

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

UN SYSTÈME D'INTELLIGENCE ARTIFICIELLE POUR LE JEU DE PLATEAU
AXIES & ALLIES

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN INFORMATIQUE

PAR

ALEXANDRE BUSTROS

JUILLET 2009

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

TABLE DES MATIÈRES

LISTE DES FIGURES.....	iv
RÉSUMÉ	vi
INTRODUCTION	1
CHAPITRE I - L'INTELLIGENCE ARTIFICIELLE ET LES JEUX	2
1.1 Les échecs	2
1.1.1 Complexité du plateau de jeu	4
1.1.2 Complexité de la prise de décisions	4
1.2 Le jeu de Go	4
1.2.1 Complexité du plateau de jeu	6
1.2.2 Complexité de la prise de décision	7
1.3 Conclusion	7
CHAPITRE II - LE JEU AXIS & ALLIES	8
2.1 Aspect historique du jeu.....	9
2.2 Les règles du jeu.....	10
2.2.1 Achat de nouvelles unités.....	11
2.2.2 Mouvement offensif.....	11
2.2.3 Phase de combat.....	12
2.2.4 Mouvement défensif.....	13
2.2.5 Placer les nouvelles unités	13
2.3 Les différentes unités	13
2.3.1 Les troupes terrestres.....	14
2.3.2 Les troupes navales	15
2.3.3 Troupes aériennes.....	16
2.3.4 Troupes neutres	17

2.4 Exemple d'un tour de jeu.....	18
2.5 Conclusion sur les règles.....	18
CHAPITRE III - LE RÔLE DE L'IA POUR LE JEU D'AXIS & ALLIES.....	19
3.1 Hasard	19
3.2 Non social	20
3.3 La complexité du plateau de jeu.....	21
3.4 Les questions à répondre face au jeu.....	22
CHAPITRE IV - LES SOLUTIONS CLASSIQUES POUR LE JEU AXIS & ALLIES	24
4.1 Réseaux de neurones	24
4.1.1 Implémentations.....	26
4.1.2 La couche d'entrée	26
4.1.3 La couche de sortie.....	28
4.1.4 Limitations et faisabilité.....	29
4.1.5 Conclusion	29
4.2 Système expert	30
4.2.1 Implémentations.....	30
4.2.2 Limitations et faisabilité.....	31
4.2.3 Conclusion	32
4.3 Algorithme génétique.....	33
4.3.1 Implémentations.....	34
4.3.2 Limitations et faisabilité.....	36
4.3.3 Conclusion	37
4.4 Algorithme Min-Max	37
4.4.1 Implémentations	38
4.4.2 Limitations et faisabilité.....	40
4.4.3 Conclusion	41

4.5 Choix du système	41
CHAPITRE V – GRANDES LIGNES DE LA MISE EN OEUVRE	42
5.1 Architecture du programme	42
5.1.1 Le système Web	43
5.1.2 Le système d'IA.....	43
5.1.3 La couche des objets utilitaires	45
5.2 Les écrans.....	46
5.3 La représentation des données	49
5.3.1 Les sommets.....	50
5.3.2 Les arêtes.....	51
5.3.3 Mise en oeuvre	53
5.4 Les règles du système expert.....	53
5.4.1 La phase d'achat de nouvelles troupes.....	54
5.4.2 La phase des mouvements de combat	55
5.4.3 La phase des mouvements pacifiques	60
5.4.4 La phase des placements des nouvelles unités	61
CHAPITRE VI RÉSULTATS OBTENUS.....	63
6.1 Résultats obtenus.....	63
6.2 Test contre les humains.....	73
6.3 Conclusion sur les tests	74
CONCLUSION.....	75
BIBLIOGRAPHIE	76
ANNEXE	77

LISTE DES FIGURES

Figure		Page
2.1	Le plateau de jeu d'Axis & Allies...	10
2.2	Représentation d'un mouvement offensif...	12
2.3	Représentation d'un combat...	13
4.1	Représentation d'un réseau de neurones...	25
4.2	La couche d'entrée d'un réseau de neurones conçue pour Axis & Allies...	27
4.3	Un individu dans un algorithme génétique conçu pour Axis & Allies....	35
4.4	Représentation de l'algorithme min & max ...	38
4.5	Un tour de jeu pour l'algorithme min & max conçu pour Axis & Allies...	40
5.1	Représentation de haut niveau du système d'IA ...	43
5.2	Écran de base du jeu d'axis & allies (version française)...	47
5.3	Écran de modification d'une valeur (version française)...	48
5.4	Action exécuté par le système d'IA lors d'un tour de jeu (version anglaise)...	49
5.5	Carte de la nomenclature des terrains...	51
5.6	Exemple d'une portion de la carte ...	52
5.7	Représentation logique de la figure 5.6 ...	53
6.1	Résultats de la première partie où tous les joueurs sont joués par le système Iron Blitz...	65
6.2	Résultats de la deuxième partie où tous les joueurs sont joués par le système Iron Blitz...	65
6.3	Résultats de la troisième partie où tous les joueurs sont joués par le système Iron Blitz...	65
6.4	Résultats de la quatrième partie où tous les joueurs sont joués par le système Iron Blitz...	66
6.5	Résultats de la cinquième partie où tous les joueurs sont joués par le système Iron Blitz...	66
6.6	Résultats de la première partie où l'Allemagne est jouée par mon système et tous les autres joueurs sont joués par le système Iron Blitz....	67
6.7	Résultats de la deuxième partie où l'Allemagne est jouée par mon système et tous les autres joueurs sont joués par le système Iron Blitz....	67
6.8	Résultats de la troisième partie où l'Allemagne est jouée par mon système et tous les autres joueurs sont joués par le système Iron Blitz....	67
6.9	Résultats de la quatrième partie où l'Allemagne est jouée par mon système et tous les autres joueurs sont joués par le système Iron Blitz....	68

