

# Projet d'AP4B: Power Sim

Osée Tchappi  
Adrien Burgun  
Tarabai Gambara  
Jiang YiWen

Automne 2021

## Résumé

Pour ce projet d'AP4B, nous avons décidé de choisir le sujet "Power Sim".

Ce document décrit notre démarche de conception du projet, à la vue de l'implémentation du jeu en Java dans les semaines à venir.

Lors de cette démarche de conception, nous avons été amenés à écrire des diagrammes UML, ceux-ci sont également présentés dans ce document.

Certains diagrammes (Cas d'utilisation - Section 2 et Diagramme de Classe - Section 3) ont été découpés en plusieurs sous-diagrammes pour la lisibilité ; les diagrammes complets, ainsi que le code source de ce rapport, peuvent être trouvés sur GitHub : <https://github.com/adri326/ap4b-project/>.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Cas d'utilisations</b>	<b>3</b>
<b>3</b>	<b>Diagramme de Classe</b>	<b>7</b>
<b>4</b>	<b>Description Textuelle</b>	<b>13</b>
<b>5</b>	<b>Diagrammes de séquence</b>	<b>19</b>
<b>6</b>	<b>Diagramme de communication</b>	<b>23</b>

# 1 Introduction

Nous avons choisi parmi les sujets proposés celui intitulé « Power Sim », car il nous a plus inspiré et qu'il nous donne plus de liberté en termes de systèmes que nous souhaitons mettre en place.

Nous nous sommes rassemblé chaque vendredi pour mettre en commun notre travail et pour distribuer le travail pour la semaine à venir. Nous avons d'abords collecté des idées indépendamment, puis nous avons construit un diagramme de cas d'utilisation et mis en place la repository pour le projet.

Nous avons ensuite travaillé en parallèle sur différents diagrammes : le diagramme de classe, les diagrammes de séquence, la description textuelle des cas d'utilisation et le diagramme de communication.

Nous avons décidé de modéliser le sujet de la manière suivante :

- Le joueur construit une ville, qui est constituée de bâtiments se trouvant sur une grille.
- Le joueur peut placer, modifier et supprimer des bâtiments.
- Des habitants s'installent lorsque des ressources (électricité, routes) sont disponibles : ils construisent des maisons et d'autres bâtiments là où il y a de la place.
- L'accent est placé sur les simulations faisant vivre la ville : la pollution, le transfert de ressources entre les bâtiments et la satisfaction des habitants sont simulés.
- Alors que la ville grandit et que le joueur construit des bâtiments, des technologies sont débloquées, le permettant d'améliorer des bâtiments ou de placer de nouveaux bâtiments : de cette manière, le joueur peut produire de l'énergie plus proprement et plus efficacement.

## 2 Cas d'utilisations

Notre phase de conception a débuté avec la mise en place d'un diagramme de cas d'utilisation.

Trois acteurs se dessinent :

- **Le joueur** ; ses interactions sont marquées par des entrées clavier/souris, ainsi que son besoin de percevoir les informations du jeu. Le joueur place les bâtiments qu'il veut, il choisit les options, il observe la ville évoluer.
- **La simulation** ; celle-ci a des interactions à caractère périodique avec le jeu : sa plus importante est l'étape de simulation, qui est divisée en différents aspects du jeu. (pollution, immigration, production d'énergie, etc.)
- **La banque** ; celle-ci gère l'argent du joueur : elle offre au joueur un solde initial, ainsi que des prêts s'il n'a plus d'argent.



