

Compte Rendu Sprint 1

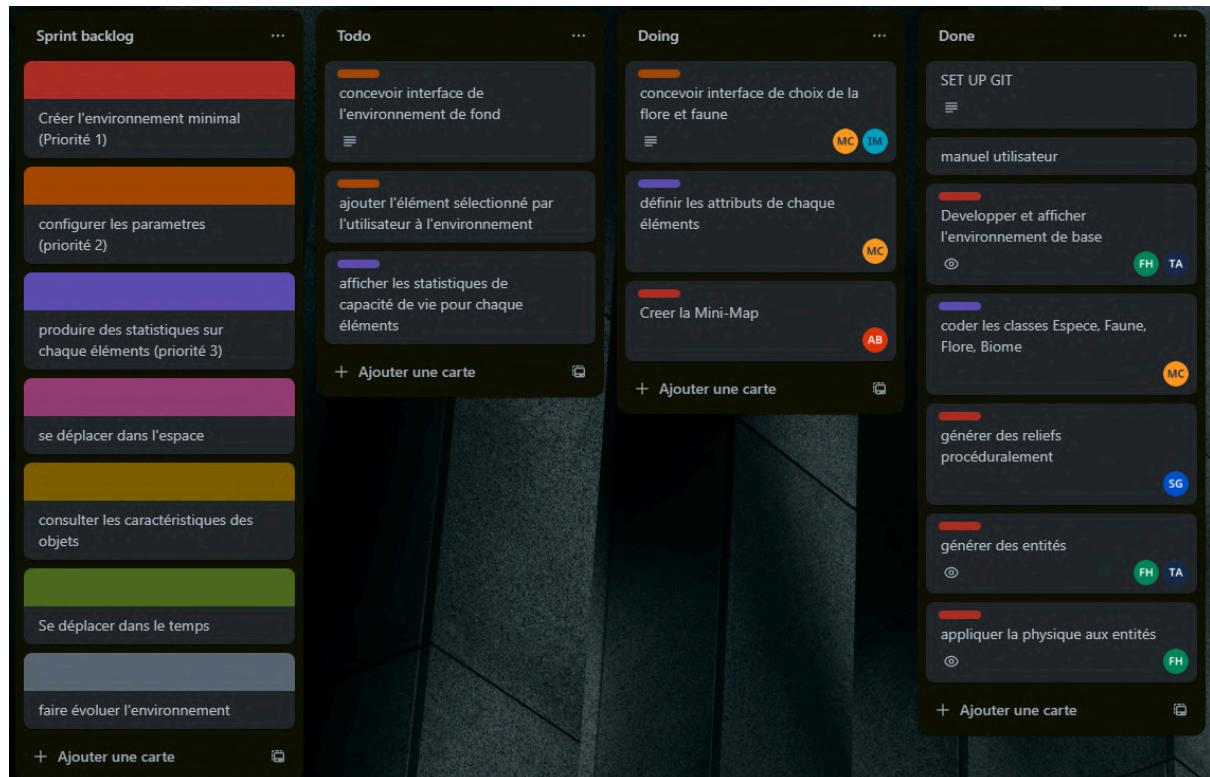
2025/04/14, Remise des livrables Itération 1 (rapports individuels + code + manuel utilisateur + rapport général).

1. Objectifs Sprint 1	2
1.1. Mêlée : Mise en place du sprint 1	2
2. Revue du Sprint 1	3
2.1. Définition approfondie des fonctionnalités	3
2.2. Création et visualisation de l'environnement minimal	3
2.3. La configuration des paramètres	5
2.4. Gestion des entités et intégration de la physique	5
2.5. Génération procédurale du monde	6
2.6. La production des statistiques pour chaque éléments et environnements	7
3. Prochaines étapes	8
3.1 Objectifs principaux	8
3.2 Objectifs annexes	8

1. Objectifs Sprint 1

1.1. Mêlée : Mise en place du sprint 1

Ce premier sprint avait pour but de mettre en place les fondations du projet, tant sur le plan technique que fonctionnel. Il s'agissait de concevoir un environnement minimal dans lequel les fonctionnalités futures pourront s'implémenter progressivement.



