UNIVERSITÉ DE LORRAINE | IUT nancy Charlemagne Scolarité

S10

SOKOBAN & IA

Rapport final

Octobre 2020 / Avril 2021

Anthony BRIOT - Benoit NICOL - Lucas SAKER - Quentin VRIGNON

Tuteur : Vincent THOMAS

Table des matières

I - Introduction

A - Objet du document

Ce document est le rapport final de notre projet tutoré. Il a pour but de présenter et d'expliquer notre projet, de sa phase de conception à la phase finale de réalisation. Nous allons présenter en détail la conception, la réalisation mais également l'organisation du projet qui s'est faite en suivant la méthode Agile (SCRUM).

B - Présentation du sujet

Le sujet de notre projet est donc la création d'une version du jeu Sokoban, jeu japonais, avec l'ajout d'une IA qui permet la résolution d'un niveau du jeu.

- Le jeu Sokoban :

Sokoban est un jeu japonais dans lequel on incarne un personnage et dont l'objectif est de placer des caisses sur des emplacements précis.

Le jeu est en 2D avec une vue de haut. Il y a 4 déplacements possibles, haut, bas, gauche et droite.

Le personnage peut bouger une caisse uniquement en étant sur une case adjacente et en tentant de se déplacer en direction de la caisse, à condition que rien ne bloque ce mouvement, c'est-à-dire qu'il ne faut pas qu'une autre caisse ou alors qu'un mur se trouve derrière cette caisse.

Pour réussir un niveau, il faut parvenir à placer toutes les caisses sur les emplacements.

- L'IA :

Tout d'abord, I.A. est un sigle qui signifie Intelligence Artificielle. Il s'agit donc d'une réalisation humaine (dans notre cas il s'agit d'un programme) permettant de simuler l'intelligence humaine et de réaliser une tâche demandée de manière automatique.

Dans notre cas, l'IA permet de trouver la solution la plus optimale pour résoudre un niveau du jeu Sokoban. En d'autres termes, il s'agit de trouver les mouvements que doit faire le personnage pour placer les caisses sur les emplacements.

Pour ce faire, notre IA doit utiliser un algorithme de recherche.

Un algorithme de recherche est un type d'algorithme qui permet de trouver une ou plusieurs solutions à un problème de recherche donné. Il en existe plusieurs, chacun ayant une façon de "rechercher" différente.

Dans le cas de notre projet, nous avons choisi l'algorithme de recherche de chemin A* car c'est un des algorithmes les plus connus et les plus "simples".

Cet algorithme recherche le chemin entre un nœud initial et un nœud final, tous deux donnés au lancement de l'algorithme (un nœud étant un état dans le chemin).

A* fonctionne avec deux listes de nœuds : une liste dite "ouverte" et une liste dite "fermée".

La liste ouverte contient les nœuds qu'il faut encore explorer (elle peut donc contenir le nœud final qu'il faut atteindre) et la liste fermée contient tous les nœuds qui ont déjà été explorés (afin de ne pas passer deux fois au même endroit).

On parcourt donc la liste ouverte et, pour chaque nœud, on vérifie s'il correspond à l'objectif, puis on recherche les nœuds suivants que l'on ajoute dans la liste ouverte et enfin, on ajoute le nœud courant dans la liste fermée.

Pour finir, A* fonctionne avec une heuristique. Il s'agit d'une fonction qui permet de calculer la distance restante entre l'objectif et un nœud en particulier. Elle sert à trier la liste ouverte afin de déterminer les nœuds qui semblent les plus proches de l'objectif.

L'heuristique est une sorte de guide pour l'IA car elle lui indique si un chemin est meilleur qu'un autre.

Nous avons donc dû comprendre puis implémenter toutes ces spécificités dans notre projet.

A terme, l'implémentation de notre IA doit permettre la vulgarisation de l'IA (dans le cadre de fêtes de la science, par exemple).

C - Présentation de l'équipe et répartition des rôles

L'équipe de notre projet "Sokoban et IA" est composée de 4 étudiants en 2ème année de DUT Informatique : Anthony BRIOT, Benoît NICOL, Lucas SAKER et Quentin VRIGNON. Pour la répartition des rôles, nous avons décidé de travailler en équipe sur toutes les parties de conception afin de confronter les idées de chacun et de créer un consensus. En ce qui concerne la partie développement, nous avons opté pour la création de binômes que l'on a changé suivant les différentes itérations.

D - Déroulement du projet

Le projet est composé de 3 phases :

- Création d'une nouvelle version du jeu Sokoban.
- Création d'un algorithme de recherche et implémentation d'une I.A. permettant de résoudre les niveaux de Sokoban.
- Permettre la vulgarisation de l'I.A. à travers notre application.

<u>1ère phase : Création d'une nouvelle version de Sokoban</u>

➔ Cette phase est constituée d'une première partie de conception, puis d'une partie de création (codage). La conception est primordiale afin de pouvoir avoir une idée précise de ce que l'on veut coder.

<u>2ème phase : Création d'un algorithme de recherche et implémentation d'une I.A. permettant de résoudre les niveaux de Sokoban</u>

➔ Pour cette phase, il est tout d'abord nécessaire de comprendre ce qu'est un algorithme de recherche. Ensuite, il faut décider de l'algorithme de recherche que l'on va utiliser (en l'occurrence A*), comprendre son fonctionnement, et en produire une version simplifiée.
Une fois que cela est fait, il faut coder cet algorithme dans le projet, ce qui consiste à créer une I.A., et l'implémenter au code déjà existant du jeu.

<u>3ème phase : Permettre la vulgarisation de l'I.A. à travers l'application</u>

➔ Cela consiste à faciliter l'affichage du fonctionnement de l'I.A.. Par exemple, il est possible d'ajouter des boutons permettant de "dérouler" le chemin trouvé par l'I.A. menant à la solution déplacement par déplacement, en avant et en arrière.

