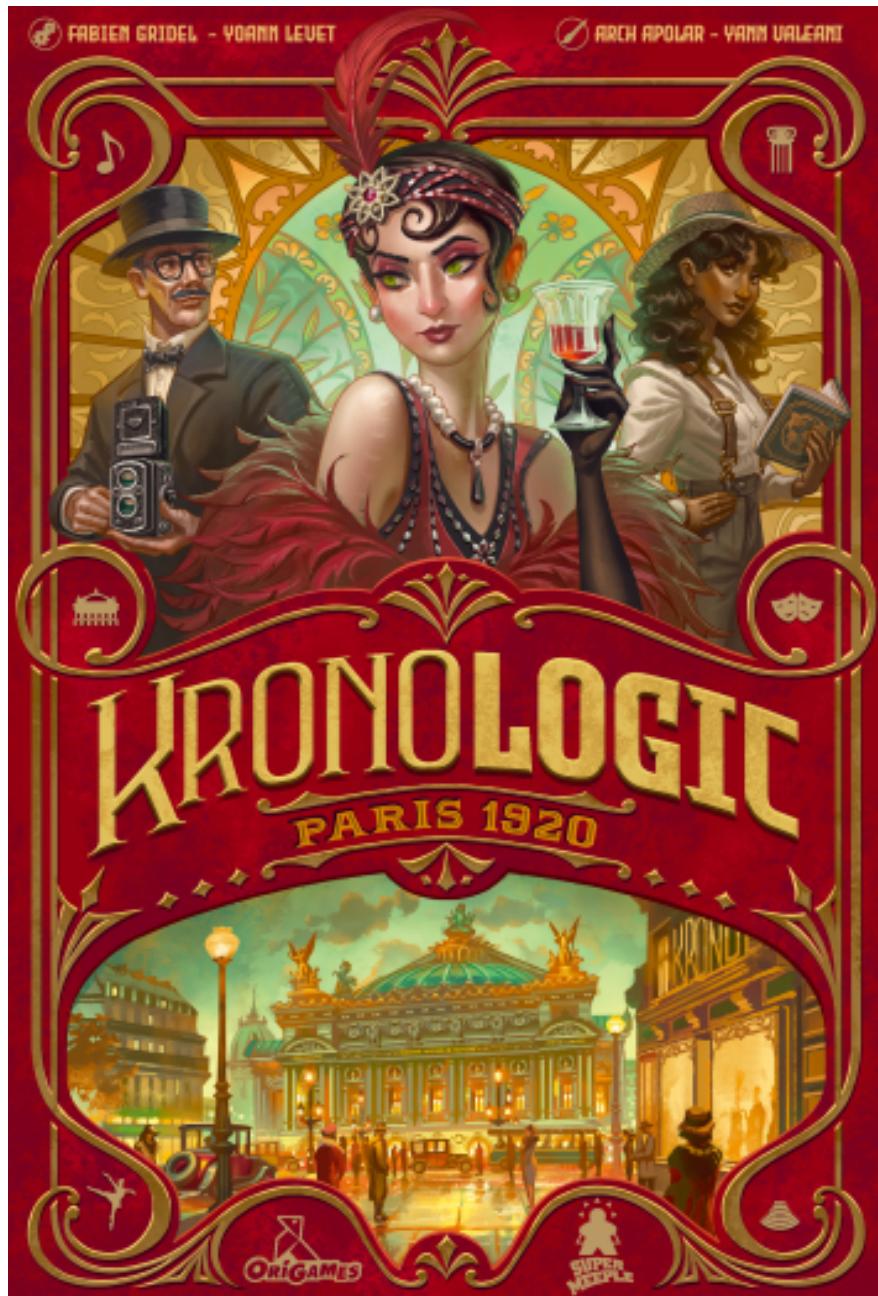


Projet tutoré - Rapport final KRONOLOGIC



UNIVERSITÉ
DE LORRAINE



IUT nancy Charlemagne
Informatique

Table des matières

Table des matières	3
1. Introduction	4
1.1. Objet du document	4
1.2. Présentation du projet	4
1.3. Planning du déroulement du projet	4
1.4. Répartition du travail	8
2. Analyse	11
2.1. Découpage fonctionnel	11
2.2. Modèles UML utilisés	11
2.3. Évolution du projet par rapport à l'étude préalable	12
3. Réalisation	14
3.1. Architecture logicielle	14
3.2. Tests	15
3.2.1. Jeu	15
3.2.2. IA	15
3.4. Les choix de conception	16
3.5. Difficultés rencontrées	17
3.6. Fonctionnalités critiques	17
3.6.1. Les intelligences artificielles	17
3.6.2. Le générateur de scénarios	45
4. Conclusion	48
5. Annexes	49
5.1. Mode d'emploi pour installer à partir de github	49
5.2. Mode d'emploi pour lancer le jeu (et fonctionnement)	50

1. Introduction

1.1. Objet du document

Ce document constitue le rapport final du projet Kronologic, réalisé dans le cadre du projet tutoré de troisième année de BUT Informatique. Il a pour objectif de présenter le déroulement complet du projet, de l'analyse initiale à la réalisation finale, en passant par les choix techniques, les difficultés rencontrées et les résultats obtenus. Il servira de trace écrite de notre démarche de conception, de développement et d'intégration des différentes fonctionnalités de l'application.

1.2. Présentation du projet

Le projet tutoré "Kronologic" vise à concevoir une intelligence artificielle (IA) capable d'accompagner ou de remplacer un joueur dans un jeu de déduction complexe. Le jeu de plateau Kronologic, créé par Fabien Gridel et Yoann Levet et édité par SuperMeeple en 2024, plonge les joueurs dans une enquête criminelle où ils doivent reconstituer les événements autour d'un meurtre en collectant des indices et en déduisant l'identité du coupable. Pour ce faire, le joueur doit analyser les lieux, les moments clés et les déplacements des personnages, tout en respectant des règles strictes de déduction.

Les objectifs de ce projet sont :

- ❖ Assimiler les mécanismes et règles du jeu et les intégrer dans une interface intuitive et accessible ;
- ❖ Développer une IA capable d'assister le joueur en réalisant des déductions logiques à partir des indices, et garantir que ses raisonnements sont corrects ;
- ❖ Concevoir une IA joueuse qui puisse analyser les indices et identifier rapidement le coupable ;
- ❖ Élaborer un générateur de scénarios permettant de créer de nouvelles enquêtes avec des indices variés, en s'assurant que chaque problème proposé offre une solution unique, accessible et exigeant une combinaison astucieuse d'informations pour être résolu.

1.3. Planning du déroulement du projet

Le projet s'est articulé autour de sept itérations principales, chacune ayant des objectifs précis. La planification initiale a dû être ajustée à plusieurs reprises en fonction des imprévus techniques et des évolutions du projet.

Itération 1 :

L'objectif principal de cette première itération est de poser les fondations techniques du projet en développant les éléments fondamentaux du jeu.

- ❖ **Implémentation du jeu** : Définition des mécanismes de base du jeu et mise en place de l'environnement de développement.
- ❖ **Encodage des données du jeu** : Modélisation des données du jeu, telles que les personnages, les lieux et les indices, en structurant ces informations de manière efficace.

- ❖ **Pose de question** : Développement de la première fonctionnalité du jeu : la pose de question (avec *Lieu et Temps* et *Lieu et Personnage*).
- ❖ **Architecture MVC** : Adoption de l'architecture Modèle-Vue-Contrôleur (MVC) afin de garantir une séparation claire entre la logique métier, l'interface utilisateur et les contrôleurs.
- ❖ **Développement de l'interface graphique** : Création d'une première version de l'interface utilisateur, avec une mise en place des éléments visuels principaux.
- ❖ **Suivi du tutoriel Choco-Solver** : Approfondissement des connaissances sur Choco-Solver, un outil de résolution de contraintes, afin de mieux comprendre son intégration dans le projet.
- ❖ **Réalisation d'un prototype Choco-Solver** : Développement de prototypes utilisant Choco-Solver pour valider la faisabilité des mécanismes de déduction du jeu.

Itération 2 :

Cette phase est axée sur l'amélioration de l'affichage et sur l'introduction des fonctionnalités interactives du jeu.

- ❖ **Réalisation du changement d'affichage** : Amélioration de l'interface pour mieux visualiser les informations et offrir une expérience utilisateur plus fluide.
- ❖ **Formulation d'une déduction** : Développement du mécanisme permettant au joueur de réaliser une déduction avec les indices qu'il a récolté et ainsi essayer de gagner.
- ❖ **Mise en place du Drag and Drop des pions** : Ajout d'une fonctionnalité permettant aux joueurs de déplacer les pions sur les zones de jeu via un système de glisser-déposer.
- ❖ **Implémentation des pop-ups** : Création de fenêtres contextuelles fournissant des informations utiles, comme des résultats de déduction ou des indices débloqués.
- ❖ **Développement de l'intelligence artificielle (IA)** : Mise en place du Modèle Choco-Solver et de la première version de l'IA de déduction.
- ❖ **Définition des variables et contraintes initiales** : Mise en place des différentes variables et des contraintes initiales du temps 1.
- ❖ **Élaboration des contraintes de règles** : Mise en place des contraintes de déplacements (salles adjacentes et déplacement obligatoire).
- ❖ **Automatisation de l'ajout des contraintes en fonction des indices** : À chaque nouvel indice découvert, des nouvelles contraintes sont ajoutées à l'IA pour réduire les possibilités des positions des personnages.

Itération 3 :

Cette phase se concentre sur l'amélioration de l'expérience utilisateur, l'ajout de nouvelles mécaniques et l'optimisation de l'intelligence artificielle.

- ❖ **Affichage d'un pop-up expliquant l'éénigme** : Mise en place d'un récapitulatif interactif expliquant le contexte et les objectifs du jeu.
- ❖ **Affichage des règles** : Intégration d'une section expliquant les règles du jeu de manière claire et accessible.

- ❖ **Gestion des pions et des notes du joueur** : Introduction d'un système permettant aux joueurs de gérer leurs pions et de prendre des notes avec ceux-ci.
- ❖ **Création des zones de Drag and Drop (placement automatique des pions)** : Amélioration du système de glisser-déposer en définissant des zones précises pour un placement automatique.
- ❖ **Implémentation du placement automatique des pions au temps 1** : Ajout d'un système positionnant automatiquement les pions au début du jeu.
- ❖ **Réalisation de tests** : Validation exhaustive des fonctionnalités du jeu et de l'IA par des tests approfondis pour détecter et corriger d'éventuels bugs.
- ❖ **Synchronisation des deux affichages de prise de notes (Carte et Tableau)** : Amélioration de l'interface pour assurer une cohérence entre les deux différentes façons de visualiser les indices.
- ❖ **Contraintes des indices publics de l'IA de déduction Choco-Solver** : Gestion de l'ajout des contraintes des indices publics.
- ❖ **Ajout des variables du coupable** : Intégration des variables du personnage coupable, du lieu du crime et du temps du crime.
- ❖ **Ajout de la contrainte du coupable** : Mise en place de règles assurant que le coupable est le seul présent avec la victime au moment du crime.
- ❖ **Affichage du coupable** : Ajout d'un affichage permettant au joueur de voir qui est désigné comme coupable lorsque les indices sont suffisants.
- ❖ **Implémentation de l'IA de déduction heuristique** : Mise en place d'une seconde IA de Déduction utilisant des algorithmes heuristiques.
- ❖ **Définition des domaines en booléen** : Représentation en booléens pour simplifier le traitement des données.
- **Définition des positions initiales** : Mise en place des contraintes de placement initial de début de partie.

Itération 4 :

Cette phase se concentre sur la synchronisation des éléments du jeu et l'ajout de nouvelles mécaniques permettant d'enrichir l'expérience du joueur.

- ❖ **Synchronisation des pions de nombre** : Assurer que les pions de nombre affiché soient toujours cohérents avec les informations en jeu. Assurer que la *Vue Carte* et la *Vue Tableau* représentent les mêmes données.
- ❖ **Gestion des pions d'absences et d'hypothèses (données et affichage)** : Ajout d'un système permettant de poser des pions d'absences et d'hypothèses (ou d'hypothèse d'absence) de manière dynamique.
- ❖ **Réalisation du film de la partie** : Création d'un récapitulatif visuel retracant les étapes clés de la partie.
- ❖ **Réalisation du film du joueur** : Création d'un récapitulatif visuel retracant les notes prises par le joueur à chaque question lui permettant de retracer ses prises de décision.
- ❖ **Amélioration de l'IA de déduction heuristique** : L'IA de déduction heuristique ne prenait en compte que les positions initiales. Elle est à présent capable de raisonner sur les nouveaux indices récupérés à chaque question.

- ❖ **Ajout d'un nouvel indice** : L'IA de déduction heuristique effectue de la réduction de domaines en fonction des indices récoltés.
- ❖ **IA Assistance Choco-Solver** : Intégration d'un système d'assistance utilisant Choco-Solver pour aider le joueur.
- ❖ **Correction des déductions du joueur** : Mise en place d'une correction des déductions du joueur en comparaison des déductions de l'IA de déduction.

Itération 5 :

Cette itération marque l'aboutissement du développement des IA et la finalisation de nombreuses mécaniques d'assistance pour le joueur.

- ❖ **Finalisation de l'IA de déduction heuristique** : l'IA prend désormais en compte l'ensemble des contraintes du jeu. La propagation des domaines n'est plus limitée aux temps adjacents mais s'étend à tous les temps, grâce à un algorithme de propagation de contraintes.
- ❖ **Développement de l'IA d'assistance Choco-Solver** : cette IA simule la pose de chaque question possible et évalue son impact sur la réduction de l'espace des solutions. Elle identifie la question la plus pertinente à poser en fonction de la réduction de domaines obtenue.
- ❖ **Interaction fluide IA-joueur** : une interface d'assistance a été intégrée, permettant au joueur de demander de choisir parmi les différentes IA.
- ❖ **Finalisation de l'interface graphique** : plusieurs bugs ont été corrigés, notamment ceux liés au placement automatique des pions, aux conflits d'affichage lorsque plusieurs pions identiques se trouvent dans la même salle, et aux interactions entre la Vue Carte et la Vue Tableau. L'application est devenue visuellement plus cohérente et agréable à utiliser.

Itération 6 :

Cette phase s'est concentrée sur le développement de fonctionnalités avancées et l'autonomisation complète du raisonnement du jeu.

- ❖ **Comparaison des IA de Déduction Choco / Heuristique** : des tests ont été effectués pour comparer la performance, la rapidité et la capacité de résolution de chacune des IA sur différents types de questions et de raisonnements. Ces comparaisons ont permis de mieux comprendre leurs forces respectives (réduction de domaines, recherche du coupable,...).
- ❖ **Développement de l'IA joueuse** : un nouveau mode a été ajouté, dans lequel une IA résout entièrement l'enquête sans intervention humaine. Elle enchaîne les questions les plus pertinentes, choisies automatiquement grâce à l'IA d'assistance, jusqu'à pouvoir identifier le coupable.
- ❖ **Générateur de scénarios** : une fonctionnalité a été ajoutée pour produire automatiquement des scénarios valides au format JSON. Chaque scénario respecte les règles fondamentales du jeu (un seul moment où le détective est seul avec une autre personne, pas de meurtre au temps 1, etc.). Ce générateur est intégré dans l'interface, via un pop-up au lancement permettant de choisir entre une enquête classique ou générée aléatoirement.

Itération 7 :

Dernière phase du projet, cette itération s'est consacrée aux tests finaux, à la stabilisation de l'ensemble de l'application et à la documentation.

- ❖ **Tests sur le générateur de scénarios** : une classe de tests automatiques a été développée afin de vérifier la validité logique des scénarios produits. Chaque fichier JSON est vérifié : respect des déplacements possibles, détection correcte du meurtre, et conformité stricte avec le modèle utilisé dans le lecteur JSON.
- ❖ **Intégration finale des IA d'assistance et comparaison** : les IA d'Assistance sont désormais compatibles avec tous les scénarios, qu'ils soient prédéfinis ou générés. L'IA Heuristique d'Assistance a été finalisée pour proposer les questions les plus pertinentes. Des comparaisons des performances ont été effectués.
- ❖ **Stabilisation de l'application** : des ajustements finaux ont été effectués sur l'ensemble de l'interface graphique, avec des corrections de bugs mineurs et une meilleure gestion des erreurs et actions invalides. Le résultat est une application fluide, stable et agréable à utiliser.
- ❖ **Rédaction du rapport final et préparation de la soutenance** : une synthèse complète du projet a été rédigée, mettant en valeur les choix réalisés, les étapes de développement, les IA implémentées et les résultats obtenus. Une présentation claire et structurée a été préparée pour la soutenance finale.

