

## **Rendu du rapport de conception**

### **Sujet :** **Sim Power**

Ecrit par:

**AZINCOURT** Vincent

**DUBOIS** Jérôme

**GUILLEUX** Céleste

### **Informations complémentaires :**

Département Informatique: AP4A-P21

**BARBIER** Julien - **GECHTER** Franck - **KAS** Mohamed

# Sommaire

<b>Introduction :</b>	<b>2</b>
<b>I - Diagramme de cas d'utilisation</b>	<b>4</b>
<b>II - Les scénarii</b>	<b>5</b>
Construire une installation	5
Acheter des ressources à des tiers	6
Ajouter et retirer des ressources au joueur	6
Mettre à jour la zone de jeu	7
<b>III - Diagrammes de séquence</b>	<b>8</b>
Construire une installation	8
Mettre à jour la zone de jeu	9
<b>IV - Diagramme d'activité</b>	<b>10</b>
<b>V - Diagramme de classe</b>	<b>10</b>
Diagramme de classe du modèle :	10
Diagramme de classe du contrôleur :	12
<b>VI - Structure et changements du code</b>	<b>12</b>
<b>VII - Eléments graphiques</b>	<b>14</b>
<b>Conclusion</b>	<b>17</b>

# Introduction :

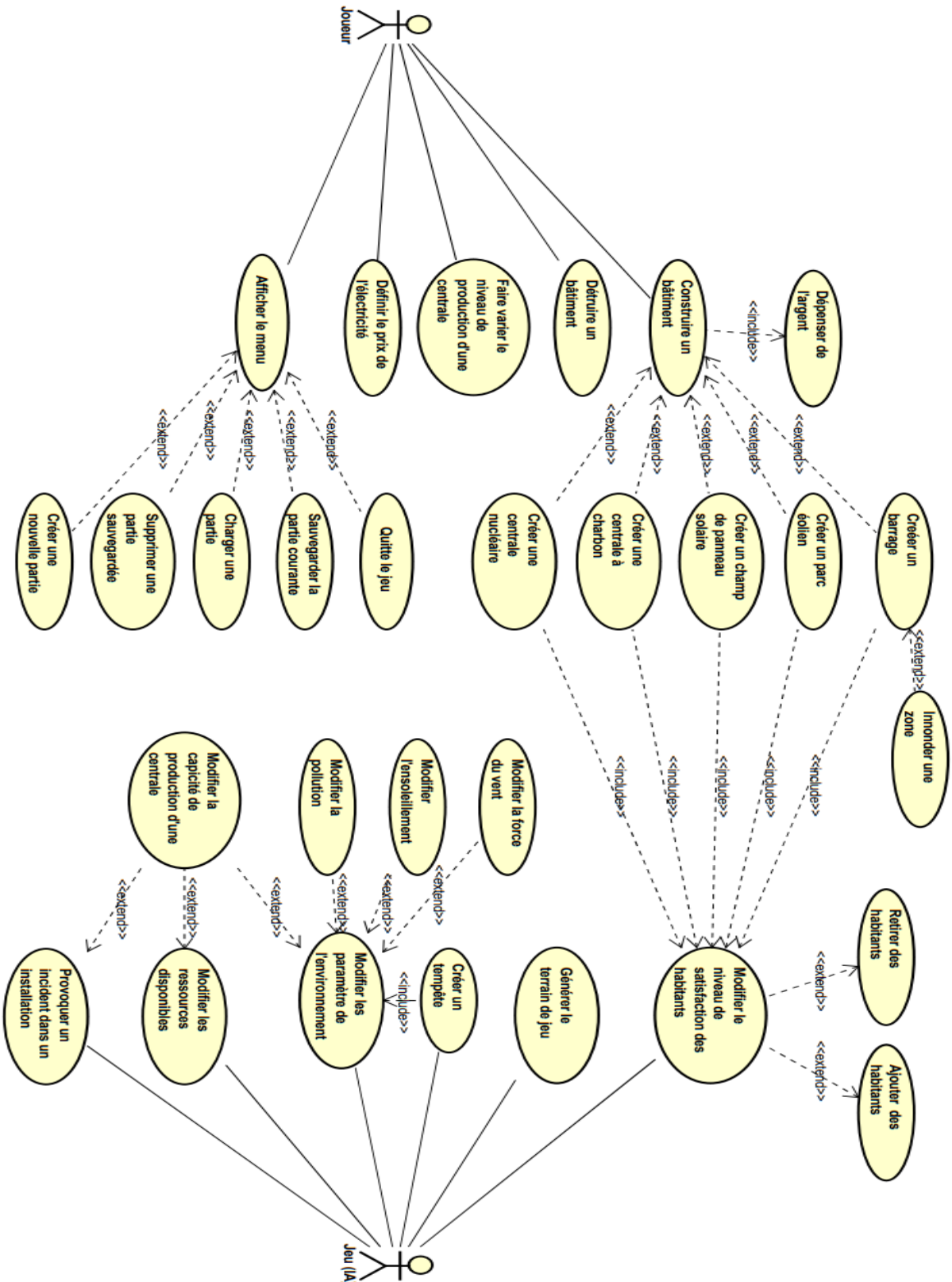
Ce document regroupe tous nos documents de conception et d'analyse du projet à savoir des descriptions textuelles des différentes utilisations possibles du site, des diagrammes des scénarios, d'objet, de séquence et d'activité, et de classe ainsi que quelques subtilités apportées à notre code.