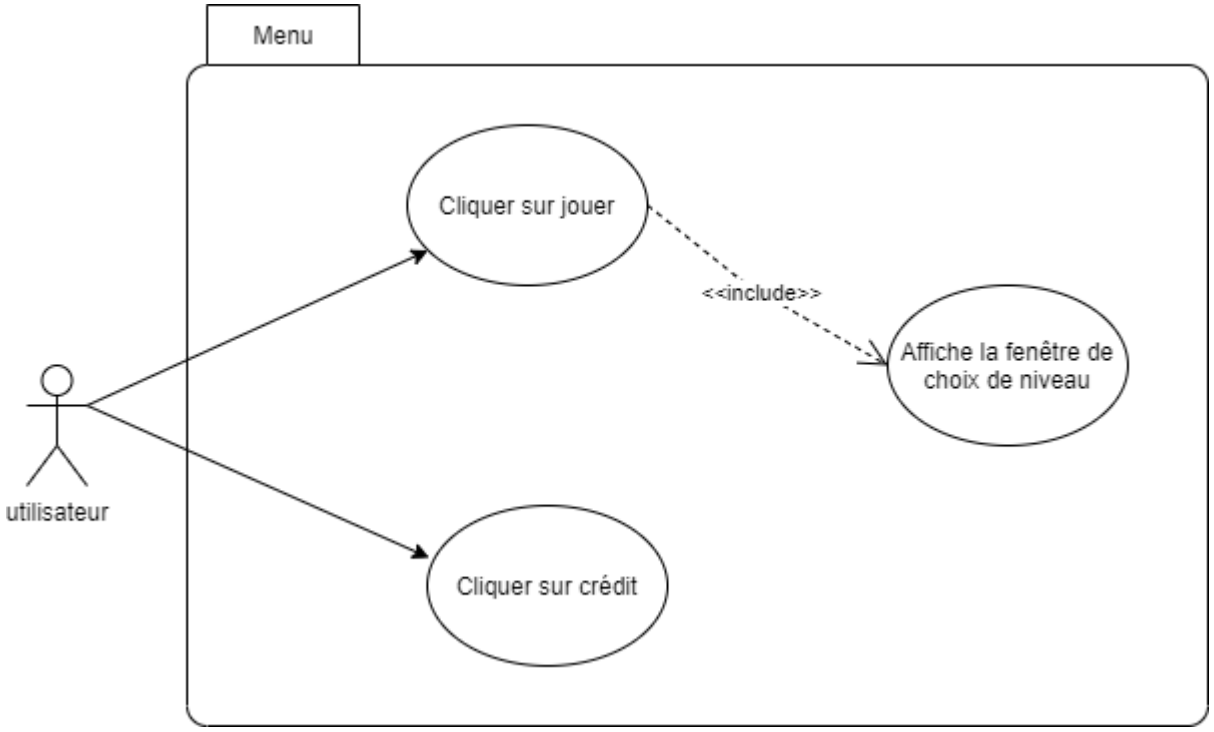
II - Analyse

A - Conception initiale

1) Diagrammes de cas d'utilisation

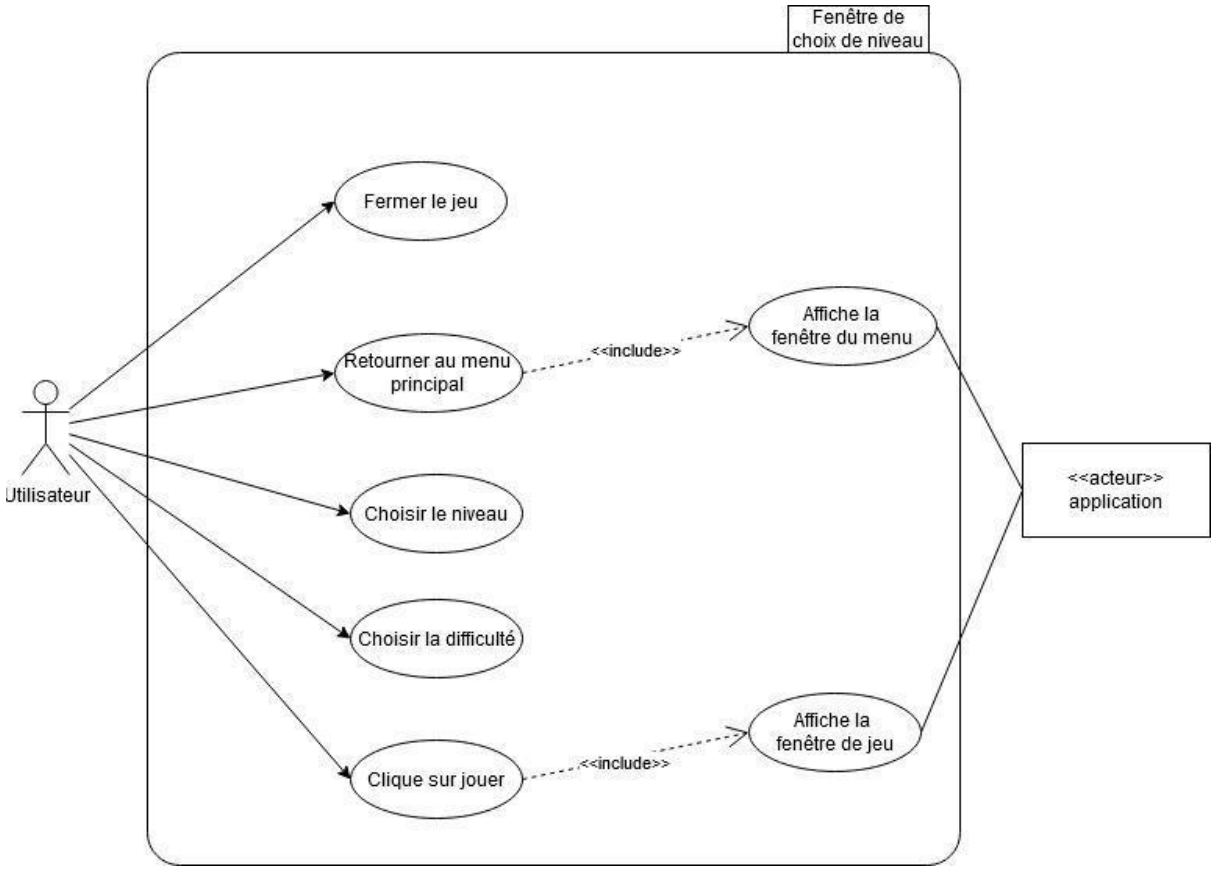
Lors de la phase de conception initiale, nous avons choisi de réaliser plusieurs diagrammes dits de "cas d'utilisation" afin de réfléchir au meilleur modèle possible pour notre interface graphique, tout en intégrant les différentes fonctionnalités.

