

Kronologic

Dietrich Corentin,
Dziezduk Mathieu,
Mougin Enzo



Sommaire

- Introduction
- La Programmation par Contraintes
- Liste des fonctionnalités
- Maquette de l'application
- Diagrammes
- Planning des itérations
et rôle de chacun
- Bilan itération 1
- Démonstration Choco-Solver



Introduction

Objectifs du projet:

- Assimilation et intégration des règles
- IA déduction
 - dont une IA de déduction heuristique
 - et une IA de déduction Choco-Solver
- IA assistante
 - dont une IA assistante heuristique
 - et une IA assistante Choco-Solver
- IA joueuse
- Générateur de scénarios



La Programmation par Contraintes

- **Les variables**

$P_{i,t}$: Position du personnage i au temps t ($1 \leq t \leq 6$).

- **Les domaines**

Les salles sont numérotées de 1 à 6, donc la variable $P_{i,t}$ a un domaine : $\{1, 2, 3, 4, 5, 6\}$.

- **Les contraintes**

Contraintes de règles

1. Déplacement obligatoire
2. Déplacement dans des salles adjacentes

Contraintes de début de partie

1. Au début de la partie ($t = 1$),
la position initiale de
chaque personnage est annoncée

Contraintes au cours de la partie

1. Public (Temps choisi)
2. Privé (Temps choisi)
3. Public (Personnage choisi)
4. Privé (Personnage choisi)

Fonctionnalités liées à : l'interface utilisateur

- Créer une page d'accueil
- Créer l'interface de jeu
- Permettre au joueur de poser une question
- Permettre au joueur de formuler une hypothèse
- Permettre au joueur de formuler des déductions
- Permettre au joueur d'accéder aux déductions de l'IA
- Permettre au joueur de demander de l'aide à l'IA assistante
- Permettre au joueur de changer d'affichage
- Permettre au joueur de visualiser les règles
- Permettre au joueur de visualiser le film des déplacements des personnages

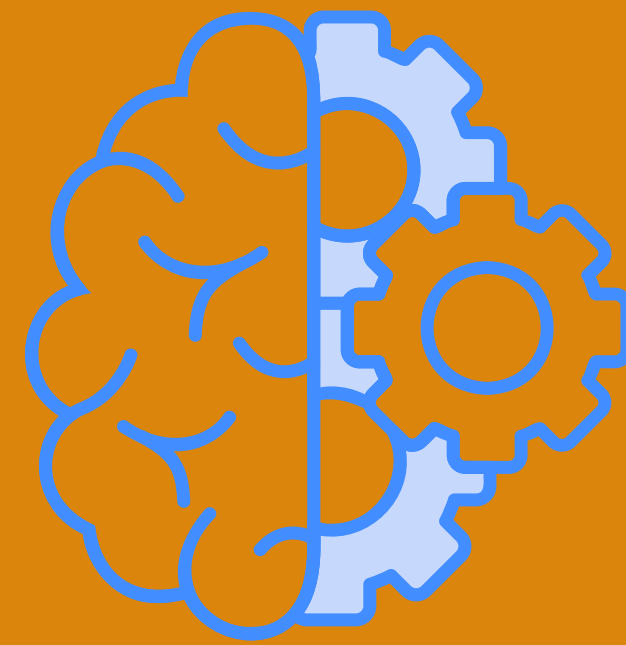
Fonctionnalités liées à : l'IA de déduction

On disposera de deux IA de déduction : une heuristique et une utilisant Choco-Solver. Bien qu'ayant des fonctionnements différents, elles devront posséder les mêmes fonctionnalités.

Analyser les indices collectés et en tirer des conclusions logiques



Proposer une hypothèse logique basée sur les indices



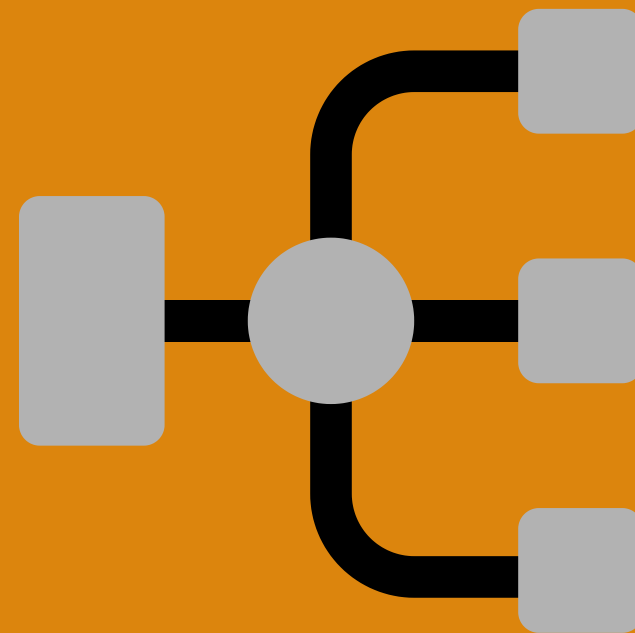
Fonctionnalités liées à : l'IA d'assistance

On disposera également de deux IA d'assistance, une heuristique et une utilisant Choco-Solver. Ayant le même objectif, elles posséderont les mêmes fonctionnalités.

