

TELECOM NANCY - PREMIÈRE ANNÉE

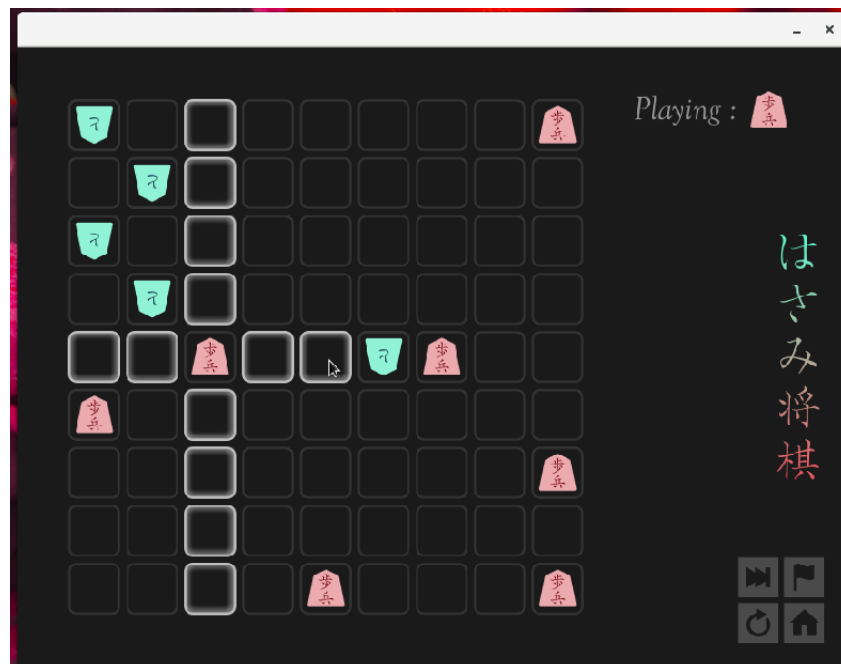
MODULE CSH : INITIATION AU C ET AU SHELL

Rapport du Projet de C

HASAMI SHOGI

ASSELIN-BOULLÉ CHARLOTTE, CALVET JEAN ET DENAUW PIERRE

À rendre le Mercredi 31 Mai 2017



Capture 0 : Interface de notre jeu en cours de partie.

Table des matières

1	Introduction	2
2	État de l'art	3
2.1	Présentation du jeu et de son environnement	3
2.1.1	Environnement de jeu	3
2.1.2	Déroulement de la partie	3
2.2	Règles et but du jeu	3
2.2.1	But du jeu : conditions de victoire ou de défaite	3
2.2.2	Règles du jeu	4
2.3	Recherches sur l'intelligence artificielle	5
3	Notre jeu	6
3.1	Modalité pour y joueur : version finale	6
3.1.1	Lancer le jeu	6
3.1.2	Règles que nous avons choisies	6
3.1.3	Y jouer : manuel de l'utilisateur	6
3.2	Démarche	8
3.2.1	Deux joueurs	8
3.2.2	Un joueur : intelligence artificielle	8
3.2.3	L'interface utilisateur	9
3.3	Idées d'amélioration à apporter	10
4	Gestion de projet	11
4.1	Fiche de définition	11
4.2	Répartition du travail	12
4.3	Gestion du temps	12
4.4	Difficultés rencontrées	13
5	Conclusion	14
6	Sitographie	15
7	Annexes	16
7.1	Compte-rendus de réunion	16

1 Introduction

Dans le cadre de notre projet de C nous devons réaliser un jeu de plateau faisant intervenir une stratégie ludique. Il s'agissait de jeux de stratégie combinatoire abstrait de caractéristiques identiques : 2 joueurs jouant à tour de rôle, chaque joueur connaissant l'état du plateau à chaque instant et ne faisant pas intervenir le hasard. Dans notre cas il s'agissait de programmer le jeu Hasami Shogi, ce jeu ayant été présenté comme un jeu aux règles simples mais à l'aspect stratégique complexe.

Il ne s'agissait pas seulement de programmer un jeu. Il s'agissait aussi et surtout de réfléchir à la modélisation que nous allions utiliser pour le faire aux structures adéquates. Il nous fallait déterminer nos ambitions : nous avions pour mission de proposer un mode "joueur seul" qui opposerait un joueur à une intelligence artificielle. C'était donc à nous de la penser et de faire la part des choses entre une intelligence réaliste, qui serait très développée, et une intelligence simple qui ne ferait que de l'aléatoire. Enfin, nous avions aussi pour mission de fournir une interface graphique fonctionnelle.

Ce projet avait pour vocation première de mettre en application nos connaissances dans le module et de les approfondir. Mais il s'agissait aussi de nous faire réfléchir par nous même à des solutions plus élaborées que celles dont nous avons l'habitude. De plus il nous a permis de nous familiariser d'avantage avec le travail sous forme de projet le tout en équipe.

Ce rapport a pour but d'expliquer succinctement notre démarche pour parvenir à l'avancement final de notre projet. Une première étape de renseignement puis de réflexions et prise de conscience de ce qu'il y a à faire, suivie d'une seconde étape d'implémentation en différentes phases. Le rapport comporte aussi l'aspect gestion de projet nécessaire à l'élaboration de ce jeu, notre organisation et les difficultés que nous avons rencontrées.