Diagramme de cas d'utilisation du menu principal



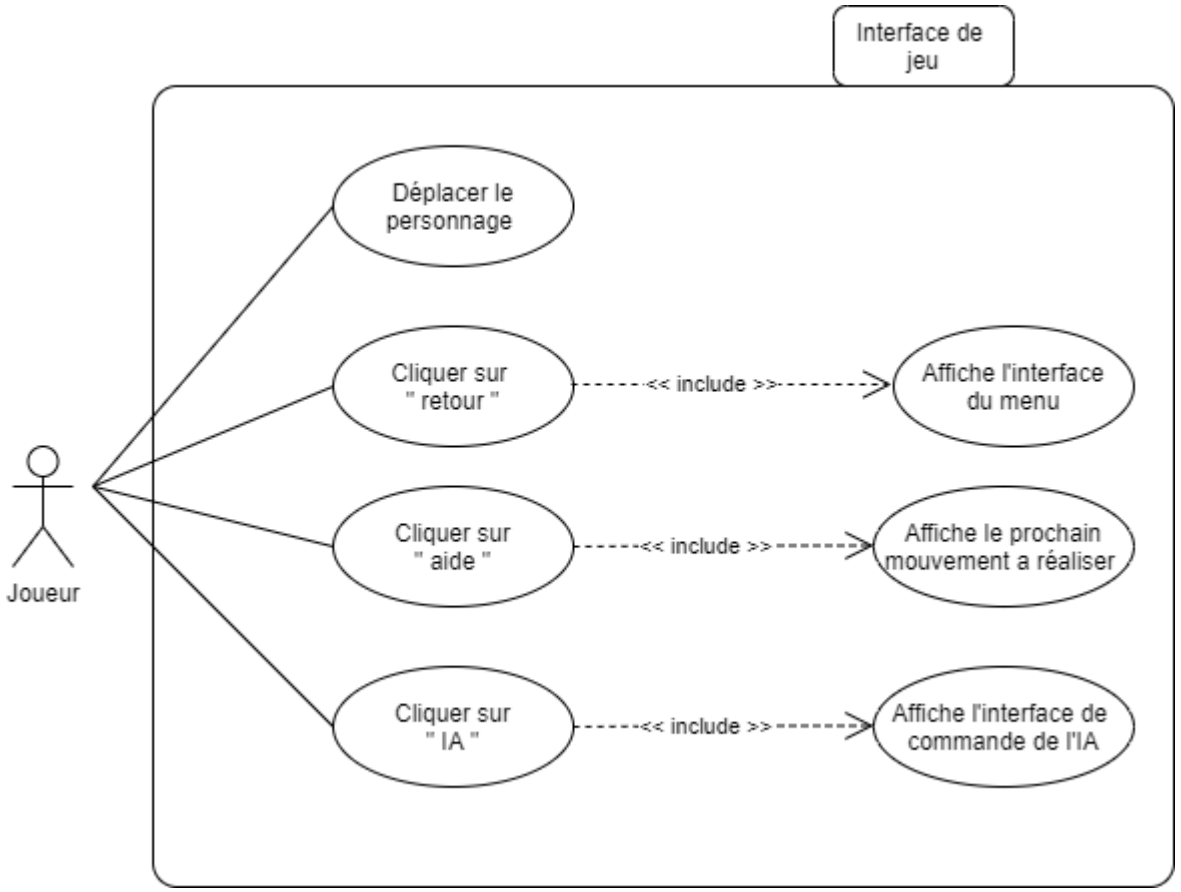
- L'utilisateur peut cliquer sur le bouton "Crédits" afin d'afficher la fenêtre des crédits avec le détail des créateurs du projet.
- Il peut cliquer sur le bouton "Jouer", afin d'afficher la fenêtre de sélection de niveau pour jouer.