Recommander une question optimale à poser



Simuler les conséquences de différentes questions



Corriger les déductions du joueur



Partie optionnelle

IA joueuse

Raisonner sur les indices collectés et en tirer des déductions logiques

Poser les questions permettant d'obtenir les informations les plus pertinentes pour la complétion de l'enquête

Générer des hypothèses intermédiaires qui seront ajustées en fonction des nouveaux indices

Éliminer les options impossibles au fur et à mesure afin de pouvoir raisonner sur un ensemble d'informations plus petit

Formuler une déduction une fois qu'elle a trouvé le coupable, le temps et le lieu du crime

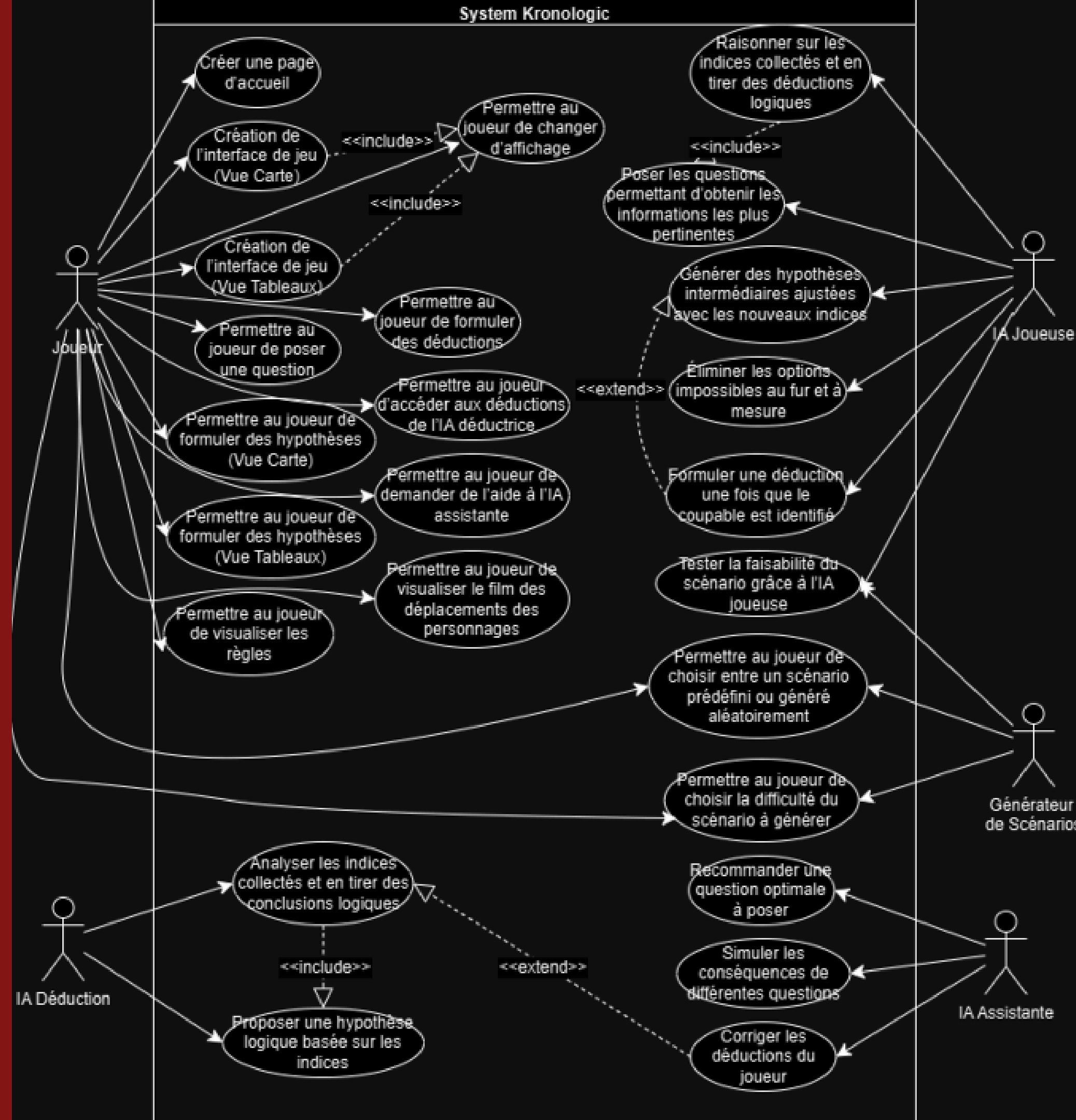
Générateur de scénarios

Permettre au joueur de choisir entre un scénario prédéfini ou un nouveau scénario généré aléatoirement

Permettre au joueur de choisir la difficulté du scénario à générer

Tester la faisabilité du scénario grâce à l'IA joueuse avant de le proposer au joueur (une seule solution possible et trouvable)