Les trois fonctionnalités prioritaires identifiées pour ce sprint étaient :

- **La création et l'affichage d'un environnement minimal** (sans caractéristiques spécifiques)
- **La configuration des paramètres** (faune, flore, biomes)
- **La production des statistiques de base** sur les éléments ajoutés à l'environnement

2. Revue du Sprint 1

2.1. Définition approfondie des fonctionnalités

Dans un premier temps, ce sprint nous a permis de fixer de manière claire et définitive les fonctionnalités principales de l'application ainsi que le déroulement général du projet.

Initialement, notre idée était de permettre à l'utilisateur de choisir un environnement parmi quatre biomes prédéfinis (forêt, montagne, désert, prairie), puis d'y ajouter librement des espèces de faune et de flore, qu'elles soient adaptées ou non à ce milieu. L'objectif était alors d'observer le niveau de compatibilité de chaque espèce avec l'environnement choisi.

Cependant, au fil du développement et de nos échanges, nous avons fait évoluer cette approche afin de renforcer la cohérence écologique du système.

Désormais, l'utilisateur pourra personnaliser les caractéristiques de l'environnement : humidité, densité de végétation, température, relief, etc.

À partir de ces paramètres, un biome est généré dynamiquement. Le système sélectionne ensuite automatiquement des espèces adaptées, selon leur taux de compatibilité avec les conditions définies, puis les fait évoluer.

Ce changement d'orientation vise à **simuler un écosystème crédible et autonome**, dans lequel les espèces évoluent naturellement en fonction de leur milieu. Cela permet une expérience plus immersive et plus proche de la logique de **simulation procédurale réaliste**.

2.2. Crédit et visualisation de l'environnement minimal

L'objectif de permettre à l'utilisateur de lancer l'application sur un environnement de base a été atteint.

L'environnement initial est volontairement neutre (sans caractéristique de biomes ni éléments ajoutés), servant de point de départ à toute simulation.

Ce socle permet d'intégrer progressivement les éléments dynamiques (faune, flore, évolutions, etc.) lors des sprints suivants.