Pour notre projet, nous avons choisi le sujet Sim Power. Nous avons opté pour un projet qui a des objectifs précis mais qui laisse un certain degré de liberté dans la conception et la gestion des éléments. L'objectif de ce jeu est de simuler de façon ludique l'acheminement en électricité de la zone de jeu. Ainsi, le joueur a la possibilité d'alimenter toute la population située sur la zone de jeu tout en minimisant le coût de l'énergie pour les habitants ainsi que la pollution (et/ou nuisance sonore). La zone de jeu est une grille composée de cases présentant des caractéristiques propres (construction, pollution, climat...).

Nous avons tout d'abord réalisé les diagrammes de cas d'utilisation qui permettent d'identifier toutes les fonctionnalités de l'utilisateur et du jeu. Après avoir établi les scénarii, nous avons développé des diagrammes de séquence et d'activité, pour la représentation temporelle des interactions entre objets pour une opération donnée soit une analyse temporelle des scénarii. Et enfin, le diagramme de classe qui permet de structurer l'ensemble des ressources du système définie comme une ensemble de relation entre classes.

On utilisera pour ce projet la version de Java 11 avec JavaFX et IntelliJ comme IDE. En complément, on utilisera le moteur de production Gradle.

# I - Diagramme de cas d'utilisation



## II - Les scénarii

### 1. Construire une installation

Description	Construire un bâtiment (centrale, entrepôt, "grande batterie", mine)
Acteurs	Le joueur Le système de jeu Sim Truc
Evènement déclencheur	Le joueur souhaite ajouter un nouveau bâtiment sur le terrain de jeu
Scénario de base	Le joueur sélectionne le bâtiment approprié dans la liste des bâtiments Le joueur sélectionne un emplacement pour construire le bâtiment Les ressources du joueur sont dépensées si elles sont suffisantes Le bâtiment est construit et applique ses bonus ou malus à la zone et à la production du joueur
Variantes	Pas assez de ressources à l'étape 3 : Le bâtiment n'est pas construit et l'action est interrompue
Préconditions	
Post-conditions	
Contraintes	