Diagramme de cas d'utilisation de la fenêtre de choix de niveau



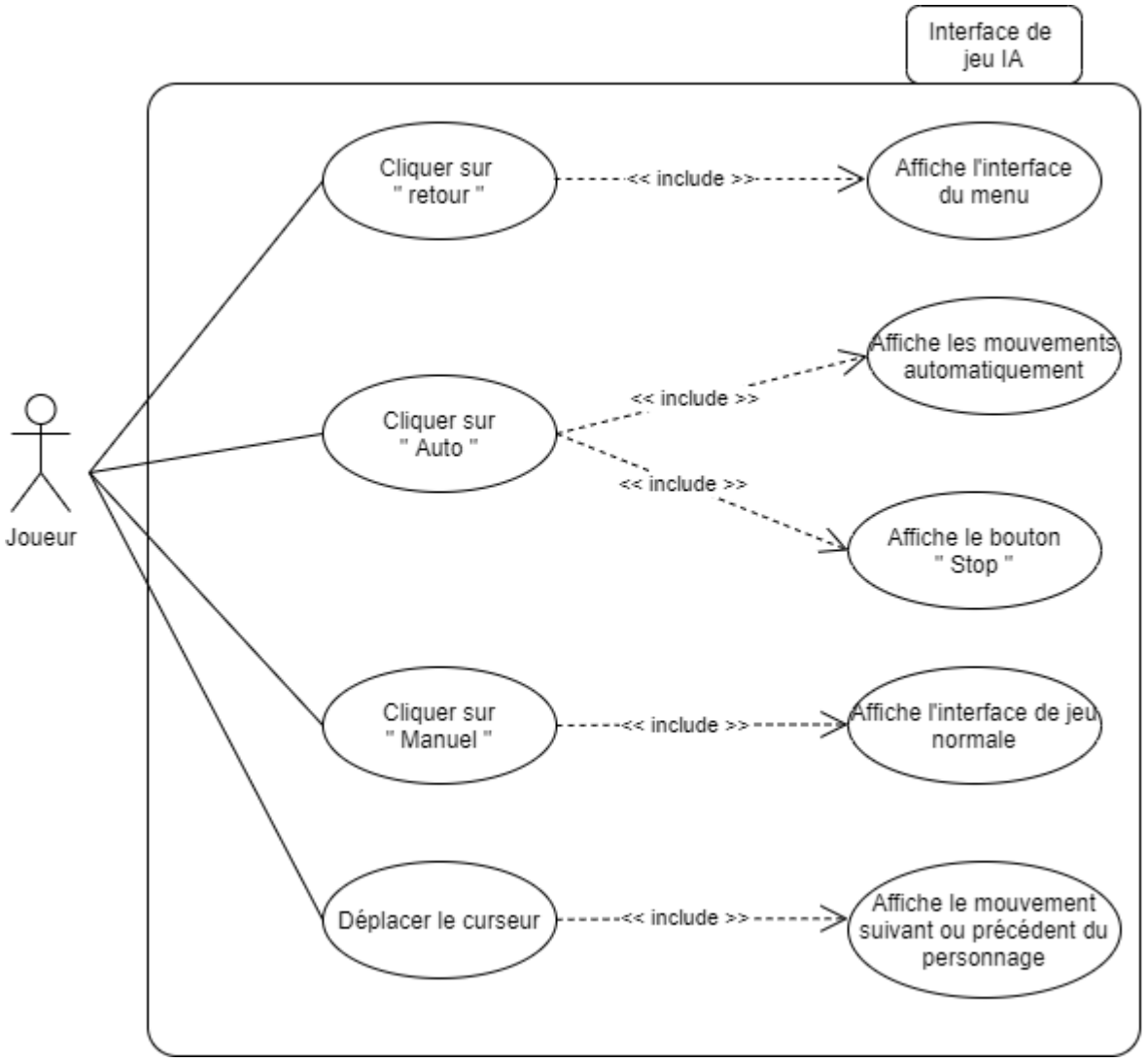
- L'utilisateur peut fermer la fenêtre du jeu à tout moment.
- Il peut cliquer sur "Retour" afin de retourner en arrière, c'est-à-dire au menu principal.
- Il peut choisir son niveau et peut également choisir la difficulté de celui-ci.
- Il peut cliquer sur jouer afin d'afficher la fenêtre de jeu avec le niveau qu'il a choisi.
- Les fenêtres sont affichées par l'application lors de l'appui sur le bouton.

Diagramme de cas d'utilisation de la fenêtre de jeu (mode sans IA)



- ❏ L'utilisateur peut déplacer le personnage avec les touches du clavier.
- ❏ Il peut également cliquer sur "Retour" pour retourner à la page du menu principal.
- ❏ Il peut cliquer sur "Aide" pour afficher le prochain mouvement à réaliser (calculé par l'IA)
- ❏ Il peut cliquer sur "IA" afin d'afficher l'interface de commande de l'IA sur le bas de l'écran.

Diagramme de cas d'utilisation de la fenêtre de jeu (mode avec IA)



- ❏ L'utilisateur peut cliquer sur "Retour" pour retourner au menu principal.
- ❏ Il peut également cliquer sur ''Auto'' afin d'afficher automatiquement les mouvements que l'IA a préalablement calculés et le bouton devient alors un bouton ''Stop'' permettant d'arrêter les déplacements de l'IA.
- ❏ Il peut cliquer sur "Manuel" pour retourner à l'interface de jeu sans l'IA.
- ❏ Il peut, pour finir, déplacer le curseur de gauche à droite pour afficher le mouvement suivant ou précédent de la solution trouvée par l'IA.

2) Recensement et évaluation des risques

Risques relationnels

- Problèmes liés à la situation sanitaire actuelle. Risque Moyen
- Problèmes liés à la communication entre les membres du groupe. Risque Moyen
- Problèmes liés à une mauvaise compréhension de la demande. Risque Moyen

Risques techniques

- Problèmes liés à la conception de base du jeu. Risque Moyen
- Problèmes lors de la création du jeu de base. Risque Faible
- Problèmes lors de la création de l'IA. Risque Élevé
- Problèmes lors de l'intégration de l'IA dans le jeu. Risque Élevé
- Problèmes lors de la création de la fonction heuristique. Risque Moyen

B - Découpage fonctionnel du projet

Nous avons travaillé tout au long de notre projet en méthode Agile (SCRUM). Cette méthode consiste à diviser le projet en plusieurs parties (itérations). Chaque itération possède un ou plusieurs objectifs à atteindre et une version fonctionnelle du produit doit pouvoir être livrée au client à la fin de chaque itération.

1) Itération 1

Au cours de l'itération 1, nous avons réalisé une première version du jeu. Dans cette version, les fonctionnalités proposées sont les suivantes :

- Affichage d'une interface graphique.
- Choix entre plusieurs niveaux différents.
- Lancer le jeu en mode joueur.
- Affichage du niveau avec un joueur, des murs, des caisses et des emplacements.
- Affichage d'un compteur de déplacements.
- Possibilité de déplacer le joueur qui peut déplacer les caisses.
- Affichage d'une fenêtre de victoire lorsqu'on a fini le niveau, avec la possibilité de soit rejouer le niveau, soit de quitter le jeu.

La partie graphique à été mise en place en parallèle de la partie fonctionnelle pure du jeu. En ce qui concerne la partie I.H.M. (Interface Homme-Machine), nous avons choisi de mettre en place le patron de conception M.V.C. (Modèle Vue Contrôleur).

Nous avons travaillé en équipe lors de cette itération.