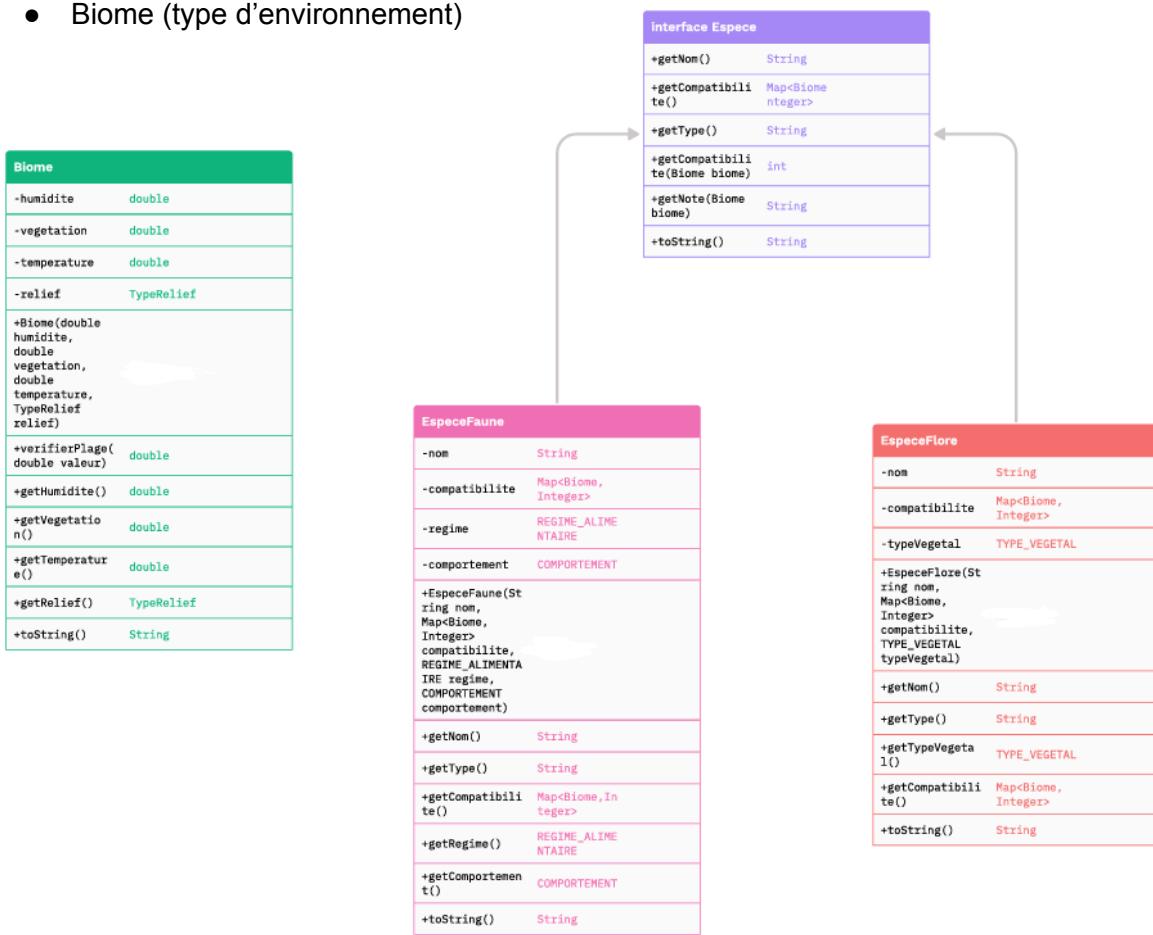


2.3. La configuration des paramètres

Nous avons implémenté les **structures de données nécessaires** pour modéliser les espèces (faune et flore), les biomes, et leurs compatibilités.

Les classes suivantes sont opérationnelles :

- Espace (interface générique pour faune/flore)
- EspaceFaune (faune avec régime et comportement)
- EspaceFlore (flore avec type végétal)
- Biome (type d'environnement)