1.4. Répartition du travail

A) Itération 1

Tâches	Mathieu	Corentin	Enzo
Implémentation du jeu			X
Encodage des données du jeu		X	X
Pose de question		X	
Architecture MVC		X	X
Développement de l'interface graphique		X	X
Suivi du tutoriel Choco-Solver	X		
Réalisation d'un prototype Choco-Solver	X		

B) Itération 2

Tâches	Mathieu	Corentin	Enzo
Réalisation du changement d'affichage			X
Formulation d'une déduction		X	X
Mise en place du Drag and Drop des pions		X	

<i>Implémentation des pop-ups</i>		X	
<i>Développement de l'intelligence artificielle</i>	X		
<i>Définition des variables et contraintes initiales</i>	X		
<i>Elaboration des contraintes de règles</i>	X		
<i>Automatisation de l'ajout des contraintes en fonction des indices</i>	X		

C) Itération 3

Tâches	Mathieu	Corentin	Enzo
<i>Affichage d'un pop-up expliquant l'énigme</i>			X
<i>Affichage des règles</i>		X	X
<i>Affichage des films (dans le terminal)</i>			X
<i>Gestion des pions et des notes du joueur</i>		X	X
<i>Création des zones de Drag and Drop</i>		X	
<i>Implémentation du placement automatique des pions au temps 1</i>		X	
<i>Réalisation de tests</i>	X		X
<i>Synchronisation des deux affichages de prise de notes (Carte et Tableau)</i>			X
<i>Contraintes des indices publics de l'IA de déduction Choco-Solver</i>	X		
<i>Ajout des variables du coupable</i>	X		
<i>Ajout de la contrainte du coupable</i>	X		
<i>Affichage du coupable</i>	X		
<i>Implémentation de l'IA de déduction heuristique</i>	X		
<i>Définition des domaines en booléen</i>	X		
<i>Définition des positions initiales</i>	X		

D) Itération 4

Tâches	Mathieu	Corentin	Enzo
<i>Synchronisation des pions de nombre</i>			X

Gestion des pions d'absences et d'hypothèses (données et affichage)		X	
Réalisation du film de la partie			X
Réalisation du film du joueur			X
Continuation de l'IA de déduction heuristique		X	
Mise à jour lors d'un nouvel indice		X	
IA Assistance Choco-Solver	X		
Correction des déductions du joueur	X		

E) Itération 5

Tâches	Mathieu	Corentin	Enzo
Finalisation de l'IA de déduction heuristique		X	
Développement de l'IA d'assistance Choco-Solver	X		
Interaction fluide IA-Joueur	X		
Finalisation de l'interface graphique			X

F) Itération 6

Tâches	Mathieu	Corentin	Enzo
Comparaison des IA de Déduction Choco / Heuristique			X
Développement de l'IA joueuse	X		
Générateur de scénarios		X	

G) Itération 7

Tâches	Mathieu	Corentin	Enzo
Tests sur le générateur de scénarios		X	
Intégration finale des IA d'Assistance et comparaison	X		X
Stabilisation de l'application	X		X
Rédaction du rapport final		X	
Préparation de la soutenance	X	X	X

2. Analyse

2.1. Découpage fonctionnel

Module	Description
Données	<i>Gestion des données du jeu et lecture des scénarios depuis un fichier JSON.</i>
IA	<i>Contient les différentes IA du projet : l'IA de déduction, l'IA d'assistance et l'IA joueuse.</i>
Jeu	<i>Regroupe tous les éléments du jeu, y compris les personnages, les lieux, les indices, le déroulement d'une partie, etc...</i>
MVC	<i>Structure dédiée à l'affichage et aux interactions avec le joueur, en assurant la séparation entre la logique du jeu et l'interface utilisateur.</i>

2.2. Modèles UML utilisés

Diagramme de classes : pour montrer la structuration des différents modules (Jeu, IA, MVC et Données) du projet en modélisant les relations entre les classes principales. Ce qui a permis d'organiser l'ensemble du projet.

Diagramme de séquence : pour illustrer les interactions entre les différentes classes et méthodes du projet, notamment entre le joueur et l'interface graphique pour la gestion des questions et des hypothèses, ainsi qu'entre la partie et la gestion des indices pour mettre à jour les connaissances au fur et à mesure du jeu. Ce qui nous a aidés à clarifier la logique des échanges et à optimiser les appels aux différentes méthodes des classes impliquées.

Diagramme de cas d'utilisation : pour illustrer les interactions principales entre le joueur, (poser des questions, formuler des hypothèses, interagir avec les différentes composantes du système), l'IA de Déduction (analyse les indices et propose des hypothèses logiques en fonction des informations disponibles), l'IA d'Assistance (aide le joueur en recommandant des questions optimales et en corrigeant ses déductions si nécessaire), l'IA Joueuse (simule une partie en s'aidant de l'IA d'Assistance) et le Générateur de scénarios (permet de créer des scénarios d'enquête).

Diagramme d'activité : pour représenter le déroulement du jeu en se concentrant sur les actions du joueur et du système. Ce qui permet de détailler le déroulement général d'une partie, depuis le lancement de l'application jusqu'à la fin du jeu, les différentes interactions possibles, telles que poser une question, formuler une hypothèse ou consulter les indices, l'impact des choix du joueur sur l'état du jeu, notamment l'ajout et la mise à jour des indices en fonction des réponses obtenues, puis l'utilisation de l'IA Déduction et de l'IA Assistante, qui permettent d'affiner les hypothèses et d'aider le joueur dans son raisonnement.

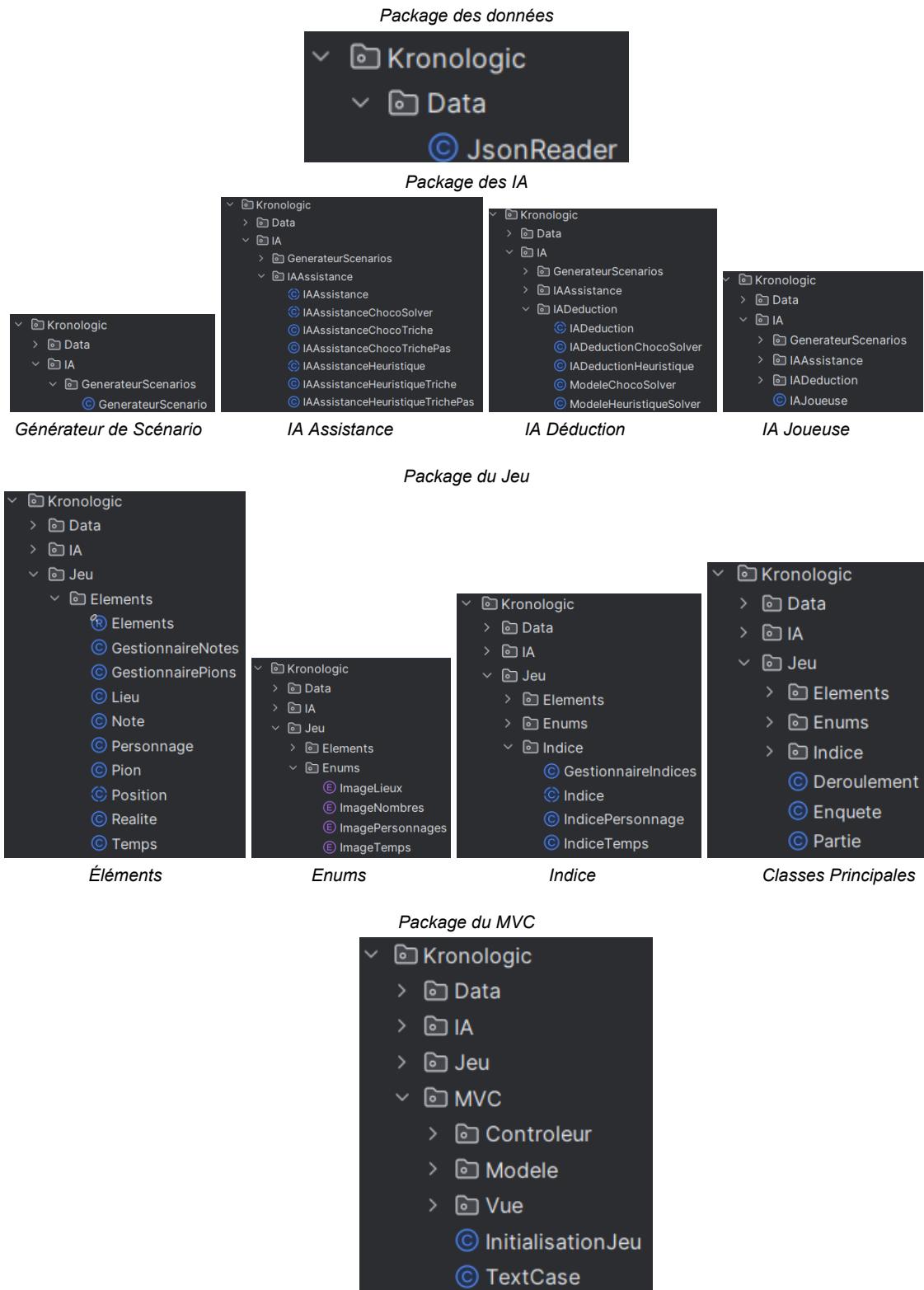
2.3. Évolution du projet par rapport à l'étude préalable

Tâches	Itération prévue	Itération réalisée
Apprentissage de Choco-Solver	Itération 1	Itération 1
Prototypes de Choco-Solver	Itération 1	Itération 1
Encodage des données du jeu	Itération 1	Itération 1
Architecture MVC	Itération 1	Itération 1
Commencer le développement de l'interface graphique	Itération 1	Itération 1
Terminer l'interface graphique	Itération 2	Itération 4
Pose de question	Itération 2	Itération 1
Visualisation des indices	Itération 2	Itération 2
Réalisation d'hypothèses	Itération 2	Itération 4
Réalisation d'une déduction	Itération 2	Itération 2
Implémenter les notions de victoire et de défaite	Itération 2	Itération 2
Développer IA de Déduction Choco-Solver	Itération 3/4	Itération 3
Développer IA de Déduction Heuristique	Itération 3/4	Itération 5/6
Comparer les performances des deux IA de Déduction	Itération 3/4	Itération 6
Intégrer les suggestions de l'IA dans l'interface graphique	Itération 3/4	Itération 6
Développer IA d'Assistance Choco-Solver	Itération 5/6	Itération 5/6
Développer IA d'Assistance Heuristique	Itération 5/6	Itération 6/7
Comparer les performances des deux IA d'Assistance	Itération 5/6	Itération 7
Permettre une interaction fluide entre l'IA d'Assistance et le Joueur	Itération 5/6	Itération 4
IA Joueuse	Optionnelle	Itération 6/7
Générateur de scénario	Optionnelle	Itération 6/7

Dans l'ensemble, le projet a suivi une progression fidèle à l'étude préalable, tout en s'adaptant de manière agile aux contraintes rencontrées. Les premières itérations, centrées sur l'apprentissage de Choco-Solver, l'encodage des données du jeu et la mise en place de l'interface graphique, ont été réalisées conformément au planning initial. Cependant, certaines tâches se sont avérées plus complexes que prévu, entraînant un léger décalage dans le calendrier. C'est notamment le cas de la finalisation de l'interface et des fonctionnalités comme la formulation d'hypothèses ou l'intégration fluide de l'IA, qui ont nécessité un travail supplémentaire lors des itérations 4 à 6. À l'inverse, d'autres aspects ont pu être implémentés plus tôt, comme la pose de questions ou la déduction de base. Les parties optionnelles, telles que l'IA joueuse et le générateur de scénarios, ont été intégrées dans les dernières itérations, illustrant une bonne gestion du temps et une montée en compétence progressive. Enfin, bien que certaines tâches aient été décalées, l'ensemble des objectifs initiaux – y compris les IA multiples, l'interaction avec le joueur, et la création de scénarios conformes – a été atteint, ce qui témoigne d'une évolution positive et maîtrisée du projet par rapport à son étude préalable.

3. Réalisation

3.1. Architecture logicielle

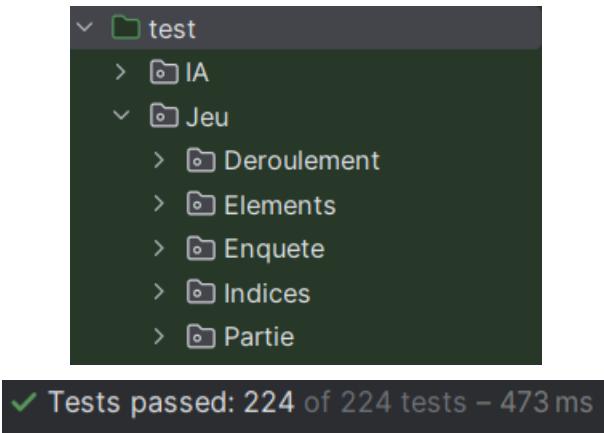


Le projet Kronologic repose sur une architecture en couches organisée selon le modèle MVC (Modèle-Vue-Contrôleur). La séparation des responsabilités est nette : la logique métier du jeu est contenue dans le package Jeu, l'affichage et l'interaction utilisateur dans MVC, et les intelligences artificielles dans le package IA, subdivisé selon leurs fonctions (déduction, assistance, jeu autonome, génération de scénarios). Le package Data permet la lecture structurée des fichiers JSON décrivant les enquêtes. Cette architecture modulaire facilite l'évolution du code, le test unitaire des composants, et l'ajout de nouvelles IA ou fonctionnalités sans impacter l'ensemble de l'application.

3.2. Tests

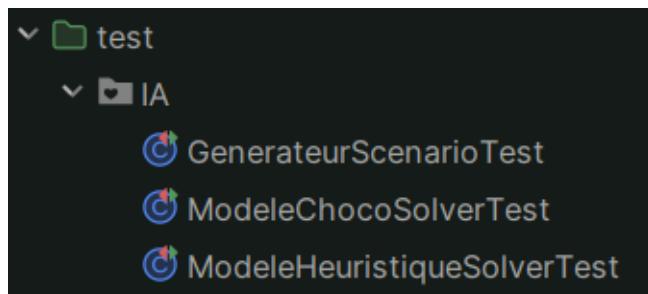
3.2.1. Jeu

Pour les tests réalisés sur les bases du jeu et ces fonctionnalités, nous avons pris soin d'être exhaustif pour nous assurer du bon déroulement du développement. Voici ce que nous avons testé :



3.2.2. IA

- ❖ L'IA de Déduction Choco-Solver a été testée dès les premières itérations pour vérifier la cohérence des contraintes posées, la pertinence des réductions de domaines et l'obtention de résultats attendus.
- ❖ L'IA Heuristique a été soumise à des tests unitaires ciblant différents cas (nombre de passages, présence à un lieu, propagation des contraintes), afin de garantir la fiabilité du raisonnement logique à partir d'indices partiels.
- ❖ Le générateur de scénarios a également été testé automatiquement, avec des validations sur les déplacements, les adjacences, le respect des règles du jeu (dont l'unicité de la scène du meurtre), et la conformité du format JSON produit.



3.4. Les choix de conception

- **Architecture MVC**

Nous avons adopté une architecture Modèle-Vue-Contrôleur (MVC) dès l'itération 1. Elle permet une séparation claire des responsabilités :

- ❖ **Modèle** : déroulement de la partie, gestion des indices, personnages...
- ❖ **Vue** : interface utilisateur (carte, tableau, pop-ups...),
- ❖ **Contrôleur** : gestion des interactions (clics, déplacements, assistance IA...).

Cela nous a permis de tester, maintenir et faire évoluer les composants indépendamment, et de faciliter l'ajout de nouvelles fonctionnalités comme les IA ou le générateur de scénarios.

- **Principe de Responsabilité Unique (SRP)**

Nous avons respecté le Single Responsibility Principle, chaque classe ayant une seule responsabilité bien définie :

- ❖ Les classes IAJoueuse, IAAccompagnement, IADeduction sont séparées,
- ❖ La lecture des fichiers JSON est centralisée dans le package Data,

Ce découpage a permis une grande modularité, notamment dans la comparaison des IA et l'ajout d'un nouveau mode autonome sans impacter le reste de l'application.

- **Polymorphisme**

L'utilisation du polymorphisme nous a permis d'unifier l'interaction avec les différentes IA via des interfaces communes :

- ❖ Par exemple, l'interface IAssistance permet de choisir entre les IA Choco ou Heuristique, avec ou sans triche.

Ce choix facilite les tests, les comparaisons et l'évolution future du projet, notamment pour ajouter d'autres types d'IA ou de stratégies.

3.5. Difficultés rencontrées

Le **placement automatique des pions** a constitué un défi majeur dans le développement du jeu, car il implique de respecter des contraintes spatiales et logiques tout en offrant une expérience utilisateur fluide. Cette problématique a évolué au fil des itérations du projet, chaque étape permettant d'affiner la solution.

Problèmes identifiés :

- ❖ **Définition des positions initiales** : Il fallait déterminer comment et où placer automatiquement les pions au début du jeu en respectant les règles du scénario.
- ❖ **Interaction avec le joueur** : Assurer une transition fluide entre le placement automatique et le déplacement manuel des pions.
- ❖ **Synchronisation des affichages** : Garantir une cohérence entre la carte et le tableau des positions.
- ❖ **Placement des pions aux bons endroits** : Le placement automatique des pions a été un véritable défi. Notre problème principal a été de bien les placer dans les bonnes sous-zones. Cela s'est résolu en revoyant notre architecture et notre structure MVC.
- ❖ **Compréhension de Choco-Solver** : L'apprentissage initial de Choco-Solver a constitué un obstacle important, car il s'agissait non seulement de comprendre le paradigme de programmation par contraintes, mais aussi de le traduire efficacement en modèles pour notre jeu. La définition des bonnes variables, des domaines, et des contraintes spécifiques au jeu a demandé plusieurs itérations et tests.
- ❖ **Réduction des domaines avec l'IA heuristique** : L'implémentation d'une IA fonctionnant sans solveur a soulevé de nombreuses questions liées à la propagation des contraintes, à la gestion des adjacences, et à la détection du coupable. Il a fallu concevoir un système de domaines booléens cohérent, capable d'évoluer à chaque nouvel indice, tout en restant performant et fiable. Le plus complexe a été d'assurer une propagation correcte à travers tous les temps du jeu, en prenant en compte les déplacements valides et les incompatibilités révélées par les indices.

Malgré ces difficultés, toutes ont été progressivement surmontées grâce à un travail d'équipe rigoureux, une amélioration continue du code, et l'intégration de nombreux tests.

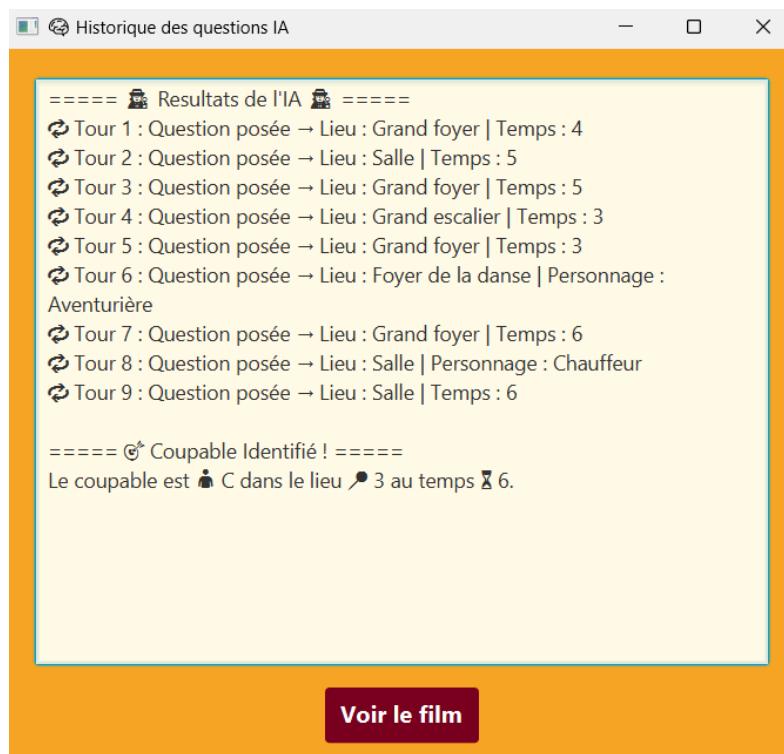
3.6. Fonctionnalités critiques

Il s'agit des fonctionnalités les plus importantes et les plus enrichissantes du projet. Nous allons les aborder plus en détail afin de montrer la façon dont nous les avons développées ainsi que leur intérêt dans un projet tel que Kronologic.