2) Itération 2

Au cours de l'itération 2, nous avons étudié l'algorithme A* afin de le comprendre. Nous avons ensuite réalisé un algorithme simplifié A* en partant d'un algorithme que nous avons trouvé sur wikipedia et en l'adaptant au jeu Sokoban.

Grâce à cet algorithme, le binôme Anthony / Quentin a pu réaliser un diagramme de classe incluant une partie propre à l'IA au reste du projet. L'implémentation de l'IA fut la partie la plus longue et complexe du projet, mais au final, nous avons réussi à avoir une IA fonctionnelle.

Pour ce faire, nous avons dû modifier notre conception préalable pour des soucis d'optimisation de la rapidité de recherche ce qui nous a pris un peu de temps.

En parallèle, le binôme Benoît / Lucas s'est occupé de l'amélioration de l'ergonomie de l'application en prenant en compte les ajouts liés à l'IA.

3) Itération 3

Au cours de l'itération 3, étant donné sa courte durée, nous avons décidé de travailler en équipe.

Nous nous sommes donc employés à implanter une fonction heuristique permettant de diminuer le nombre d'état testés et donc de diminuer le temps de recherche.

Nous avons aussi tenté de diminuer le nombre d'état en classant la liste de positions des caisses afin qu'elles soient interchangeables, mais le tri de cette liste était au final plus long que le temps de test des états supprimés.

Nous avons par ailleurs peaufiné l'ergonomie de l'application et ajouté certaines fonctionnalités à l'interface graphique.

Nous avons notamment ajouté une fonctionnalité qui permet l'affichage du nombre de nœuds constituant la liste ouverte et la liste fermée. Nous avons aussi ajouté un bouton qui permet de stopper la recherche de la solution par l'IA (avec une méthode liée au Threads Java).

Au cours du temps restant, nous avons travaillé sur le rendu du projet et la soutenance.

C - Evolution par rapport à l'étude préalable de décembre

Description des diagrammes de classes

Le premier diagramme de classe, réalisé lors de la phase de conception initiale, pose les bases de notre projet. Nous avons mis en évidence les principales classes du jeu en mettant de côté les patrons de conceptions et la partie IA. **(Cf. diagramme de classes de l'étude préalable)**

A la fin de l'itération 1, nous avions donc un jeu et une interface graphique suivant le fonctionnement du patron MVC. Nous avions également mis en place le patron stratégie pour les différents types d'utilisateurs de l'application. Cependant, ces classes ont été supprimées par la suite car nous avons trouvé une méthode plus efficace pour réaliser cette distinction. **(Cf. diagramme de classes à la fin de l'itération 1)**

Lors de l'itération 2, nous avons implémenté le package IA à l'interface graphique et au jeu de base afin de permettre son utilisation dans le jeu. La classe IA constitue le centre de ce package. Elle permet la recherche de solution en utilisant les autres classes du package. **(Cf. diagramme de classe de l'itération 2 avec ajout de l'IA)**

Le lancement de l'IA posait problème car il s'opérait sur le même "Thread" Java que l'interface graphique, ce qui avait pour conséquence de bloquer la fenêtre active. C'est pourquoi nous avons séparé l'IA et l'interface graphique dans deux "Thread" Java différents lors de l'itération 3 afin de pouvoir réaliser cette recherche sans bloquer la bloquer l'application. Nous avons également implémenté l'heuristique dans la classe noeud. **(Cf. diagramme 3)**

Nous avons donc sensiblement gardé la même conception depuis l'itération 1 avec un diagramme se développant avec l'ajout d'un certain nombre de méthodes et quelques modifications de la conception qui n'impactent pas la conception globale.