2 État de l'art

Avant de commencer tout travail de programmation il s'agissait de se renseigner sur ce que nous avions à faire et donc d'étudier le jeu de l'Hasami Shogi, aussi bien d'un point de vue théorique que pratique. Nous avons donc commencé par faire des recherches sur le jeu en lui-même sur différents sites internet (*Cf. Sitographie*). Ensuite nous avons commencé à vouloir jouer pour mieux appréhender les règles mais aussi la réflexion qu'il pouvait y avoir derrière (dans le but, plus tard, d'avoir à réfléchir sur l'intelligence artificielle). Nous avons donc téléchargé divers applications de jeu pour jouer individuellement, mais aussi emprunté un plateau avec les pions adéquates pour jouer entre nous et échanger sur les stratégies à adopter.

2.1 Présentation du jeu et de son environnement

L'Hasami Shogi est un jeu de plateau stratégique originaire du Japon. Il oppose deux joueurs ayant chacun des pions différents. Il a la particularité d'avoir deux stratégies de jeux possibles : par avancement ou par affrontement.

Il existe différents types d'Hasami Shogi avec différentes règles possibles, nous allons brièvement les exposer.

2.1.1 Environnement de jeu

Le jeu nécessite un plateau de 9x9 cases. Il existe bien entendu des variantes avec des plateaux plus grands (ou plus petits par exemple sur un échiquier). Chaque joueur possède 9 ou 18 pions (soit une ou deux rangées).

La partie commence avec les pions de chacun des joueurs de part et d'autres du plateau, alignés dans la zone de départ.

2.1.2 Déroulement de la partie

Le joueur qui commence est généralement prédéfini : c'est souvent le joueur ayant les pions blancs, ou celui en bas du plateau.

Les joueurs jouent à tour de rôle.

A chaque tour le joueur peut passer son tour ou déplacer un pion. A l'issue du déplacement ce dernier peut se retrouver dans une situation où il "mange" un (voire plusieurs) pion(s) adverse(s). Cf. paragraphe 2.2.2 concernant les règles de ces possibilités.

La partie se finit si l'un des joueurs quitte/abandonne, si l'un des joueurs gagne ou perd et si il y a match nul.

2.2 Règles et but du jeu

2.2.1 But du jeu : conditions de victoire ou de défaite

Le but du jeu est de gagner ou mettre en situation d'échec l'adversaire. Ces deux buts décrivent les deux stratégies possibles à adopter.

- La stratégie d'avancement, offensive. Le but est ici d'aligner consécutivement 5 de nos pions, que ce soit verticalement, horizontalement ou en diagonale. Il existe toutefois une condition : les pions étant dans la zone initiale (ie la première rangée de notre côté) ne sont pas considérés. Il est à noter que cette possibilité de stratégie n'est pas toujours possible en effet elle n'est

considéré que dans le cas où les joueurs ont initialement 18 pièces. Il s'agit là de la variante la plus répandue : le Dai Hasami Shogi.

- La stratégie d'affrontement, défensive. Il ne s'agit pas de gagner à proprement parler mais plutôt d'empêcher l'adversaire de gagner et ce en lui capturant suffisamment de pions pour qu'il ne puisse plus prétendre à la victoire. Dans le cas où l'on considère l'alignement possible il faut avoir mangé suffisamment de pions pour qu'il lui en reste moins de 5 (donc en manger au moins 14). Dans le cas où la seule victoire possible est par défaite de l'adversaire, il ne doit resté qu'un seul pion à ce dernier.

Il existe souvent une condition de match nul : lorsqu'un même schéma de déplacement se produit trois fois consécutivement.

2.2.2 Règles du jeu

Certaines actions sont soumises à des règles :

- le déplacement : il s'agit de celui de la Tour aux échecs, il s'effectue en ligne ou en colonne, d'autant de cases que le joueur le souhaite sous condition qu'il ne rencontre pas d'obstacle.
- le saut : il est parfois possible de passer par dessus un pion (adverse et/ou à soit suivant les versions). En effet lorsque deux jetons sont juxtaposés il est possible de passer au dessus sous réserve que la case de l'autre côté soit libre, bien sûr. En revanche aucun mouvement avant ou après le saut n'est possible.
- manger un pion : il est possible (et souvent nécessaire) de "manger" un (ou plusieurs) pion(s) adverse(s). Pour ce faire il faut qu'il(s) se retrouve(nt) encadré(s) verticalement ou horizontalement par deux de vos pions. On appelle ça une configuration "en sandwich" (c'est d'ailleurs la traduction du mot japonais "Hasami"). Il y a donc deux cas de figure : horizontal ou vertical (*Cf. figure ci-dessous*)



Les deux configurations en sandwich : le pion bleu est "mangé"

Selon les versions du jeu on a la possibilité de manger un pion dans un coin en le bloquant (ie en lui étant adjacent des deux côtés)

Parfois nous pouvons nous faire manger nous même si il y a une succession : une configuration du type A B A B donnerait A _ _ B

A noter que dans certaines versions pour manger un pion il faut le sauter. Là encore parfois il est possible d'en sauter plusieurs consécutivement, d'autre fois non. Néanmoins ces versions semblent moins pertinentes.

2.3 Recherches sur l'intelligence artificielle

Simuler une intelligence est un des enjeux phares de ces dernières années, tant dans le domaine du jeu que dans les domaines scientifiques, médicaux ou militaires. De nombreux algorithmes ont vu le jour et sont en constante évolution. Quelques exemples d'algorithme :

- Algorithme minimax/negamax avec élagage alpha bêta et table de transposition
- Réseau de neurones
- Algorithme de scoring (Hummingbird, Google)
- Depth search first
- Breadth First Search

Le but d'une intelligence artificielle est de simuler le comportement réel d'un humain face à une situation donnée, voire d'aller plus loin. Les algorithmes les plus performants à ce jour obtiennent des résultats impressionnants en terme de rapidité et de résultats. Ces progrès suivent également l'évolution des technologies informatiques et l'augmentation des puissances de calcul permettent de repousser les possibilités.