6.10	Résultats de la cinquième partie où l'Allemagne est jouée par mon système et tous les autres joueurs sont joués par le système Iron Blitz...	68
6.11	Capture d'écran de la première partie où tous les joueurs sont joués par le système Iron Blitz	69
6.12	Capture d'écran de la deuxième partie où tous les joueurs sont joués par le système Iron Blitz	70
6.13	Capture d'écran de la première partie où l'Allemagne est joués par mon système et tous les autres joueurs sont joués par le système Iron Blitz.....	71
6.14	Capture d'écran de la deuxième partie où l'Allemagne est joués par mon système et tous les autres joueurs sont joués par le système Iron Blitz	72

RÉSUMÉ

Les chercheurs en intelligence artificielle se sont toujours intéressés aux problèmes des jeux de société. Ces jeux sont depuis longtemps perçus par les humains comme des façons de prouver leur supériorité intellectuelle face à leurs adversaires. Bobby Fisher, ancien champion du monde d'échecs, a dit : « C'est juste vous et votre adversaire sur l'échiquier, et vous essayez de prouver quelque chose. » Il a expliqué aussi : « Le moment que je préfère le plus dans une rencontre, c'est celui où je sens que la personnalité de l'adversaire se brise¹. »

Il est naturel que les programmeurs voulant prouver l'efficacité de leurs techniques de programmation se confrontent aux mêmes jeux. L'objectif de ce mémoire est de proposer un système d'intelligence artificielle au jeu de société Axis & Allies. Il existe un système propriétaire qui joue présentement à ce jeu. Il a été créé par Avalon Hill, une entreprise qui s'occupe de la distribution du jeu. Selon les opinions recueillies sur les forums de discussions, le système d'intelligence artificielle qui existe n'est pas au niveau des joueurs expérimentés. Un joueur a posté le commentaire suivant : « *Iron Blitz has an ALI (Artificial Lack of Intelligence)*². » Il sera donc question dans le présent travail de proposer une meilleure solution à ce problème.

Mots-clés : Intelligence Artificielle, Axis & Allies, Système expert, Jeux de société

¹ DICCIT ECHECS. 2008. « Liste par thème: L'adversaire », In *DICCIT Echecs*. En ligne. <<http://diccit-echecs.awardspace.com/theme.html?cat=8>>. Consulté le 20 janvier 2009.

² axisandallies.org. s.d. « Iron Blitz », In *Forum d'Axis & Allies*. En ligne. <<http://www.axisandallies.org/forums/index.php?topic=5774.0>>. Consulté le 20 janvier 2009.

INTRODUCTION

Tous les amateurs de jeu le savent : il est de plus en plus fréquent de se confronter à un système d'intelligence artificielle lors d'un jeu, que ce soit pour jouer aux dames contre un adversaire virtuel, pour affronter aux échecs un système spécialisé ou pour jouer rapidement au poker sur Internet. L'intelligence artificielle dans les jeux fait maintenant partie d'une réalité facilement accessible.

Le but de cette recherche est de trouver une manière efficace pour qu'un ordinateur puisse remplacer un joueur dans le jeu de société Axis & Allies. Dans le premier chapitre, il sera question d'analyser les différents progrès des algorithmes connus dans les jeux de société dits « classiques ». Dans le deuxième chapitre, une présentation générale du jeu de société Axis & Allies sera effectuée. Dans le troisième chapitre, les questions et les spécificités de l'intelligence artificielle propre au jeu Axis & Allies seront soulevées et expliquées.

Dans le quatrième chapitre, quatre solutions du domaine de l'intelligence artificielle seront étudiées, soit les réseaux de neurones, les systèmes experts, les algorithmes génétiques et l'algorithme Min-Max. Suite à cette étude théorique, une mise en oeuvre de la méthode la plus prometteuse sera réalisée.

Les grandes lignes de la mise en oeuvre seront expliquées dans le chapitre cinq et, dans le chapitre six, les quelques tests et résultats seront présentés. Il existe déjà un système informatique qui joue à Axis & Allies, ce système sera donc utilisé comme baromètre pour notre recherche. Une fois la mise en oeuvre terminée, les deux systèmes s'affronteront, et il sera donc possible d'en analyser les résultats. L'objectif souhaité est évidemment que le nouvel algorithme remporte la victoire contre celui du système existant.

CHAPITRE I – L'INTELLIGENCE ARTIFICIELLE ET LES JEUX

De nombreux jeux de société sont aujourd'hui disponibles sur ordinateur, qu'il s'agisse des échecs, du Monopoly ou du jeu Risk. Ces différents jeux proposent des défis qui sont différents, et le programmeur devra savoir quelle technique utiliser pour qu'un système d'intelligence artificielle fournisse un bon rendement. Dans ce chapitre, nous exposerons les problèmes que proposent le jeu d'échecs et le jeu de Go en IA. Pour chacun de ces jeux, nous décrirons un système de pointe et actuel qui a réussi à bien se classer dans les dernières compétitions.

1.1 Les échecs

Le jeu d'échecs tel qu'il est connu aujourd'hui a été inventé au XVI^e siècle. Depuis longtemps les hommes se sont intéressés à inventer des machines capables de jouer aux échecs. Outre les blagues où un homme se cachait dans une boîte pour jouer au jeu, les premières machines capables de jouer aux échecs datent du début du siècle précédent. « In 1912 Torres built his chess machine "Ajedrecista", using electromagnets under the board to play the endgame rook and king against the lone king³. ». Il s'agit d'une simulation de fin de partie dans laquelle l'automate possède un roi et un pion, tandis que le joueur perdant n'a qu'un roi. L'automate gagnera, peu importe les mouvements du joueur.

Évidemment, quand les ordinateurs modernes ont fait leur apparition, plusieurs systèmes informatiques ont été créés par différents programmeurs en utilisant différentes techniques. Parmi les plus connus, on entend souvent parler de l'ordinateur Deep Blue d'IBM. Deep Blue

³ Jiménez, Ramon, 2004. « Torres y Quevedo's rook endgame automaton », in *chessbase*. En ligne. <<http://www.chessbase.com/newsdetail.asp?newsid=1799>>. Consulté le 20 janvier 2009

est devenu célèbre en battant le champion de l'époque Garry Kasparov, en mai 1997⁴. IBM consolide sa victoire en affirmant que son ordinateur était le plus puissant, tandis que Kasparov affirme avoir été triché, que l'ordinateur était aidé par un joueur humain⁵. Deep Blue serait-il très différent des machines ayant un être humain à l'intérieur ? Ce n'est pas là une question à laquelle nous pouvons répondre.