Diagramme de Cas d'Utilisations



Kronologic

Jouer

IA Joueuse

Quitter



☒ Afficher les présences



Joueur

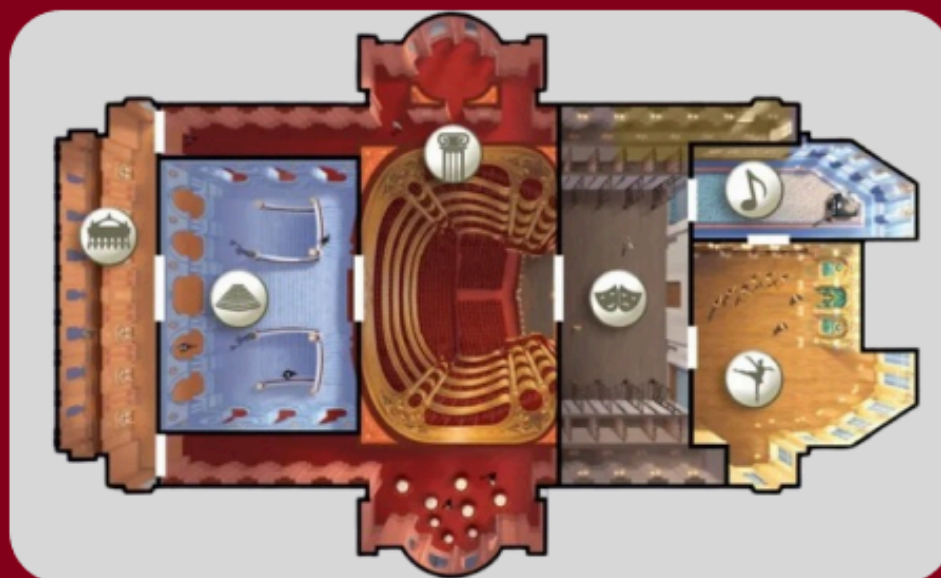


Partie

☒ Afficher les absences



1

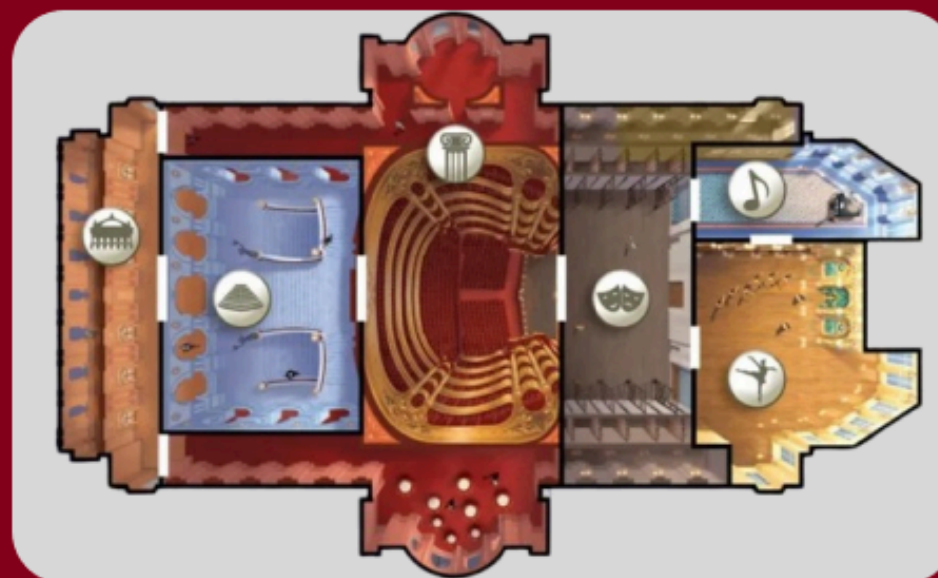


Historique des indices :

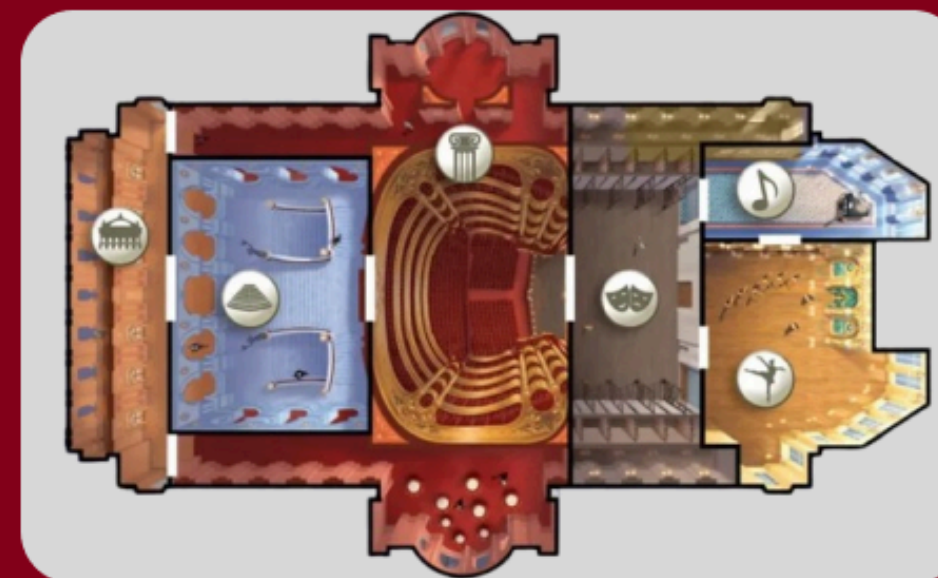
Tour 2 : Il y a 0 personne La Salle au Temps 1.

Tour 1 : L'Aventurière est passée 0 fois dans La Salle.

