivées.



V/ Bilan

En fin de compte, nous avons réussi à implémenter la majorité des fonctionnalités auxquelles nous avions pensé. On est satisfait de la représentation de ce à quoi pourrait s'apparenter l'entretien d'un potager au Kenya, et avons notamment découvert des écosystèmes insoupçonnés, ce qui nous a beaucoup motivés. Mis à part la gestion Github de notre projet, nous pouvons aussi être satisfait de notre organisation qui nous a permis de rendre le travail à temps. Cette organisation nous a aussi permis d'équilibrer la charge de travail et de la répartir de la manière la plus efficace, tout en s'aidant entre nous. C'est notamment Evan qui a mené la danse au niveau du code et expliqué clairement tout ce qu'il faisait, ce qui a permis à Simon de s'y intéresser et d'aider malgré ses difficultés.