Pour analyser un système d'IA qui joue à un jeu de société, il faut à la base parler de l'ordinateur qui exécutera le programme et de l'algorithme qui sera utilisé. Dans le cas de Deep Blue, l'ordinateur avait été créé sur mesure et avait une puissance de calcul de 300 millions de coups par seconde⁶.

La philosophie d'IBM consistait à utiliser la puissance de calcul brute pour évaluer le plus d'options possible. L'algorithme utilisé pour rendre le choix des mouvements était basé sur l'algorithme Min-Max. L'algorithme exact n'a pas été dévoilé, et Deep Blue a été démantelé peu de temps après sa victoire. L'algorithme Min-Max consiste à calculer toutes les options de jeu sur un nombre prédéterminé de tours (une définition plus détaillée sera présentée au chapitre 4). Cet algorithme est un classique des jeux de société, d'ailleurs dans les systèmes les plus efficaces d'aujourd'hui, il est encore utilisé. Il se joue un tournoi qui se nomme « *World Computer Chess Championship* ». Le dernier champion a été le programme

⁴ Saletan, William. «Chess Bump: The triumphant teamwork of humans and computers.». Slate. En Ligne. May 2007. <http://www.slate.com/id/2166000/pagnum/all/#p2>. Consulté le 21 janvier 2009.

⁵ Vikram, Jayanti. 2003. *Game Over: Kasparov and the Machine*. Film 16mm, coul., 84 min 50 s. National Film Board of Canada

⁶ IBM. s.d., «IBM : The making of Deep Blue », In *IBM : research* En Ligne. <http://www.research.ibm.com/deepblue/meet/html/d.3.1.shtml>. Consulté le 21 janvier 2009.

« Rybka »⁷. Pour des raisons évidentes de concurrence, le code de ce programme n'est pas connu. Par contre, on sait que ce programme utilise, à la base, l'algorithme Min-Max⁸.

1.1.1 Complexité du plateau de jeu

Le jeu d'échecs se joue sur un plateau de 8 par 8, ce qui fait un total de 64 cases. Chaque tour de jeu, le nombre de possibilités de jeu peut aller de 0 à 218. En moyenne, il y en a 35+-5⁹. Plus la partie avance, plus les pièces disparaissent, ce qui réduit la complexité de la prise de décisions.

1.1.2 Complexité de la prise de décisions

Un système d'intelligence artificielle qui joue au jeu d'échecs doit en fait répondre à la question « Quelle pièce dois-je bouger et de quelle position vers quelle position ? »

1.2 Le jeu de Go

Le jeu de Go a été inventé en Chine. La date exacte de sa création n'est pas connue, mais selon certaines légendes chinoises, il semble que le jeu daterait de plus de 4 000 ans. L'engouement pour créer un système d'intelligence artificielle pour ce jeu est très actuel. Contrairement aux échecs, où les machines rivalisent contre les champions du monde, le jeu de Go constitue encore un domaine où la supériorité est réservée aux joueurs humains. Il y a même une récompense de 40 000 000 NT dollars offerte par l'institution « *Taiwanese Banker Ing Chang-Ki* ». Cette récompense n'a toujours pas été réclamée, parce que personne

⁷ ICGATournaments. 2007. «ICGATournaments : Amsterdam 2007», In *ICGA*, En ligne. <<http://www.grappa.univ-lille3.fr/icga/event.php?id=40>>, Consulté le 21 janvier 2009.

⁸ Rybka. 2008. «Rybka user area : Release notes » In *RybkaChess*. En ligne. <<http://rybkachess.com/index.php?auswahl=Release+notes>>. Consulté le 21 janvier 2009.

⁹ Shannon, Claude E. 1950. « Programming a Computer for Playing Chess ». *Philosophical Magazine*, Ser.7, Vol. 41, No. 314.

n'est encore capable de construire un ordinateur capable de jouer en ayant un calibre professionnel contre un être humain, et ce, sans handicap¹⁰.

Pourtant, les règles du jeu de Go sont assez simples. Il s'agit d'un type de pièces qu'il faut placer pour encercler l'adversaire. Ce qui fait du jeu de Go un défi pour les ordinateurs, c'est la grandeur du plateau de jeu. Les algorithmes traditionnels qui calculent toutes les positions se heurtent à une plaque de jeu beaucoup trop grande. L'algorithme Min-Max, qui fonctionne bien pour les échecs, se heurte à la grandeur du plateau du jeu de Go. D'autres options sont alors intéressantes.

L'organisme *ICGA Tournaments* réalise des compétitions dans lesquelles les ordinateurs peuvent se confronter les uns les autres et ce dans différents jeux, notamment au jeu de Go. Nous allons brièvement analyser les algorithmes utilisés dans les programmes ayant gagné des tournois dans les quatre dernières années seront analysés. Ces programmes se classent parmi les meilleurs programmes de Go. Il sera donc question des programmes suivants : Mo Go, GNU Go, HandTalk III et Go Intellect¹¹.

La toute dernière approche pour résoudre le problème du jeu de Go est basée sur le hasard. On parle souvent de la méthode Monte Carlo comme représentant une solution en elle-même, mais c'est un peu un abus de langage. L'approche de Monte Carlo est une approche de résolution de problèmes se basant sur le hasard. C'est un concept qui englobe plusieurs méta-heuristiques, par exemple, les algorithmes génétiques. Ce qui définit une heuristique comme faisant partie de l'approche Monte Carlo, c'est l'utilisation du hasard pour générer des solutions qui sont ensuite évaluées, classées et modifiées selon l'algorithme

¹⁰ Macintyre, Ben. 2007. "Why computers can't surpass Go and collect \$1 million". The Times, 29 juin 2007.