## 2. Acheter des ressources à des tiers

Description	Acheter des ressources à des tiers
Acteurs	Le joueur Le système de jeu Sim Power
Evènement déclencheur	Le joueur n'a pas assez de ressources
Scénario de base	Le joueur accède à l'interface d'achat et de vente de ressources Le joueur sélectionne une ressource à vendre ou à acheter Le joueur finalise la transaction en sélectionnant un montant
Variantes	Si le joueur n'a pas assez de ressources à vendre, ou pas assez d'argent pour acheter, le bouton de validation sera grisé à l'étape 3
Préconditions	
Post-conditions	
Contraintes	

## 3. Ajouter et retirer des ressources au joueur

Description	Ajouter et retirer des ressources au joueur
Acteurs	Le système de jeu Sim Power
Evènement déclencheur	Tick
Scénario de base	Pour chaque bâtiment : Le système de jeu vérifie l'état du bâtiment Le système de jeu retire les ressources consommées par le bâtiment au joueur Le système de jeu ajoute les ressources du bâtiment aux ressources du joueur
Variantes	A l'étape 2, si le bâtiment est dans un état invalide, le bâtiment ne produit ni ne dépense et la boucle s'arrête pour lui A l'étape 3, si le bâtiment n'a pas assez de ressources, il ne produit pas et applique une pénalité au joueur A l'étape 4, si les réserves du joueur sont dépassées, l'excédent est perdu
Préconditions	Le jeu n'est pas en pause