3.6.1. Les intelligences artificielles

Dans un premier temps, il est important de rappeler que nous nous situons dans un projet de recherche et d'exploration, donc les IA sont le véritable cœur du projet. Il nous a été demandé d'en développer plusieurs, répondant chacunes à un besoin spécifique :

- ❖ **IA de Déduction** : Permet d'analyser les indices collectés et effectuer des déductions logiques. Elle se concentre sur la formulation d'hypothèses solides et la réduction des suspects en fonction des informations disponibles.
- ❖ **IA d'Assistance** : Guide le joueur dans son processus de réflexion, en fournissant des suggestions basées sur les indices collectés.
- ❖ **IA Joueuse** : Version autonome qui est capable de jouer seule et de résoudre une enquête sans intervention humaine. Elle se base sur l'IA d'assistance en répétant son processus jusqu'à trouver la bonne réponse.
- ❖ **Algorithme de génération de scénarios** : Crée des scénarios aléatoires respectant l'ensemble des contraintes du jeu et disposant d'une seule solution trouvable.



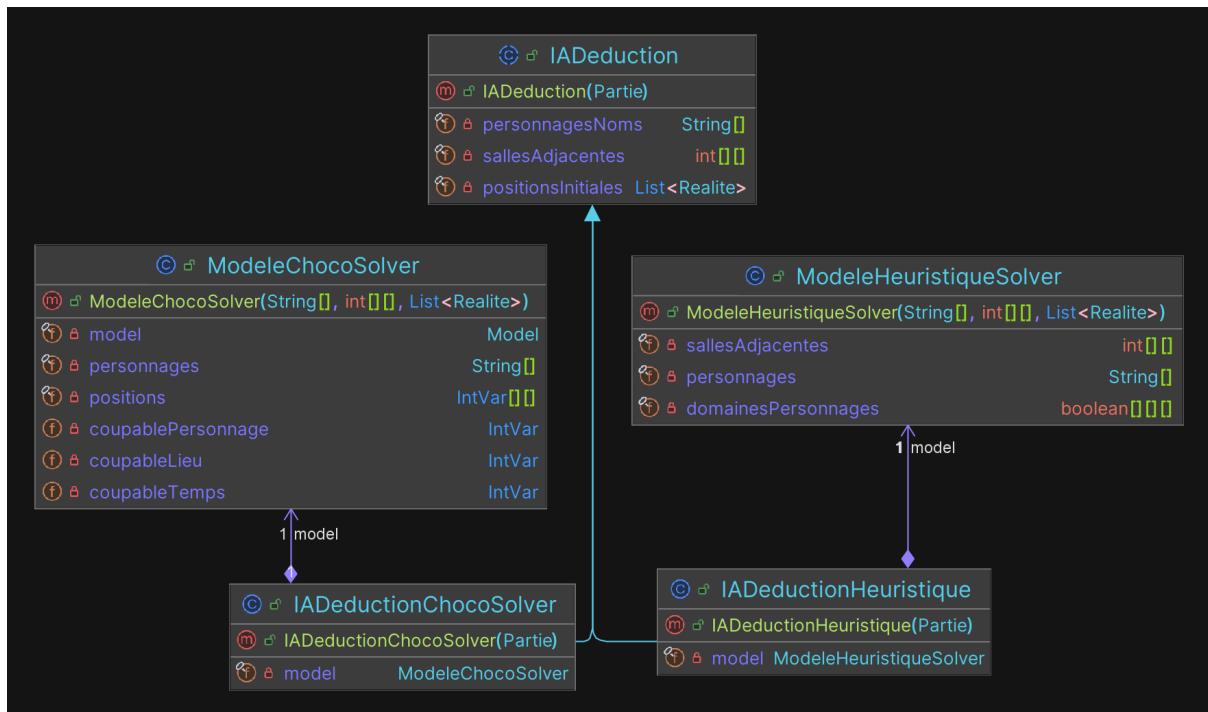
Exemple de partie jouée par l'IA joueuse de façon autonome

Chaque type d'IA a également été développée via deux approches différentes, une utilisant la librairie Choco-Solver et une se basant sur une heuristique que nous avons créé.

1) Les IA de Déduction

L'IA de Déduction Choco-Solver :

Cette IA a pour but de raisonner comme un joueur en analysant les indices fournis et en appliquant de la programmation par contraintes pour réduire progressivement l'ensemble des possibilités et identifier le coupable, à l'aide des règles du jeu.



1. Pourquoi Choco-Solver ?

Choco-Solver est une bibliothèque de programmation par contraintes, utilisée pour résoudre des problèmes complexes en modélisant les variables, leurs domaines de valeurs possibles, et les contraintes qui restreignent leurs relations.

Dans notre cas, nous l'avons choisi car il permet de :

- ❖ Gérer efficacement les contraintes du jeu, comme les déplacements autorisés, la présence des personnages et les indices obtenus.
- ❖ Réduire l'espace de recherche en éliminant rapidement les configurations impossibles.

2. Définition des contraintes

Pour permettre à l'IA de raisonner efficacement, nous avons défini plusieurs types de contraintes basées sur les règles du jeu et les indices obtenus. Ces contraintes permettent de réduire progressivement l'espace des possibilités.

a. Variables utilisés

P_{i,t} : Position du personnage i au temps t ($1 \leq t \leq 6$). Les salles sont numérotées de 1 à 6, donc la variable P_{i,t} a un domaine : {1, 2, 3, 4, 5, 6}.

CoupablePersonnage (CP) : CP $\in \{A, B, C, D, J, S\} \setminus \{D\}$
Le domaine exclut le détective (D) car il est la victime.

CoupableLieu (CL) : CL $\in \{1, 2, 3, 4, 5, 6\}$
Le domaine correspond aux salles possibles où le meurtre peut avoir eu lieu.

CoupableTemps (CT) : CT $\in \{2, 3, 4, 5, 6\}$
Le domaine exclut t=1 car les informations initiales sont toujours disponibles à ce moment-là, et le meurtre ne peut pas avoir eu lieu au temps initial.

b. Contraintes du jeu

Contraintes de règles

- **Déplacement obligatoire :**
Chaque personnage doit se déplacer entre t et t+1 :

$$\forall i \in \{A, B, C, D, J, S\}, \forall t \in \{1, \dots, 5\}, P_{i,t} \neq P_{i,t+1}$$

- **Déplacement dans des salles adjacentes :**
Les déplacements doivent respecter la connectivité des salles.

$$\forall i, t : P_{i,t+1} \in \text{adj}(P_{i,t})$$

Par exemple, si la salle 1 est reliée à 2 et 3, alors : P_{i,t+1} $\in \{2, 3\}$ si P_{i,t} = 1

Contraintes de début de partie

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

- A est dans la salle 6
- B est dans la salle 1
- C est dans la salle 1
- D est dans la salle 2
- J est dans la salle 4
- S est dans la salle 5

Pour chaque personnage i, une contrainte fixe est appliquée :

$$P_{A,1} = 6, P_{B,1} = 1, P_{C,1} = 1, \text{ etc...}$$

Contraintes au cours de la partie

Les contraintes apparaissent au fur et à mesure de la partie et dépendent des indices fournis.

- **Public (Temps choisi) :**

On sait le nombre de personnages se trouvant dans une salle donnée au temps t.

Pour un lieu L et un temps t, nous faisons une somme sur tous les personnages $i \in \{A, B, C, D, J, S\}$.

$$\sum_i 1_{P_i,t=L} = N$$

$1_{P_i,t=L}$ vaut 1 si le personnage i est dans le lieu L, sinon 0. N est le nombre total de personnages dans le lieu L au temps t.

- **Privé (Temps choisi) :**

Lorsqu'un temps t est choisi, l'indice privé donne soit :

- Une possibilité de rejouer (pas modélisé en contraintes)
- L'identité d'un personnage i dans un lieu L a ce temps :

$$P_{i,t} = L$$

- **Public (Personnage choisi) :**

On sait le nombre de fois où un personnage i s'est trouvé dans un lieu L à travers tous les temps t. Pour un personnage i et un lieu L, nous faisons une somme sur tous les temps $t \in \{1, 2, 3, 4, 5, 6\}$.

$$\sum_t 1_{P_i,t=L} = N$$

Où $P_{i,t}$ est la position du personnage i au temps t. $1_{P_i,t=L}$ vaut 1 si le personnage i est dans le lieu L au temps t, sinon 0. N est le nombre total de fois où le personnage i a été dans le lieu L.

- **Privé (Personnage choisi) :**

Lorsqu'un personnage i est choisi, l'indice privé donne soit :

- Une possibilité de rejouer (pas modélisé en contraintes)
- Un temps t où ce personnage s'est trouvé dans un lieu L :

$$P_{i,t} = L$$

Contraintes coupable

Le coupable était seul avec le détective au moment du meurtre. Pour un temps t, une salle L, et un personnage P, le meurtre est défini comme suit :

$$\text{Si } \sum_{i \in \{A, B, C, J, S\}} 1_{P_i,t=L} = 2 \text{ et } P_{D,t} = L, \text{ alors } CT = t, CL = L, CP = P$$

Où :

$1_{P_i,t=L}$ vaut 1 si $P_{i,t}=L$, sinon 0.

$P_{D,t} = L$ indique que le détective est dans la salle L au temps t.

c. Réduction de domaine

Nous avons utilisé une propagation de domaine plutôt qu'une résolution directe des solutions afin de seulement avoir les domaines (plus utile pour le joueur).

3. Fonctionnement de l'IA avec Choco-Solver

L'IA évolue à chaque indice révélé et réduit progressivement l'ensemble des possibilités afin d'identifier le coupable.

1) Initialisation :

- ❖ Chargement de tous les personnages, les salles, les temps et les contraintes de déplacement.
- ❖ Assignation des positions initiales connues.

2) Ajout des indices révélés pendant la partie :

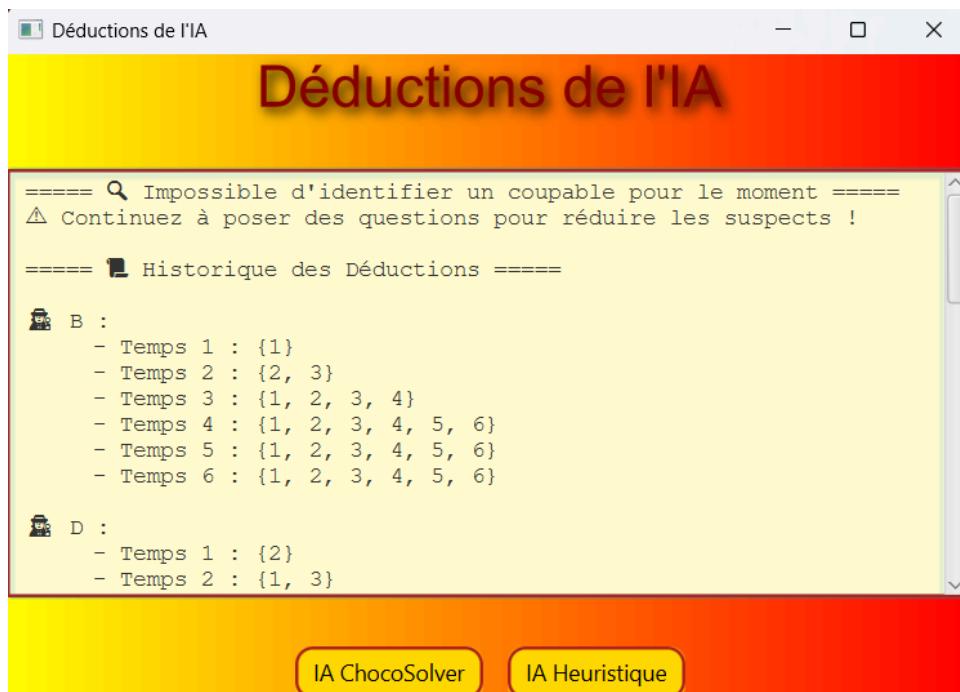
- ❖ Contraintes sur les lieux, les temps et les nombres de passages des personnages.
- ❖ Modification des domaines des variables en fonction des indices publics et privés.

3) Propagation des contraintes :

- ❖ Mise à jour des possibilités en supprimant les solutions impossibles.

4) Déduction finale :

- ❖ Vérification des variables coupables : si elles sont instanciées à une unique valeur, alors le crime est résolu.



L'IA de Déduction Heuristique :

A) Objectif de l'IA

L'IA de déduction heuristique de *Kronologic* repose sur une approche basée sur des règles logiques et des réductions de domaine pour déterminer où se trouvaient les personnages à différents moments, et ainsi identifier le coupable, le lieu du crime et le moment exact du meurtre. Contrairement à une approche basée sur un solveur de contraintes, cette IA utilise des heuristiques spécifiques pour exclure progressivement les options impossibles.

B) Représentation des données

L'IA stocke les informations sous la forme d'un tableau de booléens à trois dimensions :

domainesPersonnages[temps][personnage][lieu] = true ou false

- ❖ *temps* : Le moment dans la partie (6 étapes de temps).
- ❖ *personnage* : L'un des 6 personnages en jeu.
- ❖ *lieu* : Un des 6 lieux possibles.

Chaque cellule de ce tableau représente la possibilité qu'un personnage soit présent dans un lieu donné à un moment donné.

Au départ, toutes les combinaisons sont considérées comme possibles, puis l'IA réduit progressivement les domaines en appliquant des règles précises.

C) Les différentes règles à appliquer

L'IA applique différentes règles afin de réduire les domaines à chaque fois qu'une nouvelle question est posée.

1. Positions initiales

Lors du démarrage, les positions initiales connues des personnages sont enregistrées :

- ❖ On connaît la position de tous les personnages au temps 1, ce qui permet de réduire tous les domaines correspondants au temps 1.
- ❖ Les règles de déplacement sont appliquées : un personnage doit obligatoirement changer de lieu entre deux temps successifs.

Exemple : Si le détective est dans la Salle A au Temps 1, il ne peut pas y être au Temps 2 (il doit obligatoirement se déplacer).

2. Règles de déplacement

Le jeu impose des règles strictes sur les déplacements des personnages :

- ❖ Obligation de mouvement : Comme dit précédemment, un personnage ne peut pas rester au même endroit entre deux temps successifs.

- ❖ Salles adjacentes : Un personnage ne peut se déplacer que vers une salle adjacente à celle où il se trouve.

L'IA met donc à false toutes les positions impossibles dans le tableau domainesPersonnages en fonction des connexions entre les salles et des positions connues des personnages.

Exemple : Si un personnage est dans le Grand Foyer au Temps 1, il peut seulement se rendre au GrandEscalier ou à la Salle au Temps 2 (car ce sont les salles adjacentes). L'IA met alors à false les autres lieux pour ce personnage au Temps 2.

3. Règles sur les nombres de passages et le nombre de personnes dans une salle

Lorsqu'un indice révèle le nombre de fois où un personnage est passé dans un lieu ou le nombre total de personnages présents dans un lieu à un moment donné, l'IA ajuste son modèle.

Elle met à jour les temps de passage obligatoires et supprime les lieux non compatibles pour les autres moments. Si un nombre exact de passages ou de présences est connu, l'IA s'assure que seuls ces événements restent possibles dans les domaines restants.

Lorsque toutes les présences dans un lieu sont connues à un moment donné, l'IA peut en déduire que les autres personnages n'y étaient pas et restreindre ainsi leurs domaines de déplacement.

D) Recherche du coupable

L'un des objectifs finaux de l'IA heuristique est d'identifier, à partir des indices recueillis, le coupable, le lieu du crime et le moment exact du meurtre. Pour cela, l'IA applique une logique basée sur l'unicité de la scène du crime : le coupable est le seul personnage à avoir été seul avec le détective à un instant donné, dans un lieu donné.

La méthode `essayerDeTrouverCoupable()` scanne l'ensemble des combinaisons [temps, lieu, personnage] pour repérer un cas unique où :

- ❖ Le détective est certainement présent à un moment t dans un lieu l (c'est-à-dire que tous les autres lieux sont exclus pour lui à t).
- ❖ Un autre personnage est certainement présent au même endroit, au même moment.
- ❖ Aucun autre personnage n'est certainement présent dans ce lieu à cet instant.

Ce cas est considéré comme la scène du crime, et le personnage concerné est alors identifié comme coupable.