¹¹ Rybka. 2008. «Rybka user area : Release notes » In RybkaChess. En ligne. <<http://www.grappa.univ-lille3.fr/icga/game.php?id=12>>. Consulté le 21 janvier 2009.

choisi. Le système Mo Go, gagnant de la première place du championnat ICGA en 2007, utilise ce genre de méthode.

D'autres systèmes se basent sur la création d'un système axé sur les connaissances des humains, c'est-à-dire que des experts explicitent leurs connaissances sous forme de règles dans un système. Le système analyse les données du jeu et exécute les règles. Les programmes Go Intellect, gagnant de 2005, et Hand Talk III, gagnant de 2004, utilisent cette heuristique.

Finalement, la même technique que celle utilisée aux échecs, celle de l'algorithme Min-Max qui calcule les coups à venir et qui demande une force de calcul considérable, a été utilisée dans le jeu de Go. L'algorithme Gnu Go fonctionne avec cette méthode et il a remporté le championnat en 2006¹².

1.2.1 Complexité du plateau de jeu

Le jeu de Go se joue sur un plateau de taille 19 par 19. Les pierres sont les seules pièces. Chaque joueur dispose du nombre de pièces qu'il désire, ces pierres doivent être jouées sur l'une des 361 intersections.

On peut comparer la complexité du jeu d'échec avec celle du jeu de go :

Le nombre de coups possibles dans une position typique avoisine en moyenne 300 au Go contre 40 aux Échecs, le nombre de coups joués dans une partie typique avoisine 250 au Go contre 100 aux échecs et enfin la taille très approximative de l'arbre des possibilités du jeu de Go est environ de 10^{600} alors que celui du jeu d'échecs n'est seulement (!) que de 10^{150} environ¹³.

¹² Gelly, Sylvain. et Jongwane, Joanna. « MoGo, maître du jeu de Go ? ». interstices. En ligne. (15 décembre 2007). http://interstices.info/jcms/c_21056/mogo-maitre-du-jeu-de-go. Consultée le 21 janvier 2009.

¹³ Bouzy, Bruno, et Victorri Bernard. 2002. "Le jeu de Go et l'intelligence artificielle", Paris, Université de Paris 5.

Étant donné les règles de placement des nouvelles pièces, les possibilités augmentent au fur et à mesure que les pièces se placent sur le plateau de jeu.

1.2.2 Complexité de la prise de décision

Le système d'intelligence artificielle doit répondre à la question : Où dois-je placer ma nouvelle pièce ?

1.3 Conclusion

Dans ce travail, l'objectif est de construire un système d'intelligence artificielle pour le jeu d'Axis & Allies. Quoique populaire, ce jeu a été au siècle dernier et n'a pas eu autant d'initiés que les jeux dits classiques comme le Go ou les échecs, pour ne nommer que certains qui seront cités dans ce mémoire.

La popularité de ces jeux a mené plusieurs personnes à développer un système d'intelligence artificielle simulant la prise de décision d'un humain. Lors de l'analyse des solutions possibles, quand les similarités entre un jeu classique et le jeu d'Axis & Allies le permettront, nous analyserons les résultats des différentes techniques d'intelligence artificielle sur les jeux classiques pour en mesurer le succès. En d'autres mots, les échecs et le jeu de Go seront nos systèmes baromètre lors de l'analyse des solutions possibles. L'objectif est de profiter de l'expertise existante dans la création de système d'intelligence artificielle dans ces jeux.

CHAPITRE II – LE JEU AXIS & ALLIES

Tout d'abord, puisque ce projet porte sur un jeu de société, il est nécessaire de commencer par expliquer le jeu en question. Axis & Allies est un jeu de société qui se veut une simulation de la Deuxième Guerre mondiale. Le jeu se joue sur une carte du monde, où chaque joueur déplace ses différentes unités militaires, un peu comme dans le très connu jeu de Risk, mais avec plus de règles et, donc, avec plus de complexité.

Axis & Allies est un jeu de plateau. Voici une description des jeux de plateaux :

Jeu de société qui se joue sur un plateau de jeu, représentant un parcours ou une carte, sur lequel se déplacent des pions, selon des règles ou selon le hasard, qui peuvent parfois s'affronter ou accumuler des points en collectant des ressources matérialisées par des marqueurs.[...] Il s'agit généralement de jeux faisant appel à nos capacités de réflexion, d'observation ou de négociation, où l'on joue avec des pions (qui représentent les joueurs), des jetons et des figurines, et où l'on intègre une part de hasard par l'utilisation de cartes et de dés qui influent sur le déroulement des parties.

Dans ce type de jeu, l'action se déroule sur une surface délimitée, le plateau de jeu, comportant des étapes à franchir afin d'arriver à la victoire. Ainsi, le premier joueur à avoir accompli le parcours, ou le dernier à avoir encore des pions, ou celui qui a accumulé le plus grand nombre de points, ou toute autre condition spécifique, est déclaré vainqueur. Parmi les jeux de plateau les plus connus, on trouve le Monopoly, Risk et Clue¹⁴.

Dans le cas d'Axis & Allies, le nombre de joueurs est limité à cinq. Chaque joueur incarne une « nation » influente de la Deuxième Guerre mondiale. Les nations représentées par le jeu sont les suivantes :

- La Russie ;
- L'Allemagne ;

¹⁴ Office québécois de la langue française. 2006. Le grand dictionnaire terminologique. En ligne. <http://www.granddictionnaire.com>. Consulté le 20 janvier 2009.

- Le Royaume-Uni ;
- Le Japon ;
- Les États-Unis.

L'objectif du jeu est évidemment de gagner la guerre. Il y a deux alliances qui s'affrontent, soit celle de l'Axe et celle des alliés. Les alliés comprennent l'Empire britannique, les Russes et les Américains, et la puissance de l'Axe est composée des Allemands et des Japonais. Il est important de mentionner que la victoire n'est pas une victoire individuelle, mais bien par alliance, c'est-à-dire que le joueur qui joue l'Allemagne gagne si l'Axe gagne ; il n'y a pas de victoire personnelle. Il est facile de savoir si un joueur progresse ou s'il régresse en regardant la valeur de son territoire.