De l'étude des jeux découle un pan des mathématiques, la théorie des jeux. Ainsi, en considérant les jeux comme concept, en les catégorisant, en les classifiant, on en tire des théorèmes. Et ainsi on peut les appliquer de manière concrète à la résolution de plateau de jeu. Par exemple, l'algorithme que l'on utilisera découle du Théorème du minimax de von Neumann de 1928. Celui ci est applicable aux jeux à sommes nulles, c'est à dire la majorité des jeux de plateau à 2 joueurs sans composantes aléatoires. De l'étude mathématiques et conceptuelles, on peut ainsi trouver des règles de résolution qui simule l'intelligence par l'anticipation, comme le ferait un être humain.

Si le but premier était donc de simuler des comportements rationnels face à des situations de jeux, les applications sont aujourd'hui beaucoup plus diversifiées. Des voitures autonomes de TeslaTM aux recherches presque instantanées de GoogleTM sur un réseau de milliards de sites Internet en passant par les algorithmes d'affichages de la publicité en fonction de vos sites visités, l'intelligence artificielle est partout de nos jours, et de plus en plus dans notre quotidien avec "l'Internet des objets" et le "tout connecté".

Les dernières avancées dans ce domaine concernent le machine learning, c'est à dire la capacité d'une machine à apprendre des informations qu'elle reçoit. On passe alors d'un intelligence artificielle pour un jeu qui applique une stratégie simple à une machine qui apprend de ses adversaires, comprend les stratégies et finit par devenir un adversaire redoutable et imprévisible. Mais on peut trouver d'autres applications, comme les intelligences de GoogleTM qui reconnaissent les images ou même apprennent à reconnaître des dessins et les améliorer. Très récemment des articles de recherche ont même été publiés sur des ordinateurs de recherche destinés à apprendre sur l'apprentissage par le machine learning, en calculant les performances d'apprentissage à partir de variations de paramètres.

Notre choix c'est porté sur une variation de l'algorithme minimax, le Negamax, en y appliquant l'élagage alpha bêta. Nous l'expliquerons ci-dessous.

3 Notre jeu

Nous allons maintenant vous faire part de ce que nous avons implémenté et pourquoi nous avons fait certains choix.

3.1 Modalité pour y joueur : version finale

Pour jouer à notre jeu il est nécessaire de posséder la librairie SDL2 (et SDL2_image).

3.1.1 Lancer le jeu

Il suffit ensuite de compiler en tapant "make" dans le terminal (attention il est possible que vous deviez nettoyer le dossier en faisant un "make clean") puis de lancer le script play comme illustré sur la *Capture 1*. Ensuite il n'y a plus qu'à se laisser porter par l'interface utilisateur.

```
[charlotte@plume ProjectC]$ make
gcc -Wall -c c-src/main.c -o c-out/main.o
gcc -Wall -c c-src/board.c -o c-out/board.o
gcc -Wall -c c-src/gameNoUI.c -o c-out/gameNoUI.o
gcc -Wall -c c-src/gameUI.c -o c-out/gameUI.o
gcc -Wall -c c-src/find.c -o c-out/find.o
gcc -Wall -o c-bin/play c-out/* -lSDL2 -lSDL2_image
[charlotte@plume ProjectC]$ sh play.sh
```

Capture 1 : les commandes pour lancer le jeu dans un terminal

3.1.2 Règles que nous avons choisies

Le joueur qui commence la partie est le joueur au pion rouge (celui du bas). Nous avons choisis la règle de "manger" en "sandwich" car elle nous paraissait plus intéressante. Ainsi pour qu'un joueur puisse manger un pion il faut que ce pion adverse se retrouve entre deux pions de ce joueur (sur la même ligne ou la même colonne) avec adjacence. Nous avons gardé la possibilité d'en manger plusieurs d'un coup.

De plus notre pion n'est pas mangé si c'est nous qui le déplaçons entre deux pions du joueur adverse. (la configuration A B A B donnera A _ A B si c'est le joueur A qui a amené le dernier pion, A B _ B si c'est le joueur B).

Par ailleurs, comme nous l'expliquerons plus bas dans le rapport, nous avons décidé de prendre la version simple du jeu et non le Dai, ainsi aligner les pions ne changeant pas l'issue de la partie permet pas de gagner.

3.1.3 Y jouer : manuel de l'utilisateur

A l'ouverture du menu principal, l'utilisateur doit choisir ce qu'il souhaite faire (*Cf. Capture 2*) :

- lancer une partie à deux joueurs en cliquant sur "Jouer à 2"
- lancer une partie en solo (en jouant donc contre l'intelligence artificielle) en cliquant sur le niveau de difficulté qu'il souhaite : facile, medium ou difficile (suivant le niveau l'intelligence parcourt plus de coups et s'assure d'avoir plus de chance de gagner)
- accéder à la fenêtre des règles de jeu (et aides pour savoir comment jouer)
- quitter (croix ou bouton correspondant)



Capture 2 : la fenêtre menu avec les différentes options

Lors de la partie le joueur (qui a la main) a différentes options qui s'offrent à lui (*Cf. Capture 3*) :

- cliquer sur un bouton d'option :
 - celle de passer son tour : flèche d'accélération
 - celle d'abandonner : le drapeau
 - celle de recommencer la partie (de même type) : la flèche qui boucle sur elle-même
 - celle de retourner au menu principal : la maison
- jouer son tour :
 1. clic gauche sur un pion pour le sélectionner (les cases où le pion peut se déplacer s'illumine)
 2. clic droit pour relâcher le pion, puis clic gauche pour en sélectionner un nouveau
 3. clic gauche sur une case illuminée pour déplacer le dit pion

Lorsqu'il a déplacé son pion, si la position d'arrivée lui permet de manger un (ou plusieurs) pion(s) adverse(s), le(s) pion(s) disparaî(ssen)t. Si cette disparition entraîne le fait que l'adversaire n'ait plus qu'1 ou 0 pion, alors il est dérouteré et le joueur qui vient de jouer est déclaré gagnant.