2.4. Gestion des entités et intégration de la physique

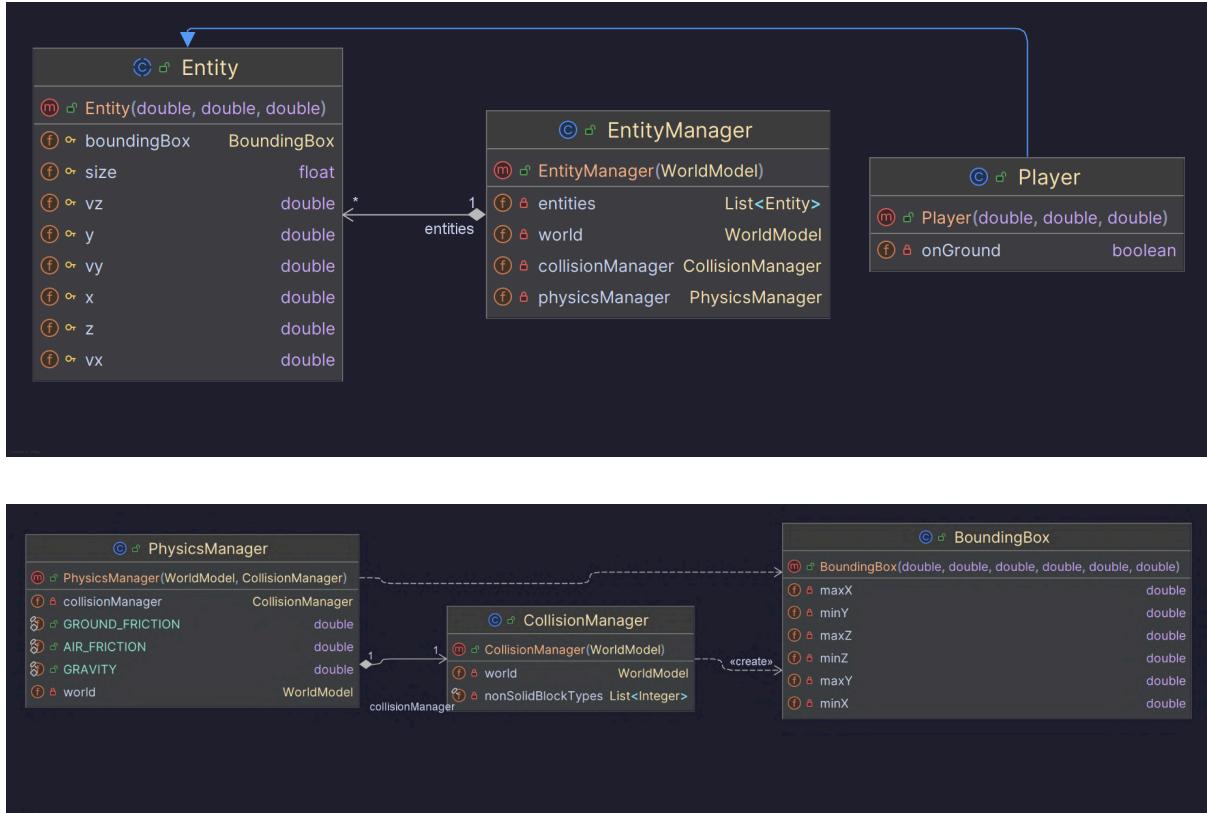
Nous avons commencé à implémenter la gestion des entités dans le moteur. Bien que cette partie ne soit pas encore finalisée, la structure mise en place est solide et servira de base fiable pour la suite du développement.

Chaque entité peut désormais être associée à une composante physique, ce qui permet d'appliquer des lois de la physique (gravité, collisions, déplacements, etc.) à son comportement dans le monde. Cette gestion sera notamment utilisée pour représenter le joueur, qui pourra ainsi se déplacer de manière réaliste dans l'environnement, en étant

affecté par la physique du jeu. Actuellement seule la classe Player existe, mais n'est pas finalisée.

Elle servira également à modéliser le comportement des animaux, qui pourront se déplacer et évoluer dynamiquement dans le monde. À terme, cette gestion d'entité pourrait aussi être utilisée pour représenter la faune et la flore, mais cela dépendra de la manière dont nous choisirons de les implémenter. Ce dernier point reste encore à définir.