S'il y a moins de cinq joueurs, une ou plusieurs personnes contrôlent plus d'une des nations d'une même alliance, par exemple, s'il n'y a que deux joueurs, l'un d'eux jouera l'Axe et l'autre jouera les alliés.

2.1 Aspect historique du jeu

Le jeu se déroule en 1942, une année charnière de ce véritable désastre qu'a constitué la Deuxième Guerre mondiale.

L'année 1942 représente pour la 2^e Guerre mondiale ce qu'avait été l'année 1917 pour la 1^{re} Guerre mondiale, un renversement de situation décisif pour l'issue du conflit. Le tournant de 1942 découle de la mondialisation du conflit qui modifie sensiblement le rapport des forces. Il est marqué d'abord par les dernières offensives victorieuses de l'Axe, puis par les coups d'arrêt et les premières contre-offensives de la Grande Alliance¹⁵.

En effet, les Allemands semblent être en pleine possession de leurs moyens, ils ne sont pas encore écrasés en Russie, et les Américains ne sont pas encore entrés en guerre.

¹⁵ CRDP de Champagne-Ardenne, 2000-2005. Le monde en guerre en 1942. En Ligne. <<http://www.crdp-reims.fr/memoire/bac/2GM/sujets/01mondeenguerre.htm> >. Consulté le 20 janvier 2009.

C'est à ce moment qu'il est possible de refaire l'histoire, mais cette fois, en ne sacrifiant la vie de personne.

2.2 Les règles du jeu

La manière la plus simple de faire comprendre le jeu est d'expliquer comment un tour complet de jeu se déroule.

Avant de parler d'un tour de jeu, décrivons la situation initiale. Le jeu se joue sur une carte du monde divisée en territoires qui ne correspondent pas à des pays. La carte se trouve à la figure 2.1.

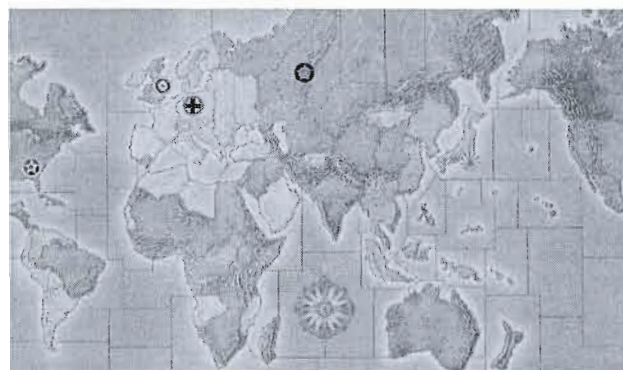


Figure 2.1 : Le plateau de jeu d'Axis & Allies.

Chaque couleur correspond à une nation. Le brun pâle désigne les Anglais, le jaune les Japonais, le gris les Allemands, et le brun rougeâtre indique les Russes. Chaque nation commence en ayant des unités militaires dans la majorité de ses territoires. Le nombre et le type d'unités militaires sont définis par les règles, au début du jeu. Donc, comme aux échecs, le début d'une partie d'Axis & Allies est toujours le même, ce sont les décisions des joueurs et le hasard des dés qui façonnent la partie.

L'ordre du jeu est décidé par les règles. La première nation à jouer sera toujours la Russie. Ensuite, joueront, dans l'ordre, les Allemands, les Britanniques, les Japonais et les

Américains, ce qui conclut un tour. Au tour suivant, les Russes joueront à nouveau, et cet ordonnancement sera respecté jusqu'à la fin.

Les joueurs disposent également de billets qui sont des indices de production industrielle appelés IPC, ce qui équivaut tout simplement à de l'argent durant le jeu. Au début du jeu, les nations disposent d'un montant prédéterminé d'argent. Dès la fin du premier tour, ce montant d'argent sera calculé en fonction du nombre de territoires conquis par la nation. Donc, plus une nation conquiert de territoires, plus elle pourra acheter d'unités au tour suivant.

Chaque nation jouera un tour de jeu qui comporte les phases de jeu suivantes.

2.2.1 Achat de nouvelles unités

La première chose qu'un joueur peut faire est d'acheter de nouvelles unités militaires. Ce qui fait tout l'intérêt de ce jeu est la présence de diverses unités militaires ayant des coûts différents et des effets différents. Nous discuterons un peu plus loin des différents types d'unités. Lorsque les unités sont achetées, elles sont mises de côté jusqu'à la phase de déploiement de nouvelles unités.

2.2.2 Mouvement offensif

Le joueur qui joue peut effectuer autant de mouvements offensifs qu'il le désire, c'est-à-dire qu'il peut déplacer une ou plusieurs unités militaires sur un territoire qu'il ne contrôle pas.

À la figure 2.2, nous avons l'exemple d'un mouvement offensif du joueur russe envers le joueur japonais. Puisque le territoire japonais est occupé, il y aura vraisemblablement un combat à la phase de combat.

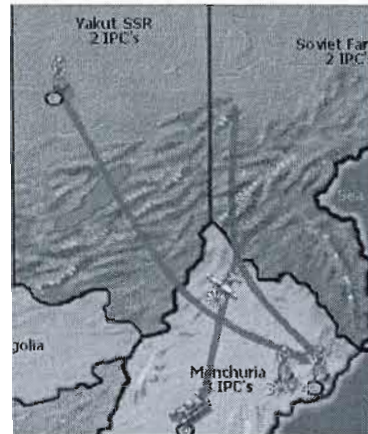


Figure 2.2 : Représentation d'un mouvement offensif

2.2.3 Phase de combat

Pour chaque territoire où se trouvent ensemble des unités militaires appartenant à l'Axe et des unités militaires appartenant aux alliés, il y aura un combat.