3.2 Démarche

3.2.1 Deux joueurs

Initialement nous avons choisit de programmer un jeu de Dai Hasami Shogi : un plateau de 9x9 cases, 18 pions de part et d'autres et la possibilité de gagner en alignant 5 pions.

Il a d'abord fallut créer des structures pour les pions et pour le plateau.

Ensuite il s'agissait de faire une boucle qui alterne entre les joueurs et affiche par un printf un tableau entre chaque tour et ce tant qu'ils n'ont pas quitté et gagné ou perdu.

Pour chaque tour nous avons un printf scanf qui demandait ce que le joueur voulait faire, sachant qu'il fallait vérifier s'il ne se trompait pas de pion (en ne prenant pas un de ses pions) et que le déplacement était possible. Il fallait ensuite vérifier que nous n'étions pas dans une situation où un pion était mangé (et si oui s'il en restait suffisamment à l'adversaire pour que la partie continue) et si 5 pions du joueurs étaient alignés.

Lorsqu'il a fallut rebasculé à la version classique de l'Hasami Shogi (qui simplifie le travail de valuation pour l'élaboration de l'intelligence artificielle) il a fallut changer une bonne partie du code puisque ça impactait les structures et les boucles : nous avons moins de pions et plus la possibilité de gagner par alignement.

3.2.2 Un joueur : intelligence artificielle

Lors de la recherche pour implémenter l'intelligence artificielle nous nous sommes rendu compte qu'il était nécessaire de revenir sur nos choix : faire une intelligence artificielle avec le jeu de Dai complexifiait grandement la mise en place d'une intelligence artificielle basique comme celle que nous avons décidé d'utiliser.

En effet, ce genre d'intelligence artificielle repose sur l'exploration des mouvements futurs, et de l'évaluation de la "valeur d'un mouvement". Or, avec 2 conditions de victoires, et donc 2 conditions de défaite, l'évaluation devient plus complexe, et l'optimisation encore plus. C'est pourquoi nous avons décidé de revoir notre jeu pour changer les règles.

Algorithmiquement, le plus dur n'était pas la compréhension de la démarche, que ce soit la forme simple de l'algorithme, la version Negamax ou l'élégage alpha-beta. Ce qui a été gênant était l'utilisation des bonnes structures et représentations mémoire. L'utilisation des pointeurs est également quelque chose de très nouveau. Mais avec de l'entre-aide et de l'acharnement, le fonctionnement a été compris et nous avons pu arriver à un résultat convenable.

Comme évoqué ci-dessus, le principe algorithmique que nous avons choisi est le negamax avec élégage alpha bêta. C'est une variante du minimax. Cet algorithmique est une évaluation d'un mouvement par les conséquences qu'il peut engendrer. Pour chaque mouvement possible, on parcourt l'ensemble des mouvements possibles après, et l'on répète cette opération un certain nombre de fois. Plus on anticipe loin, plus l'intelligence est performante. L'évaluation repose sur un principe simple : chaque joueur cherche à maximiser ses gains : ainsi, comme on se met à la place de l'ordinateur, celui-ci prend la meilleure des décisions, sachant que son adversaire, le joueur, prendra lui aussi la meilleure des décisions de son point de vue.

Partant de ce principe, il faut ensuite une fonction lorsque dans la recherche on atteint une feuille, c'est à dire si la partie est finit ou si on atteint la profondeur maximum choisit selon la difficulté. Dans nos règles, un joueur prend l'avantage s'il prend les pièces adverses. Ainsi, l'évaluation basique consiste à faire la différence entre le nombre de pièces adverses et les siennes. Pour les fins de partie, on choisi de mettre un nombre de point pour une victoire, qui est supérieur à ce qu'on peut espérer au maximum pour tout autre plateau, et on prend l'opposé en cas de défaite.

Ainsi, arrivé aux feuilles, on obtient les évaluations des plateaux, et on remonte l'arbre des appels récurifs en prenant alternativement le minimum ou le maximum de ses fils.

Le negamax est une simplification plus sémantique que pour les performances. En effet, cela part du principe que $\max(A,B) = -\min(-A,-B)$. Ainsi, on retire la disjonction de cas autour du joueur dont c'est le tour, et on se contente d'une seule boucle.

C'est l'élagage alpha bêta qui va améliorer les performances de notre intelligence artificielle. Cela repose sur le fait que lors de la remontée de l'arbre, il est inutile de continuer à explorer des branches de l'arbre qui ne vont aboutir à rien de toute façon. Ainsi, dans l'appel récursif, on ajoute des bornes. Celles-ci sont initialisées à $\alpha = -\infty$ et $\beta = \infty$. Au fur et à mesure, lorsque l'on parcourt les branches, on change les bornes afin de les rétrécir, et de ne pas parcourir des noeuds inutiles.

D'autres améliorations sont possibles, avec des tables de transpositions pour la mémorisation par exemple. Ainsi, avec une table, on peut mémoriser certains noeuds qui sont parcourus plusieurs fois.