Diagramme de classes de l'étude préalable

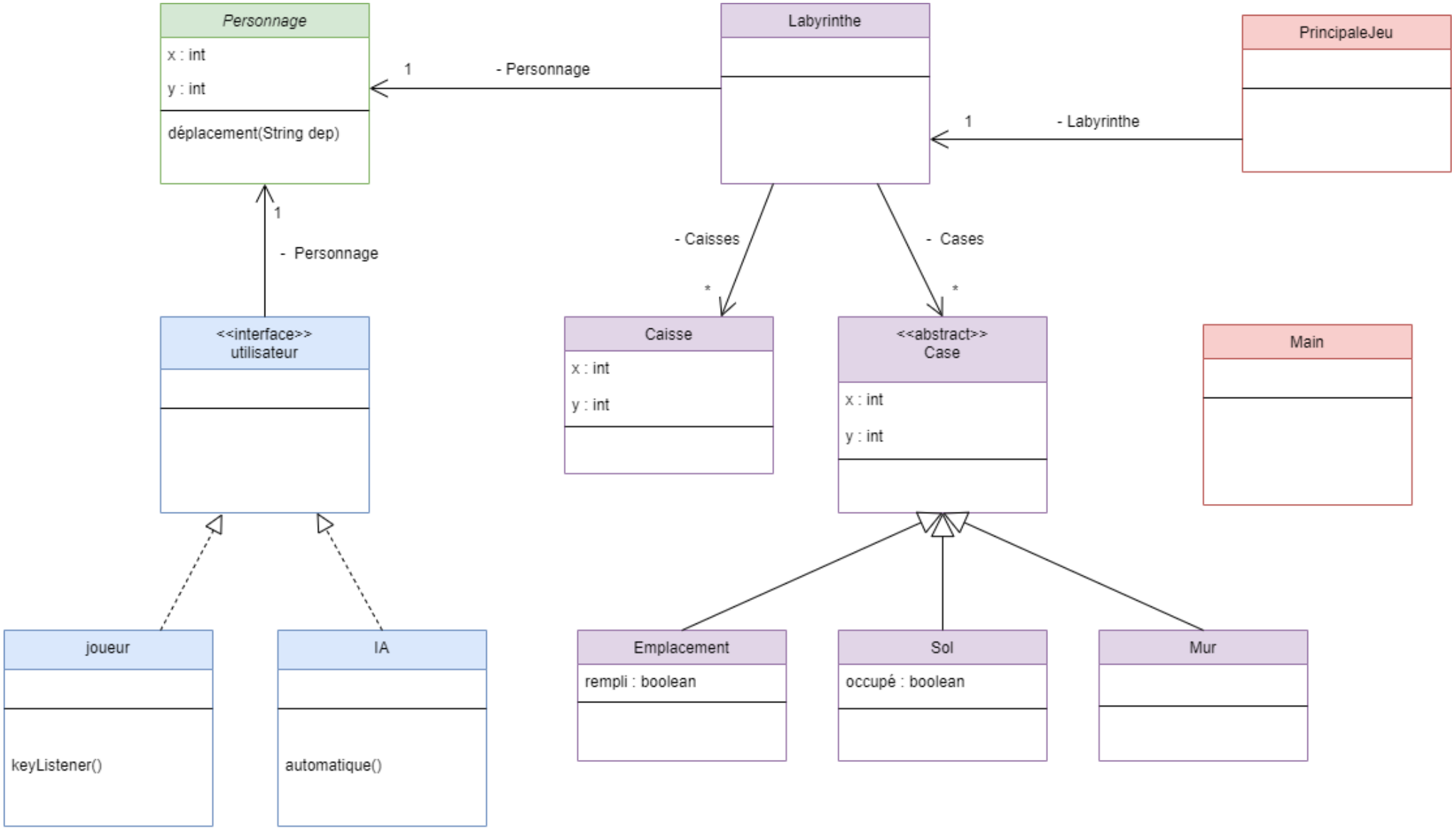


Diagramme de classes à la fin de l'itération 1

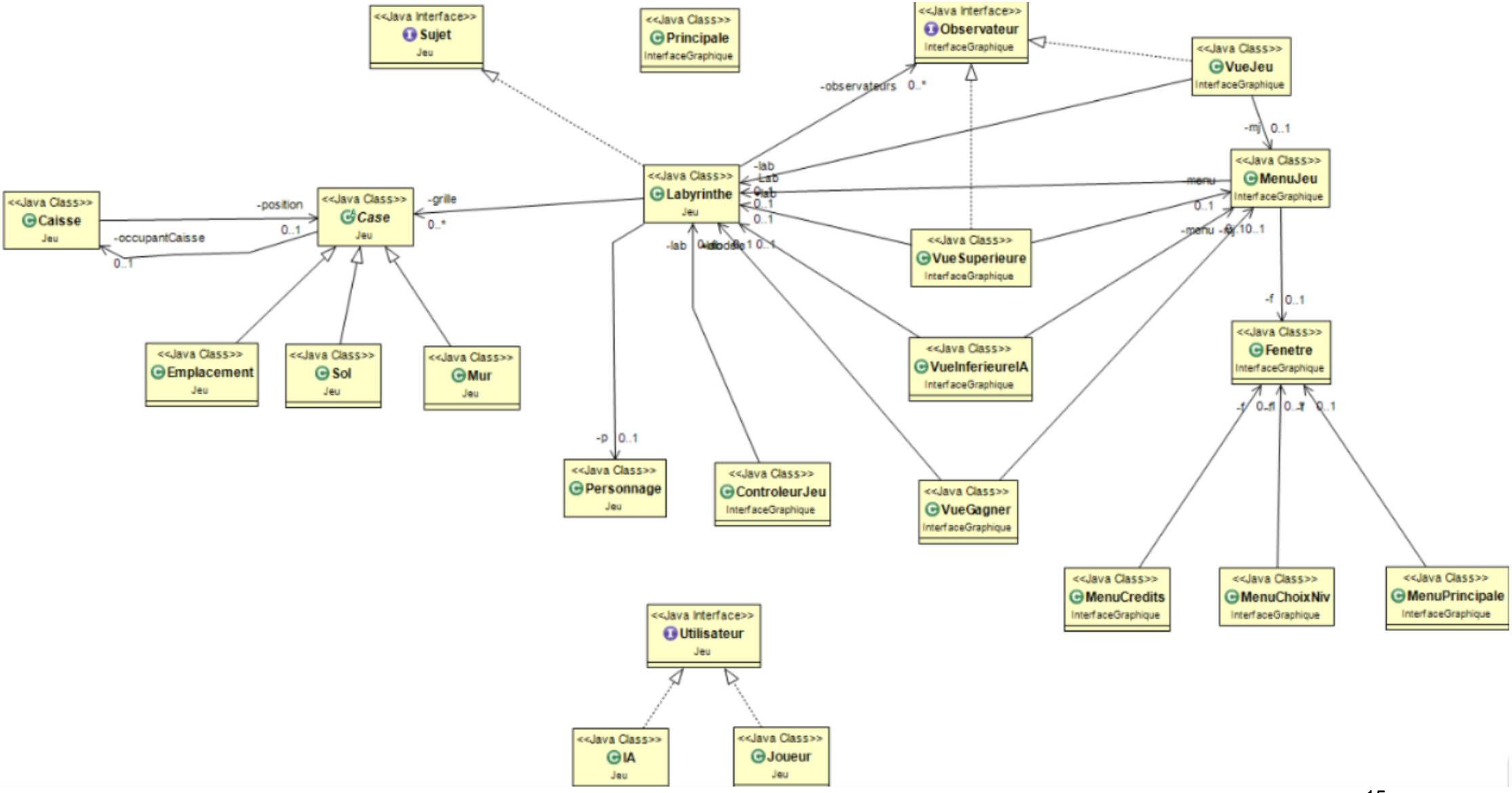