FIGURE 1 – Les cas d'utilisation associés au joueur

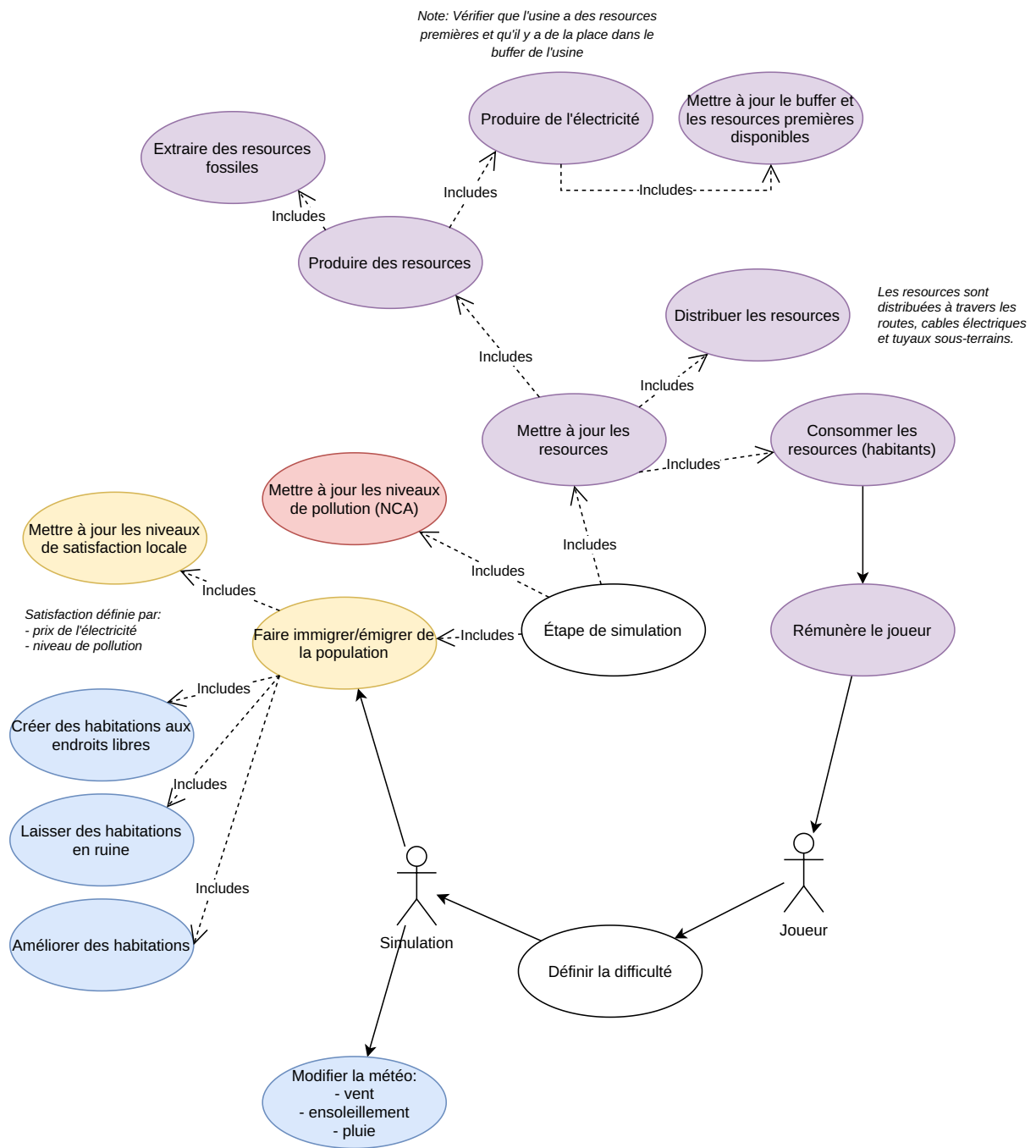


FIGURE 2 – Les cas d'utilisation associés à la simulation

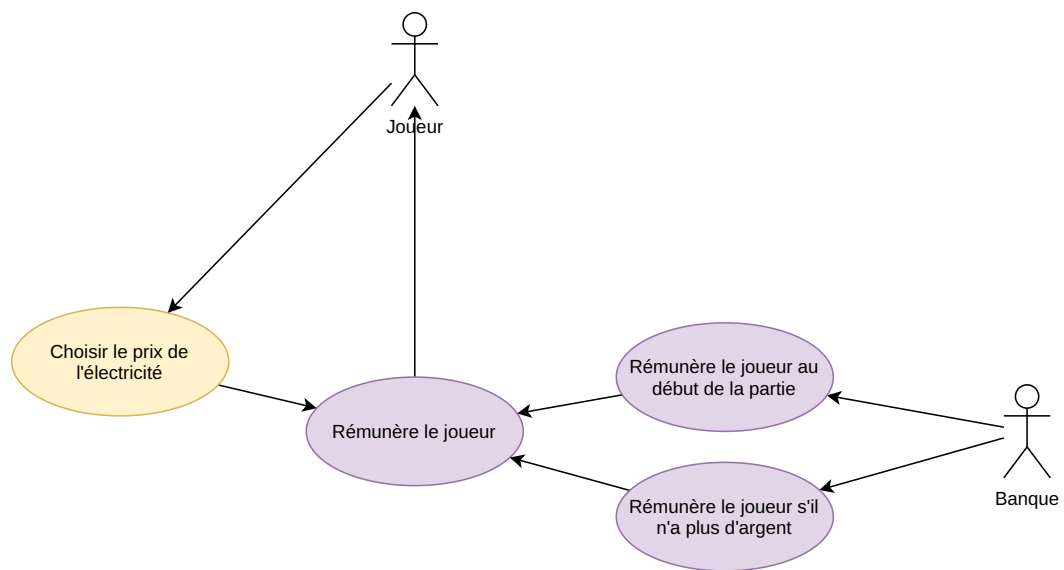


FIGURE 3 – Les cas d'utilisation associés à la banque

### 3 Diagramme de Classe

Nous avons construit un diagramme de classe, implémentant les besoins émis dans le diagramme de cas d'utilisation et servant de base aux diagrammes de séquences venant par la suite.

Dans cette conception du projet, la classe **GameState** est la classe principale de la mécanique du jeu : celle-ci stocke l'entièreté des informations d'une partie. Les différents composants de cette classe correspondent aux différents besoins du projet :

- La classe **Tile** (Figure 5, contenue dans la classe **Map**) est commune à tous les bâtiments posés sur l'espace de jeu : celle-ci est étendue pour implémenter la logique et les spécificités des différents bâtiments (Figure 6).
- La classe **Bank** gère la partie "banque" : rémunérer le joueur au début, faire un prêt, etc.
- La classe **Simulation** gère le coeur de la simulation du jeu : l'immigration, la génération de ressources et la pollution.
- La classe **Weather** contient une petite simulation de météo et du cycle jour/nuit.
- La classe **TechTree** (Figure 7) contient les technologies que le joueur débloque au fil de la partie : celles-ci sont caractérisées par des pré-requis (**Requirement**) et des récompenses (**Reward**).

Les classes spécifiques à la partie graphisme sont incluses dans ce rapport mais seront amenées à être modifiées et étendues lors de la phase d'implémentation du projet. Celles-ci se trouvent dans la Figure 8