La comparaison des IA de Déduction :

La comparaison entre les deux intelligences artificielles repose exclusivement sur le temps qu'elles mettent à réduire leurs domaines de recherche ou à identifier le coupable. Ce critère a été retenu car nos deux IA de déduction suivent un processus similaire pour effectuer ces réductions de domaine. Voici un aperçu :

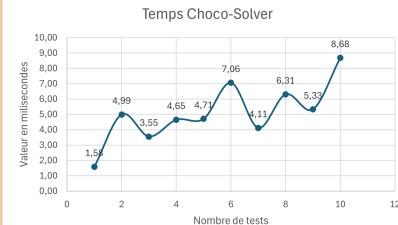
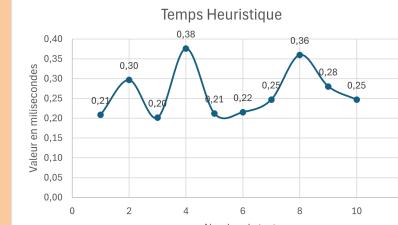
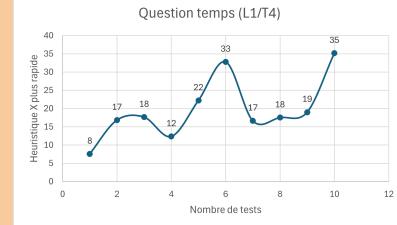
	Choco-Solver	Heuristique
Initialisation	<pre> █ A : - Temps 1 : {6} - Temps 2 : {4, 5} - Temps 3 : {3, 4, 5, 6} - Temps 4 : {1, 2, 3, 4, 5, 6} - Temps 5 : {1, 2, 3, 4, 5, 6} - Temps 6 : {1, 2, 3, 4, 5, 6} █ B : - Temps 1 : {1} - Temps 2 : {2, 3} - Temps 3 : {1, 2, 3, 4} - Temps 4 : {1, 2, 3, 4, 5, 6} - Temps 5 : {1, 2, 3, 4, 5, 6} - Temps 6 : {1, 2, 3, 4, 5, 6} </pre>	<pre> █ A : - Temps 1 : {6} - Temps 2 : {4, 5} - Temps 3 : {3, 4, 5, 6} - Temps 4 : {1, 2, 3, 4, 5, 6} - Temps 5 : {1, 2, 3, 4, 5, 6} - Temps 6 : {1, 2, 3, 4, 5, 6} █ B : - Temps 1 : {1} - Temps 2 : {2, 3} - Temps 3 : {1, 2, 3, 4} - Temps 4 : {1, 2, 3, 4, 5, 6} - Temps 5 : {1, 2, 3, 4, 5, 6} - Temps 6 : {1, 2, 3, 4, 5, 6} </pre>
Question avec réduction sur A : Foyer de la Danse	<pre> █ A : - Temps 1 : {6} - Temps 2 : {4} - Temps 3 : {3, 6} - Temps 4 : {4} - Temps 5 : {6} - Temps 6 : {5} </pre>	<pre> █ A : - Temps 1 : {6} - Temps 2 : {4} - Temps 3 : {3, 6} - Temps 4 : {4} - Temps 5 : {6} - Temps 6 : {5} </pre>
Question avec réduction sur B : Grand Escalier	<pre> █ B : - Temps 1 : {1} - Temps 2 : {2} - Temps 3 : {1, 3} - Temps 4 : {2} - Temps 5 : {1, 3} - Temps 6 : {2} </pre>	<pre> █ B : - Temps 1 : {1} - Temps 2 : {2} - Temps 3 : {1, 3} - Temps 4 : {2} - Temps 5 : {1, 3} - Temps 6 : {2} </pre>

En d'autres termes, chacune des IA applique les mêmes règles et stratégies pour éliminer progressivement les possibilités jusqu'à parvenir à une conclusion. Ainsi, la seule variable qui diffère entre elles est la rapidité avec laquelle elles effectuent ces opérations. Cette approche permet une évaluation objective de leur performance, en se concentrant sur l'efficacité algorithmique et la gestion du temps de calcul, sans introduire d'autres facteurs pouvant fausser la comparaison.

En mesurant précisément la durée nécessaire à chaque IA pour accomplir ces tâches, nous obtenons une indication claire de leur performance respective en termes de vitesse de raisonnement et d'optimisation des calculs.

Pour cela, nous allons comparer l'efficacité des IA de déduction pour la réduction des domaines dans plusieurs cas. Tout d'abord, nous allons observer les différences concernant la réduction des domaines dans les différents cas :

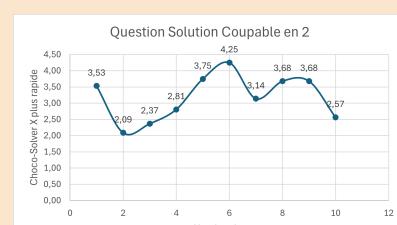
Réduction des domaines	Choco-Solver	Heuristique	Combien de fois plus efficace																																																																		
Initialisation	<p>Temps Choco-Solver</p> <table border="1"> <thead> <tr> <th>Nombre de tests</th> <th>Valeur en millisecondes</th> </tr> </thead> <tbody> <tr><td>1</td><td>96,49</td></tr> <tr><td>2</td><td>89,90</td></tr> <tr><td>3</td><td>81,83</td></tr> <tr><td>4</td><td>82,99</td></tr> <tr><td>5</td><td>81,61</td></tr> <tr><td>6</td><td>95,20</td></tr> <tr><td>7</td><td>94,49</td></tr> <tr><td>8</td><td>96,33</td></tr> <tr><td>9</td><td>103,16</td></tr> <tr><td>10</td><td>80,02</td></tr> </tbody> </table>	Nombre de tests	Valeur en millisecondes	1	96,49	2	89,90	3	81,83	4	82,99	5	81,61	6	95,20	7	94,49	8	96,33	9	103,16	10	80,02	<p>Temps Heuristique</p> <table border="1"> <thead> <tr> <th>Nombre de tests</th> <th>Valeur en millisecondes</th> </tr> </thead> <tbody> <tr><td>1</td><td>1,82</td></tr> <tr><td>2</td><td>1,63</td></tr> <tr><td>3</td><td>1,42</td></tr> <tr><td>4</td><td>1,46</td></tr> <tr><td>5</td><td>1,40</td></tr> <tr><td>6</td><td>2,20</td></tr> <tr><td>7</td><td>1,66</td></tr> <tr><td>8</td><td>1,63</td></tr> <tr><td>9</td><td>1,74</td></tr> <tr><td>10</td><td>1,37</td></tr> </tbody> </table>	Nombre de tests	Valeur en millisecondes	1	1,82	2	1,63	3	1,42	4	1,46	5	1,40	6	2,20	7	1,66	8	1,63	9	1,74	10	1,37	<p>Initialisation - Comparaison Choco-Solver Heuristique</p> <table border="1"> <thead> <tr> <th>Nombre de tests</th> <th>Heuristique X fois plus rapide</th> </tr> </thead> <tbody> <tr><td>1</td><td>53</td></tr> <tr><td>2</td><td>55</td></tr> <tr><td>3</td><td>58</td></tr> <tr><td>4</td><td>57</td></tr> <tr><td>5</td><td>58</td></tr> <tr><td>6</td><td>43</td></tr> <tr><td>7</td><td>57</td></tr> <tr><td>8</td><td>59</td></tr> <tr><td>9</td><td>59</td></tr> <tr><td>10</td><td>58</td></tr> </tbody> </table>	Nombre de tests	Heuristique X fois plus rapide	1	53	2	55	3	58	4	57	5	58	6	43	7	57	8	59	9	59	10	58
Nombre de tests	Valeur en millisecondes																																																																				
1	96,49																																																																				
2	89,90																																																																				
3	81,83																																																																				
4	82,99																																																																				
5	81,61																																																																				
6	95,20																																																																				
7	94,49																																																																				
8	96,33																																																																				
9	103,16																																																																				
10	80,02																																																																				
Nombre de tests	Valeur en millisecondes																																																																				
1	1,82																																																																				
2	1,63																																																																				
3	1,42																																																																				
4	1,46																																																																				
5	1,40																																																																				
6	2,20																																																																				
7	1,66																																																																				
8	1,63																																																																				
9	1,74																																																																				
10	1,37																																																																				
Nombre de tests	Heuristique X fois plus rapide																																																																				
1	53																																																																				
2	55																																																																				
3	58																																																																				
4	57																																																																				
5	58																																																																				
6	43																																																																				
7	57																																																																				
8	59																																																																				
9	59																																																																				
10	58																																																																				
... en moyenne	90,20 ms	1,63 ms	Heuristique 56 fois plus rapide que Choco-Solver																																																																		
Question Lieu : Grand Foyer - Temps 6	<p>Temps Choco-Solver</p> <table border="1"> <thead> <tr> <th>Nombre de tests</th> <th>Valeur en millisecondes</th> </tr> </thead> <tbody> <tr><td>1</td><td>6,65</td></tr> <tr><td>2</td><td>7,91</td></tr> <tr><td>3</td><td>5,35</td></tr> <tr><td>4</td><td>5,80</td></tr> <tr><td>5</td><td>5,58</td></tr> <tr><td>6</td><td>5,17</td></tr> <tr><td>7</td><td>5,18</td></tr> <tr><td>8</td><td>5,44</td></tr> <tr><td>9</td><td>5,05</td></tr> <tr><td>10</td><td>6,08</td></tr> </tbody> </table>	Nombre de tests	Valeur en millisecondes	1	6,65	2	7,91	3	5,35	4	5,80	5	5,58	6	5,17	7	5,18	8	5,44	9	5,05	10	6,08	<p>Temps Heuristique</p> <table border="1"> <thead> <tr> <th>Nombre de tests</th> <th>Valeur en millisecondes</th> </tr> </thead> <tbody> <tr><td>1</td><td>0,28</td></tr> <tr><td>2</td><td>0,24</td></tr> <tr><td>3</td><td>0,24</td></tr> <tr><td>4</td><td>0,25</td></tr> <tr><td>5</td><td>0,24</td></tr> <tr><td>6</td><td>0,29</td></tr> <tr><td>7</td><td>0,29</td></tr> <tr><td>8</td><td>0,26</td></tr> <tr><td>9</td><td>0,21</td></tr> <tr><td>10</td><td>0,25</td></tr> </tbody> </table>	Nombre de tests	Valeur en millisecondes	1	0,28	2	0,24	3	0,24	4	0,25	5	0,24	6	0,29	7	0,29	8	0,26	9	0,21	10	0,25	<p>Question temps (L1/T6)</p> <table border="1"> <thead> <tr> <th>Nombre de tests</th> <th>Heuristique X fois plus rapide</th> </tr> </thead> <tbody> <tr><td>1</td><td>24</td></tr> <tr><td>2</td><td>32</td></tr> <tr><td>3</td><td>22</td></tr> <tr><td>4</td><td>23</td></tr> <tr><td>5</td><td>24</td></tr> <tr><td>6</td><td>18</td></tr> <tr><td>7</td><td>18</td></tr> <tr><td>8</td><td>21</td></tr> <tr><td>9</td><td>24</td></tr> <tr><td>10</td><td>24</td></tr> </tbody> </table>	Nombre de tests	Heuristique X fois plus rapide	1	24	2	32	3	22	4	23	5	24	6	18	7	18	8	21	9	24	10	24
Nombre de tests	Valeur en millisecondes																																																																				
1	6,65																																																																				
2	7,91																																																																				
3	5,35																																																																				
4	5,80																																																																				
5	5,58																																																																				
6	5,17																																																																				
7	5,18																																																																				
8	5,44																																																																				
9	5,05																																																																				
10	6,08																																																																				
Nombre de tests	Valeur en millisecondes																																																																				
1	0,28																																																																				
2	0,24																																																																				
3	0,24																																																																				
4	0,25																																																																				
5	0,24																																																																				
6	0,29																																																																				
7	0,29																																																																				
8	0,26																																																																				
9	0,21																																																																				
10	0,25																																																																				
Nombre de tests	Heuristique X fois plus rapide																																																																				
1	24																																																																				
2	32																																																																				
3	22																																																																				
4	23																																																																				
5	24																																																																				
6	18																																																																				
7	18																																																																				
8	21																																																																				
9	24																																																																				
10	24																																																																				
... en moyenne	5,82 ms	0,26 ms	Heuristique 23 fois plus rapide que Choco-Solver																																																																		
Question Personnage : Salle - Déetective	<p>Temps Choco-Solver</p> <table border="1"> <thead> <tr> <th>Nombre de tests</th> <th>Valeur en millisecondes</th> </tr> </thead> <tbody> <tr><td>1</td><td>4,40</td></tr> <tr><td>2</td><td>3,54</td></tr> <tr><td>3</td><td>4,81</td></tr> <tr><td>4</td><td>4,08</td></tr> <tr><td>5</td><td>4,37</td></tr> <tr><td>6</td><td>6,87</td></tr> <tr><td>7</td><td>3,69</td></tr> <tr><td>8</td><td>4,26</td></tr> <tr><td>9</td><td>3,88</td></tr> <tr><td>10</td><td>4,13</td></tr> </tbody> </table>	Nombre de tests	Valeur en millisecondes	1	4,40	2	3,54	3	4,81	4	4,08	5	4,37	6	6,87	7	3,69	8	4,26	9	3,88	10	4,13	<p>Temps Heuristique</p> <table border="1"> <thead> <tr> <th>Nombre de tests</th> <th>Valeur en millisecondes</th> </tr> </thead> <tbody> <tr><td>1</td><td>0,24</td></tr> <tr><td>2</td><td>0,19</td></tr> <tr><td>3</td><td>0,20</td></tr> <tr><td>4</td><td>0,25</td></tr> <tr><td>5</td><td>0,34</td></tr> <tr><td>6</td><td>0,29</td></tr> <tr><td>7</td><td>0,23</td></tr> <tr><td>8</td><td>0,21</td></tr> <tr><td>9</td><td>0,22</td></tr> <tr><td>10</td><td>0,26</td></tr> </tbody> </table>	Nombre de tests	Valeur en millisecondes	1	0,24	2	0,19	3	0,20	4	0,25	5	0,34	6	0,29	7	0,23	8	0,21	9	0,22	10	0,26	<p>Question personnage (L3/D)</p> <table border="1"> <thead> <tr> <th>Nombre de tests</th> <th>Heuristique X fois plus rapide</th> </tr> </thead> <tbody> <tr><td>1</td><td>18</td></tr> <tr><td>2</td><td>19</td></tr> <tr><td>3</td><td>24</td></tr> <tr><td>4</td><td>17</td></tr> <tr><td>5</td><td>13</td></tr> <tr><td>6</td><td>24</td></tr> <tr><td>7</td><td>16</td></tr> <tr><td>8</td><td>20</td></tr> <tr><td>9</td><td>18</td></tr> <tr><td>10</td><td>16</td></tr> </tbody> </table>	Nombre de tests	Heuristique X fois plus rapide	1	18	2	19	3	24	4	17	5	13	6	24	7	16	8	20	9	18	10	16
Nombre de tests	Valeur en millisecondes																																																																				
1	4,40																																																																				
2	3,54																																																																				
3	4,81																																																																				
4	4,08																																																																				
5	4,37																																																																				
6	6,87																																																																				
7	3,69																																																																				
8	4,26																																																																				
9	3,88																																																																				
10	4,13																																																																				
Nombre de tests	Valeur en millisecondes																																																																				
1	0,24																																																																				
2	0,19																																																																				
3	0,20																																																																				
4	0,25																																																																				
5	0,34																																																																				
6	0,29																																																																				
7	0,23																																																																				
8	0,21																																																																				
9	0,22																																																																				
10	0,26																																																																				
Nombre de tests	Heuristique X fois plus rapide																																																																				
1	18																																																																				
2	19																																																																				
3	24																																																																				
4	17																																																																				
5	13																																																																				
6	24																																																																				
7	16																																																																				
8	20																																																																				
9	18																																																																				
10	16																																																																				

... en moyenne	4,40 ms	0,24 ms	Heuristique 18 fois plus rapide que Choco-Solver
Question avec le plus de réduction Grand Foyer - Temps 4			
... en moyenne	5,10 ms	0,26 ms	Heuristique 20 fois plus rapide que Choco-Solver

Finalement, ces résultats nous permettent de conclure sans ambiguïté que notre algorithme d'IA de Déduction Heuristique est plus rapide que l'IA de Déduction Choco-Solver. Dès la première itération, nous avions déjà constaté que les algorithmes heuristiques avaient généralement un avantage en termes de rapidité par rapport à Choco-Solver.

Cette différence de performance peut s'expliquer par la nature même de Choco-Solver : bien qu'il soit particulièrement efficace pour identifier une solution unique parmi un grand volume de données, il peut se révéler plus lent lorsqu'il s'agit de réduire progressivement les domaines possibles.

Afin de vérifier cette hypothèse, nous allons maintenant analyser les performances des deux IA face aux questions menant à l'identification du coupable. Plus précisément, nous mesurerons le temps que chacune met pour parvenir à cette conclusion, afin de confirmer ou d'inflimer notre théorie.

Réduction des domaines	Choco-Solver	Heuristique	Combien de fois plus efficace
Réponse en 2 : Salle - Temps 6 / Salle - Chauffeur			
... en moyenne	3,42 ms	10,60 ms	Choco-Solver 3 fois plus rapide que l'Heuristique

Réponse en 2 : Salle - Temps 6 / Salle - DéTECTIVE Salle - Chauffeur	<p>Temps Heuristique</p> <table border="1"> <thead> <tr> <th>Nombre de tests</th> <th>Valeur en millisecondes</th> </tr> </thead> <tbody> <tr><td>1</td><td>2.95</td></tr> <tr><td>2</td><td>3.68</td></tr> <tr><td>3</td><td>3.67</td></tr> <tr><td>4</td><td>3.43</td></tr> <tr><td>5</td><td>3.17</td></tr> <tr><td>6</td><td>4.67</td></tr> <tr><td>7</td><td>3.54</td></tr> <tr><td>8</td><td>4.88</td></tr> <tr><td>9</td><td>3.35</td></tr> <tr><td>10</td><td>3.14</td></tr> </tbody> </table>	Nombre de tests	Valeur en millisecondes	1	2.95	2	3.68	3	3.67	4	3.43	5	3.17	6	4.67	7	3.54	8	4.88	9	3.35	10	3.14	<p>Temps Heuristique</p> <table border="1"> <thead> <tr> <th>Nombre de tests</th> <th>Valeur en millisecondes</th> </tr> </thead> <tbody> <tr><td>1</td><td>10.89</td></tr> <tr><td>2</td><td>11.32</td></tr> <tr><td>3</td><td>13.16</td></tr> <tr><td>4</td><td>13.63</td></tr> <tr><td>5</td><td>10.54</td></tr> <tr><td>6</td><td>10.87</td></tr> <tr><td>7</td><td>8.86</td></tr> <tr><td>8</td><td>9.25</td></tr> <tr><td>9</td><td>10.18</td></tr> <tr><td>10</td><td>9.89</td></tr> </tbody> </table>	Nombre de tests	Valeur en millisecondes	1	10.89	2	11.32	3	13.16	4	13.63	5	10.54	6	10.87	7	8.86	8	9.25	9	10.18	10	9.89	<p>Question Solution Coupable en 3</p> <table border="1"> <thead> <tr> <th>Nombre de tests</th> <th>Choco-Solver x plus rapide</th> </tr> </thead> <tbody> <tr><td>1</td><td>3.69</td></tr> <tr><td>2</td><td>3.07</td></tr> <tr><td>3</td><td>3.56</td></tr> <tr><td>4</td><td>3.97</td></tr> <tr><td>5</td><td>3.33</td></tr> <tr><td>6</td><td>2.32</td></tr> <tr><td>7</td><td>2.50</td></tr> <tr><td>8</td><td>1.90</td></tr> <tr><td>9</td><td>3.04</td></tr> <tr><td>10</td><td>3.35</td></tr> </tbody> </table>	Nombre de tests	Choco-Solver x plus rapide	1	3.69	2	3.07	3	3.56	4	3.97	5	3.33	6	2.32	7	2.50	8	1.90	9	3.04	10	3.35
Nombre de tests	Valeur en millisecondes																																																																				
1	2.95																																																																				
2	3.68																																																																				
3	3.67																																																																				
4	3.43																																																																				
5	3.17																																																																				
6	4.67																																																																				
7	3.54																																																																				
8	4.88																																																																				
9	3.35																																																																				
10	3.14																																																																				
Nombre de tests	Valeur en millisecondes																																																																				
1	10.89																																																																				
2	11.32																																																																				
3	13.16																																																																				
4	13.63																																																																				
5	10.54																																																																				
6	10.87																																																																				
7	8.86																																																																				
8	9.25																																																																				
9	10.18																																																																				
10	9.89																																																																				
Nombre de tests	Choco-Solver x plus rapide																																																																				
1	3.69																																																																				
2	3.07																																																																				
3	3.56																																																																				
4	3.97																																																																				
5	3.33																																																																				
6	2.32																																																																				
7	2.50																																																																				
8	1.90																																																																				
9	3.04																																																																				
10	3.35																																																																				
... en moyenne	3,65 ms	10,86 ms	Choco-Solver 3 fois plus rapide que l'Heuristique																																																																		

Ainsi, l'IA de Déduction Choco-Solver se révèle finalement plus rapide que l'IA de Déduction Heuristique lorsqu'il s'agit d'identifier le coupable. Comme mentionné précédemment, Choco-Solver excelle dans la recherche d'une solution unique parmi un grand ensemble de possibilités, et cette supériorité devient particulièrement évidente lorsqu'on la compare à l'IA de Déduction Heuristique.