3.2.3 L'interface utilisateur

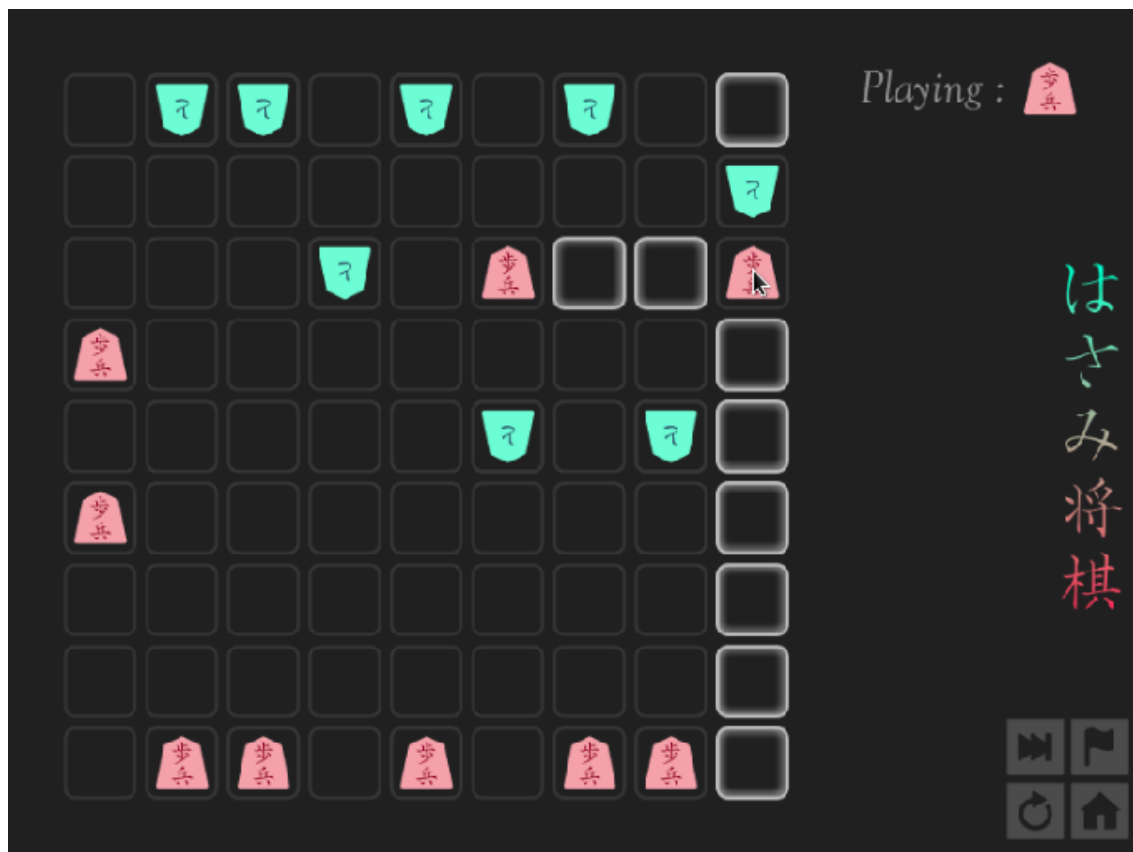
Tout d'abord nous voulons remercier amplement M. Syméon CARLE, qui nous a très gentiment proposé puis élaboré (sur inkscape) une maquette graphique pour notre jeu. Très moderne, belle et épurée, nous ne pouvons que lui en être reconnaissant.

Aussi la prise en main de SDL2 fut quelque peu ralenti par les tentatives d'installation de la librairie pour le moins laborieuse mais à force d'acharnement, de beaucoup de tests et d'aide son fonctionnement est devenu plus facile. La difficulté principale étant surtout de coller une image sur une fenêtre.

Il fallait aussi élaborer et gérer des structures adaptées (et ne pas oublier de les libérer !)

Pour l'aspect dynamique que nous avons voulu donner (pion déplacé mais aussi illumination des cases, nous avons conclu qu'il serait plus simple de refaire le tout à chaque fois plutôt que de mouvoir les objets.

Cependant la difficulté n'était pas forcément dans l'utilisation de la librairie en elle-même ni dans la gestion des structures utiles. La véritable difficulté était la gestion de la représentation mémoire qui a été faite jusqu'ici : il s'agissait de représentation matricielle qui lorsqu'elle était manipulée ou transposée nous donnait graphiquement des choses tout à fait singulières.



Capture 3 : L'interface graphique de la partie en cours

3.3 Idées d'amélioration à apporter

Le projet se voulait déjà très ambitieux mais nous avons eu le loisir de songer à des améliorations possibles :

- Trouver une intelligence artificielle plus poussée. Lui donner la possibilité de passer. Bien que celle-ci soit déjà relativement efficace.
- Donner la possibilité de sauvegarder la partie en cours et de la recharger plus tard.
- Que le type d'Hasami Shogi (Dai ou non) soit une option pour le joueur (et donc trouver une intelligence artificielle en conséquence).

4 Gestion de projet

Le projet s'étendant sur une période assez longue (3mois) et étant ambitieux, il s'agissait de bien s'organiser pour réussir à rendre un livrable fonctionnel et à la hauteur de nos ambitions. Nous nous sommes accordé pour nous voir très régulièrement et échanger sur le projet au jour le jour. Néanmoins nous avons fait quelques réunions régulières pour nous mettre à jour sur les échanges oraux et décider des choses à faire. Vous pouvez retrouver les compte-rendus de ces réunions en annexe. Pour une question d'habitude, nous avons utilisé GitHub et installé un mirroring sur la Forge (puisque'il fallait qu'il y ait une activité et un rendu sur la forge). Enfin nous avons beaucoup programmer en groupe pour s'entraider et puis partager nos idées car avoir un double voire triple regard sur la même chose s'avoue souvent bénéfique.