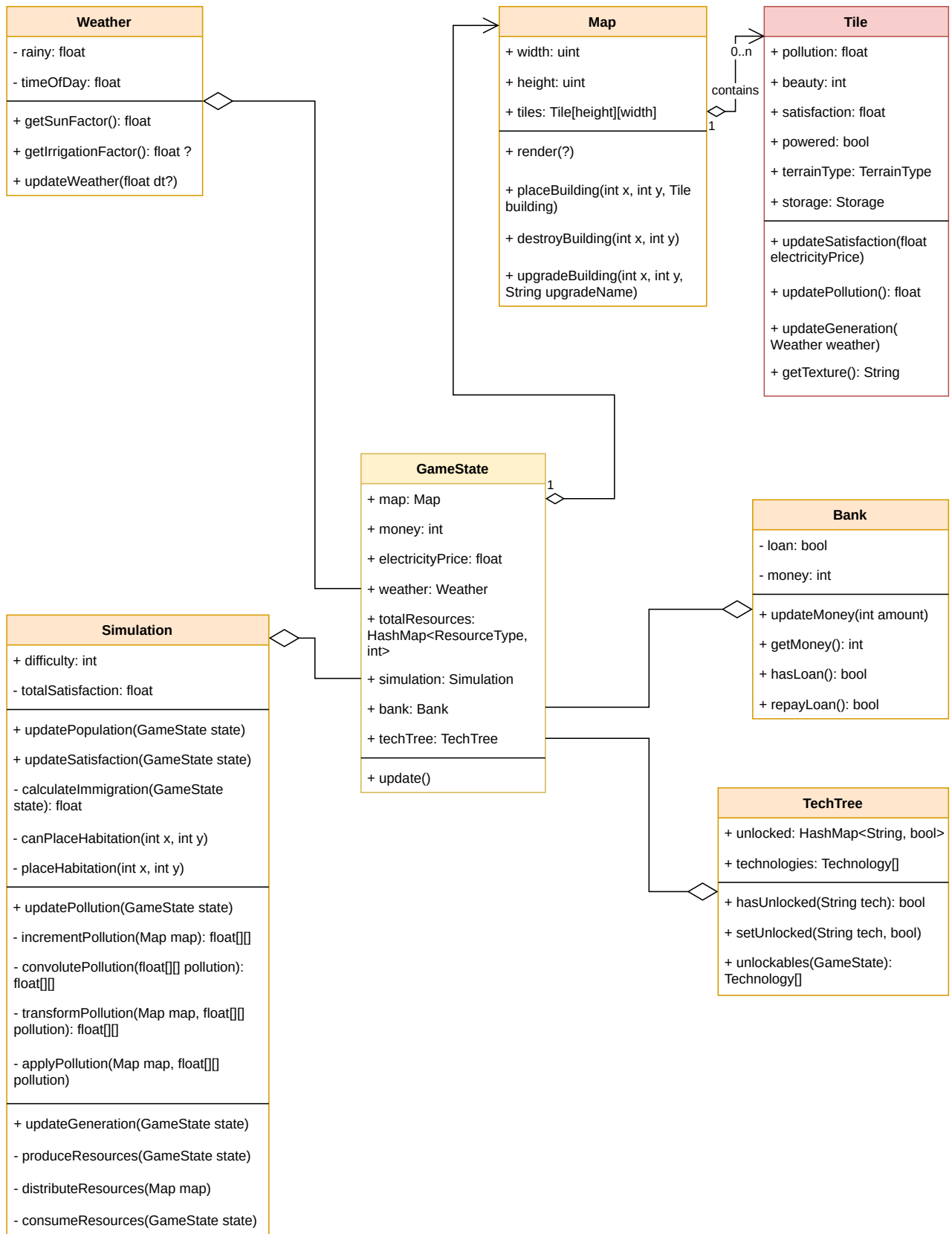


FIGURE 4 – La classe principale `GameState` et les classes qu'elle contient.



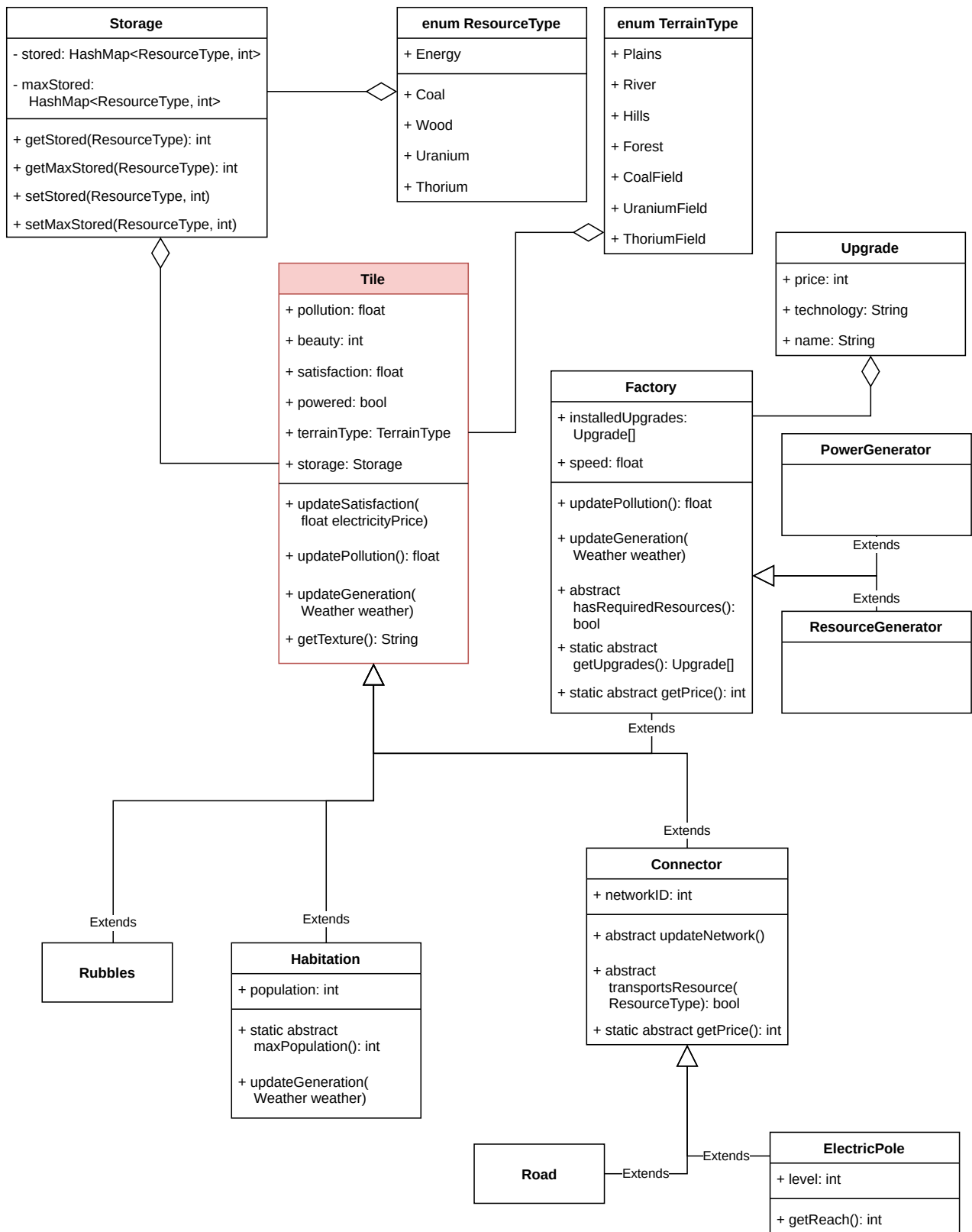


FIGURE 5 – La classe **Tile** et ses principaux descendants.