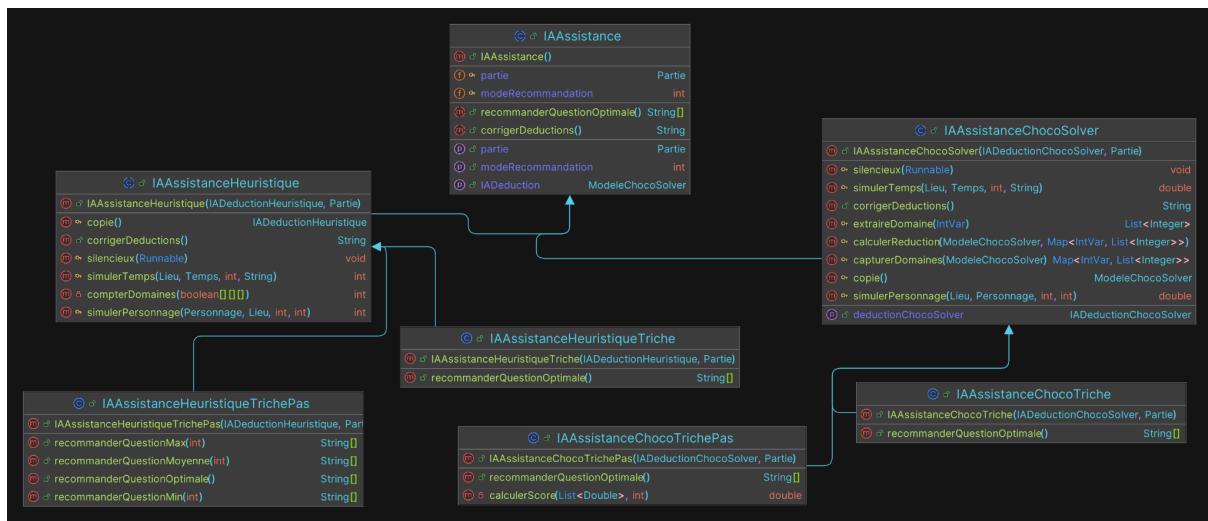
Cette différence de performance peut s'expliquer par le fait que, dans ce cas précis, le volume de données à traiter n'est pas excessivement important. Choco-Solver parvient donc à optimiser plus efficacement la recherche de la solution finale, compensant ainsi sa relative lenteur dans la réduction des domaines. Nous pouvons donc confirmer notre hypothèse.

2) Les IA d'Assistance

L'IA d'Assistance Choco-Solver :

A. Objectif de l'IA

L'IA d'assistance Choco a pour rôle de suggérer la meilleure question possible au joueur à chaque tour. Elle ne cherche pas directement à trouver le coupable, mais à réduire au maximum les domaines des variables du modèle ChocoSolver. Plus la réduction est importante, plus l'indice est jugé pertinent. Elle propose également une fonctionnalité de correction des déductions du joueur, en comparant les notes prises avec les conclusions logiques du solveur.



Fonctionnement

Deux variantes ont été implémentées :

❖ IA qui triche (IAAssistanceChocoTriche)

Elle parcourt tous les indices disponibles dans le scénario (temps ou personnage), connus à l'avance, et calcule pour chacun la réduction réelle du domaine du modèle ChocoSolver. Elle sélectionne celui ayant le plus grand impact sur les variables.

❖ IA qui ne triche pas (IAAssistanceChocoTrichePas)

Elle ne connaît pas les réponses à l'avance. Elle simule toutes les combinaisons possibles d'infos publiques/privées pour chaque question et calcule la réduction associée.

Trois stratégies de sélection sont disponibles :

➤ **Min** : question dont la pire réponse est la moins mauvaise,

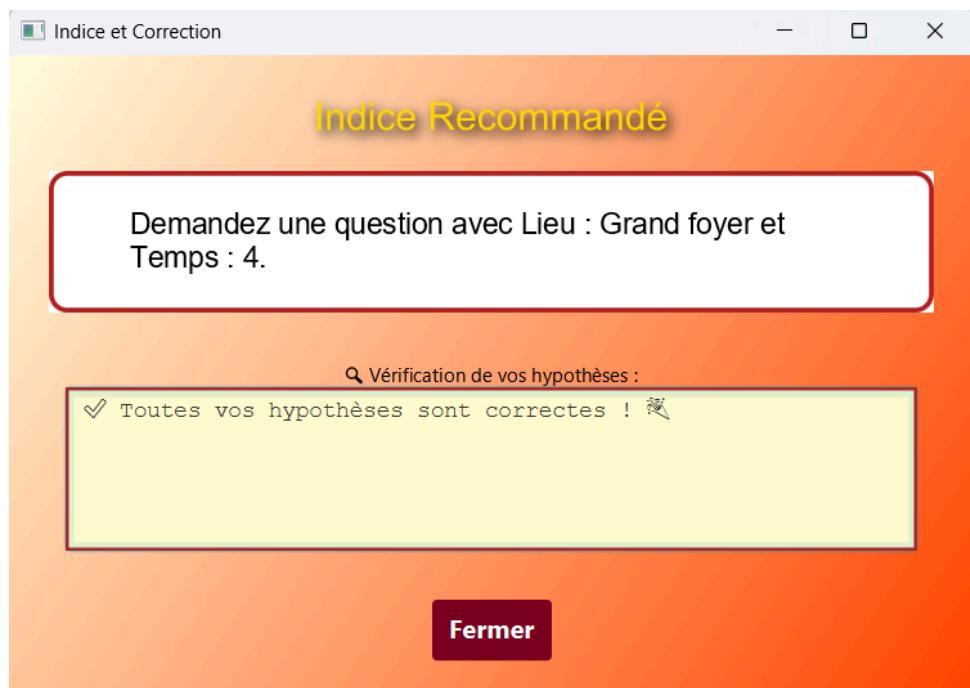
➤ **Max** : question dont la meilleure réponse est la plus efficace,

➤ **Moyenne** : question avec la meilleure réduction moyenne.

Dans les deux cas, une copie temporaire du modèle ChocoSolver est utilisée à chaque simulation, pour ne pas altérer la partie en cours.

Correction des déductions

L'IA peut également être utilisée pour corriger les hypothèses du joueur. En comparant les notes prises avec les déductions de l'IA de déduction ChocoSolver, elle identifie les incohérences (présence/absence/hypothèses erronées).



L'IA d'Assistance Heuristique :

Objectif de l'IA

Similaire à l'IA d'Assistance Choco-Solver.

Fonctionnement

Similaire à l'IA d'Assistance Choco-Solver sauf que l'IA de Déduction utilisée est l'Heuristique.

Correction des déductions

L'IA compare les notes du joueur avec le modèle booléen à trois dimensions utilisé par l'IA de Déduction Heuristique. Elle signale toute incohérence entre les hypothèses du joueur et les positions encore possibles dans le modèle :

- ❖ Présence déclarée alors qu'impossible,
- ❖ Absence notée alors que possible,
- ❖ Hypothèse fausse car exclue du domaine.

La comparaison des IA d'Assistance :

Contrairement aux deux IA de Déduction, notre analyse des IA d'Assistance se concentre sur leur efficacité en termes de temps d'exécution et de réduction de domaine. Nous allons évaluer les performances des quatre types d'IA d'Assistance (Triche, Min, Max et Moyenne) en utilisant Choco-Solver ainsi que notre approche heuristique. L'objectif est d'analyser leur efficacité respective et de tirer des conclusions à partir des résultats observés.

À noter que, pour l'ensemble de nos tests, la première itération sera systématiquement ignorée, car nous avons constaté qu'elle présente un temps d'exécution anormalement élevé par rapport à la moyenne.

Pour l'initialisation :

Temps	Comparaison	Combien de fois plus efficace																																			
Triche	<p>Initialisation - Temps - Triche</p> <table border="1"> <caption>Data for Initialisation - Temps - Triche</caption> <thead> <tr> <th>Nombre d'itération</th> <th>Choco-Solver (ms)</th> <th>Heuristique (ms)</th> </tr> </thead> <tbody> <tr><td>1</td><td>450</td><td>50</td></tr> <tr><td>2</td><td>250</td><td>40</td></tr> <tr><td>4</td><td>180</td><td>30</td></tr> <tr><td>6</td><td>120</td><td>20</td></tr> <tr><td>8</td><td>100</td><td>40</td></tr> <tr><td>10</td><td>80</td><td>20</td></tr> </tbody> </table>	Nombre d'itération	Choco-Solver (ms)	Heuristique (ms)	1	450	50	2	250	40	4	180	30	6	120	20	8	100	40	10	80	20	<p>Comparaison du temps</p> <table border="1"> <caption>Data for Comparaison du temps</caption> <thead> <tr> <th>Nombre d'itération</th> <th>Rapport</th> </tr> </thead> <tbody> <tr><td>1</td><td>8.0</td></tr> <tr><td>2</td><td>6.5</td></tr> <tr><td>4</td><td>5.0</td></tr> <tr><td>6</td><td>2.5</td></tr> <tr><td>8</td><td>1.0</td></tr> <tr><td>10</td><td>12.5</td></tr> </tbody> </table>	Nombre d'itération	Rapport	1	8.0	2	6.5	4	5.0	6	2.5	8	1.0	10	12.5
Nombre d'itération	Choco-Solver (ms)	Heuristique (ms)																																			
1	450	50																																			
2	250	40																																			
4	180	30																																			
6	120	20																																			
8	100	40																																			
10	80	20																																			
Nombre d'itération	Rapport																																				
1	8.0																																				
2	6.5																																				
4	5.0																																				
6	2.5																																				
8	1.0																																				
10	12.5																																				
... en moyenne	<ul style="list-style-type: none"> - Pour Choco-Solver : 180 ms - Pour Heuristique : 36.6 ms 	Heuristique 6 fois plus rapide que Choco-Solver																																			
Min	<p>Initialisation - Temps - Min</p> <table border="1"> <caption>Data for Initialisation - Temps - Min</caption> <thead> <tr> <th>Nombre d'itération</th> <th>Choco-Solver (ms)</th> <th>Heuristique (ms)</th> </tr> </thead> <tbody> <tr><td>1</td><td>1600</td><td>0</td></tr> <tr><td>2</td><td>1600</td><td>0</td></tr> <tr><td>4</td><td>1600</td><td>0</td></tr> <tr><td>6</td><td>1500</td><td>0</td></tr> <tr><td>8</td><td>1700</td><td>0</td></tr> <tr><td>10</td><td>1500</td><td>0</td></tr> </tbody> </table>	Nombre d'itération	Choco-Solver (ms)	Heuristique (ms)	1	1600	0	2	1600	0	4	1600	0	6	1500	0	8	1700	0	10	1500	0	<p>Min - Comparaison du temps</p> <table border="1"> <caption>Data for Min - Comparaison du temps</caption> <thead> <tr> <th>Nombre d'itération</th> <th>Rapport</th> </tr> </thead> <tbody> <tr><td>1</td><td>12.0</td></tr> <tr><td>2</td><td>12.5</td></tr> <tr><td>4</td><td>13.5</td></tr> <tr><td>6</td><td>13.0</td></tr> <tr><td>8</td><td>14.5</td></tr> <tr><td>10</td><td>12.5</td></tr> </tbody> </table>	Nombre d'itération	Rapport	1	12.0	2	12.5	4	13.5	6	13.0	8	14.5	10	12.5
Nombre d'itération	Choco-Solver (ms)	Heuristique (ms)																																			
1	1600	0																																			
2	1600	0																																			
4	1600	0																																			
6	1500	0																																			
8	1700	0																																			
10	1500	0																																			
Nombre d'itération	Rapport																																				
1	12.0																																				
2	12.5																																				
4	13.5																																				
6	13.0																																				
8	14.5																																				
10	12.5																																				
... en moyenne	<ul style="list-style-type: none"> - Pour Choco-Solver : 1596 ms - Pour Heuristique : 123 ms 	Heuristique 13 fois plus rapide que Choco-Solver																																			

Max	<p>Initialisation - Temps - Max</p> <table border="1"> <thead> <tr> <th>Nombre d'itération</th> <th>Choco-Solver (ms)</th> <th>Heuristique (ms)</th> </tr> </thead> <tbody> <tr><td>2</td><td>~1600</td><td>100</td></tr> <tr><td>4</td><td>~1700</td><td>100</td></tr> <tr><td>6</td><td>~1600</td><td>100</td></tr> <tr><td>8</td><td>~1500</td><td>100</td></tr> <tr><td>10</td><td>~1600</td><td>100</td></tr> </tbody> </table>	Nombre d'itération	Choco-Solver (ms)	Heuristique (ms)	2	~1600	100	4	~1700	100	6	~1600	100	8	~1500	100	10	~1600	100	<p>Max - Comparaison du temps</p> <table border="1"> <thead> <tr> <th>Nombre d'itération</th> <th>Heuristique X fois plus rapide</th> </tr> </thead> <tbody> <tr><td>2</td><td>~11.5</td></tr> <tr><td>4</td><td>~14.5</td></tr> <tr><td>6</td><td>~13.5</td></tr> <tr><td>8</td><td>~12.5</td></tr> <tr><td>10</td><td>~14.0</td></tr> </tbody> </table>	Nombre d'itération	Heuristique X fois plus rapide	2	~11.5	4	~14.5	6	~13.5	8	~12.5	10	~14.0
Nombre d'itération	Choco-Solver (ms)	Heuristique (ms)																														
2	~1600	100																														
4	~1700	100																														
6	~1600	100																														
8	~1500	100																														
10	~1600	100																														
Nombre d'itération	Heuristique X fois plus rapide																															
2	~11.5																															
4	~14.5																															
6	~13.5																															
8	~12.5																															
10	~14.0																															
... en moyenne	<ul style="list-style-type: none"> - Pour Choco-Solver : 1589 ms - Pour Heuristique : 124 ms 	Heuristique 13 fois plus rapide que Choco-Solver																														
Moyenne	<p>Initialisation - Temps - Max</p> <table border="1"> <thead> <tr> <th>Nombre d'itération</th> <th>Choco-Solver (ms)</th> <th>Heuristique (ms)</th> </tr> </thead> <tbody> <tr><td>2</td><td>~1600</td><td>100</td></tr> <tr><td>4</td><td>~1700</td><td>100</td></tr> <tr><td>6</td><td>~1600</td><td>100</td></tr> <tr><td>8</td><td>~1500</td><td>100</td></tr> <tr><td>10</td><td>~1700</td><td>100</td></tr> </tbody> </table>	Nombre d'itération	Choco-Solver (ms)	Heuristique (ms)	2	~1600	100	4	~1700	100	6	~1600	100	8	~1500	100	10	~1700	100	<p>Moyenne - Comparaison du temps</p> <table border="1"> <thead> <tr> <th>Nombre d'itération</th> <th>Heuristique X fois plus rapide</th> </tr> </thead> <tbody> <tr><td>2</td><td>~10.5</td></tr> <tr><td>4</td><td>~13.5</td></tr> <tr><td>6</td><td>~12.5</td></tr> <tr><td>8</td><td>~11.5</td></tr> <tr><td>10</td><td>~15.0</td></tr> </tbody> </table>	Nombre d'itération	Heuristique X fois plus rapide	2	~10.5	4	~13.5	6	~12.5	8	~11.5	10	~15.0
Nombre d'itération	Choco-Solver (ms)	Heuristique (ms)																														
2	~1600	100																														
4	~1700	100																														
6	~1600	100																														
8	~1500	100																														
10	~1700	100																														
Nombre d'itération	Heuristique X fois plus rapide																															
2	~10.5																															
4	~13.5																															
6	~12.5																															
8	~11.5																															
10	~15.0																															
... en moyenne	<ul style="list-style-type: none"> - Pour Choco-Solver : 1607 ms - Pour Heuristique : 126 ms 	Heuristique 13 fois plus rapide que Choco-Solver																														

Comme pour les IA de Déduction, l'Heuristique se révèle plus rapide. Toutefois, dans le cas des IA qui ne trichent pas, elle est encore plus rapide. Cela pourrait suggérer que Choco-Solver est moins performant lorsqu'il s'agit de simuler l'ensemble des combinaisons d'indices possibles.

Cependant, les questions générées pour les IA non tricheuses sont rarement comparables entre elles, ce qui rend difficile l'évaluation objective de la "meilleure" solution. Pour aller plus loin, nous allons donc analyser et comparer leurs scores de réduction.

Mais avant cela, nous allons vérifier notre première hypothèse concernant les temps d'exécution.