4.1 Fiche de définition

Pour bien déterminer le projet nous avons établie une fiche de définition de celui-ci pour mettre à plat ce qu'il y avait à faire et les moyens à notre disposition

Titre du projet	Hasami Shogi
Objectif Final	Implémenter un jeu d'Hasami Shogi avec intelligence artificielle et interface graphique.
Contexte	Projet effectué dans le cadre du module de C, en adéquation avec nos compétences. Ce projet a débuté le 17 Février et doit être terminé le jour de la soutenance le 7 juin.
Livrables	Code source à rendre le 31 Juin sur la forge. Rapport à déposer sur Arche avant le 31 Juin.
Moyens	Connaissances acquises au cours du module. Documentation illimitée que représente Internet. Tutorats proposés par deux 3ème année, Julien DÉOUX et Romain ORHAND.
Les risques	Ne pas terminer le projet à temps suite à une mauvaise estimation du temps à consacrer à certaines tâches. Ne pas organiser des réunions hebdomadaires à cause d'emplois du temps surchargés.
Les acteurs	Chef de projet : Jean CALVET Les autres acteurs : Pierre DENAUW et Charlotte ASSELIN BOULLÉ

4.2 Répartition du travail

Nous avons commencé par définir les rôles de chacun, et en particulier statué que Jean CALVET serait notre chef de projet.

Pour nous répartir le travail nous avons donné une tâche majeure de programmation à chacun de nous trois :

Charlotte devait s'occuper de l'interface graphique ainsi que de l'aspect secrétariat.

Jean se chargeait du jeu en lui-même dans sa partie 2 utilisateurs sur le même ordinateur puis a grandement assisté Charlotte pour l'élaboration de l'interface graphique, ainsi que aidé Pierre en ce qui concerne l'intelligence artificielle.

Quant à Pierre il s'occupait de comprendre, choisir puis programmer l'intelligence artificielle et donc la partie joueur seul contre l'ordinateur.

Tâches / Acteurs	Charlotte		Jean		Pierre	
. Étude de l'existant	R A	5h	A	3h	A	6h
Partie deux joueurs	C	2h	R A	20h	A	8h
Intelligence artificielle	I	-	C	4h	R A	28h
Interface graphique	R A	13h	A	20h	I	-
Compte-rendu	R A	21h	C	3h	C	5h
Total	A	41h	R A	50h	A	47h

Matrice RACI représentant notre répartition du travail

R pour Responsable, A pour Acteur(e), C pour Consulté(e) et I pour Informé(e)

4.3 Gestion du temps

Pour bien gérer notre projet et notamment ne pas nous laisser submerger par le travail ni délaisser le projet au profit du Projet de POO-SD en parallèle ou des cours en général il nous a semblé important de nous imposer des échéances.

Nous avons donc décider de le jalonner en 3 grandes parties chacune nous donnant un jeu fonctionnel avec une caractéristique en plus. Elles se chevauchent toutes bien car pour que l'une puisse être définitivement implémenter il fallait que au moins une partie des précédentes le soit. En particulier les partie 2 et 3 n'ont de sens que lorsque la partie 1 est finie et fonctionnelle. Enfin nous avons une dernière étape d'amélioration. Ces jalonnements sont volontairement très vagues pour permettre une plus grande flexibilité, et le délai se veut conséquent car nous n'avions pas connaissance de la quantité de travail que ça pouvait représenter et préférons voir large.

Vous remarquerez que les étapes correspondent à la répartition des tâches et la découpe de nos explications.

1. Il s'agit en tout premier lieu de faire un jeu basique. Ainsi le premier jâlon, avec une date limite fin Avril (au retour des vacances de Pâques) devait permettre de proposer un jeu qui pourrait

opposer deux joueurs. Le tout se ferait directement sur le terminal. En parallèle il fallait que Charlotte étudie SDL et que Pierre se renseigne et réfléchisse à quelle intelligence artificielle implémenter, et comment.

2. Mi Mai le jâlon concernant l'interface utilisateur devait être finie.
3. De même le jâlon d'intelligence artificielle s'étalait sur le mois de Mai.
4. Pour finir la dernière phase était, comme expliqué précédemment, le moment des améliorations si nous avions le temps pour. L'ajout d'un menu, la recherche d'une intelligence plus recherchée, la création de niveaux, etc...

4.4 Difficultés rencontrées

- Sur la forme :
 - Tout d'abord nous avons eu pas mal de problèmes dus à l'environnement de développement. En particulier nous avions tous les trois des systèmes d'exploitation différents, Jean sous Windows, Pierre sous Mac OS et Charlotte sous LINUX GNU/Fedora, ce qui faisait qu'il y avait parfois des conflits, par exemple dans l'importation de la librairie SDL2. Jean a ainsi installé une machine virtuelle Arch/Linux, pour pouvoir utiliser les fonctionnalités UNIX ainsi que CLion sous Windows pour bénéficier de la facilité et de l'habitude d'utilisation. Concernant les librairies, il est nécessaire d'inclure "SDL2_image/SDL_image" sous Mac mais "SDL2/SDL_image" sous Linux (fichier c-head/gui.h, ligne 11). Pour Windows Jean a du apprendre à utiliser CMake.
 - L'utilisation du Git a encore quelques zones obscures pour nous, tout du moins le fait de travailler ensemble sur les mêmes fichiers soulève souvent des conflits.
- Sur le fond :
 - Ne pas oublier de mettre à jour le Makefile ou encore d'inclure les fichiers nécessaires (avec le chemin d'accès). Erreur d'étourderie qui nous valait de nous arracher les cheveux au moment de compiler.
 - Utiliser la librairie SDL2. L'importer, la comprendre mais aussi réussir à lier la représentation en mémoire matricielle avec celle graphique.
 - (surtout) la mise en forme du code : représentation mémoire des différents éléments, utilisation des structures, pointeurs et autres spécificités de la programmation en C qui nous étaient encore un peu flous.
 - Pierre a eu des difficultés pour mettre au point son intelligence artificielle. En effet, d'une part s'est posé le problème du stockage mémoire. En effet, la recherche des mouvements possibles aurait pu se faire en utilisant un seul plateau constamment mis à jour en fonction des mouvements possibles. Mais cela posait des problèmes de représentations et d'algorithmique. D'un autre côté, l'application de l'algorithme fut moins aisé que prévu. Même si le principe est facile à comprendre, les fonctions d'évaluation du plateau sont à élaborer soit même et il faut beaucoup d'essais et de tâtonnement.