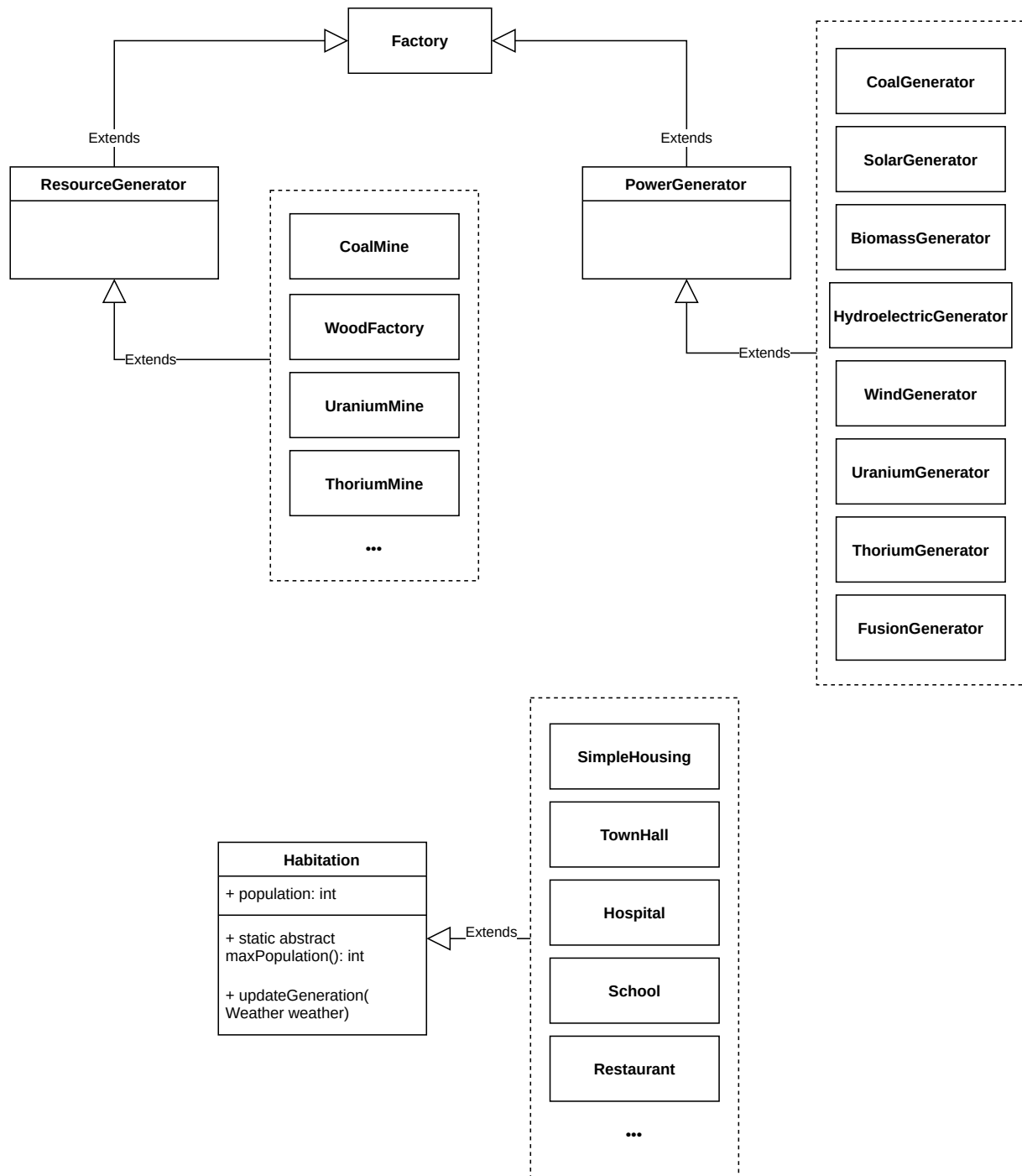


FIGURE 6 – Détail de Figure 5 : les classes descendant de Factory et de Habitation.

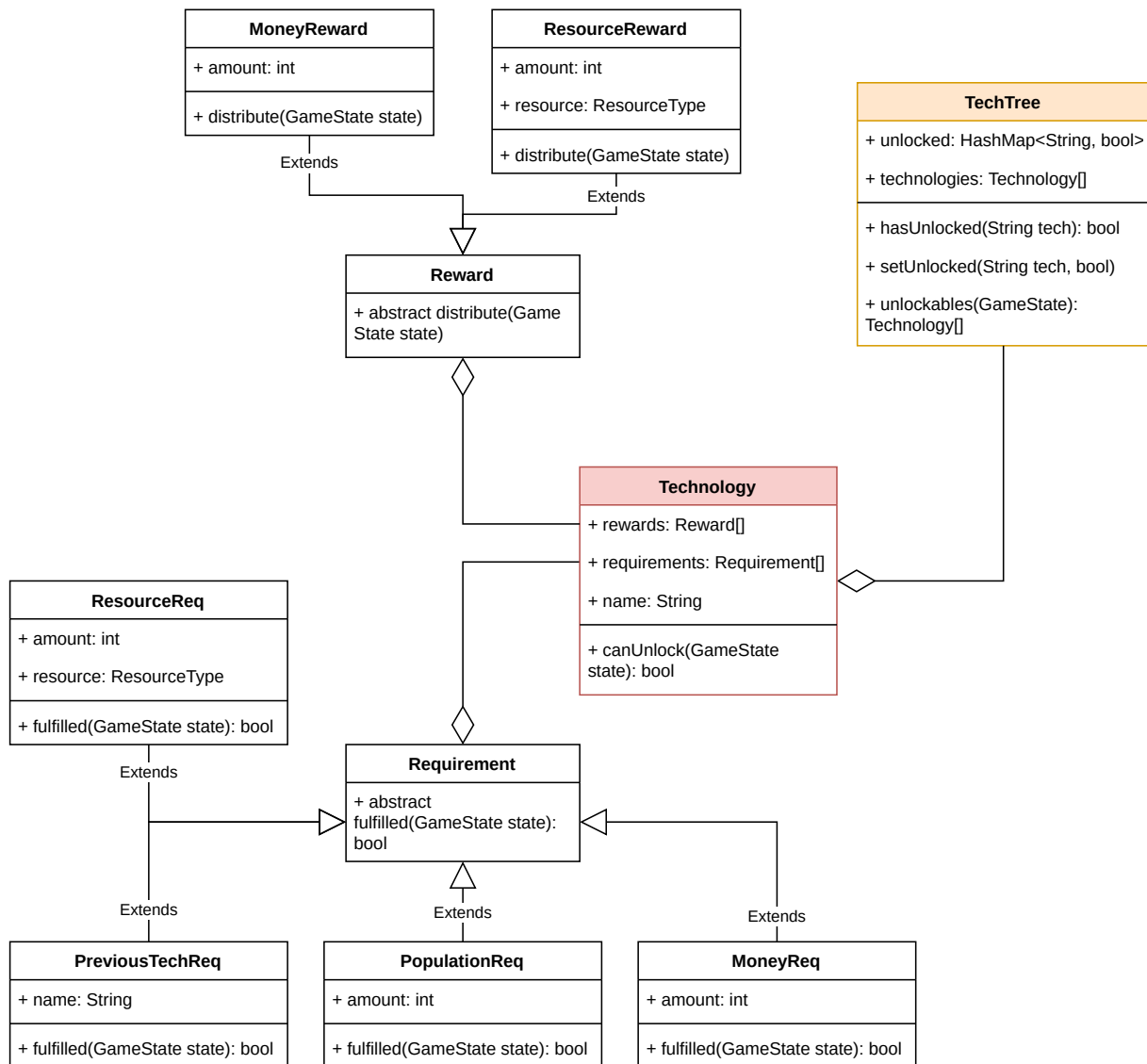


FIGURE 7 – La classe TechTree et ses principaux descendants.