Une des caractéristiques de ce jeu est que chaque unité se comporte différemment en défense qu'elle ne le fait en attaque. C'est pourquoi il est important de planifier une attaque avec les bonnes unités.

Le fonctionnement des combats est assez simple. L'attaquant lance les dés pour savoir quelle unité réussira son attaque, l'attaquant fera de même par la suite. Puis, les deux adversaires doivent détruire des unités militaires en fonction des pertes qu'ils ont subies. Chaque unité militaire réussissant son attaque fait perdre une unité militaire au joueur adverse. Les unités qui survivent recommencent et ce jusqu'à la fin du combat.

On peut voir, à la figure 2.3, la version informatisée du plateau d'attaques et de défenses, qui rappelle les valeurs d'attaques et de défenses de chaque troupe.



Figure 2.3 : Représentation d'un combat

2.2.4 Mouvement défensif

Toutes les unités qui n'ont pas attaqué durant le tour courant peuvent se déplacer à l'intérieur de leurs territoires ou de ceux de leurs alliés.

2.2.5 Placer les nouvelles unités

À la fin du tour, les unités qui ont été achetées au début du tour sont placées sur le plateau de jeu. Les joueurs ne peuvent placer leurs nouvelles unités que sur des terrains ayant des usines.

2.3 Les différentes unités

Tel que mentionné ci-dessus, chaque unité a donc des caractéristiques différentes, ce qui fait de ce jeu une excellente étude pour un système d'intelligence artificielle. Étant donné les valeurs des différentes unités, parfois une attaque peut avoir comme intérêt de réduire la force d'offensive de l'adversaire, même si elle s'avère un véritable suicide.

Afin d'alléger la forme littéraire de ce document, nous utiliserons les expressions suivantes : « l'unité attaque à x » et « l'unité défend à y ». Ces expressions signifieront que l'unité réussira son attaque, si le joueur lance le chiffre x ou moins que le chiffre x avec un dé. Autrement dit, l'unité à x chance sur 6 de réussir son attaque et y chance sur 6 de réussir à

se défendre. Lors d'une attaque, les joueurs lancent les dés pour chaque unité en défense ou à l'attaque.

La description des unités militaires se fera en trois sous-catégories :

- Les unités terrestres ;
- Les unités navales ;
- Les unités aériennes.

2.3.1 Les troupes terrestres

Les troupes terrestres sont les seules unités qui peuvent conquérir un territoire. À la suite d'un combat, si aucune troupe terrestre ne se trouve sur le territoire ennemi, celui-ci reste sous le contrôle de l'ennemi.

Il y a deux types d'unités terrestres.

- Infanteries

L'infanterie, qui représente un régiment de soldats, forme l'unité de base. Elle ne coûte que 3 IPC à produire. Elle attaque à un et se défend à deux. Elle ne possède aucune capacité particulière. L'infanterie peut se déplacer d'une case à l'attaque ou en mouvement défensif.

Quand un joueur subit des pertes à l'attaque ou en défense, il doit retirer des unités, les bons joueurs essayant autant que possible de faire mourir les soldats avant d'avoir à détruire les tanks ou les avions. Rappelons que ceci n'est qu'un jeu.

- Tanks

Les tanks coûtent 5 IPC. Ils attaquent à trois et se défendent à deux. Il est donc avantageux d'attaquer avec des tanks. C'est l'unité d'attaque par excellence, elle ne coûte pas trop chers et elle est très efficace à l'attaque. Le tank peut se déplacer de deux cases à l'attaque ou en mouvement défensif.

Le tank possède une capacité spéciale : il peut conquérir deux territoires en un seul tour si le premier est vide, et ce, en utilisant ses deux mouvements. Dans Axis & Allies, il n'est pas nécessaire de laisser des unités sur un territoire. Une fois qu'il est conquis, le territoire appartient au joueur jusqu'à ce qu'un autre joueur le reprenne.

2.3.2 Les troupes navales

Comme leur nom le laisse entendre, les troupes navales œuvrent sur l'eau. Les troupes navales ne peuvent en aucun temps se trouver sur une zone terrestre. Puisque les troupes navales ne servent que dans des zones maritimes, elles sont plus subtiles à utiliser. Les territoires navals n'ont aucune valeur. Il n'est donc pas nécessaire de les occuper pour gagner la partie. Cependant, il est important de bien gérer les bateaux, parce qu'ils ont plusieurs caractéristiques qui influencent grandement les attaques terrestres. Toutes les unités navales peuvent se déplacer de deux cases.

- Sous-marin

Le sous-marin est l'unité maritime de base. Il coûte 8 IPC, et il attaque à deux et se défend à deux. Un sous-marin peut attaquer ou se défendre seulement si l'adversaire possède des troupes navales. En contrepartie, il peut se submerger s'il résiste au premier tour d'attaque.

- Porte-avions

Le porte-avions coûte 18 IPC, il attaque à un et se défend à trois. Il constitue une très bonne unité de défense ; il peut transporter deux chasseurs. Ces chasseurs se défendent quand le territoire du porte-avions est attaqué, ce qui aide à fortifier rapidement la défense navale.

- Frégate ou navire de guerre

Le navire de guerre coûte 24 IPC. Il attaque à quatre et se défend à quatre. Il s'agit d'une unité dispendieuse, mais qui a la capacité de bombarder les côtes à l'aide de transporteurs lors d'un débarquement.

- Transporteur

Le transporteur coûte 8 IPC, il se défend à un et n'attaque pas. Un transporteur permet de transporter deux soldats ou un tank. Avec cette unité, il est donc possible de transporter des unités entre les continents ou tout simplement de déplacer des unités plus rapidement. Les unités ont besoin d'un certain déplacement pour entrer et sortir du transport. Parfois, elles peuvent aussi faire une grande distance grâce à un transport. On nomme débarquement le fait de sortir des tanks ou des soldats du transporteur vers la terre ferme.