5 Conclusion

Ce projet fort ludique fut très enrichissant pour nous. Il nous permit en effet de mieux comprendre la programmation en C notamment la gestion des pointeurs, mais aussi de la mémoire, des structures ou encore le principe des Makefiles. Il nous a aussi permis d'explorer des possibilités que nous n'avions pas vu dans le cadre des cours, en particulier l'utilisation de la librairie graphique.

C'était aussi l'occasion pour nous de travailler en équipe et ce fut bénéfique car il y a eu beaucoup d'entre-aide (même si elle avait souvent lieu que dans un seul sens!) au sein du groupe, chose très appréciable.

En bref nous sommes heureux d'avoir élaboré ce projet en partant de rien et fières du rendu final, même si bien entendu nos aspirions à l'améliorer d'avantage.

6 Sitographie

[1] Abstract Strategy Games - Hasami Shogi

Khoa Pham

<http://abstractstrategygames.blogspot.fr/2010/10/hasami-shogi.html>

[2] Chess/Shogi/Hasami Shogi (page anglaise)

Katsutoshi Seki, 2 Janvier 1998

<http://www.chessvariants.com/shogivariants.dir/hasami.html>

[3] Braking / Hasami Shogi

<https://brainking.com/fr/GameRules?tp=73>

[4] Wikipedia, Hasami Shogi (page anglaise)

https://en.wikipedia.org/wiki/Hasami_shogi

[5] Wikibooks, Artificial Intelligence/Search/Adversarial search/Minimax Search

https://en.wikibooks.org/wiki/Artificial_Intelligence/Search/Adversarial_search/Minimax_Search

7 Annexes

7.1 Compte-rendus de réunion

Compte-rendu de la 1ère réunion (Mer 01/03/17)

Pages 17 et 18

Compte-rendu de la 2ème réunion (Mer 15/03/17)

Pages 19 et 20

Compte-rendu de la 3ème réunion, post-vacances (Mar 25/04/17)

Page 21

Compte-rendu de la 4ème réunion (Lun 15/05/17)

Pages 22 et 23

Compte-rendu de la dernière réunion (Mer 24/05/17)

Page 24

(Projet C) Compte-rendu de la 1ère réunion.

Charlotte Asselin-Boullé, Jean Calvet et Pierre Denauw

Mercredi 01 Mars 2017

Motif de la réunion: organisation générale du projet	Lieu: TELECOM Nancy Couloir E1
Présents : Charlotte, Jean et Pierre	Date, heure de début, durée : 01/03, 18h23, 1h

1 Ordre du jour

- Répartition des rôles
- Gestion du projet
- Mise au point sur certains éléments du jeu
- Choix initiaux
- Fixer les objectifs pour les semaines à venir

2 Informations échangées et décisions

Suite au mail de monsieur DA SILVA (hier, soit le 28/02) nous avons eu la confirmation du choix de notre jeu : l'Hasami Shogi.

Les rôles ont été distribués : Jean (chef de projet) s'occupe de la partie joue à 2, Charlotte fera la partie Interface graphique et l'aspect secrétariat et Pierre s'occupera de l'intelligence artificielle.

Étant donné que nous nous voyons régulièrement nous échangerons sur le projet au fur et à mesure mais nous allons quand-même organiser quelques grosses réunions de mise au point régulièrement. Par ailleurs nous allons mettre en place un GitHub pour une gestion plus agréable et nous ferons un "miroir" sur la Forge conformément aux consignes.

Le jeu est un jeu de plateau qui oppose deux joueurs. Le but est soit d'aligner 5 pions (stratégie d'attaque), soit d'empêcher l'autre de le faire en lui mangeant ses pions (stratégie de défense).

Pour l'instant nous partons sur une configuration de 9x9 cases avec 18 pions pour chaque joueur.

Il va s'agir maintenant de connaître le principe du jeu jusque dans ses subtilités avant de commencer à coder.

3 Actions à suivre

Quoi ?	Qui ?	Délai ?
Création Github/Forge, Mirroring github	Jean	pour la semaine prochaine
Se renseigner sur le jeu (règles détaillées et jouer)	tous	pour la prochaine réunion
déterminer ce qu'il y a à faire	tous	pour la prochaine réunion
Etat de l'Art	Charlotte	commencer

Date de la prochaine réunion : Mercredi 15 Mars, après nos TPs de C respectifs (et une pause bien méritée ;))

(Projet C) Compte-rendu de la 2ème réunion.

Charlotte Asselin-Boullé, Jean Calvet et Pierre Denauw

Mercredi 15 Mars 2017

Motif de la réunion: Choix pour notre jeu	Lieu: TELECOM Nancy Salle PI
Présents : Charlotte, Jean et Pierre	Date, heure de début, durée : 15/03, 16h, 35minutes

1 Ordre du jour

- Définition des "niveaux de complétion" du projet
- Choix des règles : retour sur l'état de l'Art
- Fixer les objectifs pour les semaines à venir

2 Décisions

On part sur l'idée d'un jeu simple mais fonctionnel et ludique : une interface graphique avec un plateau (de 9x9 cases), 18*2 pions. On donne la possibilité de joueur à 2 et celle de joueur seul avec l'ordinateur (donc contre une intelligence artificielle (même basique !)).