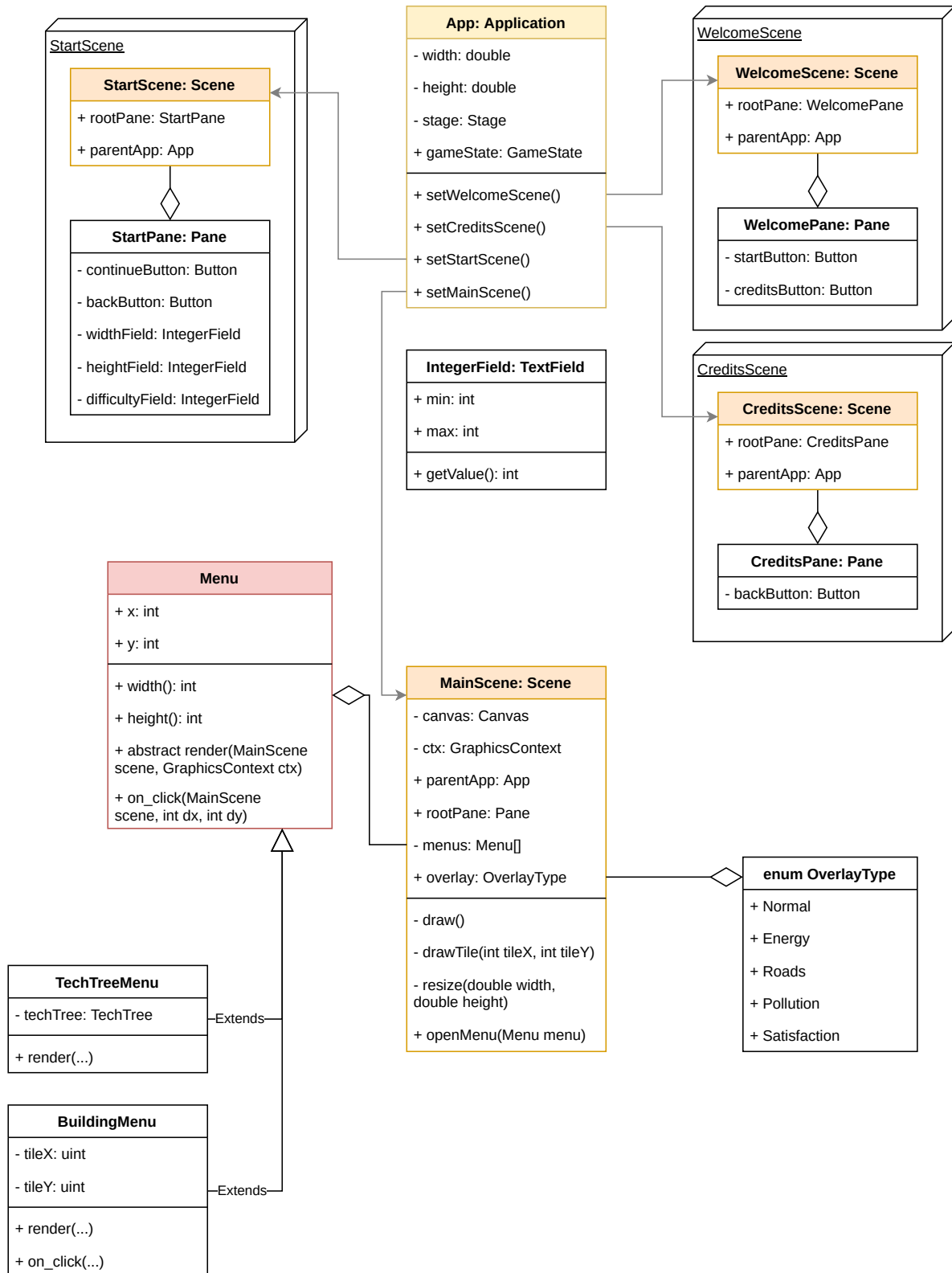


FIGURE 8 – La classe **App** et les différentes classes pour l’affichage du jeu.

## 4 Description Textuelle

En parallèle du diagramme de classe, nous avons élaboré une description textuelle de certains cas d'utilisation :

- Placer un bâtiment
- Sélectionner un bâtiment sur la carte
- Voir les informations spécifiques à un bâtiment
- Supprimer un bâtiment
- Améliorer un bâtiment
- Débloquer un bâtiment
- Créer des habitations aux endroits libres
- Connaître les informations détaillées de la ville

<b>Cas d'utilisation</b>	Placer un bâtiment
<b>Acteurs</b>	joueur
<b>Objectif</b>	Permet de créer un nouveau bâtiment sur la carte pour produire, stocker ou distribuer de l'énergie
<b>Précondition</b>	<ul style="list-style-type: none"> <li>➔ Une partie de jeu doit être en cours</li> <li>➔ Le bâtiment à mettre doit être débloqué</li> <li>➔ Le joueur doit disposer des fonds nécessaires à la construction</li> <li>➔ La zone de jeu doit avoir au moins une case libre</li> </ul>
<b>Postcondition</b>	<ul style="list-style-type: none"> <li>✓ Le bâtiment est visible sur la carte à l'endroit voulu</li> <li>✓ Les fonds du joueur ont diminué</li> <li>✓ Le bâtiment est comptabilisé dans les actifs du joueur</li> <li>✓ Les paramètres de satisfaction sont mis à jour</li> <li>✓ L'emplacement du bâtiment est mis à jour</li> </ul>
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le joueur sélectionne le bâtiment à placer</li> <li>2. Le système demande au joueur de choisir l'emplacement du bâtiment</li> <li>3. Le joueur sélectionne l'emplacement sur la carte où il souhaite poser le bâtiment</li> <li>4. Le système vérifie la validité de l'emplacement</li> <li>5. Le système génère la facture au joueur</li> <li>6. Le joueur valide la facture</li> <li>7. Le système vérifie les ressources du joueur</li> <li>8. Le système valide la transaction</li> <li>9. Le bâtiment est placé</li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>A) Le joueur ne dispose pas des fonds nécessaires <ol style="list-style-type: none"> <li>i. Cet enchaînement démarre au point 7</li> <li>ii. Le scénario reprend au point 5</li> </ol> </li> <li>B) L'emplacement choisis par le joueur est occupé <ol style="list-style-type: none"> <li>i. Cet enchaînement démarre au point 4</li> <li>ii. Le scénario reprend au point 2</li> </ol> </li> </ol>
<b>Scénario d'exception</b>	<ol style="list-style-type: none"> <li>A) Le bâtiment sollicité est bloqué</li> <li>B) Le joueur annule la création du bâtiment</li> </ol>

<b>Cas d'utilisation</b>	Sélectionner un bâtiment sur la carte
<b>Acteurs</b>	joueur
<b>Objectif</b>	Permet de consulter les détails d'un bâtiment
<b>Précondition</b>	<ul style="list-style-type: none"> <li>➔ Une partie de jeu doit être en cours</li> <li>➔ Le bâtiment doit être sur la carte</li> </ul>
<b>Postcondition</b>	✓ Les informations sur le bâtiment sont affichées
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le joueur clique sur le bâtiment à consulter sur la carte</li> <li>2. Le système génère les informations actuelles du bâtiment</li> <li>3. Le système affiche dans une interface, les informations du bâtiment</li> </ol>
<b>Scénario alternatif</b>	Pas de scénario alternatif