Après la meilleure question posée selon IA Assistance Choco-Solver Triche (Grand Foyer - Temps 4) :

Temps	Comparaison	Combien de fois plus efficace																																	
Triche	<p>1Q- Grand Foyer - Temps 4 - Temps - Triche</p> <table border="1"> <caption>Data for Triche - Comparaison du temps</caption> <thead> <tr> <th>Nombre d'itération</th> <th>Choco-Solver (ms)</th> <th>Heuristique (ms)</th> </tr> </thead> <tbody> <tr><td>0</td><td>420</td><td>50</td></tr> <tr><td>2</td><td>250</td><td>20</td></tr> <tr><td>4</td><td>180</td><td>20</td></tr> <tr><td>6</td><td>120</td><td>20</td></tr> <tr><td>8</td><td>100</td><td>20</td></tr> <tr><td>10</td><td>100</td><td>20</td></tr> </tbody> </table>	Nombre d'itération	Choco-Solver (ms)	Heuristique (ms)	0	420	50	2	250	20	4	180	20	6	120	20	8	100	20	10	100	20	<p>Triche - Comparaison du temps</p> <table border="1"> <caption>Data for Triche - Comparaison du temps</caption> <thead> <tr> <th>Nombre d'itération</th> <th>Heuristique X fois plus rapide</th> </tr> </thead> <tbody> <tr><td>2</td><td>18,00</td></tr> <tr><td>4</td><td>11,00</td></tr> <tr><td>6</td><td>4,00</td></tr> <tr><td>8</td><td>6,00</td></tr> <tr><td>10</td><td>2,00</td></tr> </tbody> </table>	Nombre d'itération	Heuristique X fois plus rapide	2	18,00	4	11,00	6	4,00	8	6,00	10	2,00
Nombre d'itération	Choco-Solver (ms)	Heuristique (ms)																																	
0	420	50																																	
2	250	20																																	
4	180	20																																	
6	120	20																																	
8	100	20																																	
10	100	20																																	
Nombre d'itération	Heuristique X fois plus rapide																																		
2	18,00																																		
4	11,00																																		
6	4,00																																		
8	6,00																																		
10	2,00																																		
... en moyenne	<ul style="list-style-type: none"> Pour Choco-Solver : 167 ms Pour Heuristique : 24 ms 	Heuristique 8 fois plus rapide que Choco-Solver																																	
Min	<p>1Q- Grand Foyer - Temps 4 - Temps - Min</p> <table border="1"> <caption>Data for Min - Comparaison du temps</caption> <thead> <tr> <th>Nombre d'itération</th> <th>Choco-Solver (ms)</th> <th>Heuristique (ms)</th> </tr> </thead> <tbody> <tr><td>0</td><td>1700</td><td>150</td></tr> <tr><td>2</td><td>1700</td><td>150</td></tr> <tr><td>4</td><td>1700</td><td>150</td></tr> <tr><td>6</td><td>1700</td><td>150</td></tr> <tr><td>8</td><td>1700</td><td>150</td></tr> <tr><td>10</td><td>1700</td><td>150</td></tr> </tbody> </table>	Nombre d'itération	Choco-Solver (ms)	Heuristique (ms)	0	1700	150	2	1700	150	4	1700	150	6	1700	150	8	1700	150	10	1700	150	<p>Min - Comparaison du temps</p> <table border="1"> <caption>Data for Min - Comparaison du temps</caption> <thead> <tr> <th>Nombre d'itération</th> <th>Heuristique X fois plus rapide</th> </tr> </thead> <tbody> <tr><td>2</td><td>9,00</td></tr> <tr><td>4</td><td>11,00</td></tr> <tr><td>6</td><td>10,50</td></tr> <tr><td>8</td><td>11,00</td></tr> <tr><td>10</td><td>11,50</td></tr> </tbody> </table>	Nombre d'itération	Heuristique X fois plus rapide	2	9,00	4	11,00	6	10,50	8	11,00	10	11,50
Nombre d'itération	Choco-Solver (ms)	Heuristique (ms)																																	
0	1700	150																																	
2	1700	150																																	
4	1700	150																																	
6	1700	150																																	
8	1700	150																																	
10	1700	150																																	
Nombre d'itération	Heuristique X fois plus rapide																																		
2	9,00																																		
4	11,00																																		
6	10,50																																		
8	11,00																																		
10	11,50																																		
... en moyenne	<ul style="list-style-type: none"> Pour Choco-Solver : 1626 ms Pour Heuristique : 150 ms 	Heuristique 11 fois plus rapide que Choco-Solver																																	
Max	<p>1Q- Grand Foyer - Temps 4 - Temps - Max</p> <table border="1"> <caption>Data for Max - Comparaison du temps</caption> <thead> <tr> <th>Nombre d'itération</th> <th>Choco-Solver (ms)</th> <th>Heuristique (ms)</th> </tr> </thead> <tbody> <tr><td>0</td><td>1600</td><td>150</td></tr> <tr><td>2</td><td>1600</td><td>150</td></tr> <tr><td>4</td><td>1600</td><td>150</td></tr> <tr><td>6</td><td>1600</td><td>150</td></tr> <tr><td>8</td><td>1600</td><td>150</td></tr> <tr><td>10</td><td>1600</td><td>150</td></tr> </tbody> </table>	Nombre d'itération	Choco-Solver (ms)	Heuristique (ms)	0	1600	150	2	1600	150	4	1600	150	6	1600	150	8	1600	150	10	1600	150	<p>Max - Comparaison du temps</p> <table border="1"> <caption>Data for Max - Comparaison du temps</caption> <thead> <tr> <th>Nombre d'itération</th> <th>Heuristique X fois plus rapide</th> </tr> </thead> <tbody> <tr><td>2</td><td>9,50</td></tr> <tr><td>4</td><td>11,50</td></tr> <tr><td>6</td><td>11,00</td></tr> <tr><td>8</td><td>11,50</td></tr> <tr><td>10</td><td>11,50</td></tr> </tbody> </table>	Nombre d'itération	Heuristique X fois plus rapide	2	9,50	4	11,50	6	11,00	8	11,50	10	11,50
Nombre d'itération	Choco-Solver (ms)	Heuristique (ms)																																	
0	1600	150																																	
2	1600	150																																	
4	1600	150																																	
6	1600	150																																	
8	1600	150																																	
10	1600	150																																	
Nombre d'itération	Heuristique X fois plus rapide																																		
2	9,50																																		
4	11,50																																		
6	11,00																																		
8	11,50																																		
10	11,50																																		
... en moyenne	<ul style="list-style-type: none"> Pour Choco-Solver : 1646 ms Pour Heuristique : 150 ms 	Heuristique 11 fois plus rapide que Choco-Solver																																	

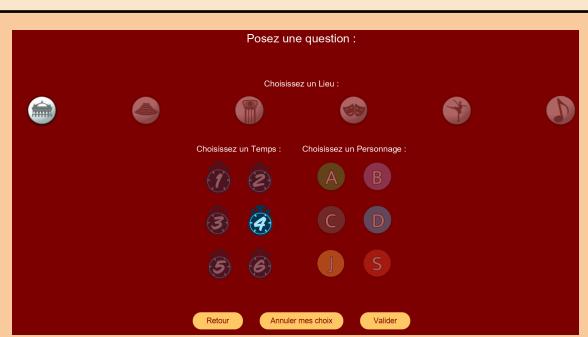
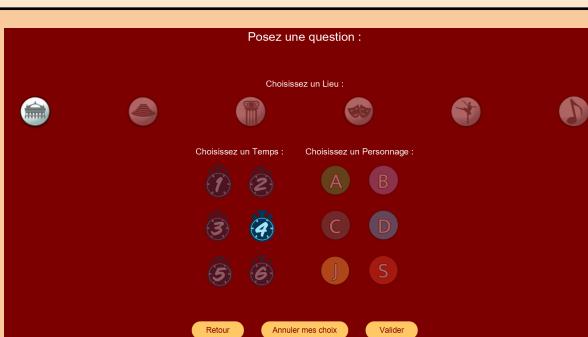
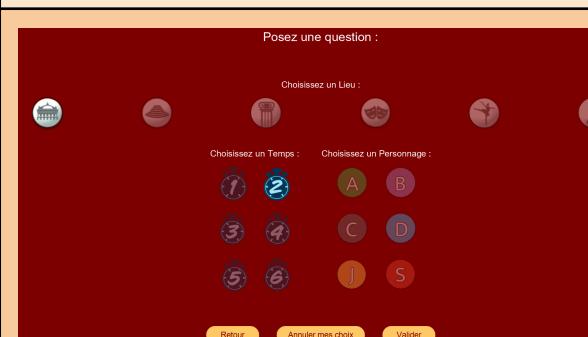
Moyenne	<p>1Q- Grand Foyer - Temps 4 - Temps - Moyenne</p> <table border="1"> <thead> <tr> <th>Nombre d'itération</th> <th>Choco-Solver (ms)</th> <th>Heuristique (ms)</th> </tr> </thead> <tbody> <tr><td>2</td><td>~1800</td><td>~150</td></tr> <tr><td>4</td><td>~1800</td><td>~150</td></tr> <tr><td>6</td><td>~1800</td><td>~150</td></tr> <tr><td>8</td><td>~1800</td><td>~150</td></tr> <tr><td>10</td><td>~1800</td><td>~150</td></tr> </tbody> </table>	Nombre d'itération	Choco-Solver (ms)	Heuristique (ms)	2	~1800	~150	4	~1800	~150	6	~1800	~150	8	~1800	~150	10	~1800	~150	<p>Moyenne - Comparaison du temps</p> <table border="1"> <thead> <tr> <th>Nombre d'itération</th> <th>Ratio (Heuristique / Choco-Solver)</th> </tr> </thead> <tbody> <tr><td>2</td><td>~10.5</td></tr> <tr><td>4</td><td>~11.5</td></tr> <tr><td>6</td><td>~11.0</td></tr> <tr><td>8</td><td>~10.5</td></tr> <tr><td>10</td><td>~11.5</td></tr> </tbody> </table>	Nombre d'itération	Ratio (Heuristique / Choco-Solver)	2	~10.5	4	~11.5	6	~11.0	8	~10.5	10	~11.5
Nombre d'itération	Choco-Solver (ms)	Heuristique (ms)																														
2	~1800	~150																														
4	~1800	~150																														
6	~1800	~150																														
8	~1800	~150																														
10	~1800	~150																														
Nombre d'itération	Ratio (Heuristique / Choco-Solver)																															
2	~10.5																															
4	~11.5																															
6	~11.0																															
8	~10.5																															
10	~11.5																															
... en moyenne	<ul style="list-style-type: none"> - Pour Choco-Solver : 1732 ms - Pour Heuristique : 154 ms 	Heuristique 11 fois plus rapide que Choco-Solver																														

Les résultats observés après la sélection de la meilleure question en termes de réduction confirment la théorie formulée précédemment.

Enfin, nous avons mené un test comparatif entre nos deux IA d'Assistance, en nous concentrant à la fois sur leur capacité de réduction et sur la pertinence des questions recommandées tout au long de la partie. Pour ce test, les questions ont été posées en suivant exclusivement les recommandations de l'IA d'Assistance basée sur Choco-Solver.

Après initialisation :

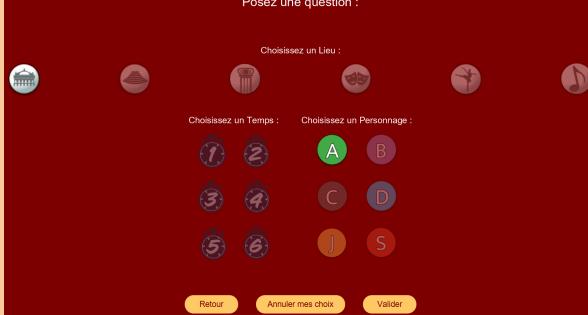
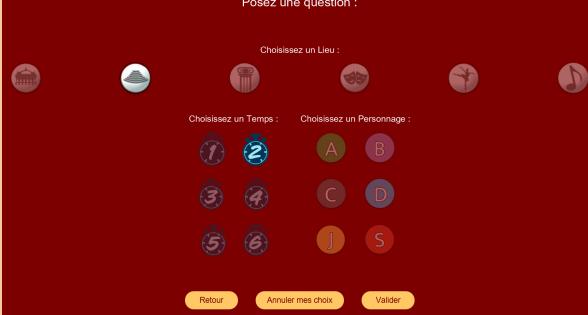
Triche	Choco-Solver		Heuristique
Réduction	20		20
Question recommandée	<p>Poser une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>Retour Annuler mes choix Valider</p> <p>Grand Foyer - Temps 4</p>		<p>Poser une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>Retour Annuler mes choix Valider</p> <p>Grand Foyer - Temps 4</p>
Min	Choco-Solver		Heuristique
Réduction	12		5

Question recommandée		
Max	Choco-Solver	Heuristique
Réduction	86	45
Question recommandée		
Moyenne	Choco-Solver	Heuristique
Réduction	25	19
Question recommandée		

Dès les premières itérations, des différences notables apparaissent entre les algorithmes des IA qui ne trichent pas. Les scores de réduction obtenus par Choco-Solver sont nettement supérieurs à ceux de l'Heuristique (*Min : 12/5, Max : 86/45, Moyenne : 25/19*).

Outre ces écarts de performance, les questions recommandées par chaque méthode diffèrent également, ce qui souligne des stratégies d'exploration distinctes. La seule exception concerne les IA tricheuses : dans ce cas précis, elles ont produit des scores de réduction identiques et recommandé la même question, suggérant un comportement parfaitement aligné.

Après la 1ère question (Grand Foyer - Temps 4) :

Triche	Choco-Solver	Heuristique
Réduction	20	20
Question recommandée	<p>Posez une question :</p>  <p>Salle - Temps 5</p>	<p>Posez une question :</p>  <p>Salle - Temps 5</p>
Min	Choco-Solver	Heuristique
Réduction	12	5
Question recommandée	<p>Posez une question :</p>  <p>Grand Foyer - Aventurière</p>	<p>Posez une question :</p>  <p>Grand Escalier - Temps 2</p>
Max	Choco-Solver	Heuristique
Réduction	77	44
Question recommandée	<p>Posez une question :</p>  <p>Grand Foyer - Temps 5</p>	<p>Posez une question :</p>  <p>Salle - Temps 3</p>
Moyenne	Choco-Solver	Heuristique
Réduction	23	18

Question recommandée	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps : Choisissez un Personnage :</p> <table border="1"> <tr><td>1</td><td>2</td><td>A</td><td>B</td></tr> <tr><td>3</td><td>4</td><td>C</td><td>D</td></tr> <tr><td>5</td><td>6</td><td>J</td><td>S</td></tr> </table> <p>Foyer de la danse - Temps 5</p>	1	2	A	B	3	4	C	D	5	6	J	S	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps : Choisissez un Personnage :</p> <table border="1"> <tr><td>1</td><td>2</td><td>A</td><td>B</td></tr> <tr><td>3</td><td>4</td><td>C</td><td>D</td></tr> <tr><td>5</td><td>6</td><td>J</td><td>S</td></tr> </table> <p>Foyer de la danse - Baronne</p>	1	2	A	B	3	4	C	D	5	6	J	S
1	2	A	B																							
3	4	C	D																							
5	6	J	S																							
1	2	A	B																							
3	4	C	D																							
5	6	J	S																							

Mêmes observations que pour l'initialisation.

Après la 2ème question (Salle - Temps 5) :

Triche	Choco-Solver	Heuristique																								
Réduction	20	20																								
Question recommandée	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps : Choisissez un Personnage :</p> <table border="1"> <tr><td>1</td><td>2</td><td>A</td><td>B</td></tr> <tr><td>3</td><td>4</td><td>C</td><td>D</td></tr> <tr><td>5</td><td>6</td><td>J</td><td>S</td></tr> </table> <p>Grand Foyer - Temps 5</p>	1	2	A	B	3	4	C	D	5	6	J	S	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps : Choisissez un Personnage :</p> <table border="1"> <tr><td>1</td><td>2</td><td>A</td><td>B</td></tr> <tr><td>3</td><td>4</td><td>C</td><td>D</td></tr> <tr><td>5</td><td>6</td><td>J</td><td>S</td></tr> </table> <p>Grand Foyer - Temps 5</p>	1	2	A	B	3	4	C	D	5	6	J	S
1	2	A	B																							
3	4	C	D																							
5	6	J	S																							
1	2	A	B																							
3	4	C	D																							
5	6	J	S																							
Min	Choco-Solver	Heuristique																								
Réduction	13	5																								
Question recommandée	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps : Choisissez un Personnage :</p> <table border="1"> <tr><td>1</td><td>2</td><td>A</td><td>B</td></tr> <tr><td>3</td><td>4</td><td>C</td><td>D</td></tr> <tr><td>5</td><td>6</td><td>J</td><td>S</td></tr> </table> <p>Grand Escalier - Journaliste</p>	1	2	A	B	3	4	C	D	5	6	J	S	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps : Choisissez un Personnage :</p> <table border="1"> <tr><td>1</td><td>2</td><td>A</td><td>B</td></tr> <tr><td>3</td><td>4</td><td>C</td><td>D</td></tr> <tr><td>5</td><td>6</td><td>J</td><td>S</td></tr> </table> <p>Foyer de la danse - Baronne</p>	1	2	A	B	3	4	C	D	5	6	J	S
1	2	A	B																							
3	4	C	D																							
5	6	J	S																							
1	2	A	B																							
3	4	C	D																							
5	6	J	S																							
Max	Choco-Solver	Heuristique																								
Réduction	63	43																								

Question recommandée	<p>Grand Foyer - Temps 6</p>	<p>Salle - Temps 3</p>
Moyenne	Choco-Solver	Heuristique
Réduction	21	14
Question recommandée	<p>Grand Escalier - Temps 4</p>	<p>Grand Foyer - Temps 2</p>

Mêmes observations que pour la première question.

Après la 3ème question (Grand Foyer - Temps 5) :

Triche	Choco-Solver	Heuristique
Réduction	14	14
Question recommandée	<p>Grand Escalier - Temps 3</p>	<p>Grand Escalier - Temps 3</p>
Min	Choco-Solver	Heuristique
Réduction	14	5

Question recommandée	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>Retour Annuler mes choix Valider</p> <p>Grand Foyer - Aventurière</p>	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>Retour Annuler mes choix Valider</p> <p>Foyer de la danse - Baronne</p>
Max	Choco-Solver	Heuristique
Réduction	40	35
Question recommandée	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>Retour Annuler mes choix Valider</p> <p>Grand Foyer - Temps 6</p>	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>Retour Annuler mes choix Valider</p> <p>Salle - Temps 3</p>
Moyenne	Choco-Solver	Heuristique
Réduction	15	12
Question recommandée	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>Retour Annuler mes choix Valider</p> <p>Grand Escalier - Temps 5</p>	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>Retour Annuler mes choix Valider</p> <p>Grand Foyer - Temps 2</p>

Mêmes observations que pour la question précédente.