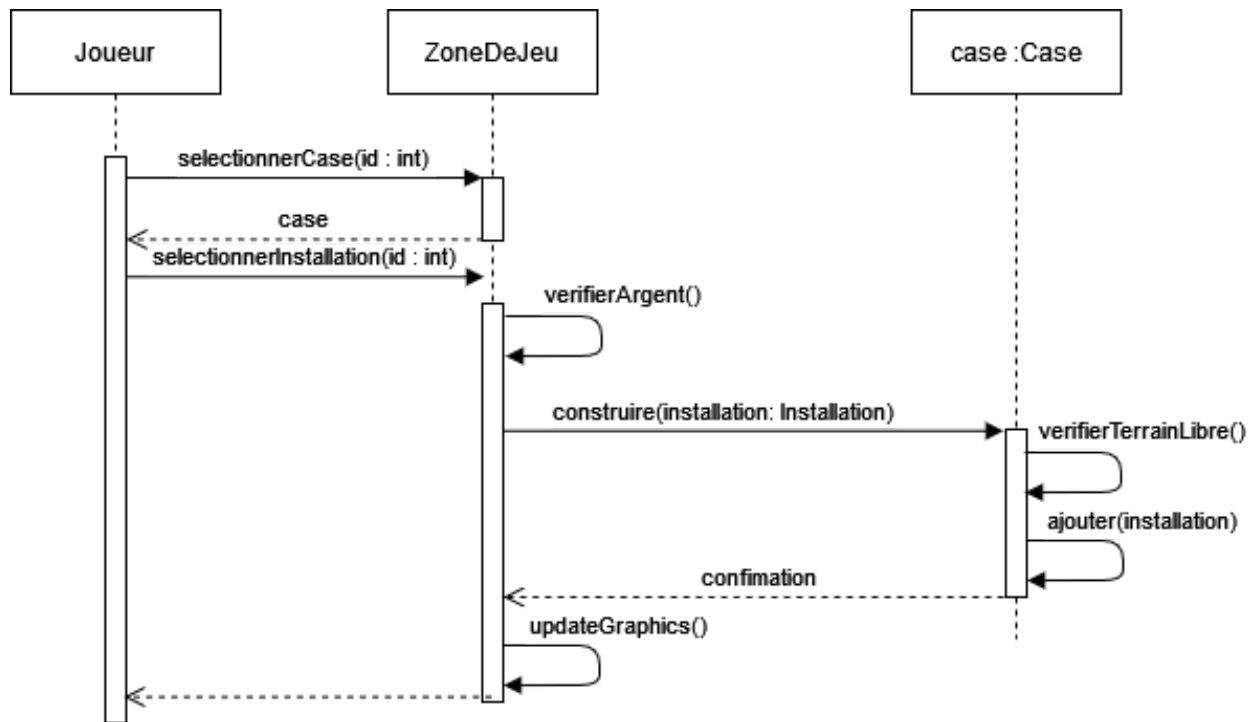
Post-conditions	
Contraintes	

#### 4. Mettre à jour la zone de jeu

Description	Mettre à jour la zone de jeu
Acteurs	Le système de jeu Sim Power
Evènement déclencheur	Tick
Scénario de base	Pour chaque case: Le système met à jour le niveau de pollution Le système met à jour les ressources disponibles en fonction des installations Le système met à jour le niveau de satisfaction des habitants
Variantes	S'il y a une tempête, le système change le climat en conséquence (vent, ensoleillement)
Préconditions	Le jeu n'est pas en pause
Post-conditions	
Contraintes	