2



3



A B C D J S

☐ Hypothèse

X

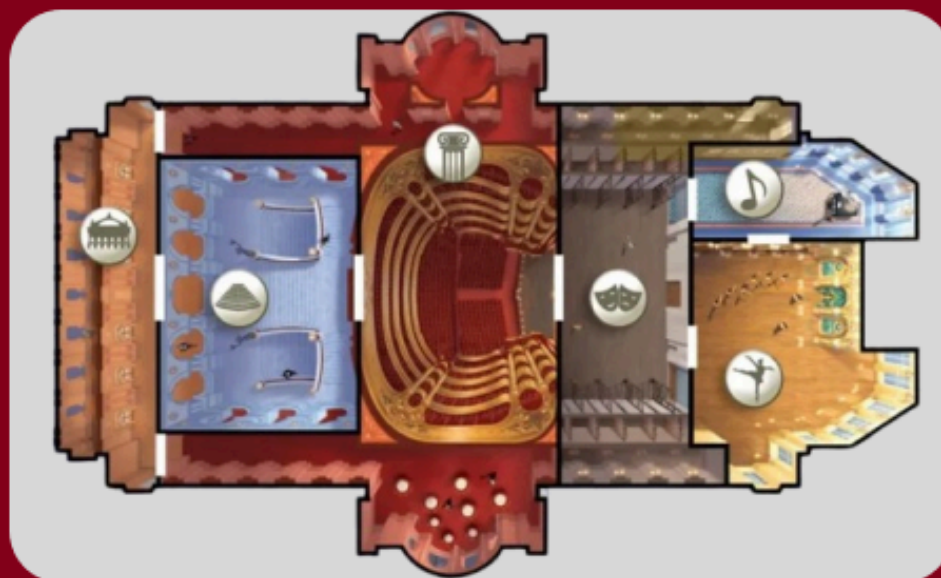
☐ Absence

☒ Masquer les hypothèses

Changer affichage

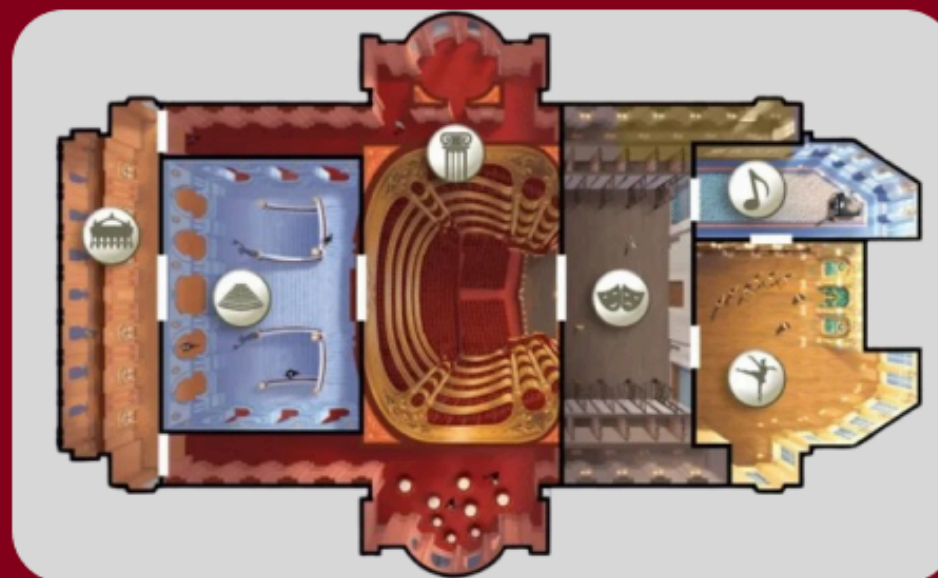
Déduction de l'IA

4



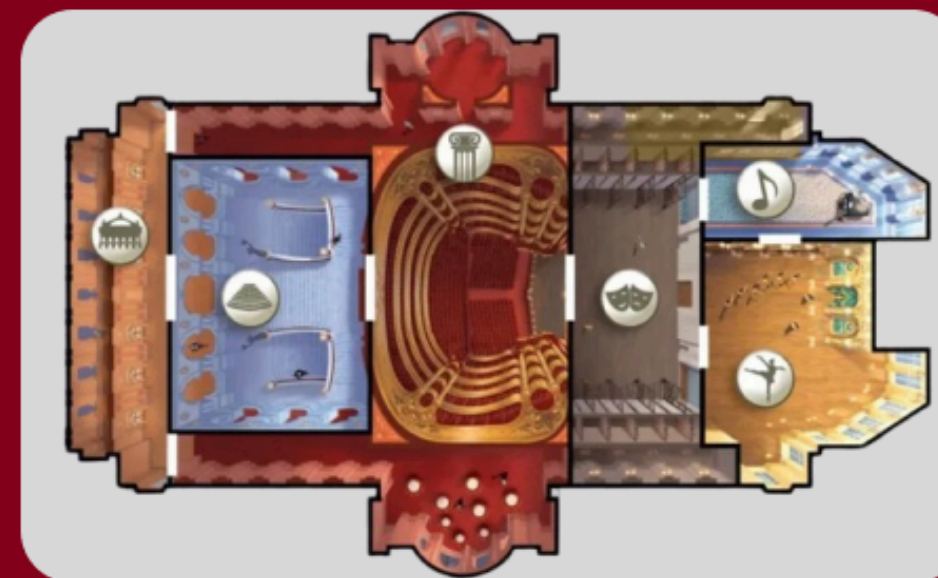
Faire une déduction

5



Poser une question

6

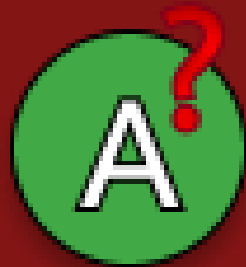


Demander un indice

Pour augmenter le nombre de personnage il faut double cliquer sur le pion. Cela vous permet de choisir un nombre entre 0 et 6.