Après la 4ème question (Grand Escalier - Temps 3) :

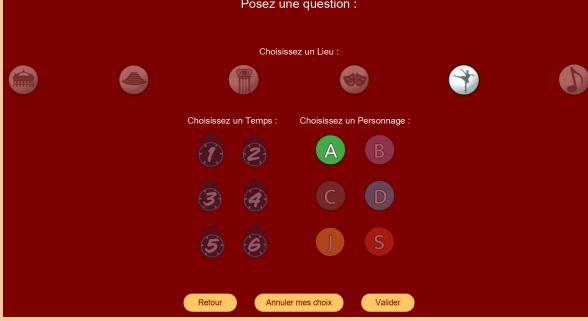
Triche	Choco-Solver	Heuristique																								
Réduction	13	13																								
Question recommandée	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps : Choisissez un Personnage :</p> <table border="1"> <tr><td>1</td><td>2</td><td>A</td><td>B</td></tr> <tr><td>3</td><td>4</td><td>C</td><td>D</td></tr> <tr><td>5</td><td>6</td><td>J</td><td>S</td></tr> </table> <p>Retour Annuler mes choix Valider</p> <p style="text-align: center;">Grand Foyer - Temps 3</p>	1	2	A	B	3	4	C	D	5	6	J	S	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps : Choisissez un Personnage :</p> <table border="1"> <tr><td>1</td><td>2</td><td>A</td><td>B</td></tr> <tr><td>3</td><td>4</td><td>C</td><td>D</td></tr> <tr><td>5</td><td>6</td><td>J</td><td>S</td></tr> </table> <p>Retour Annuler mes choix Valider</p> <p style="text-align: center;">Foyer de la danse - Aventurière</p>	1	2	A	B	3	4	C	D	5	6	J	S
1	2	A	B																							
3	4	C	D																							
5	6	J	S																							
1	2	A	B																							
3	4	C	D																							
5	6	J	S																							
Min	Choco-Solver	Heuristique																								
Réduction	14	7																								
Question recommandée	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps : Choisissez un Personnage :</p> <table border="1"> <tr><td>1</td><td>2</td><td>A</td><td>B</td></tr> <tr><td>3</td><td>4</td><td>C</td><td>D</td></tr> <tr><td>5</td><td>6</td><td>J</td><td>S</td></tr> </table> <p>Retour Annuler mes choix Valider</p> <p style="text-align: center;">Grand Foyer - Aventurière</p>	1	2	A	B	3	4	C	D	5	6	J	S	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps : Choisissez un Personnage :</p> <table border="1"> <tr><td>2</td><td>A</td><td>B</td></tr> <tr><td>3</td><td>4</td><td>C</td><td>D</td></tr> <tr><td>5</td><td>6</td><td>J</td><td>S</td></tr> </table> <p>Retour Annuler mes choix Valider</p> <p style="text-align: center;">Foyer du chant - Temps 2</p>	2	A	B	3	4	C	D	5	6	J	S	
1	2	A	B																							
3	4	C	D																							
5	6	J	S																							
2	A	B																								
3	4	C	D																							
5	6	J	S																							
Max	Choco-Solver	Heuristique																								
Réduction	32	39																								
Question recommandée	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps : Choisissez un Personnage :</p> <table border="1"> <tr><td>1</td><td>2</td><td>A</td><td>B</td></tr> <tr><td>3</td><td>4</td><td>C</td><td>D</td></tr> <tr><td>5</td><td>6</td><td>J</td><td>S</td></tr> </table> <p>Retour Annuler mes choix Valider</p> <p style="text-align: center;">Grand Foyer - Temps 6</p>	1	2	A	B	3	4	C	D	5	6	J	S	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps : Choisissez un Personnage :</p> <table border="1"> <tr><td>1</td><td>2</td><td>A</td><td>B</td></tr> <tr><td>3</td><td>4</td><td>C</td><td>D</td></tr> <tr><td>5</td><td>6</td><td>J</td><td>S</td></tr> </table> <p>Retour Annuler mes choix Valider</p> <p style="text-align: center;">Salle - Temps 2</p>	1	2	A	B	3	4	C	D	5	6	J	S
1	2	A	B																							
3	4	C	D																							
5	6	J	S																							
1	2	A	B																							
3	4	C	D																							
5	6	J	S																							
Moyenne	Choco-Solver	Heuristique																								
Réduction	14	11																								

Question recommandée	 <p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>A B C D J S</p> <p>Retour Annuler mes choix Valider</p> <p style="text-align: center;">Grand Foyer - Aventurière</p>	 <p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>2 A B 3 C D 5 J S</p> <p>Retour Annuler mes choix Valider</p> <p style="text-align: center;">Foyer du chant - Temps 2</p>
-----------------------------	--	--

Deux observations ressortent de cette analyse. Premièrement, bien que les IA qui triche obtiennent les mêmes scores de réduction, elles ne recommandent pas la même question. Cela suggère une différence dans les critères de sélection, possiblement liée au fait que Choco-Solver prend en compte un ou plusieurs éléments supplémentaires lors du choix de la question optimale.

Deuxièmement, pour les IA d'Assistance utilisant la stratégie Max, l'Heuristique obtient cette fois un score de réduction supérieur à celui de Choco-Solver, alors qu'à l'initialisation, ce dernier présentait quasiment le double de performance. Cette inversion est difficile à expliquer, d'autant plus que les IA de Déduction produisent ici une réduction de domaine équivalente. Une hypothèse serait que Choco-Solver utilise une méthode de calcul différente pour estimer la réduction, mais celle-ci reste opaque et hors de notre portée d'analyse.

Après la 5ème question (Grand Escalier - Temps 3) :

Triche	Choco-Solver	Heuristique
Réduction	13	13
Question recommandée	 <p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>A B C D J S</p> <p>Retour Annuler mes choix Valider</p> <p style="text-align: center;">Foyer de la danse - Aventurière</p>	 <p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>2 A B 3 C D 5 J S</p> <p>Retour Annuler mes choix Valider</p> <p style="text-align: center;">Foyer de la danse - Aventurière</p>
Min	Choco-Solver	Heuristique
Réduction	14	9

Question recommandée	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>A B C D J S</p> <p>Retour Annuler mes choix Valider</p> <p>Grand Foyer - Aventurière</p>	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>A B C D J S</p> <p>Retour Annuler mes choix Valider</p> <p>Foyer de la danse - Baronne</p>
Max	Choco-Solver	Heuristique
Réduction	32	39
Question recommandée	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>A B C D J S</p> <p>Retour Annuler mes choix Valider</p> <p>Grand Foyer - Temps 6</p>	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>A B C D J S</p> <p>Retour Annuler mes choix Valider</p> <p>Salle - Temps 2</p>
Moyenne	Choco-Solver	Heuristique
Réduction	14	10
Question recommandée	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>A B C D J S</p> <p>Retour Annuler mes choix Valider</p> <p>Grand Foyer - Aventurière</p>	<p>Posez une question :</p> <p>Choisissez un Lieu :</p> <p>Choisissez un Temps :</p> <p>Choisissez un Personnage :</p> <p>A B C D J S</p> <p>Retour Annuler mes choix Valider</p> <p>Grand Foyer - Temps 2</p>

Nous observons ici un phénomène similaire à celui constaté lors de la question précédente. Un point particulièrement intéressant réside dans le fait que la nouvelle question recommandée par l'IA d'Assistance Choco-Solver qui triche correspond précisément à celle suggérée par l'Heuristique au tour précédent. Cela confirme l'existence d'un ordre de préférence dans les stratégies, suggérant que Choco-Solver prend peut-être en compte des critères supplémentaires qui influencent le classement final des questions.

Conclusion

L'analyse comparative des IA d'Assistance, qu'elles soient tricheuses ou non, révèle plusieurs éléments clés. D'un point de vue rapidité d'exécution, l'Heuristique s'avère plus rapide que Choco-Solver, en particulier pour les IA qui ne trichent pas, où elle est en moyenne douze fois plus rapide. Cela laisse penser que Choco-Solver est moins adapté pour explorer rapidement un grand nombre de possibilités.

Cependant, en termes de scores de réduction, Choco-Solver se distingue clairement. Ses performances dépassent généralement celles de l'Heuristique concernant les stratégies Min, Max et Moyenne, bien que certaines exceptions existent, comme dans le cas de la stratégie Max où l'Heuristique obtient un score supérieur une fois qu'un nombre important de réductions ait été faites. Ces disparités, difficilement explicables avec les données disponibles, pourraient être dues à des différences internes dans le calcul de la réduction par Choco-Solver, auquel nous n'avons pas accès.

Autre point notable : les questions recommandées par les différentes IA d'Assistance varient, même lorsque les scores de réduction sont équivalents, ce qui suggère des logiques de sélection différentes. Cela est particulièrement visible chez les IA tricheuses, où Choco-Solver et l'Heuristique peuvent produire des résultats identiques mais recommandent des questions différentes.

Enfin, certaines convergences entre les méthodes apparaissent au fil des tours, renforçant l'idée que les deux approches peuvent parfois partager des préférences implicites, tout en les exprimant à des moments différents.

En somme, bien que l'Heuristique offre un gain de temps significatif, Choco-Solver présente des avantages en matière de qualité de réduction. Cela nous laisse un choix intéressant entre les deux IA qui pourra dépendre des priorités fixées : rapidité ou profondeur d'analyse.

3) L'IA joueuse

L'IA joueuse est une composante avancée de Kronologic, développée dans l'itération 6, dont le rôle est de résoudre entièrement une enquête de manière autonome, sans intervention humaine. Son objectif est de reproduire les actions d'un joueur expérimenté, en exploitant les capacités de l'IA d'assistance pour enchaîner les meilleures questions possibles jusqu'à identifier le coupable.

Fonctionnement

L'IA joueuse s'appuie sur une instance de l'IA d'assistance (Choco-Solver triche). Elle interagit dynamiquement avec le moteur du jeu pour :

- ❖ Poser une question optimale sélectionnée par l'IA d'assistance,
- ❖ Ajouter automatiquement les notes publiques, privées ou hypothétiques selon les réponses,
- ❖ Répéter ce processus jusqu'à ce que le coupable soit identifié.

Le mode IA joueuse est intégré dans le menu : une barre de chargement accompagne l'exécution et l'historique des déductions est affiché quand l'ia a fini ainsi qu'un film de tous les tours.

Intégration technique

La classe IAJoueuse regroupe l'ensemble de la logique du jeu automatique. Elle utilise les méthodes publiques de l'interface IAAssistance pour récupérer les questions les plus pertinentes, et interagit avec Partie pour poser les questions et manipuler les indices.

L'IA joueuse est déclenchée via un contrôleur, intégré dans l'architecture MVC. Elle peut être lancée sur un scénario classique ou généré automatiquement, grâce à une modification de la classe InitialisationJeu.

Intérêt dans le projet

L'IA joueuse joue un double rôle clé :

1. Vérification automatique des scénarios générés : elle valide que chaque scénario possède bien une solution unique, identifiable par raisonnement logique.
2. Démonstration autonome du jeu : elle offre un mode de résolution automatique qui valorise les IA développées, tout en facilitant la prise en main pour les utilisateurs novices.

3.6.2. Le générateur de scénarios

Afin d'offrir une plus grande jouabilité et de diversifier les enquêtes proposées au joueur, nous avons développé un générateur automatique de scénarios. Celui-ci permet de créer dynamiquement des fichiers JSON contenant tous les éléments nécessaires à une partie (personnages, lieux, déplacements, indices, solution), tout en garantissant la validité et la jouabilité complète du scénario.

Fonctionnement général

Le générateur produit aléatoirement :

- ❖ Les trajectoires de chaque personnage dans les différents lieux au fil des 6 temps du jeu (en respectant les connexions entre les lieux).
- ❖ Un crime unique, c'est-à-dire un moment où le détective se retrouve seul avec un autre personnage dans un lieu.
- ❖ Un ensemble cohérent d'indices privés et publics, extraits des trajectoires, au format attendu par le jeu.

Ces éléments sont ensuite regroupés dans un objet Partie, puis exportés au format JSON. Ce fichier JSON a exactement la même structure que celui décrivant l'enquête classique afin qu'il soit adapté à notre lecteur.

Contraintes vérifiées lors de la génération

Pour garantir la conformité d'un scénario, celui-ci doit suivre les règles du jeu, donc de nombreuses contraintes sont prises en compte lors de la génération :

1. Déplacements valides :

- Les personnages doivent se déplacer uniquement vers des lieux adjacents.
- Il est interdit de rester dans un même lieu deux temps d'affilée.

2. Scène du crime unique et cohérente :

- Le meurtre ne peut pas avoir lieu au temps 1.
- Il doit exister un seul moment dans toute la partie où le détective est seul avec exactement un autre personnage (le coupable).
- Ce moment devient la scène du crime et définit la solution officielle de l'enquête.

3. Indices pertinents :

- Les indices sont automatiquement générés à partir des trajectoires :
 - Indices Temps : nombre de personnes présentes dans un lieu à un instant donné.
 - Indices Personnages : nombre de passages du personnage choisi dans le lieu choisi.
- L'objectif est de permettre au joueur (ou à une IA) de retrouver la solution uniquement à partir des indices fournis.

4. Vérification par l'IA joueuse :

- Une IA joueuse résout le scénario automatiquement, en posant les meilleures questions jusqu'à identifier le coupable.
- Si elle échoue à trouver la bonne solution, le scénario est rejeté.

5. Unicité de la solution :

- Le générateur s'assure qu'il n'existe pas d'autre moment dans la partie où le détective est seul avec une autre personne.
- Cela garantit que les indices ne peuvent mener qu'à une seule solution correcte, sans fausse piste.

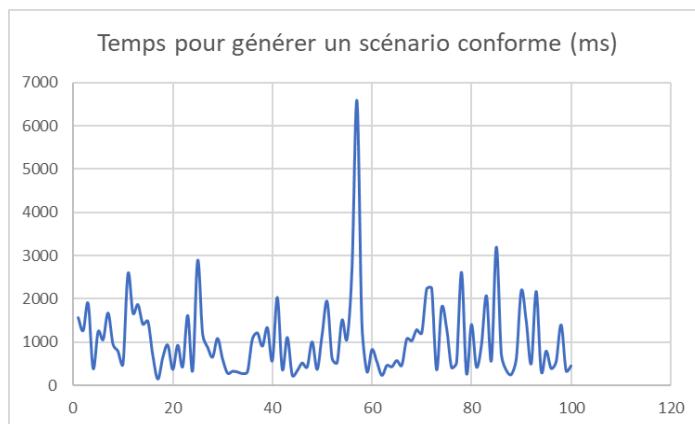
Si une (ou plusieurs) de ces contraintes n'est (ou ne sont) pas respectée(s), nous considérons le scénario comme non conforme et nous le rejettons. Nous relançons donc l'algorithme de génération jusqu'à en trouver un valide, que l'on propose ensuite au joueur.

Analyse des performances de l'algorithme

Une classe MainGenerateur a été créée afin d'effectuer des analyses sur les performances de ce dernier. Elle s'occupe de lancer un certain nombre de fois l'algorithme et recense pour chaque itération du programme dans un fichier .csv :

- **le temps** mis pour générer un scénario conforme
- **le nombre de scénarios non conformes** générés avant d'en trouver un valide

Pour les analyses ci-dessous, nous avons généré 100 scénarios conformes afin d'avoir un panel de données assez grand.

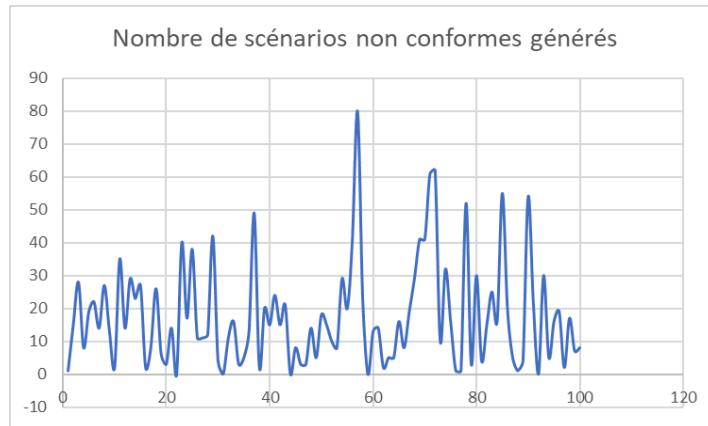


Ce graphique montre la durée nécessaire pour générer 100 scénarios valides, en millisecondes. Nous pouvons en conclure plusieurs choses :

- ❖ La plupart des scénarios sont générés en moins de 2000 ms, ce qui montre une bonne efficacité globale. Le fait de générer un scénario n'a donc pas de réel impact sur le jeu puisque cela reste fluide et ne demande pas beaucoup de temps.

- ❖ Quelques pics ponctuels se situent autour des 4000 ms voire même des 6500 ms, ce qui indique que certaines générations ont nécessité beaucoup d'essais avant d'être conformes.
- ❖ Cela reflète la nature aléatoire du processus et la complexité des règles à respecter.

Temps moyen pour générer un scénario : 1054,94 ms



Ce graphique représente, pour chaque scénario valide obtenu, le nombre de scénarios rejetés avant d'y parvenir.

- ❖ On constate que dans la majorité des cas, il faut entre 0 et 30 tentatives pour générer un scénario conforme.
- ❖ Certains cas extrêmes (ex. scénario 57) avoisinent les 80 scénarios invalides, ce qui explique les pics de temps observés dans le graphique précédent.
- ❖ Cela met en évidence la difficulté à respecter simultanément toutes les contraintes (déplacements, scène de crime unique, unicité de la solution...).

Nombre moyen de scénarios non conformes générés : 17,68

4. Conclusion

Le projet *Kronologic* nous a permis de relever de nombreux défis à la fois techniques et collaboratifs. À partir d'une idée originale mêlant jeu de déduction et modélisation informatique, nous avons conçu une application complète, interactive et modulaire, intégrant à la fois une interface utilisateur ergonomique, une logique métier solide et plusieurs intelligences artificielles spécialisées.

Ce projet nous a permis d'approfondir de nombreux domaines abordés durant notre formation : architecture logicielle (MVC), conception orientée objet, développement d'interface graphique, manipulation de fichiers JSON, test unitaire, et bien sûr, mise en œuvre de l'intelligence artificielle à travers deux approches complémentaires :

- l'une avec Choco-Solver
- l'autre avec un raisonnement heuristique personnalisé.

Les différentes fonctionnalités développées (IA d'assistance, IA joueuse, système d'hypothèses, générateur de scénarios, ...) ont toutes enrichi le gameplay tout en constituant des modules techniques ambitieux. Leur intégration fluide et cohérente témoigne de notre capacité à structurer, tester et faire évoluer un projet complexe sur le long terme.