2.3.3 Troupes aériennes

Les troupes les plus utiles, mais aussi les plus dispendieuses, sont les troupes aériennes. Elles sont communément appelées « les avions ». Elles permettent d'attaquer sur mer comme sur terre.

Les avions peuvent et doivent se déplacer dans la phase de déplacement défensif. Les avions doivent avoir assez de mouvement pour se rendre dans la zone de combat et, ensuite, pouvoir atterrir sur un territoire allié.

Par exemple, un chasseur qui peut se déplacer sur quatre cases se trouve dans un territoire A. Il veut attaquer le territoire B, qui se situe à trois cases. Il pourra le faire seulement s'il y a un territoire allié à une case du territoire B, puisqu'il ne peut se poser en terrain hostile.

- Chasseur

Le chasseur est un petit avion de combat, il coûte 12 IPC, se défend à quatre et attaque à trois. C'est un outil pratique, puisqu'il attaque et se défend bien, c'est une pièce clé du jeu. Il peut se déplacer sur quatre cases.

- **Bombardier**

Le bombardier est un gros outil de guerre, il coûte 16 IPC, il attaque à quatre, mais il se défend à un. Il peut se déplacer sur six cases.

Lors de la phase d'attaque, le bombardier peut décider de ne pas faire partie de la bataille si le territoire ennemi comporte une industrie. À la place, il fera du bombardement stratégique. Dans ce cas, le joueur lancera un dé, et le joueur adverse qui contrôle l'industrie perdra ce montant en IPC.

2.3.4 Troupes neutres

Les troupes neutres ne sont contrôlées par personne. Le joueur qui contrôle le territoire sur lequel elles se trouvent peut en faire usage.

- **Anti-aérien**

L'anti-aérien permet de descendre un avion avant l'attaque sur le territoire. Le joueur doit lancer un 1 pour que l'anti-aérien touche sa cible.

Quand un joueur contrôle l'anti-aérien, il peut décider de le déplacer ; il se déplace de un sur le territoire.

- **Industrie**

Les joueurs ne peuvent placer de nouvelles unités que sur les territoires qui ont une industrie.

2.4 Exemple d'un tour de jeu

Toute cette description est un peu lourde, conséquence du fait que les règles du jeu sont complexes. Dans le but d'alléger la compréhension, voici l'exemple d'un tour de jeu joué par un humain.

1. Le joueur choisit les unités qu'il souhaite acheter ce tour-ci.
2. Le joueur choisit quelles unités feront un mouvement offensif.
3. Le joueur lance les dés pour chaque territoire qu'il a attaqué. Le joueur défendant le territoire lance les dés pour les unités défensives. Cette phase dure jusqu'à ce que le combat soit gagné par le défendeur ou l'attaquant.
4. Le joueur choisi quelles unités feront un mouvement défensif.
5. Le joueur place les unités qu'il a achetées au début du tour sur les territoires de son choix parmi ceux qui possèdent une industrie.
6. Le joueur collecte l'argent en fonction des territoires qu'il contrôle. Cet argent lui servira à la première phase de son prochain tour de jeu.

2.5 Conclusion sur les règles

Il y a quelques éléments techniques dans les règles que les joueurs doivent connaître pour jouer sans faire de faute, mais en ce qui concerne la présentation de ce mémoire, l'explication des règles est ici suffisante pour comprendre l'enjeu qu'est la programmation d'un système d'intelligence artificielle pour ce jeu de société.

CHAPITRE III – LE RÔLE DE L'IA POUR LE JEU D'AXIS & ALLIES

Peu importe la méthode d'intelligence artificielle qui sera utilisée pour les solutionner, plusieurs problèmes restent communs. Dans ce chapitre il sera question du hasard dans le jeu, de l'absence de caractéristiques sociales dans le système d'intelligence artificielle, de la complexité du plateau de jeu ainsi que des questions à répondre pour un tour de jeu de l'intelligence artificielle.

3.1 Hasard

Ce jeu comporte des lancements de dés fréquents et en grand nombre. Il est donc nécessaire de se demander comment le hasard sera géré dans le système d'IA.

Pour Axis & Allies, il n'est pas question de pur hasard. Il est possible et important de manipuler le hasard. Tel que mentionné précédemment, les pièces se comportent différemment en fonction de leurs actions, et cela a un grand impact sur le hasard. Par exemple, une infanterie réussit son attaque si le joueur qui la contrôle obtient un six sur un dé à six faces. Mais quand cette même unité se défend, elle réussit à repousser une unité ennemie avec deux ou moins sur un dé à six faces. Dans le même ordre d'idées, un bombardier qui attaque une cible réussit son attaque si le joueur lance quatre ou moins sur un dé, mais il réussit sa défense seulement s'il lance un six sur le dé. Même s'il est question de dés et de hasard, on peut donc optimiser nos chances en utilisant les bonnes pièces en attaque et les bonnes pièces en défense, ce qui est une des finesses de ce jeu.

De plus, une fois que la bonne stratégie est utilisée, il est possible de prévoir statistiquement ce qui devrait se passer. Il est possible d'affirmer si le dé n'est pas truqué qu'une attaque d'un soldat a une chance sur six de réussir tandis que l'attaque d'un tank, une

chance sur deux. Il est donc logique que si on évalue l'attaque de deux soldats et de deux tanks, on peut dire que si les probabilités sont les suivantes de réussir des attaques :

- 2 X une chance sur six (pour les deux soldats)

et

- 2 X une chance sur deux (pour les tanks)

Dans le meilleur scénario, un lancé de dé parfait, quatre attaques seront réussies, dans le pire aucune attaque ne sera réussie. De la même manière on peut savoir statistiquement les chances de succès du défendeur. Il est question ici de probabilités mathématiques aisément calculables.