Pions d'hypothèse
de présence



Pions d'hypothèse
de nombres



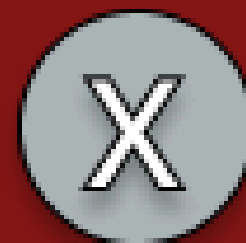
Pions d'hypothèse
d'absence



Pions de présence



Pions de nombres



Pions d'absence





Changer affichage

Déduction de l'IA

Historique des indices :

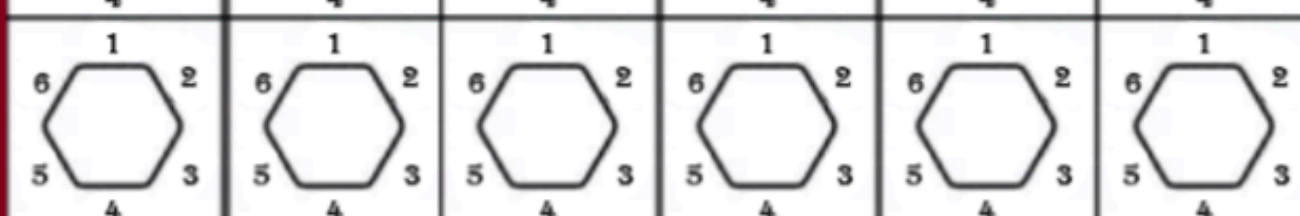
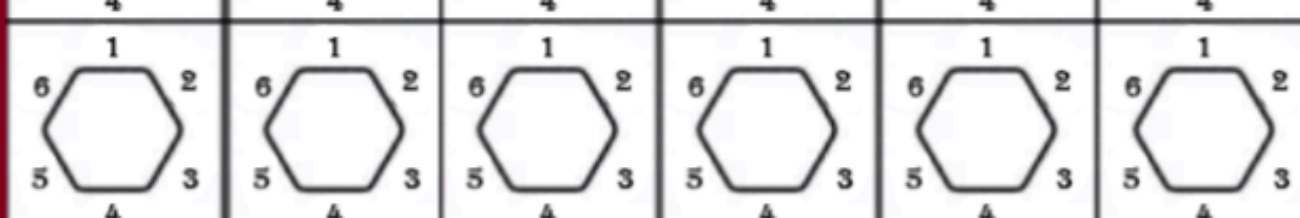
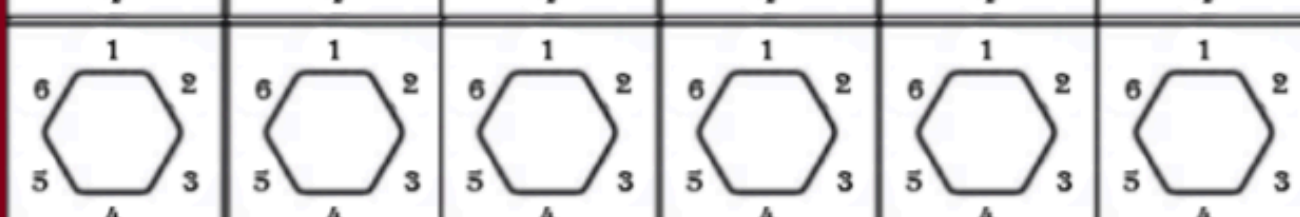
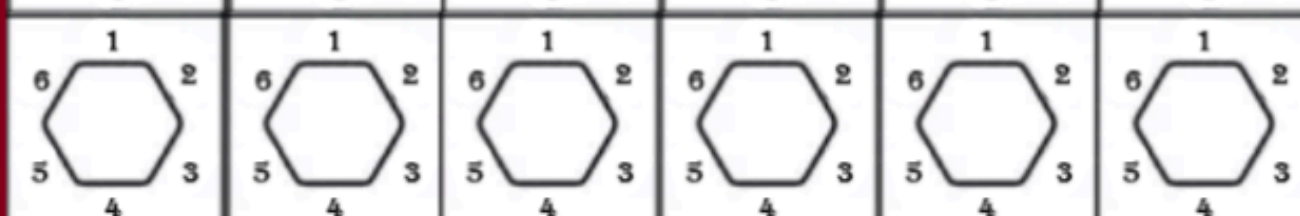
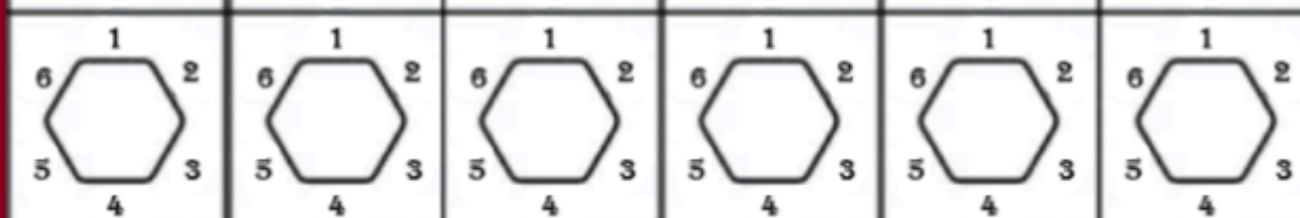
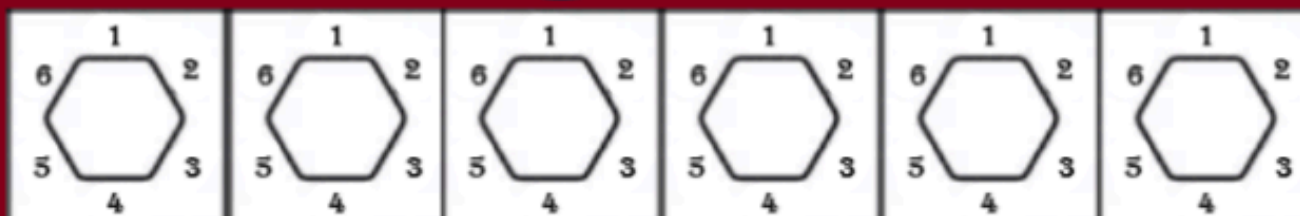
Tour 2 : Il y a 0 personne La Salle au Temps 1.