Au-delà des compétences techniques, *Kronologic* a aussi renforcé notre capacité à travailler en équipe, à nous adapter face aux imprévus, et à documenter notre travail pour le rendre transmissible. Le projet est aujourd'hui pleinement fonctionnel, et ses fondations solides ouvrent la voie à de futures extensions : nouveaux scénarios, IA plus avancées, interface web ou mobile, ou encore enrichissement du moteur de déduction.

Le projet peut tout à fait être repris et poursuivi l'année prochaine par un autre groupe d'étudiants. Nous pouvons envisager plusieurs axes d'amélioration ou d'extension :

- ❖ Développement d'une interface web ou mobile.
- ❖ Génération de scénarios de différentes difficultés.
- ❖ Ajout d'un mode multijoueur local ou en ligne.

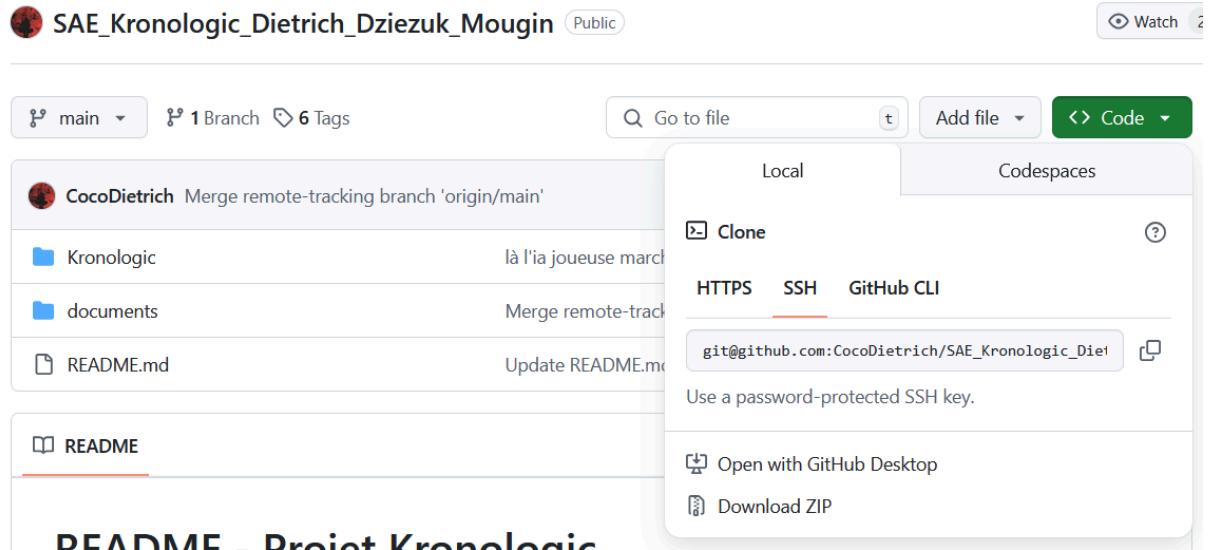
5. Annexes

5.1. Mode d'emploi pour installer à partir de github

Pour l'installation depuis github, il faut vous rendre sur l'adresse suivante :

https://github.com/CocoDietrich/SAE_Kronologic_Dietrich_Dziezuk_Mougin

Une fois sur la page, vous devez cliquer sur le bouton code, puis sur SSH et copier l'adresse.



Une fois cela réalisé, il ne vous reste plus qu'à lancer la commande git clone sur git bash :

```
enzom@LeMouj2 MINGW64 ~
$ git clone git@github.com:CocoDietrich/SAE_Kronologic_Dietrich_Dziezuk_Mougin.git|
```

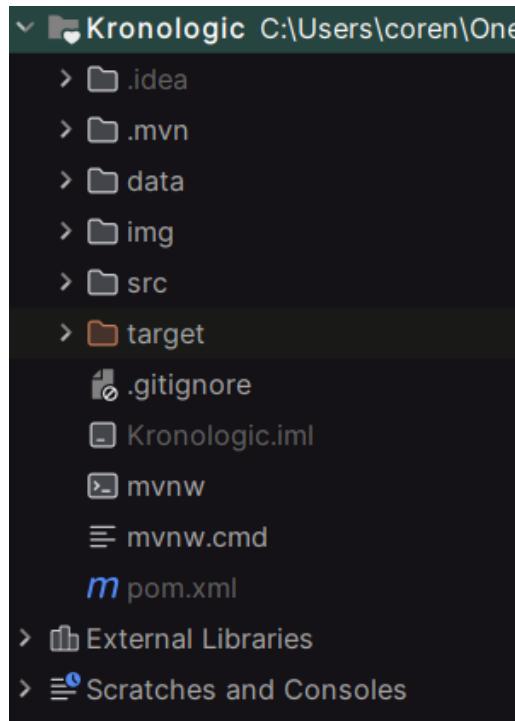
Il ne vous reste plus qu'à suivre le mode d'emploi pour lancer le jeu.

Work in progress

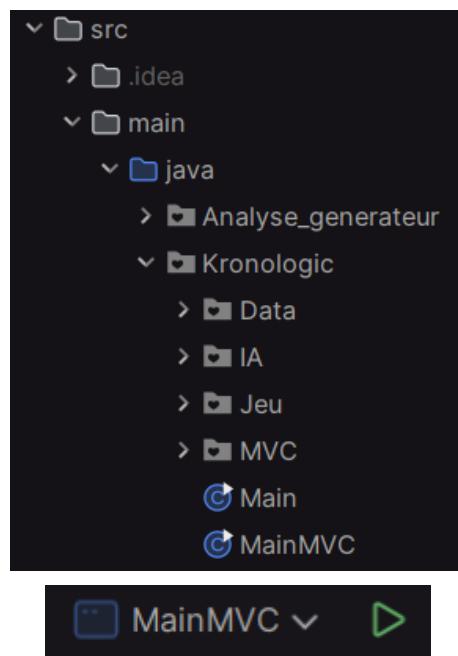
Sur le dépôt github, un dossier contenant les fichiers .jar nécessaires au fonctionnement de l'application sera accessible. Les .jar ne sont pas encore créés, mais le seul qui faudra lancer sera le *kronologic.jar*. Nous avons aussi l'idée de rendre l'application lancable sans téléchargement directement depuis github.

5.2. Mode d'emploi pour lancer le jeu (et fonctionnement)

Après avoir cloné le dépôt Github, ouvrez IntelliJ et ouvrez un nouveau projet avec comme racine le répertoire Kronologic. Vous devriez obtenir quelque chose du genre :

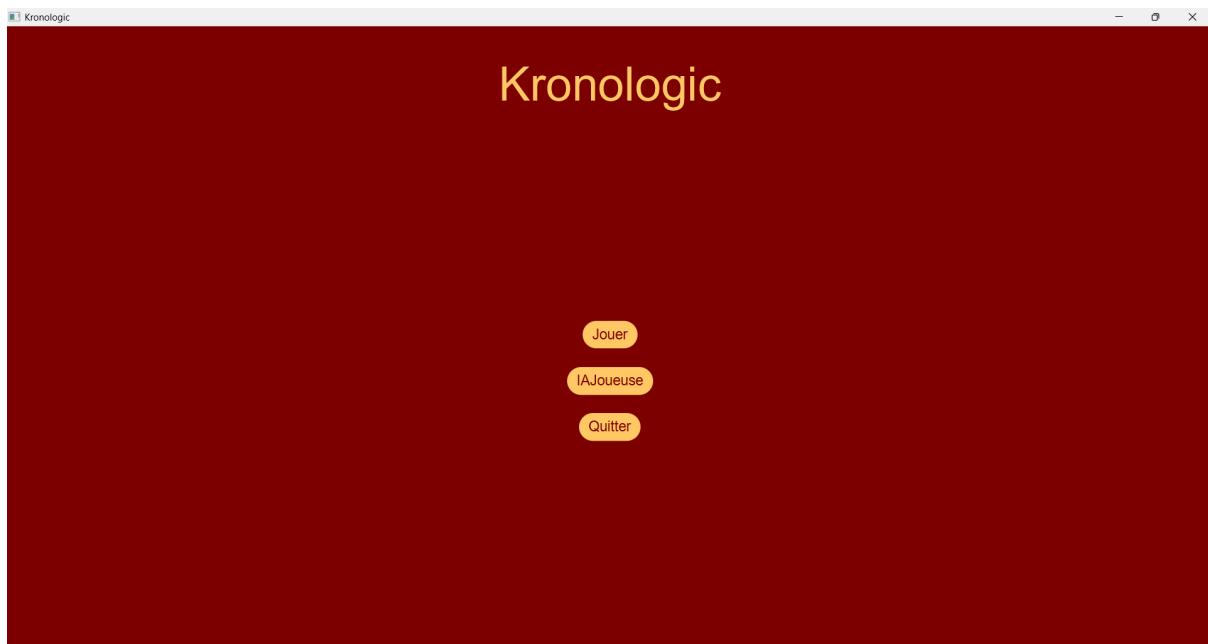


Ensute, afin de lancer l'application, il suffit d'exécuter la classe MainMVC, qui se trouve dans le répertoire src/main/java/Kronologic avec le bouton "Run" (flèche verte) :

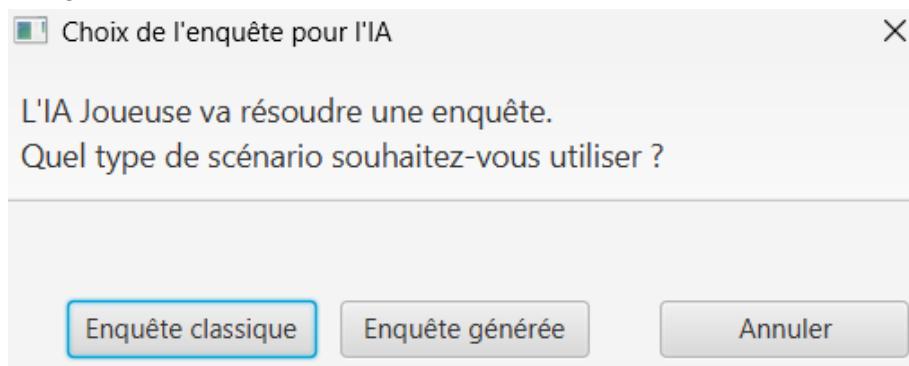


On arrive ensuite sur la page d'accueil de l'application (ci-dessous), où le joueur peut effectuer trois actions :

- Jouer une partie
- Simuler une partie grâce à l'IA joueuse qui va trouver la solution de manière autonome
- Quitter l'application



On va d'abord voir comment utiliser l'IA joueuse. Dans un premier temps, le joueur va devoir choisir s'il veut qu'elle raisonne sur le scénario classique ou un scénario généré par l'algorithme de génération de scénarios :



Après avoir choisi le type de scénario, l'IA va résoudre l'enquête d'elle-même et afficher son cheminement de pensée (les questions qu'elle a posées) ainsi que la solution qu'elle a trouvée.

===== 🗂️ Résultats de l'IA 🗂️ =====

- Tour 1 : Question posée → Lieu : Grand foyer | Temps : 4
- Tour 2 : Question posée → Lieu : Salle | Temps : 5
- Tour 3 : Question posée → Lieu : Grand foyer | Temps : 5
- Tour 4 : Question posée → Lieu : Grand escalier | Temps : 3
- Tour 5 : Question posée → Lieu : Grand foyer | Temps : 3
- Tour 6 : Question posée → Lieu : Foyer de la danse | Personnage : Aventurière
- Tour 7 : Question posée → Lieu : Grand foyer | Temps : 6
- Tour 8 : Question posée → Lieu : Salle | Personnage : Chauffeur
- Tour 9 : Question posée → Lieu : Salle | Temps : 6

===== 🕵️ Coupable Identifié ! =====

Le coupable est 🕵️ C dans le lieu 🕒 3 au temps ⏱ 6.

[Voir le film](#)

Le bouton “Voir le film” permet au joueur d'avoir accès aux déductions que l'IA au fur et à mesure de la partie en fonction des indices dont elle disposait. Par exemple, avec seulement les positions initiales, il lui est possible d'assumer qu'un personnage ne sera pas à un lieu éloigné au temps 2 (voir ci-dessous).



On va maintenant expliquer comment utiliser l'application quand le joueur joue par lui-même. Nous allons montrer l'utilité de chaque bouton, comment accéder aux différentes fonctionnalités du jeu (pose de question, film du joueur, etc...) ainsi qu'aux IA permettant d'aider le joueur.

En premier lieu, le joueur est accueilli par un pop-up lui permettant de comprendre le synopsis de l'enquête et ce qu'il va devoir faire :



Après avoir pris connaissance de l'enquête, le joueur a accès aux positions initiales des personnages, choses grâce auxquelles il peut tirer, à l'instar de l'IA joueuse, des conclusions.

De nombreux boutons sont présents au sein de l'interface, dont nous allons détailler les rôles ci-dessous :



- **Bouton de règles** (en haut à droite) : Ce bouton permet d'afficher les règles du jeu, ce qui sert de mini mode d'emploi de l'application.

Poser une question

- **Pose de question** (en bas au centre) : Permet d'arriver sur la vue de pose d'une question, où le joueur choisit un lieu, ainsi qu'un temps ou un personnage sur lequel recevoir un indice.

Faire une déduction

- **Faire une déduction** (en bas à gauche) : Permet d'arriver sur la vue de déduction, où le joueur devra choisir un triplet lieu-temps-personnage qu'il pense être la solution à l'énigme.

Film du joueur

- **Film du joueur** (en haut au centre) : Permet d'accéder à l'évolution des pions placés par le joueur au cours de la partie en fonction des indices obtenus.

Film de la partie

- **Film de la partie** (en haut au centre) : Permet de voir l'évolution des déplacements réels des personnages au cours de l'enquête, pour comparer ses notes et voir le résultat.

Changer affichage

- **Changer d'affichage** (au milieu à droite) : Permet de changer de type d'affichage et donc d'accéder à la vue sous forme de tableau du plateau de jeu. Tous les boutons y sont également présents afin d'avoir accès à toutes les fonctionnalités du jeu.

Tour 1 :

Il y a 1 personne(s) dans le lieu Grand escalier au pas de temps 2.
De plus, le personnage Baronne est dans ce lieu à ce pas de temps.

- **Historique des indices obtenus** (au milieu à gauche) : Zone de texte avec l'historique écrit des indices par rapport aux questions posées au cours de la partie en fonction des tours. Permet d'avoir un suivi des questions de la partie.



- **Pions d'hypothèses et d'absences** (au milieu) : Ces checkbox permettent au joueur de placer le type de pion qu'il souhaite :
 - ❖ Pion de présence : les deux checkbox ne sont pas cochées
 - ❖ Pion d'hypothèse : seule la checkbox d'hypothèse est cochée
 - ❖ Pion d'absence : seule le checkbox d'absence est cochée
 - ❖ Pion d'hypothèse d'absence : les deux checkbox sont cochées



- **Affichage des types de pions** (en haut au centre et au milieu) : Le joueur a la possibilité via ces checkbox d'afficher ou non les différents types de pions pour un souci de clarté.



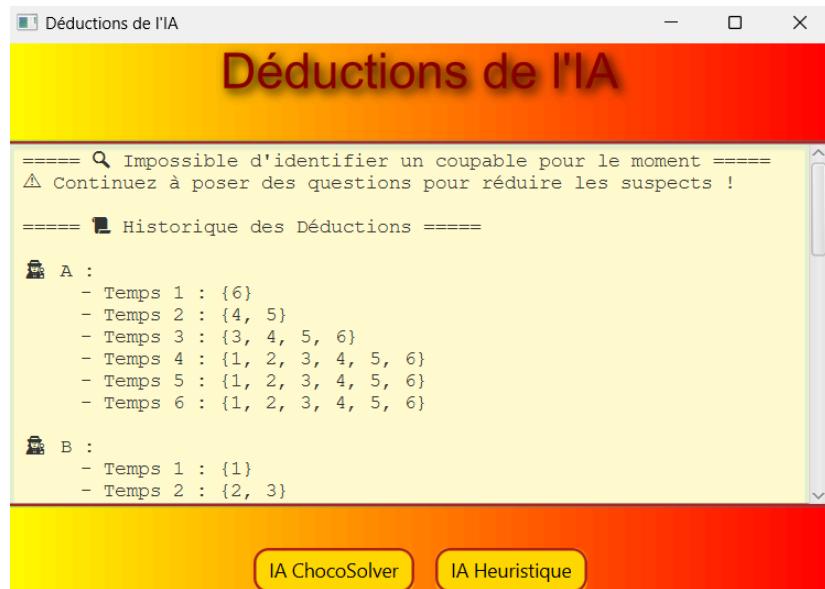
- **Pions de personnage** (au milieu) : Pions que l'on peut déplacer sur des salles représentant les personnages de l'enquête. Il ne peut y avoir qu'un pion par salle pour un temps donné par personnage. Pour enlever un pion déjà placé, il suffit de le déplacer vers une zone non valide.



- **Pion de nombre** (au milieu) : Pion également déplaçable sur les cartes. Ce pion représente un nombre de personnages, il est donc utile dans certaines situations. Par exemple, si un indice nous dit qu'il y a trois personnages dans une salle à un temps donné mais qu'on ne sait pas lesquels. Cependant, le pion n'est pas déplaçable dans l'état ci-dessus. Afin d'attribuer une valeur à ce pion, il faut d'abord double-cliquer dessus, entrer un chiffre puis valider. On peut ensuite le placer à notre guise.

Déduction de l'IA

- **Déduction de l'IA** (au milieu à droite) : Permet d'accéder aux déductions faites par les IA de déduction (Choco-Solver et heuristique) en fonction des indices obtenus par le joueur. Il peut s'en servir afin de comparer les siennes avec celles des IA.



- **Déductions de l'IA Choco-Solver et Heuristique** : Après avoir cliqué sur le bouton précédent, les déductions de l'IA de déduction Choco-Solver s'affichent automatiquement, pour changer d'approche. Il suffit de cliquer sur les boutons en bas, pour passer notamment sur l'IA de déduction heuristique.

Demander un indice

- **Demander un indice** (en bas à droite) : Permet d'accéder au menu de l'IA d'assistance. Il peut s'en servir afin de vérifier ses hypothèses mais également pour lui demander quelle est la question optimale à poser.



- **Menu de l'IA d'assistance** : Ici, le joueur va pouvoir choisir entre les différentes types d'IA afin de sélectionner celle qui lui convient le mieux. Les différences entre les IA ont été évoquées plus haut dans ce rapport, je ne vais donc pas les réexpliquer ici.