

Compte Rendu Sprint 2

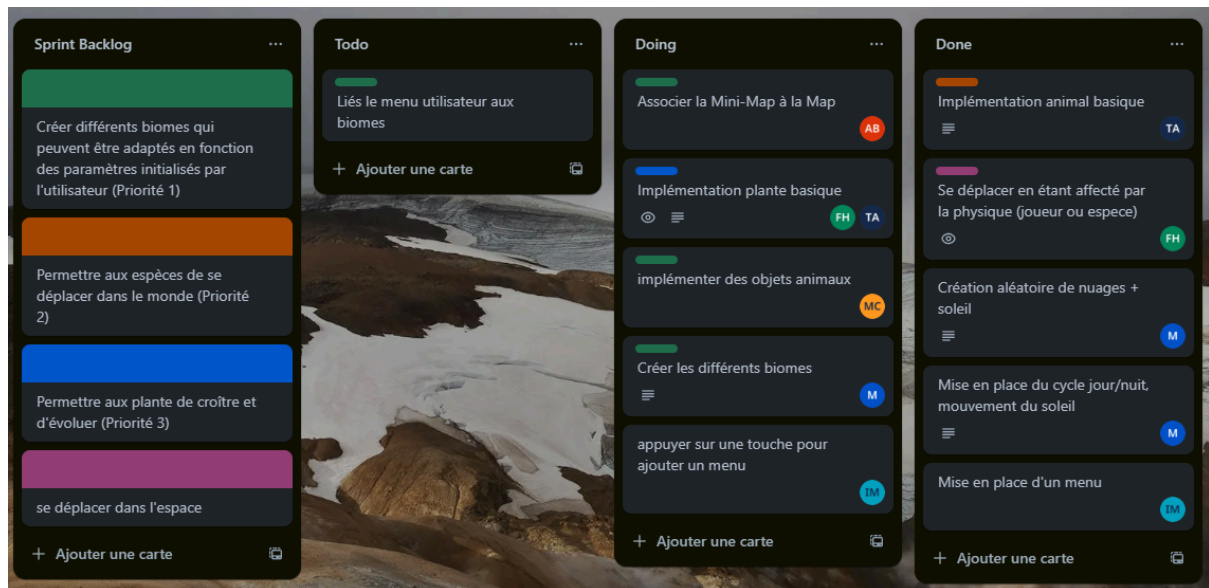
1. Objectifs Sprint 2	2
1.1. Mêlée : Mise en place du sprint 2	2
2. Revue du Sprint 2	3
2.1. Création des différents biomes	3
2.2. Espèces pouvant se déplacer	3
2.3. Permettre aux plantes de pousser	4
2.4. Améliorations diverses	5
3. Prochaines étapes	5
3.1 Objectifs principaux	5
3.2 Objectifs annexes	6

1. Objectifs Sprint 2

1.1. Mêlée : Mise en place du sprint 2

12 mai 2025

Ce deuxième sprint avait pour but principal d'implémenter les objets et entités qui allaient évoluer dans notre monde (terrain, faune et flore), ainsi que de diverses améliorations telles qu'une minimap et la gestion des paramètres via un menu.



Les trois fonctionnalités prioritaires identifiées pour ce sprint étaient :

- **Créer des différents biomes (caractéristiques terrains)**
- **Permettre aux espèces de se déplacer dans le monde**
- **Permettre aux plantes de croître**

2. Revue du Sprint 2

2.1. Création des différents biomes

Il est désormais possible de choisir entre une multitude de terrains générées procéduralement dont les paramètres influent la génération du monde.