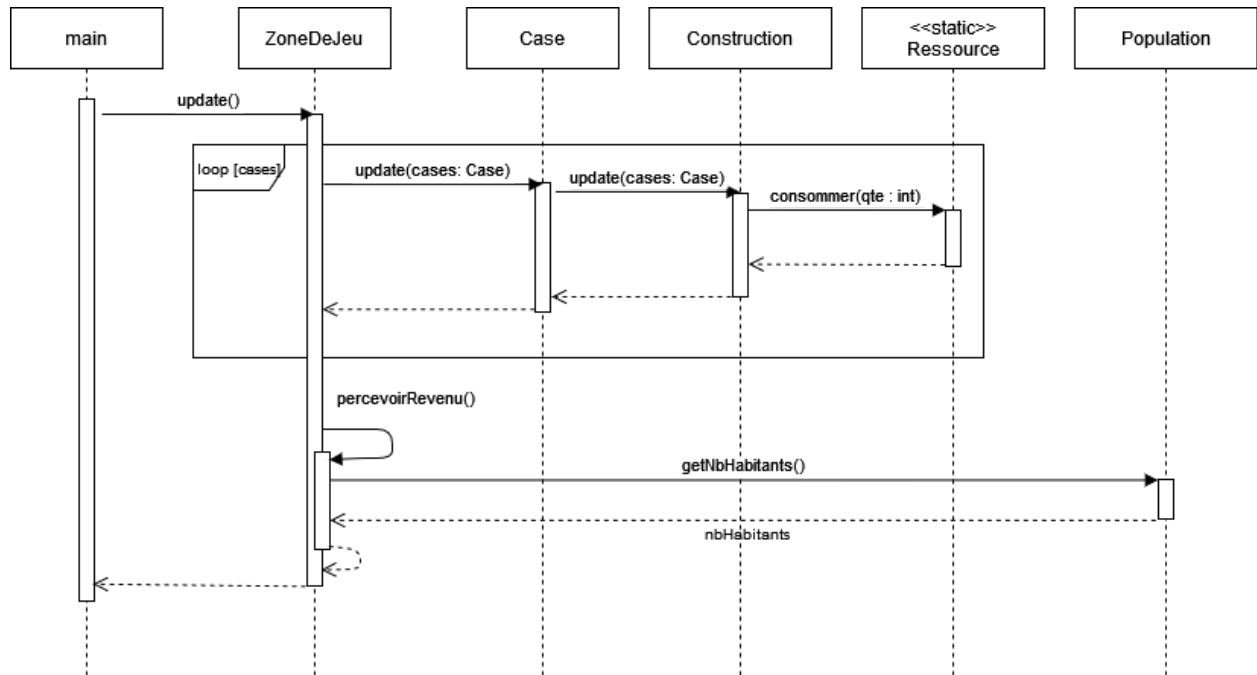
### III - Diagrammes de séquence

#### 1. Construire une installation

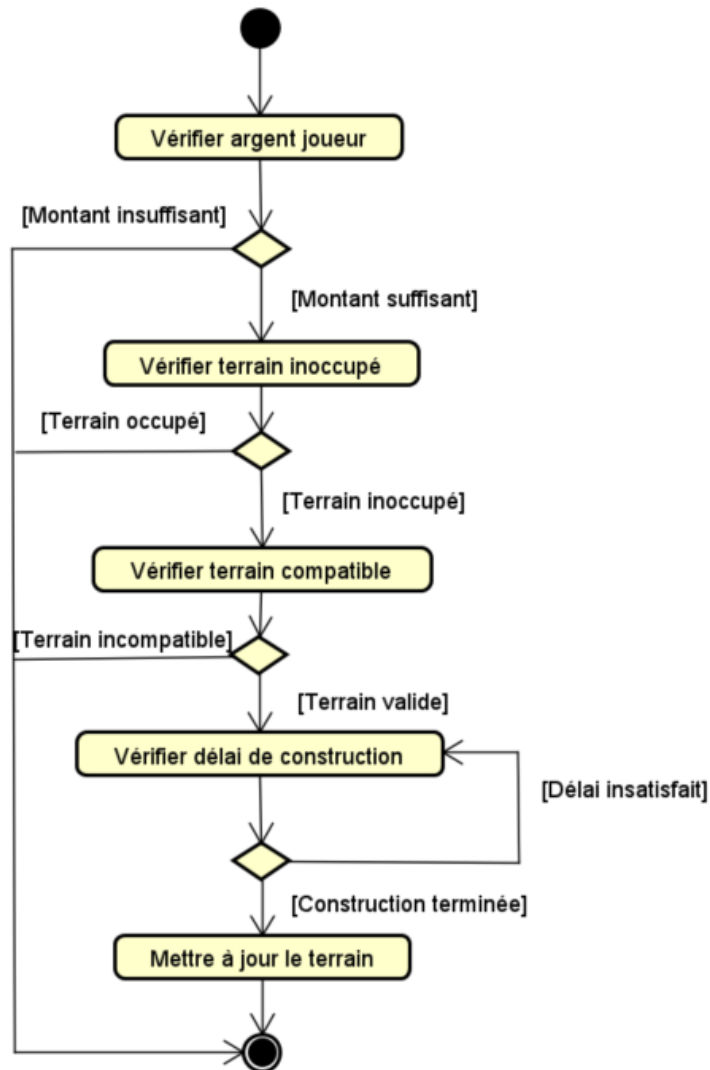




## 2. Mettre à jour la zone de jeu



## IV - Diagramme d'activité

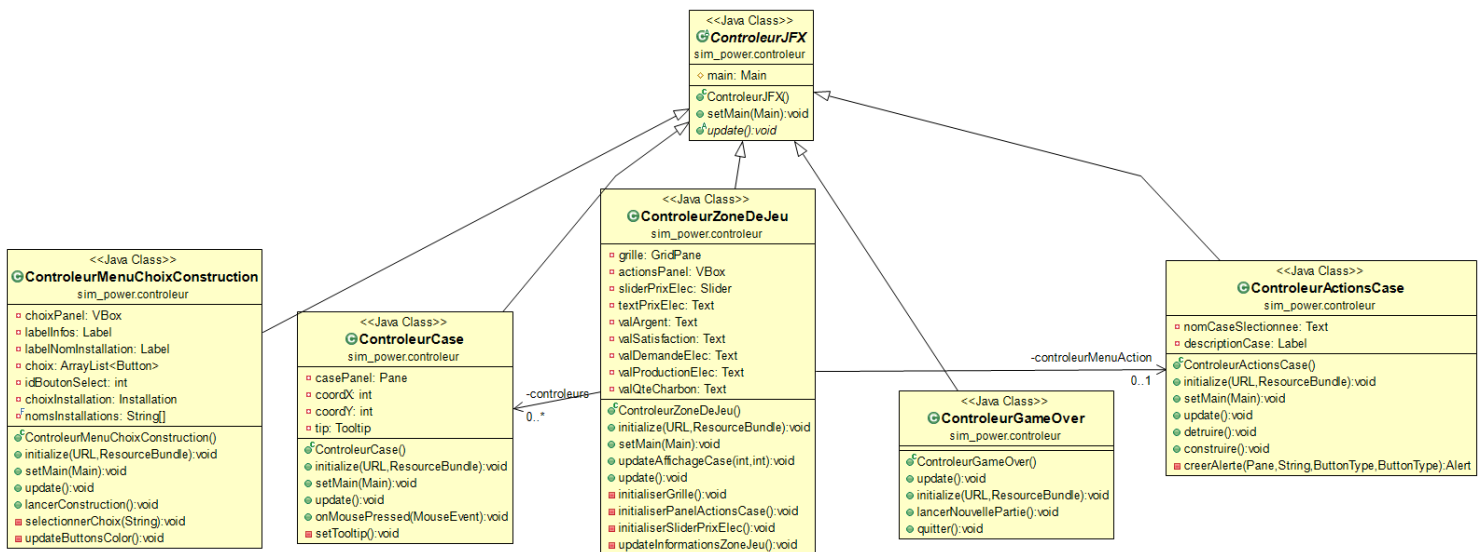


## V - Diagramme de classe

1. Diagramme de classe du modèle :



## 2. Diagramme de classe du contrôleur :



## VI - Structure et changements du code

Lors de la conception du projet, nous nous sommes rendu compte que des éléments manquaient pour, d'une part, assurer une bonne communication entre nos différentes classes et, d'autre part, proposer une description à chacune des cases de la zone de jeu. Désormais, plusieurs états sont donc possibles pour une case :

- Case vide :
  - Plaine
- Construction :
  - Barrage
  - Champs de panneau solaires
  - Champs d'éoliennes
  - Centrale nucléaire
  - Mine de charbon
  - Zone d'habitation
- Reserve :
  - Cours d'eau
  - Réserve naturelle de charbon

- 
- État critique/Accident de la case :
  - Zone inondée

Les cases de ressources sont aléatoires pour offrir une expérience de jeu unique à chaque nouvelle zone de jeu.