Tour 1 : L'Aventurière est passée 0 fois dans La Salle.

Poser une question

Demander un indice

Faire une déduction



A

B

C

D

S

J

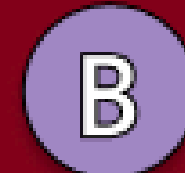


Posez une question :

Choisissez un Lieu :



Choisissez un Temps ou un Personnage :



Quitter

Valider

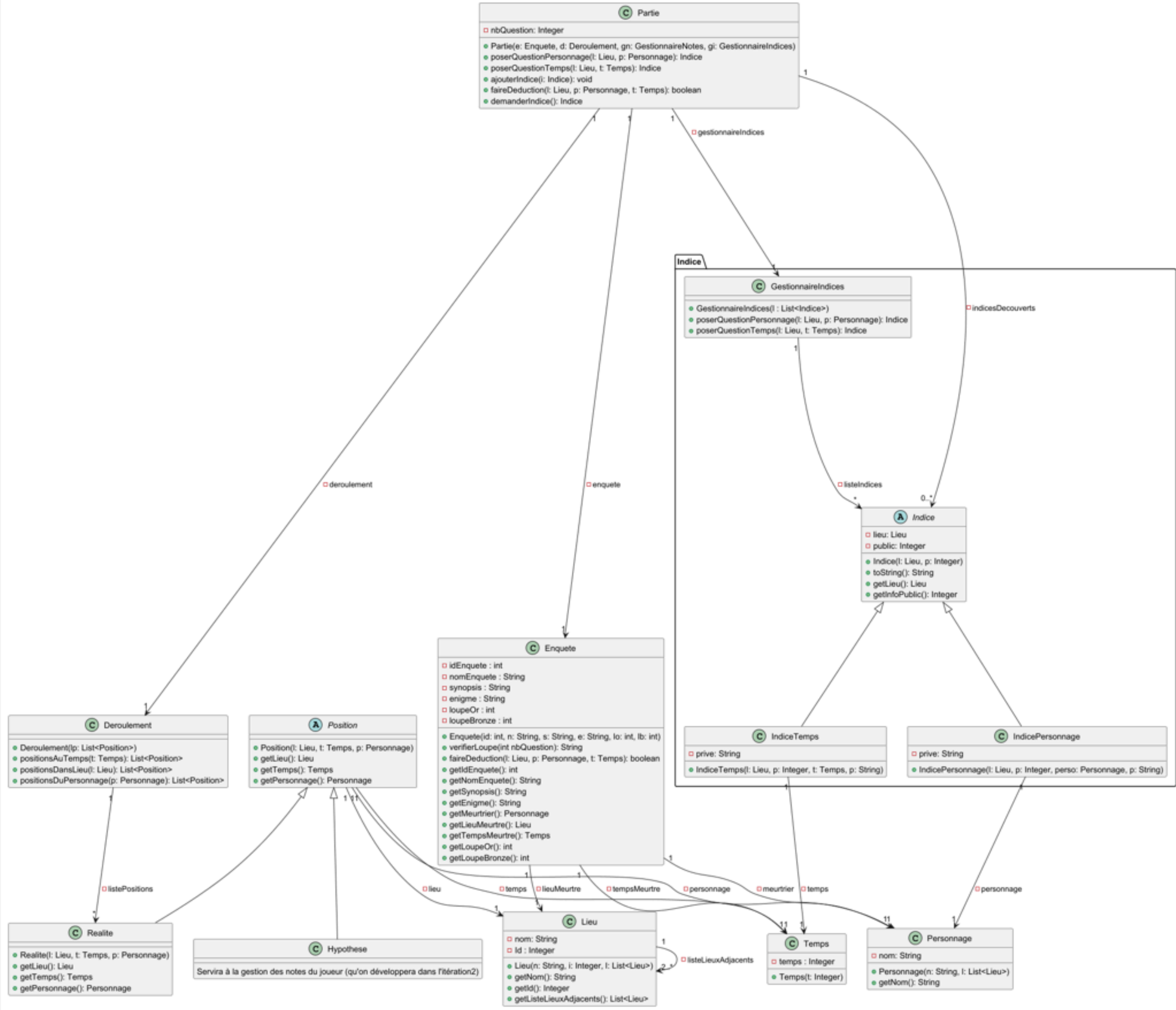


Diagramme de classe partie Kronologic

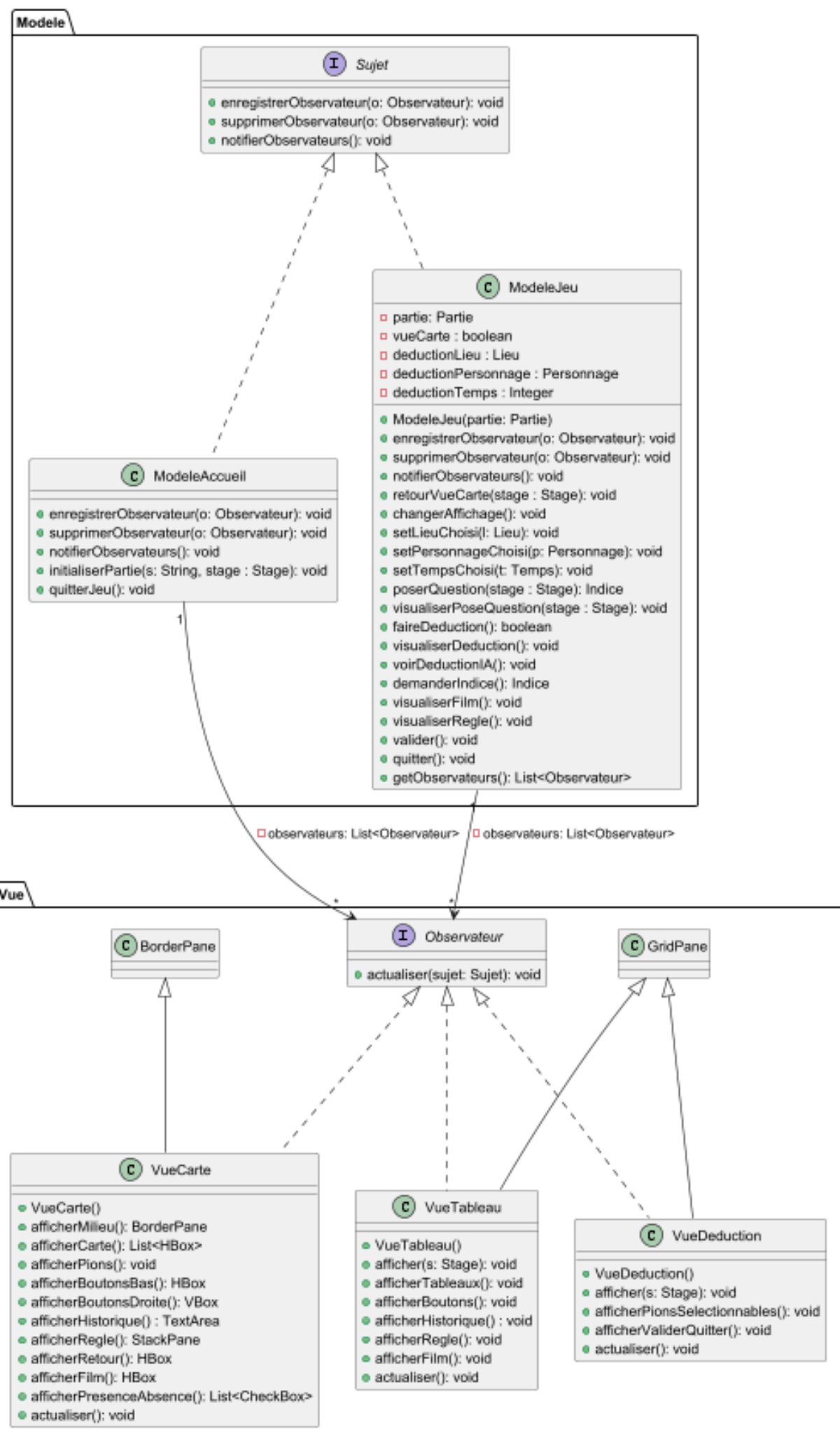


Diagramme de classe partie MVC

Diagramme de séquence (poser une question)