Diagramme de classes de l'itération 2 avec ajout de l'IA

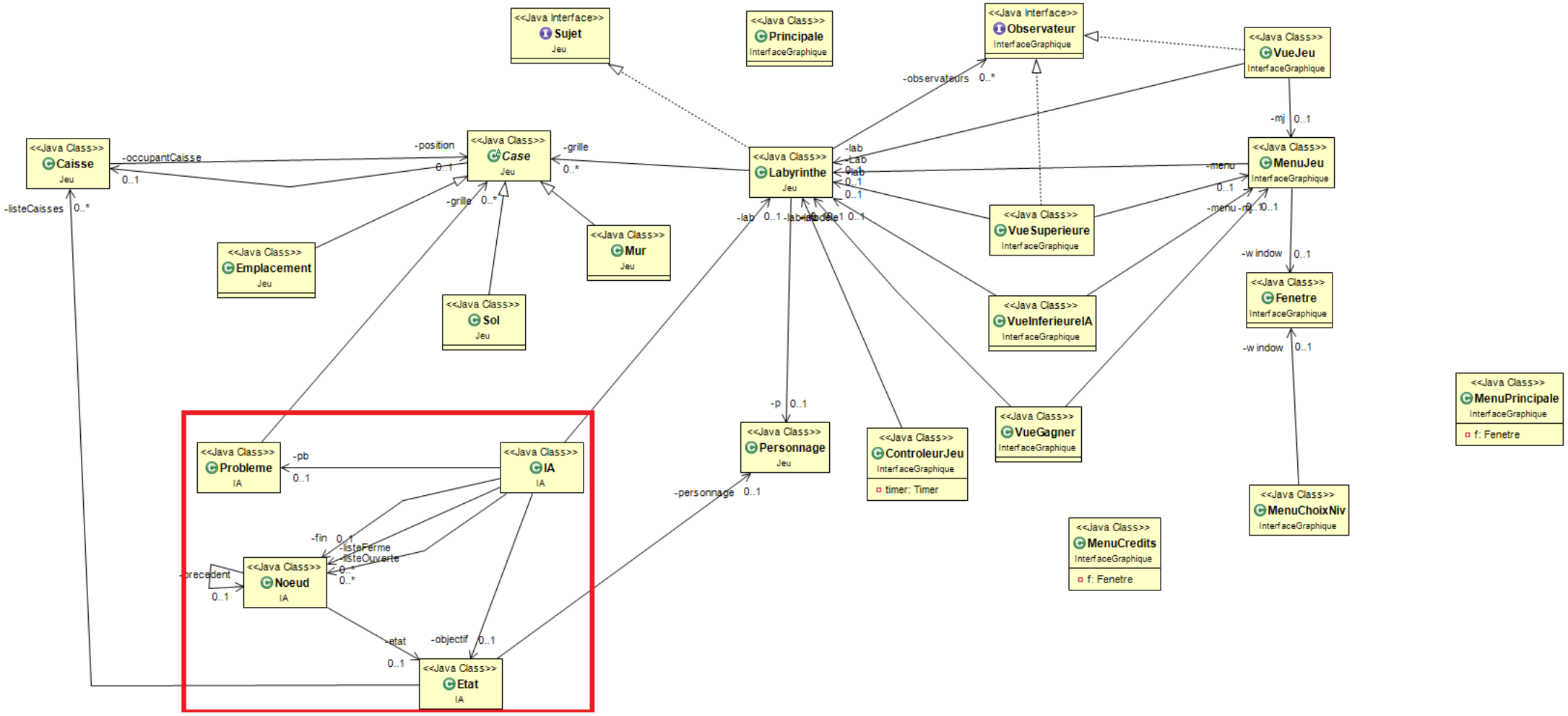
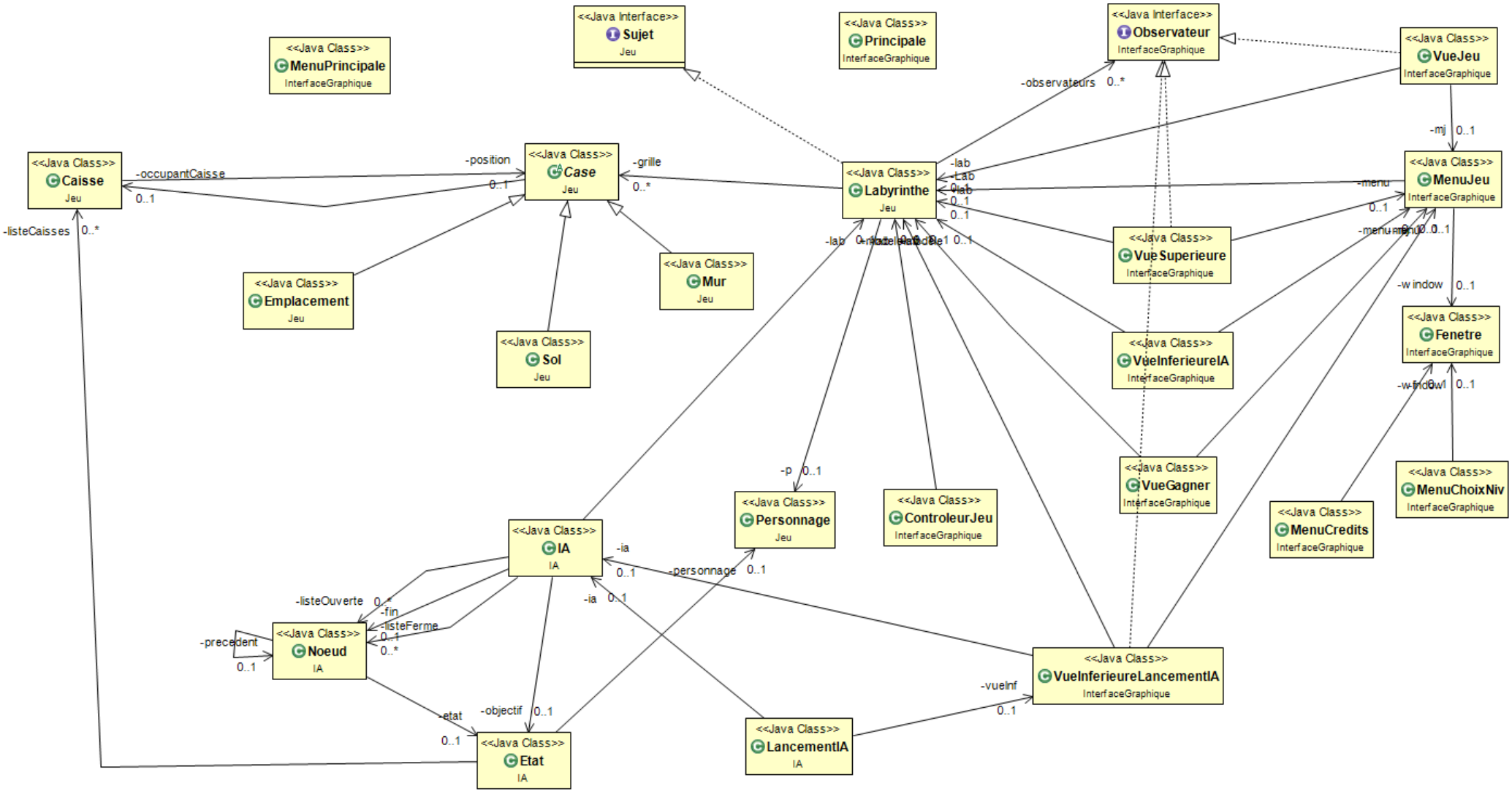