De plus, nous nous sommes focalisés sur l'incident tempête qui perturbera le déroulement de la partie et modifiera les paramètres météorologiques(vent et soleil) de la zone de jeu.

Au fur et à mesure de la progression de la partie, le joueur subit donc quelques pénalités si la satisfaction des habitants n'est pas maximisée. La satisfaction locale de l'habitant de la case, dépendant du niveau de pollution.

- Celle -ci se mesure de -125 à 125.
- Une pollution plus élevée a d'autant plus d'effets que son importance.
- Une pollution faible a peu d'impact sur la satisfaction.
- Le vent a aussi des effets mineurs sur la satisfaction.

La satisfaction globale de la zone de jeu se calcule suivant :

- La moyenne de la satisfaction de chaque habitation (expliquée juste au-dessus),
- Le prix de l'électricité
- La subvention des besoins énergétiques.

Ainsi, nous avons établi une nouvelle règle : Si le nombre d'habitation est réduit à 0, le joueur perd: Game over.

## VII - Eléments graphiques

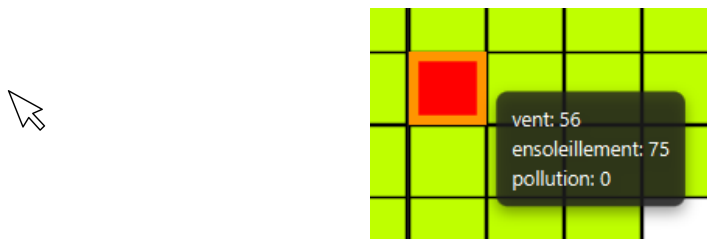
L'utilisation de javafx permet de structurer les éléments graphiques à l'aide de fichiers d'extension ".fxml" qu'on peut éditer avec l'outil SceneBuilder notamment. L'interface graphique de l'application a été décomposée en 5 fichiers FXML, soit 5 vues:

- "VueCase.fxml", qui affiche une case
- "VueActionCase.fxml", qui affiche les actions à effectuer sur une case sélectionnée
- "VueMenuChoixConstruction.fxml", qui affiche un menu qui permet de choisir une installation à construire sur une case
- "VueZoneDeJeu.fxml", c'est la vue principale qui affiche les cases, les actions disponibles, et les informations sur le jeu
- "VueGame.fxml", qui annonce la fin de la partie, et permet d'en recommencer une nouvelle

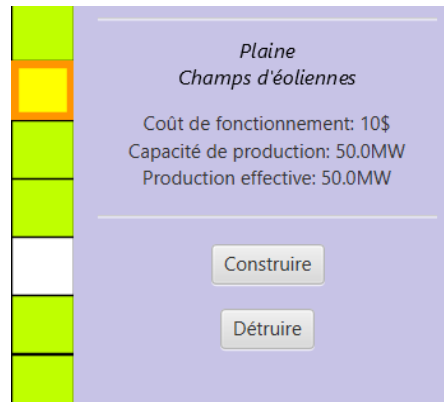
Chaque fichier est lié à une classe java, un contrôleur, qui permet au joueur d'interagir avec l'interface, et par conséquent de modifier les données du jeu (le modèle).

### Fonctionnalités par vue/contrôleur:

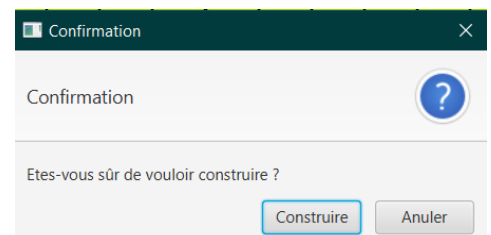
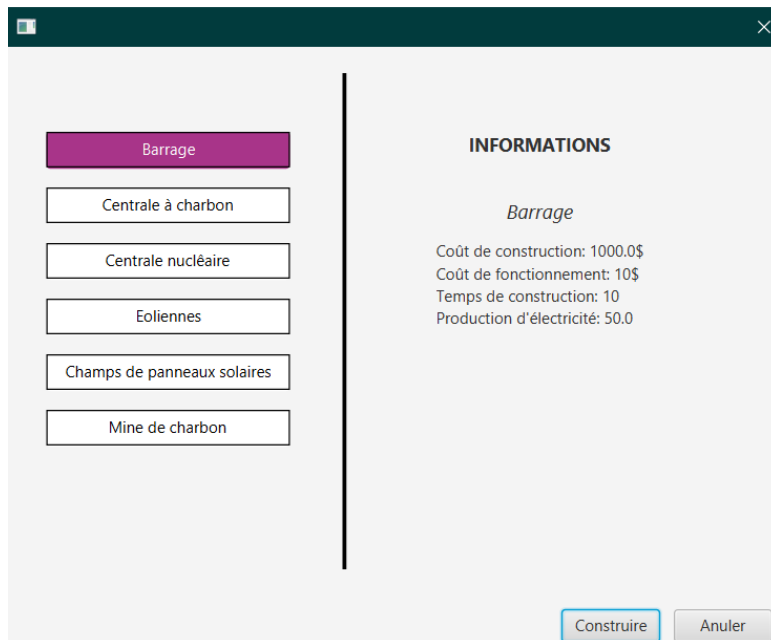
- Pour les cases
  - Si la souris survole une case, une info-bulle apparaît avec les valeurs de la météo et de la pollution.
  - Si un clic est détecté, la case est sélectionnée, elle apparaît alors avec une bordure jaune