La manière de "manger" un pion est décidée comme celle d'encadrer un pion adverse avec ses propres pions, que ce soit horizontalement ou verticalement (et non comme celle de passer par dessus vue sur de (rares) versions). L'alignement des pions doit être de 5 hors zones de départ. Un joueur peut passer son tour, se déplacer d'autant de cases qu'il veut en horizontale ou en verticale jusqu'à rencontrer un obstacle. Il peut sauter un pion si il est dans la case adjacente et ne peut aller que dans la case adjacente de l'autre côté.

3 Actions à suivre

Quoi ?	Qui ?	Délai ?
Etat de l'Art pour l'intelligence artificielle	Pierre	-
Rédiger état de l'Art (règles...)	Charlotte	-
Commencer à approvoiser SDL	Charlotte	-
Coder la partie deux joueurs	Jean	vacances d'Avril

Date de la prochaine réunion : Retour des vacances de Pâques : Mardi 25 Avril à 18h (après l'examen d'Anglais, et une pause ;p)

(Projet C) Compte-rendu de la 3ème réunion.

Charlotte Asselin-Boullé, Jean Calvet et Pierre Denauw

Mardi 25 Avril 2017

Motif de la réunion: Avancement retour des vacances	Lieu: TELECOM Nancy Salle E.16
Présents : Charlotte, Jean et Pierre	Date, heure de début, durée : 25/04, 18h, 40min

1 Ordre du jour

- Avancement de chacun des membres
- Prévision

2 Informations et décisions

Jean a quasiment fini sa partie : il ne reste que quelques petits bugs à fixer et l'améliorer un peu sur certains points notamment la recherche pour savoir si un joueur a gagné ou non.

Charlotte a enfin fini par réussir à installer SDL et commence à le prendre en main (gestion fenêtre, clics...). Jean va certainement l'aider à faire l'Interface. Elle avance aussi sur la rédaction du rapport (il ne reste plus que les parties qu'on ne peut pas écrire d'avance).

Pierre a fini d'étudier les intelligences artificielles et déjà décider ce qu'il veut faire dessus, il commence à le coder.

3 Actions à suivre

Quoi ?	Qui ?	Délai ?
Finir avec les dernières retouches	Jean	d'ici la fin de la semaine
Comprendre le code dans son entier	Charlotte et Pierre	
Démarrer la programmation de l'UI	Charlotte	au plus vite
Démarrer la programmation de l'IA	Pierre	au plus vite

Date de la prochaine réunion : Vendredi 5 Mai 16h30

(Projet C) Compte-rendu de la 4ème réunion.

Charlotte Asselin-Boullé, Jean Calvet et Pierre Denauw

Lundi 15 Mai 2017

Cette réunion est la réunion reportée prévue initialement le 5.

Motif de la réunion: S'accorder sur l'avancement	Lieu: TELECOM Nancy Salle PI
Présents : Charlotte, Jean et Pierre	Date, heure de début, durée : 15/05, 16h27, 45min

1 Ordre du jour

- Etat du projet (avancement ?)
- Nécessité de changer du Dai Hasami Shogi à l'Hasami Shogi classique pour simplifier l'IA.
- Répartition des tâches restantes

2 Informations et décisions

Jean a fini sa partie mais doit la modifier en grande partie en effet il nous est nécessaire de changer nos ambitions et de passer du Dai Hasami Shogi à l'Hasami Shogi normal pour faciliter le calcul des valuations établie dans l'intelligence artificielle.

Pierre a commencé la dite IA. c'est d'ailleurs ainsi qu'il s'est rendu compte de la difficulté très importante que l'aspect "calcul des deux victoires possibles" représentait.

L'interface graphique avance (Syméon nous a fait une maquette et va rajouter des boutons d'options ainsi qu'un Menu) Charlotte a beaucoup de mal mais l'a en main et a même commencé à regarder le dynamisme derrière.

3 Actions à suivre

Quoi ?	Qui ?	Délai ?
Changer la partie jeu	Jean	le plus rapidement
Interface graphique	Jean et Charlotte	2 semaines
continuer l'intelligence artificielle		2semaines
Mettre à jour le rapport	Charlotte	la prochaine réunion

Date de la prochaine (et dernière) réunion : Mercredi 24 Mai, 16h30
(après le TP noté de BD et une belle pause)

(Projet C) Compte-rendu de la dernière réunion.

Charlotte Asselin-Boullé, Jean Calvet et Pierre Denauw

Mercredi 24 Mai 2017

Motif de la réunion: Organisation finale	Lieu: TELECOM Nancy Salle N0.44
Présents : Charlotte, Jean et Pierre	Date, heure de début, durée : 24/05, 16h, 20min

1 Ordre du jour

- Organisation de la dernière semaine

2 Décisions

Il nous reste à rendre plus performante l'Intelligence Artificielle et corriger la partie qui prend en mémoire les possibilités et celle qui renvoie le mouvement. Par ailleurs Charlotte va se charger (aidée de Jean) de faire un Menu Graphique. Il faut aussi vérifier que tout le code est bien commenté et puis clôturer le rapport.

3 Actions à suivre

Quoi ?	Qui ?	Délai ?
Menu dans UI	Charlotte	pour Mercredi
Finir l'IA	Pierre	pour Mercredi
Aider les autres et commenter	Jean	-
Finir le rapport	Charlotte	pour Mercredi

Date de la prochaine réunion : Avant la soutenance pour se mettre au point si besoin.