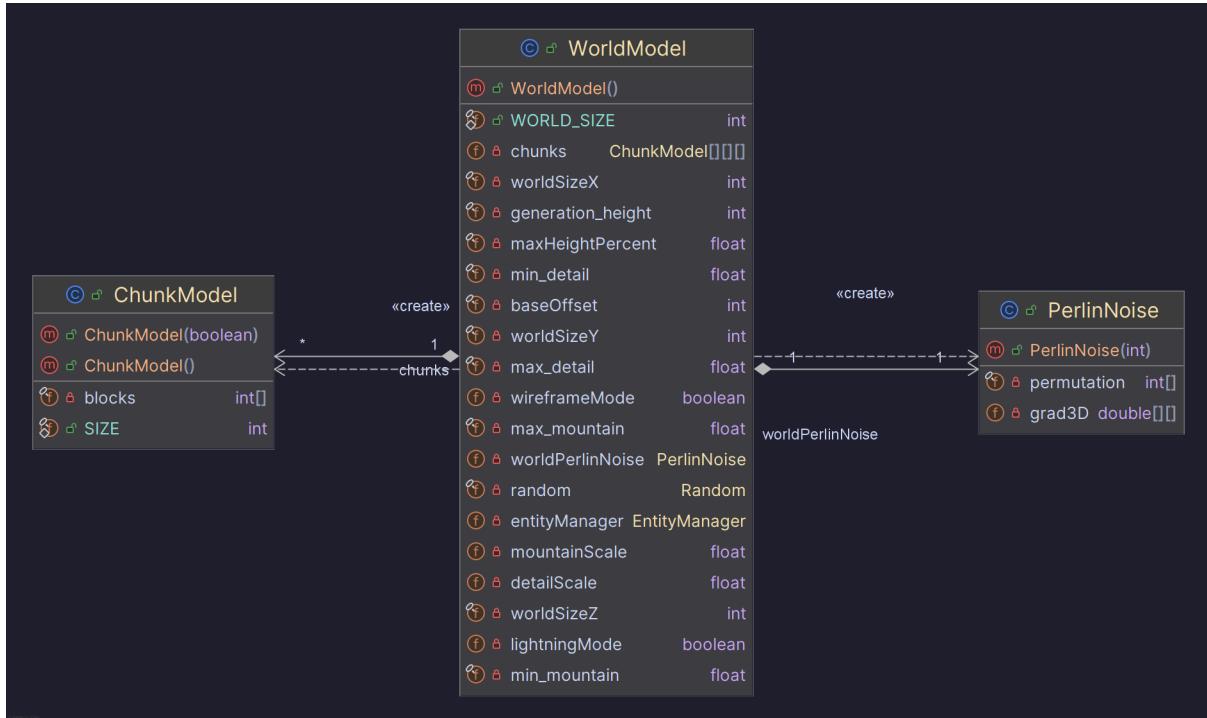
package model :



2.5. Génération procédurale du monde

Nous avons commencé à implémenter du bruit de Perlin afin de générer des paysages de manière procédurale. Cette première étape permet de créer des variations naturelles du terrain, avec des reliefs plus réalistes que de simples générateurs aléatoires.

Cependant, la gestion des biomes n'est pas encore en place. Il reste encore du travail pour intégrer différents types de biomes (plaines, montagnes, déserts, forêts, etc.) en fonction de critères géographiques cohérents. De plus, il faudra affiner les paramètres utilisés par le bruit de Perlin afin qu'ils influencent le monde de manière plus réaliste, en tenant compte de facteurs comme l'altitude, la température ou l'humidité.



2.6. La production des statistiques pour chaque éléments et environnements

Un système de compatibilité par score a été mis en place, permettant d'évaluer la pertinence d'un élément (faune/flore) dans un biome donné.

Pour chaque espèce ajoutée, l'utilisateur peut obtenir :

- Un score de compatibilité avec les biomes
- Une note qualitative : Adapté, Risque, Inadapté
- Une représentation textuelle de ses caractéristiques (type, comportement, etc.)

Ces indicateurs permettent déjà une première forme d'analyse sur la cohérence d'un élément dans un environnement configuré par l'utilisateur.

3. Prochaines étapes

3.1 Objectifs principaux

- Implémentations d'interfaces HUD :
 - Mini-carte
 - Paramètres du jeu
 - Statistiques des éléments du monde
- Gestion du joueur :
 - Implémenter son déplacement en interaction avec la physique
- Améliorer la génération du monde :
 - Intégration de biomes
 - Utilisation de paramètres réalistes pour influencer la génération
- Créations de menus de configuration :
 - Interface pour définir les paramètres de génération du monde
 - Options d'évolutions du monde
- Implémentations d'entités vivantes :
 - Faune et flore
 - Interaction avec l'environnement
 - Évolution selon des règles définies (croissance, déplacement, reproduction...)

3.2 Objectifs annexes

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