<b>Scénario d'exception</b>	Pas de scénario d'exception

<b>Cas d'utilisation</b>	Voir les informations spécifiques à un bâtiment
<b>Acteurs</b>	joueur
<b>Objectif</b>	Permet de consulter la fiche descriptive d'un bâtiment, cette dernière présentant l'utilité, les conditions de création, les améliorations possibles, le mode de fonctionnement, les ressources exploitées et éventuellement la production escomptée par niveau d'amélioration
<b>Précondition</b>	➔ Une partie de jeu doit être en cours
<b>Postcondition</b>	✓ Les informations sur le bâtiment sont affichées
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le joueur clique sur le bâtiment à consulter</li> <li>2. Le système affiche dans une interface, les informations du bâtiment</li> </ol>
<b>Scénario alternatif</b>	Pas de scénario alternatif
<b>Scénario d'exception</b>	Pas de scénario d'exception

<b>Cas d'utilisation</b>	Supprimer un bâtiment
<b>Acteurs</b>	joueur
<b>Objectif</b>	Permet au joueur de supprimer de la carte un bâtiment pour faire de l'espace et se faire un peu d'argent
<b>Précondition</b>	➔ Une partie de jeu doit être en cours ➔ Le bâtiment à supprimer doit exister sur la carte
<b>Postcondition</b>	✓ Le bâtiment est retiré des actifs du joueurs ✓ La zone éclairée par le bâtiment est mise à jour ✓ Les fonds (argent) du joueur sont mis à jour ✓ L'emplacement du bâtiment est mis à jour ✓ Le niveau de satisfaction est mis à jour ✓ Le niveau de pollution est mis à jour
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le joueur clique sur le bâtiment à supprimer sur la carte</li> <li>2. Le système affiche dans une interface, les informations du bâtiment</li> <li>3. Le joueur clique sur le bouton supprimer</li> <li>4. Le système génère la fiche présentant les conséquences de l'action (les pertes en ressources, les indemnités financières, etc...)</li> <li>5. Le joueur confirme son choix de suppression</li> <li>6. Le bâtiment est retiré de la carte</li> </ol>

<b>Scénario alternatif</b>	Pas de scénario alternatif
<b>Scénario d'exception</b>	A) Le joueur annule la suppression <ul style="list-style-type: none"> <li>• Ce scénario débute au point 4</li> </ul>

<b>Cas d'utilisation</b>	Améliorer un bâtiment
<b>Acteurs</b>	joueur
<b>Objectif</b>	Permet au joueur d'améliorer les performances d'un bâtiment
<b>Précondition</b>	<ul style="list-style-type: none"> <li>➔ Une partie de jeu doit être en cours</li> <li>➔ Le bâtiment à améliorer doit exister sur la carte</li> <li>➔ Le joueur doit disposer des fonds nécessaires à l'amélioration</li> <li>➔ Le bâtiment doit être améliorable</li> </ul>
<b>Postcondition</b>	<ul style="list-style-type: none"> <li>✓ Le bâtiment est amélioré</li> <li>✓ Les fonds (argent) du joueur sont mis à jour (diminution)</li> <li>✓ Les paramètres du bâtiment sont mis à jour</li> <li>✓ Le niveau de satisfaction est mis à jour</li> <li>✓ Le niveau de pollution est mis à jour</li> </ul>
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le joueur clique sur le bâtiment à améliorer sur la carte</li> <li>2. Le système affiche dans une interface, les informations du bâtiment</li> <li>3. Le joueur clique sur le bouton améliorer</li> <li>4. Le système génère la fiche présentant les améliorations possibles</li> <li>5. Le joueur sélectionne l'amélioration qu'il veut</li> <li>6. Le système vérifie si le joueur remplit les conditions nécessaires pour l'amélioration</li> <li>7. Le bâtiment est amélioré</li> </ol>
<b>Scénario alternatif</b>	A) Le joueur ne dispose pas des ressources nécessaires à l'amélioration <ol style="list-style-type: none"> <li>i. Ce scénario démarre au point 6</li> <li>ii. Le système informe le joueur de l'échec de l'opération</li> <li>iii. Le scénario reprend au point 4</li> </ol>
<b>Scénario d'exception</b>	A) Le joueur annule l'amélioration <ol style="list-style-type: none"> <li>1. Ce scénario débute au point 4</li> </ol>

<b>Cas d'utilisation</b>	Débloquer un bâtiment
<b>Acteurs</b>	Système (simulation), joueur
<b>Objectif</b>	Permet donner l'accès à un nouveau bâtiment



<b>Précondition</b>	<ul style="list-style-type: none"> <li>➔ Une partie de jeu doit être en cours</li> <li>➔ Le type de bâtiment précédent (dans la hiérarchie des bâtiments établis plus haut) doit être débloquent</li> <li>➔ La population doit être suffisante</li> <li>➔ Le niveau de satisfaction doit correspondre à la valeur requise</li> </ul>
<b>Postcondition</b>	✓ Le bâtiment est débloquent
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le système vérifie les conditions de débloquent du bâtiment</li> <li>2. Le système informe au joueur via une interface que le bâtiment est désormais débloquent (accessible)</li> <li>3. Le joueur clique sur le bouton Ok</li> <li>4. Le bâtiment est accessible</li> </ol>
<b>Scénario alternatif</b>	Pas de scénario alternatif
<b>Scénario d'exception</b>	A) Les conditions de déblocages ne sont pas remplies <ol style="list-style-type: none"> <li>1. Ce scénario débute au point 1</li> </ol>

<b>Cas d'utilisation</b>	Créer des habitations aux endroits libres
<b>Acteurs</b>	Système (simulation), joueur
<b>Objectif</b>	Permet d'ajouter de nouvelles habitations sur la carte
<b>Précondition</b>	<ul style="list-style-type: none"> <li>➔ Une partie de jeu doit être en cours</li> <li>➔ Le pourcentage de satisfaction requis doit être atteint</li> <li>➔ Un espace doit être libre sur la carte</li> <li>➔ Le taux de pollution requis doit être atteint</li> <li>➔ Les ressources d'énergies doivent être importantes</li> </ul>
<b>Postcondition</b>	✓ L'habitation est créée sur la carte
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le système vérifie les conditions de création d'une nouvelle habitation</li> <li>2. Le système informe au joueur via une interface qu'une habitation sera ajoutée à la carte</li> <li>3. Le joueur clique sur le bouton Ok</li> <li>4. L'habitation est ajoutée</li> </ol>
<b>Scénario alternatif</b>	Pas de scénario alternatif
<b>Scénario d'exception</b>	A) Les conditions de création ne sont pas remplies <ol style="list-style-type: none"> <li>1. Ce scénario débute au point 1</li> </ol>

<b>Cas d'utilisation</b>	Connaître les informations détaillées de la ville
<b>Acteurs</b>	joueur
<b>Objectif</b>	Permet au joueur d'obtenir la fiche détaillant les informations sur la ville
<b>Précondition</b>	➔ Une partie de jeu doit être en cours