12 mai 2025

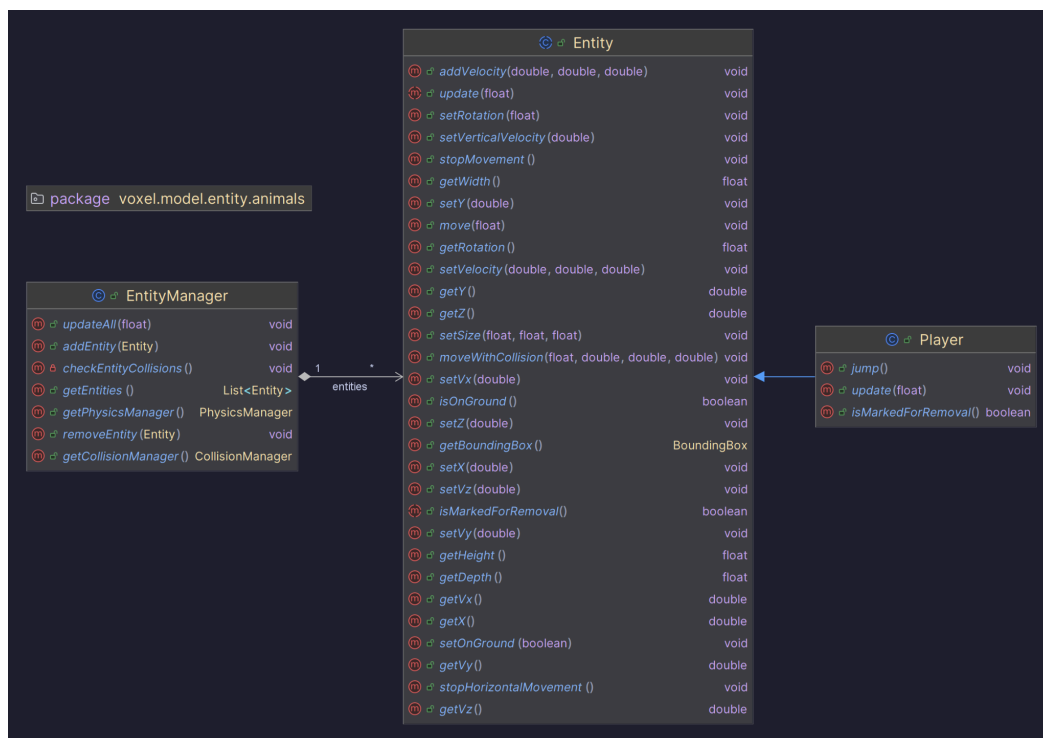
Ces derniers étant : le niveau de l'eau, la hauteur des montagnes, l'amplitude de la crête des montagnes, le type de sol.

2.2. Espèces pouvant se déplacer

Il est désormais possible de faire apparaître une espèce qui évoluera selon des règles définies ultérieurement.

Pour le moment il s'agit simplement de bloc de couleur sans modeles 3D qui se déplace selon des règles primaire.

Chaque typé d'entité hérite d'une classe abstraite Entity. Les entités sont gérées via un EntityManager.



(Contenu du package *voxel.model.entity*)

Wolf Wolf(double, double, double) isNight boolean startPatrolling() void patrol(float) void rest() void update(float) void isNight boolean markedForRemoval boolean	Owl Owl(double, double, double) isNight boolean glide(float) void update(float) void takeOff() void land() void isNight boolean markedForRemoval boolean	Sheep Sheep(double, double, double) setRotation(float, float) void moveForward(float) void update(float) void setMoving(float, float) void startGrazing() void markedForRemoval boolean
Lizard Lizard(double, double, double) startBasking() void moveForward(float) void startMoving() void move(float) void update(float) void markedForRemoval boolean	Fox Fox(double, double, double) setRotation(float, float) void moveForward(float) void update(float) void setMoving(float, float) void startHunting() void markedForRemoval boolean	Cow Cow(double, double, double) setMoving(float, float) void update(float) void jump() void setRotation(float, float) void moveForward(float) void markedForRemoval boolean
Dromedary Dromedary(double, double, double) startResting() void update(float) void advance(float) void moveForward(float) void markedForRemoval boolean	Scorpion Scorpion(double, double, double) moveForward(float) void hideUnderSand() void advance(float) void update(float) void markedForRemoval boolean	Eagle Eagle(double, double, double) update(float) void markedForRemoval boolean