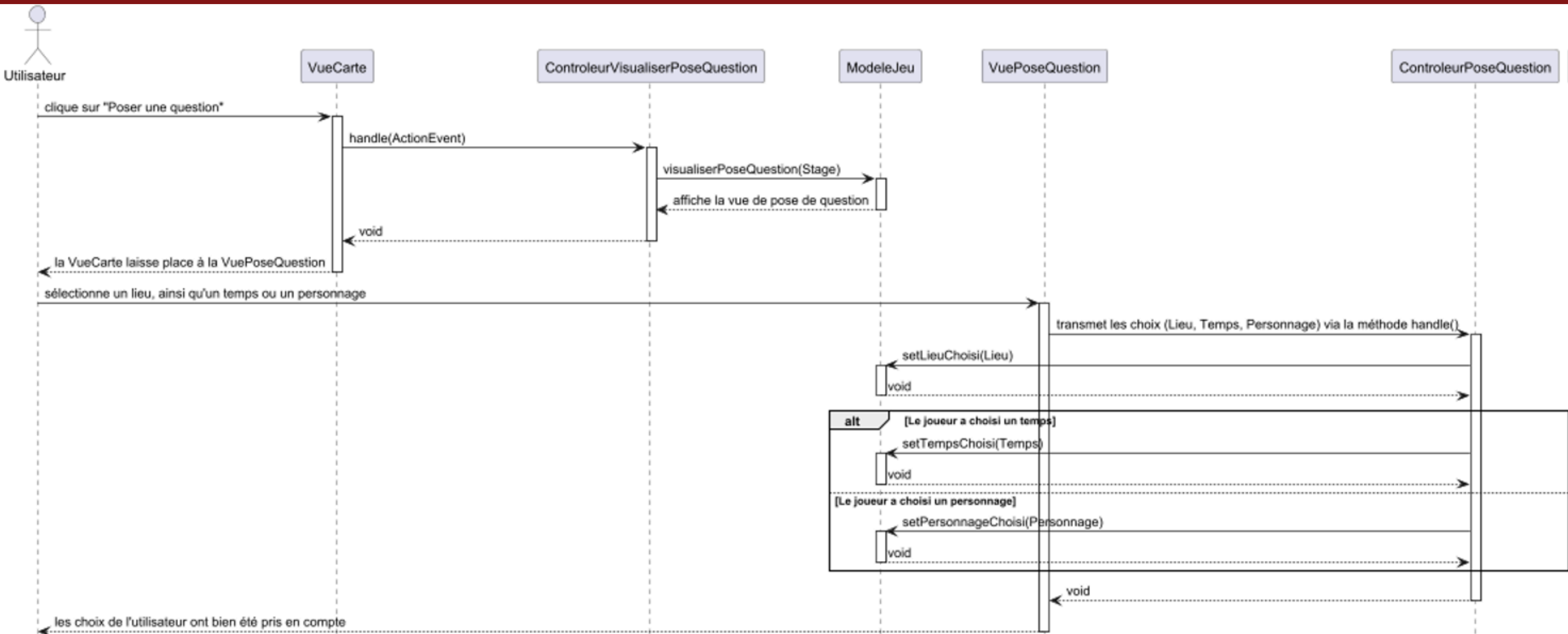
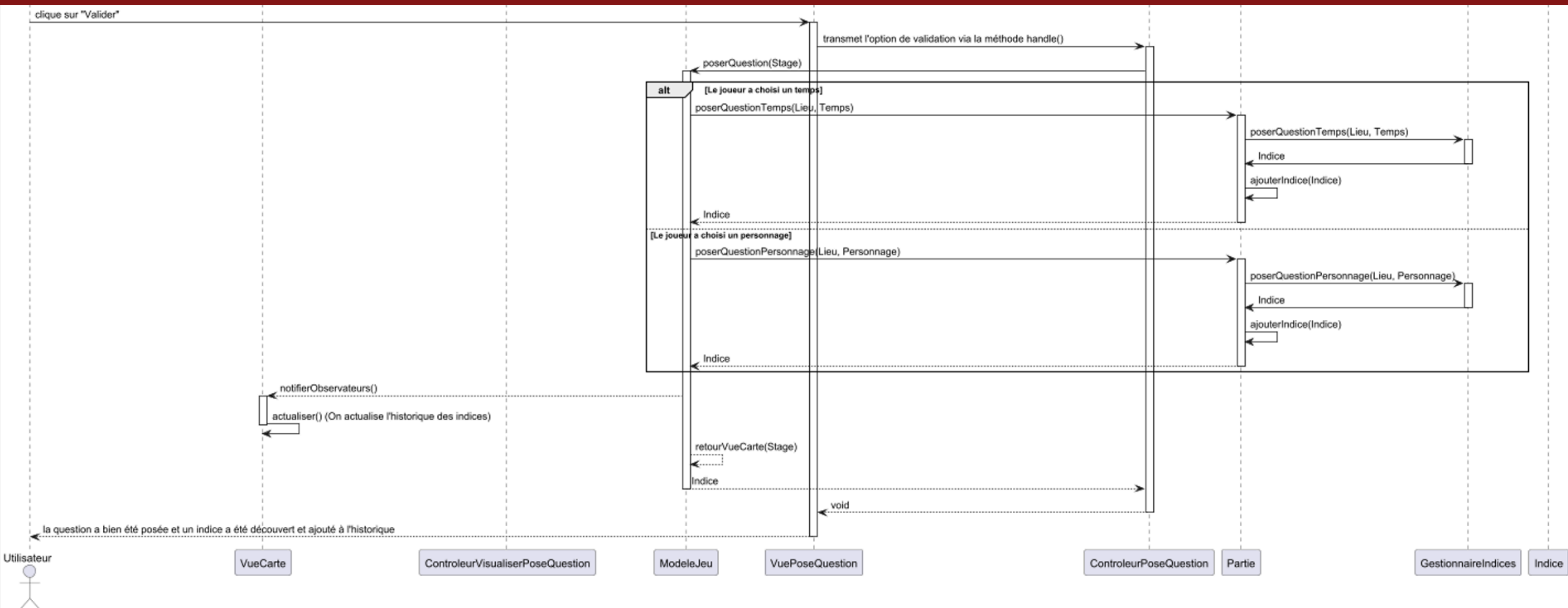


Diagramme de séquence (poser une question)





	Objectifs principaux	Rôle des membres du groupe
Itération 1	<ul style="list-style-type: none"> Compréhension de Choco-Solver Réaliser des prototypes Mettre en place les bases du jeu 	<ul style="list-style-type: none"> Mathieu : se familiariser avec Choco-Solver Enzo et Corentin : mettre en place les bases du jeu
Itération 2	<ul style="list-style-type: none"> Développer l'interface graphique Implémenter les fonctionnalités principales de jeu 	Chacun s'occupera de différentes fonctionnalités et de leurs représentations graphiques
Itérations 3 et 4 (IA de déduction)	<ul style="list-style-type: none"> Développer deux versions Comparer les performances Intégrer les suggestions de l'IA à l'interface graphique 	<ul style="list-style-type: none"> Mathieu : Implémenter l'IA reposant sur Choco-Solver Enzo et Corentin : Implémenter l'IA heuristique
Itérations 5 et 6 (IA d'assistance)	<ul style="list-style-type: none"> Développer deux versions Comparer les performances Permettre une interaction fluide entre le joueur et l'IA 	<ul style="list-style-type: none"> Mathieu : Implémenter l'IA reposant sur Choco-Solver Enzo et Corentin : Implémenter l'IA heuristique et les interactions
Itération 7	<ul style="list-style-type: none"> Corriger les éventuels bugs et erreurs Finaliser ce qui n'a pas pu être terminé 	Aucune répartition n'a été définie pour le moment, car nous ne pouvons pas nous projeter aussi loin.

Itération 1

Choco-Solver (Comparaison de performance)

Traveling Salesman Problem (TSP)

Contraintes

17 villes

- Heuristique : 327 ms
- Choco Solver : 4171 ms

30 villes :

- Heuristique : Java heap space (mémoire insuffisante)
- Choco Solver : 13295 ms

- Chaque ville doit être visitée une seule fois.
- Le trajet commence et se termine à la même ville.
- Trouver un chemin qui minimise la distance totale ou le coût total.

Présentation Interface Kronologic