Diagramme de classe de l'itération 3



III - Réalisation

A - Fonctionnalitées réalisées

Voici la liste des fonctionnalités que nous avons réalisées concernant le jeu Sokoban de base ainsi que l'interface graphique :

Fonctionnement du jeu :

- Lancer le jeu en mode joueur.
- Création des différents éléments du jeu (personnage, caisses, emplacements, murs)
- Déplacer le personnage avec les touches du clavier
- Déplacer les caisses

Différents menus :

- Création et implémentation du menu principal
- Création et implémentation du menu d'explication du jeu
- Création et implémentation du menu de choix de niveau
- Création et implémentation du menu de jeu
- Création et implémentation de l'interface de victoire

Interface graphique du jeu :

- Affichage du labyrinthe avec le personnage et les caisses
- Menu en rapport avec le jeu : affichage du nombre de mouvements du personnages, le niveau et la difficultée
- Ajout de boutons permettant de recommencer le niveau en cours ou de revenir au menu principal à partir du jeu.

Nous avons ensuite ajouté au jeu une IA permettant de résoudre un niveau du jeu Sokoban, ce qui constitue la seconde phase du projet.

La seconde grande fonctionnalité est donc la création d'une IA pour résoudre un niveau du jeu sokoban.

Nous avons donc mis en place un menu dédié à l'IA qui permet de lancer la recherche de solutions et de pouvoir gérer l'affichage de la solution trouvée ainsi que quelques fonctionnalités supplémentaires :

- Création et implémentation d'une interface permettant de lancer l'IA.
- Création et implémentation d'une interface permettant de voir l'avancement de l'IA dans la recherche de solution.
- Création et implémentation de boutons permettant de parcourir la solution trouvée par l'IA, déplacement par déplacement, en avant et en arrière, et la visualiser sur l'interface graphique du jeu.
- La possibilité de quitter l'IA quand on le veut pour continuer en mode joueur.
- Possibilité de stopper la recherche de solution par l'IA à tout moment.

B - Difficultés rencontrées

Tout au long de ce projet, nous avons rencontré certaines difficultés, plus ou moins faciles à surmonter mais qui nous ont permis d'aboutir au résultat que nous avons aujourd'hui.

Tout d'abord, nous avons dû revoir plusieurs fois la conception du jeu et notamment les diagrammes de cas d'utilisations. En effet, nous avons eu la possibilité de nous entretenir plusieurs fois avec notre tuteur afin qu'il valide ou non notre conception, ce qui nous a permis d'affiner nos diagrammes UML.

Nous avons également eu quelques difficultés lors de la conception en Java du jeu de base car il fallait réussir à créer ce jeu tout en implémentant l'interface graphique et les données du jeu avec le patron de conception M.V.C..

Jusqu'ici, toutes les difficultés rencontrées étaient plus ou moins surmontables et ne nous ont pas vraiment ralenti. Les plus grandes difficultés sont apparues lors de la création et surtout de l'implémentation de l'IA dans le code du jeu.

Lors de l'implémentation de l'IA, nous avons eu un certain nombre de problèmes pour la lier avec l'interface graphique et également des soucis de performances de l'IA ce qui bloquait la résolution de niveaux.

Malgré les nombreuses difficultés rencontrées, nous sommes parvenus à les surmonter et nous n'avons abandonné aucun ajout de fonctionnalité dans le jeu.

IV - Conclusion

Ce projet nous a beaucoup apporté en termes de compétences de travail d'équipe.

Nous avons pu nous rendre compte de l'importance d'une bonne organisation, mais également de l'utilité de la méthode Agile (SCRUM) qui permet de toujours recentrer l'orientation du projet vers l'objectif à atteindre. Cette méthode nous a par exemple permis à certains moments de changer une conception qui ne convenait pas au projet.

Malgré les difficultés liées à la situation sanitaire, nous avons réussi à nous réunir pour faire le point aussi souvent que cela était nécessaire et notre tuteur, Monsieur Vincent THOMAS, nous a aidé à faire avancer ce sujet exploratoire.

Nous avons donc réussi à mettre en place les fonctionnalités qui nous ont permis d'atteindre les objectifs attendus.

Cependant, nous aurions voulu pouvoir continuer la phase de création et d'implémentation de l'IA afin d'améliorer encore son efficacité et également développer davantage l'interface graphique dans le cadre de la phase de vulgarisation de l'IA.

Pour ce qui est de la reprise de notre projet par d'autres étudiants, après réflexion, nous pensons que celui-ci ne peut pas être repris dans le cadre d'un nouveau projet tutoré, non pas parce qu'il est terminé mais parce qu'améliorer l'algorithme de l'IA requiert un niveau Master.

Comme alternative, nous proposons un sujet similaire au nôtre avec un autre type de jeu auquel on peut implémenter une IA ou encore reprendre ce sujet en utilisant d'autres algorithmes de recherche afin de modifier l'IA et pouvoir comparer les deux.