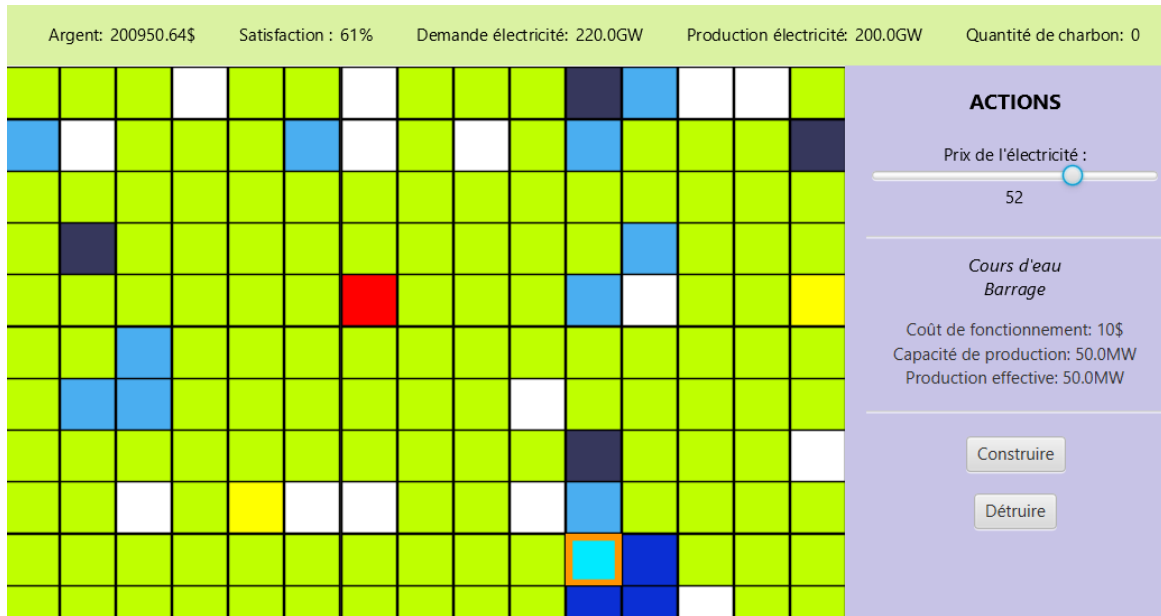
- Pour le menu latéral de modification de la case sélectionnée:
  - Le libellé de la case et une description sont afficher
  - Le bouton “construire” ouvre le menu de construction
  - Le bouton “détruire” supprime l’installation d’une case s’il y en a une



- Pour le menu de construction
  - Cliquer sur une installation la sélectionne et affiche son libellé et sa description
  - Le bouton “Confirmer”, affiche une alerte qui demande une seconde confirmation, si le joueur valide alors le bâtiment est créé et le menu disparaît
  - Le bouton “Annuler” fait disparaître le menu



- Pour la zone de jeu
  - Affichage des cases
  - Affichage des informations globales sur la partie, en haut de la fenêtre
  - Le slider permet de modifier le prix de l'électricité



- Pour le “game over”
  - Le bouton “Nouvelle partie” crée une nouvelle zone de jeu
  - Le bouton “Quitter” ferme l’application



#### Remarque :

Correction d'un bug à cause duquel les caractères accentués et spéciaux ne s'affichent pas correctement. Pour résoudre ces problèmes d'encodage, nous avons utilisé le code spécifique du caractère (codage *Unicode*)(ex: \u00EA = é).



# Conclusion

Avec ce document, nous avons pu mettre en lumière l'étendue des fonctionnalités de notre projet. Nous avons également une représentation de la structure du système sous la forme d'un diagramme de classe. L'étape de conception des diagrammes UML nous a permis d'organiser nos idées et de revoir les limites de notre projet.

L'étape de *coding* nous a permis de mettre à jour plusieurs aspects vus à l'étape précédente et de corriger les incohérences de notre projet. C'est l'application de la phase théorique du sujet.

Ainsi, nous pouvons fournir un jeu permettant de simuler de façon ludique l'acheminement en électricité de la zone de jeu. Il s'agit d'une expérience unique alliant l'aléatoire à une stratégie d'optimisation des ressources.