<b>Postcondition</b>	✓ L'interface présentant les informations sur la ville est visible
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le joueur clique sur le bouton carte</li> <li>2. Le système affiche dans une interface, un résumé de la partie</li> <li>3. Le joueur clique sur le bouton informations sur la ville</li> <li>4. Le système génère la fiche présentant les informations détaillées de la ville</li> </ol>
<b>Scénario alternatif</b>	Pas de scénario alternatif
<b>Scénario d'exception</b>	A) Le joueur annule l'amélioration <ol style="list-style-type: none"> <li>1. Ce scénario débute au point 3</li> </ol>

## 5 Diagrammes de séquence

Enfin, nous avons construit quatre diagrammes de séquence, pour détailler l'ordre d'opération des systèmes complexes du jeu :

- La fonction `update()` de `GameState`, correspondant à une étape de simulation (Figure 9)
- Les actions que le joueur peut faire par rapport aux bâtiments (Figure 10)
- Le fonctionnement de la banque (Figure 11)

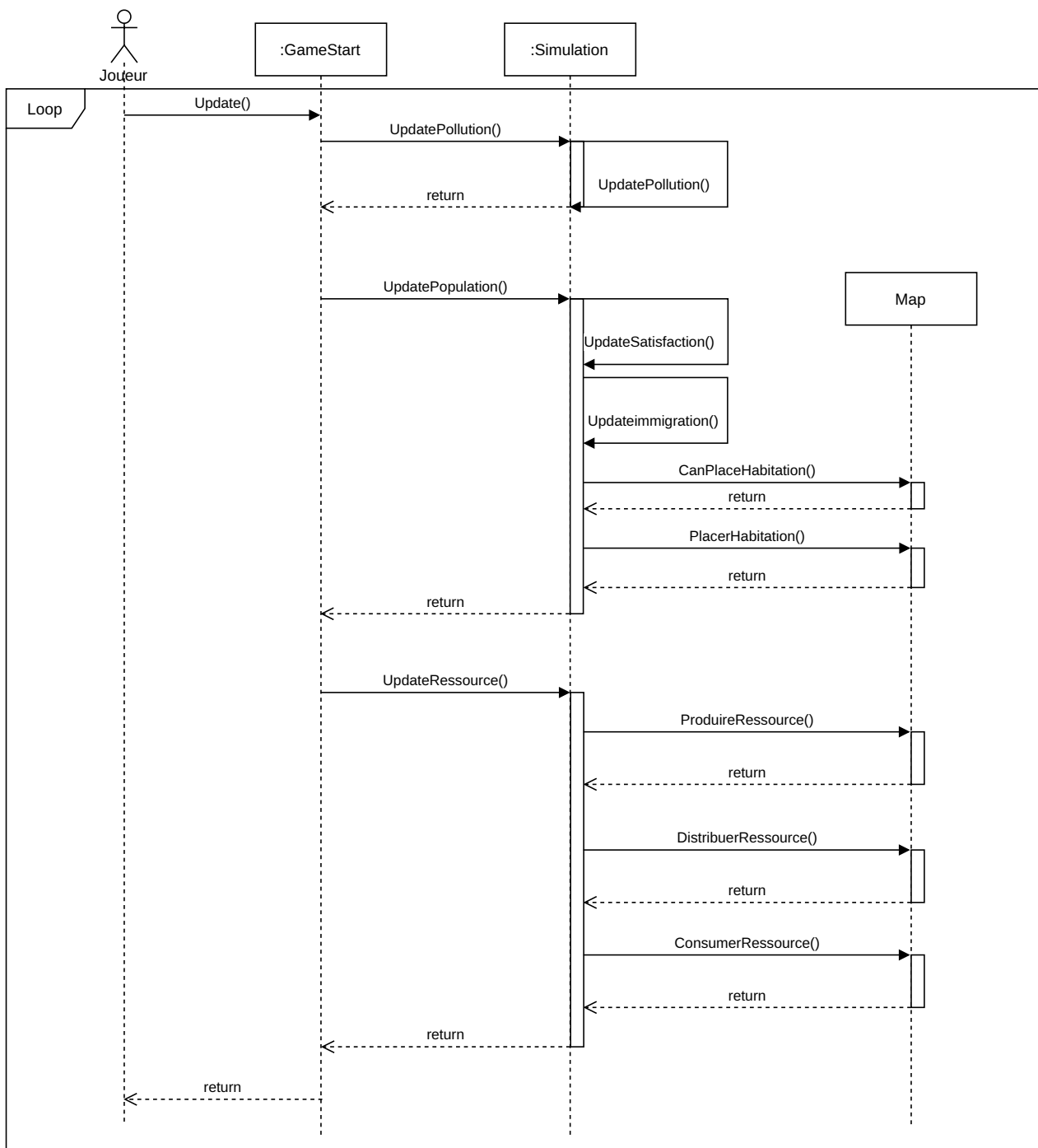


FIGURE 9 – Diagramme de séquence d'une étape de simulation

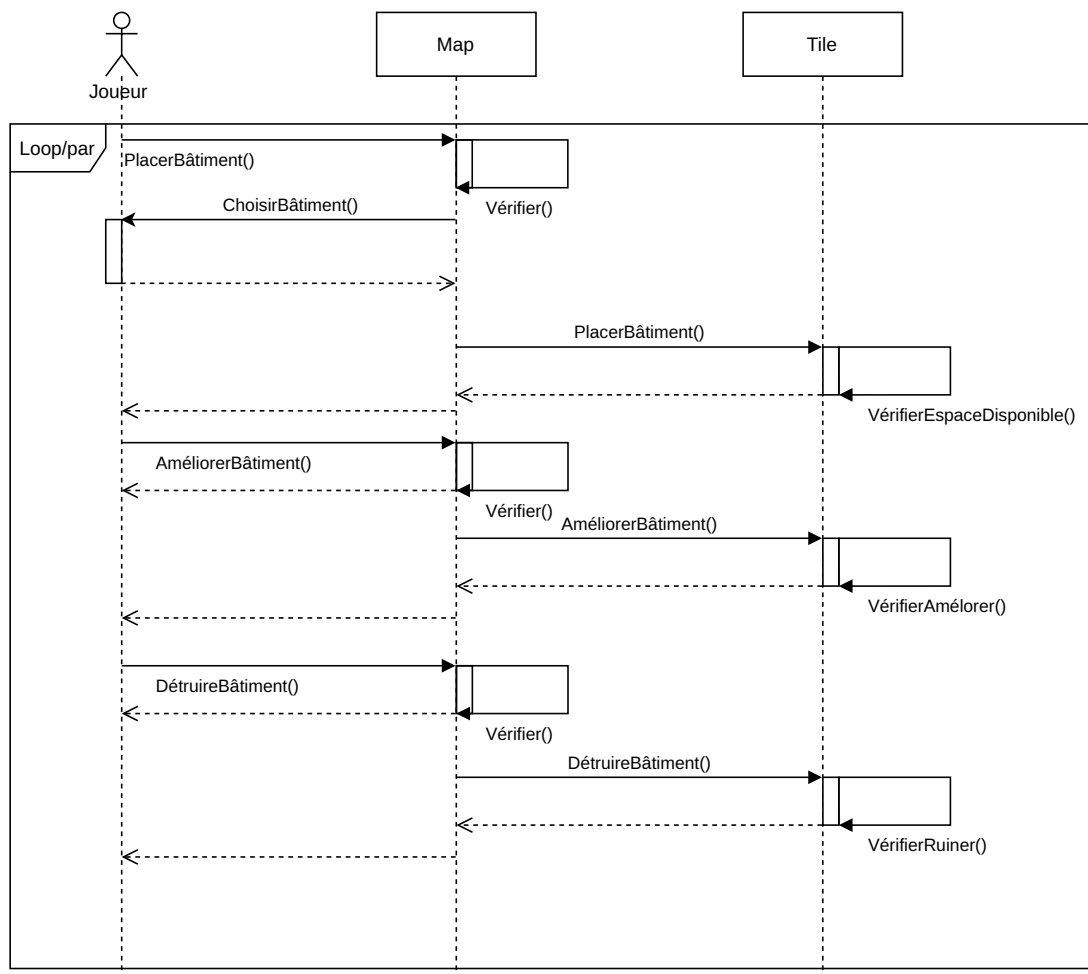


FIGURE 10 – Diagramme de séquence des actions du joueur sur les bâtiments

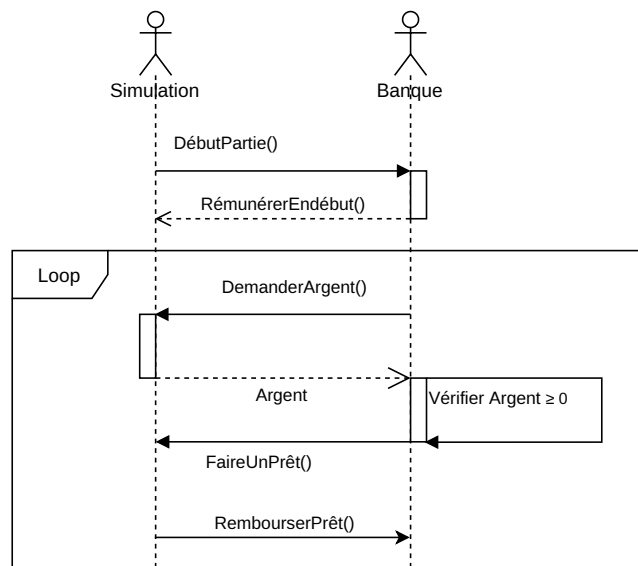


FIGURE 11 – Diagramme de séquence des actions de la banque

## 6 Diagramme de communication

Le diagramme de communication met en avant le sens des communications entre les différents objets constituant le coeur du jeu : le joueur, la simulation, les habitants et la zone de jeu.

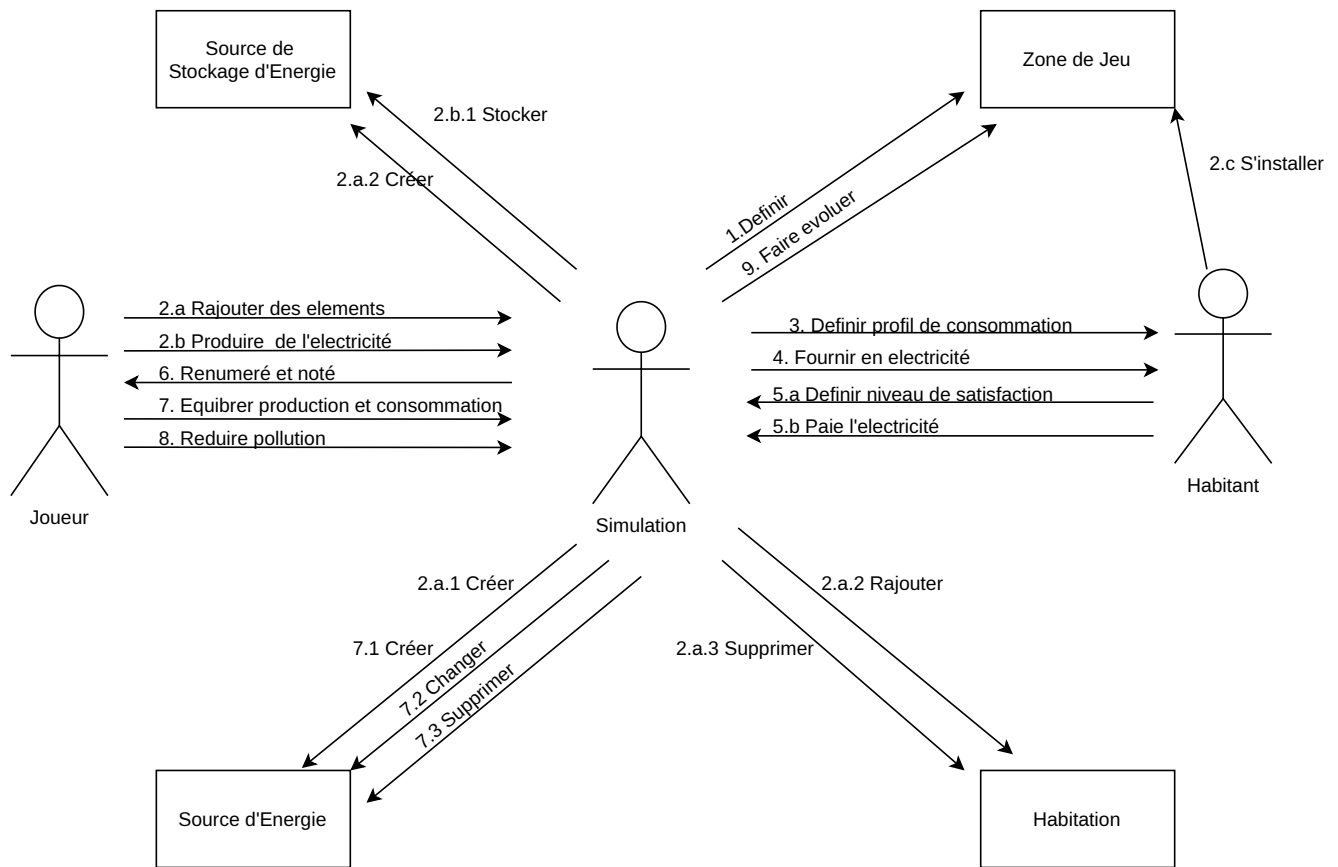


FIGURE 12 – Diagramme de communication