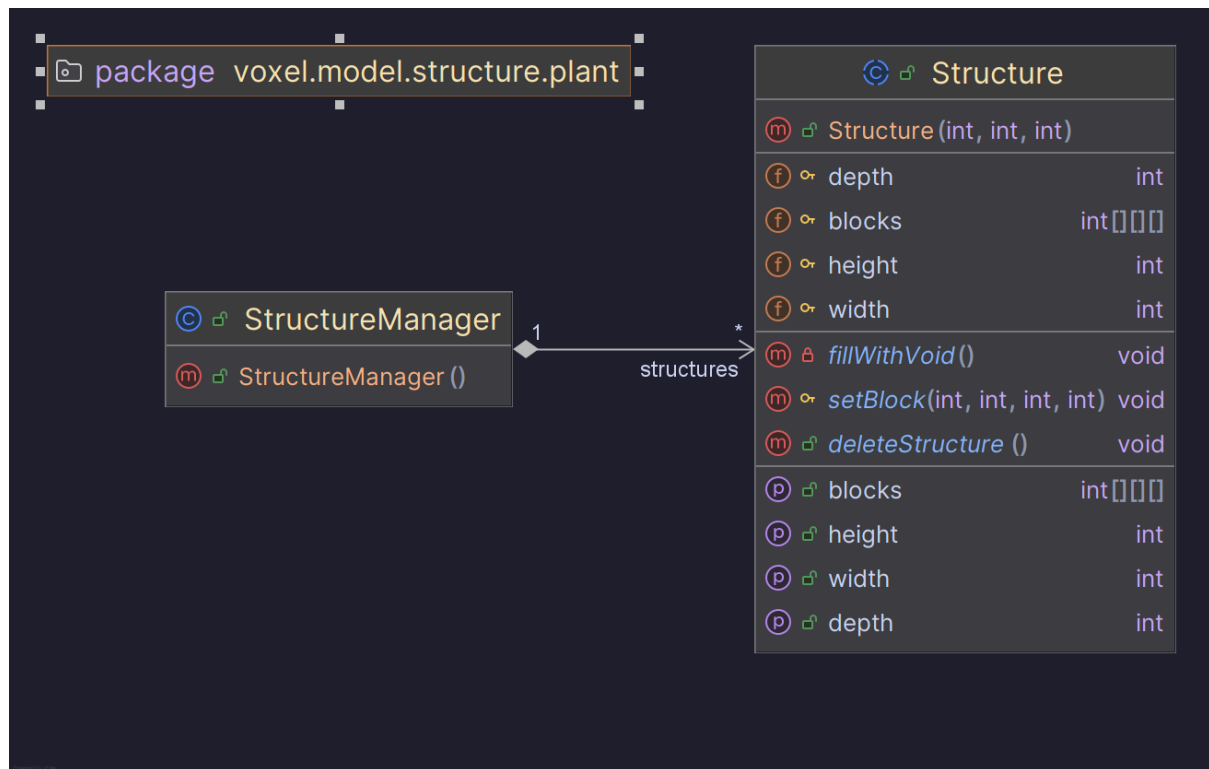
(Contenu du package *voxel.model.entity.animals*)

2.3. Permettre aux plantes de pousser

Nous avons défini une classe abstraite *Structure* qui permet de définir des formations de voxels (à différencier des entités qui peuvent se déplacer n'importe où dans le monde).

La première (et seule pour le moment) classe qui hérite de *Structure* est la classe *BasicTree* qui génère un arbre procéduralement.

L'objectif sera d'avoir plusieurs structures qui évoluent au fur et à mesure du temps en fonction de certaines règles.



2.4. Améliorations diverses

- Implémentation d'un cycle jour/nuit
- Implémentation de nuages
- Possibilité pour l'utilisateur de jouer en vue 1ère personne (être dans une entité joueur), 3ème personne, ou caméra libre

3. Prochaines étapes

3.1 Objectifs principaux

- Permettre au joueur de choisir ses paramètres
- Modèles 3D et animation pour la faune
- + de types de structures/plantes
- Implémentations d'interfaces HUD (toujours d'actualité) :
 - Mini-carte
 - Paramètres du jeu
 - Statistiques des éléments du monde
- Faire évoluer le monde et l'environnement (érosion (??))
- Évolution de la faune et de la flore selon des règles définies (croissance, déplacement, reproduction...)

3.2 Objectifs annexes

- Optimisation des performances
- Amélioration du rendu visuel (Ombres, éclairage, textures (?))...