De plus, les joueurs ont à leur disposition un nombre important de troupes et les troupes attaquent ou se défendent plusieurs fois dans un déplacement. Cela entraîne évidemment qu'un bon nombre de dés sont lancés. Selon la loi des grands nombres, plus il y a de dés lancés, plus les résultats des lancers de dés devraient se normaliser. Donc même si un tour donner un joueur est chanceux, sur la totalité de la partie après plusieurs centaines de lancers de dés, nos prévisions basées sur des probabilités permettront de bien évaluer la situation.¹⁶

Tous les mouvements qui seront effectués se baseront donc sur des évaluations statistiques de ce qui devrait se produire. Pour déterminer si le système d'IA peut effectuer une attaque pour toutes les solutions, nous avons besoin de savoir ce qui résulterait après une attaque. C'est à ce moment qu'une évaluation statistique de l'attaque sera utilisée.

3.2 Non social

Un autre aspect à prendre en considération est la tromperie : dans tous les jeux de plateau qui se jouent à plusieurs joueurs, il est possible de tenter d'influencer les autres joueurs en leur donnant des indices ou des propositions de plan qui ultimement vous

¹⁶ Henk, Tijms. 2007. *Understanding Probability: Chance Rules in Everyday Life*, Cambridge University Press.

permettront de gagner. Le système qui sera analysé ne sera pas analysé pour n'importe quel jeu de plateau, mais bien pour Axis & Allies. Évidemment, les jeux de plateau sont très différents les uns des autres. Si l'on pense à Monopoly, par exemple, il est question d'un jeu dans lequel l'aptitude sociale est importante. D'un autre côté, si l'on pense au Scrabble, il est nécessaire de connaître beaucoup de mots et de comprendre les subtilités du plateau, tandis qu'un jeu comme « Les colons de Catane » exige une bonne compréhension du système mathématique mais également des capacités sociales d'influencer les autres joueurs.

Dans le cas des trois jeux énoncés ci-dessus, le Scrabble est celui dans lequel un ordinateur aura le plus de chance d'exceller, parce qu'il n'exige aucune, en tout cas très peu d'aptitudes sociales.

Axis & Allies est un jeu qui demande des aptitudes d'analyse du plateau de jeu et aucune aptitude sociale. Les alliances étant déterminées dès le début et le jeu se jouant en deux équipes, il n'est donc pas possible de manipuler une tierce personne. Les aptitudes sociales complexes sont encore de l'ordre de la fiction pour les ordinateurs. Il est difficile d'imaginer qu'une machine essaie d'influencer votre façon de jouer sans que cela ne soit évident, tout en étant efficace.

3.3 La complexité du plateau de jeu

Le plateau de jeu d'Axis & Allies comporte 122 cases. Ces cases sont organisées sous la forme d'un graphe complexe. Chaque case peut être reliée de une à huit autres cases. Dans ces cases, il peut y avoir un nombre illimité d'unités de combat. La complexité mathématique nécessaire pour représenter ce jeu dépasse la complexité du jeu de Go, mentionné précédemment, ce qui causera des difficultés.

Sur les 122 cases, il n'y a pas de limites sur le nombre de pièces qui peuvent être présentes, ce qui, mathématiquement parlant, propose un nombre difficilement prévisible et très grand de possibilités de représentations des pièces.

3.4 Les questions à répondre face au jeu

Une difficulté majeure dans la fabrication d'un système d'intelligence artificielle qui joue à Axis & Allies est la complexité d'un tour de jeu. En effet, la majorité des jeux étudiés jusqu'à présent comportent des tours de jeu simples. Si l'on pense aux échecs, durant un tour de jeu, un joueur doit choisir quelle pièce bouger et sur quelle case la placer. Ce n'est pas aussi simple dans le cas d'Axis & Allies.

Premièrement, il y a cinq phases de jeu dans Axis & Allies, donc cinq décisions d'un ordre différent à prendre. Commençons par l'analyse de chaque phase.

- **Phase 1 : Achat de nouvelles unités**

Le système d'IA devra choisir comment dépenser son argent. Pour chacune des 10 unités, le système devra donc choisir combien il désire acheter d'unités.

- **Phase 2 : Déplacement offensif**

Le système devra choisir quelles unités militaires déplacer et dans quelle direction. Ici seulement cette décision est nettement plus complexe que celle qui doit être prise pour les échecs. Dans cette phase, le système doit choisir pour chaque unité contrôlée si elle doit bouger et sur quel territoire. La difficulté consiste en ce qu'il n'est pas question de bouger une seule unité, toutes les unités sont sujettes à bouger.

- **Phase 3 : Combat**

Le système n'aura rien à faire dans cette phase.

- **Phase 4 : Déplacement défensif**

Le système devra décider si les unités militaires n'ayant pas bougé doivent être déplacées après la phase de combat. Encore une fois, toutes les unités n'ayant pas bougé sont sujettes à être déplacées.

- **Phase 5 : Placement de nouvelles unités**

Le système doit placer les unités militaires achetées au début du tour. Il faut choisir sur quel territoire chaque unité sera placée.

Il faut comprendre que le système devra répondre aux quatre questions suivantes à chaque tour de jeu, soit :

- Quoi acheter ?
- Quelles unités militaires déplacer et dans quelle direction ?
- Comment se positionner pour la défense ?
- Où placer les nouvelles unités ?

Ces différentes réponses doivent aussi être prises les unes en fonctions des autres. Il faut que les troupes à l'attaque soient coordonnées avec les troupes en défense. Il est difficile d'imaginer de fragmenter la prise de décision en plusieurs sous-systèmes utilisant des modes de raisonnement différents.

Tout au long de l'analyse des différentes options possibles, nous porterons une attention particulière sur le fait que les systèmes doivent donner les quatre réponses.

CHAPITRE IV – LES SOLUTIONS CLASSIQUES POUR LE JEU AXIS & ALLIES

Dans ce qui suit, nous examinerons quatre solutions possibles pour un jeu tel que Axis & Allies. Les solutions qui seront analysées sont les suivantes :

- Le réseau de neurones ;
- Le système expert ;
- L'algorithme génétique ;
- L'algorithme Min-Max.

Nous analyserons les forces et les faiblesses ainsi que la faisabilité de chaque solution. Pour conclure ce chapitre, une solution sera adoptée, sa mise en oeuvre sera décrite au chapitre 5 et nous présenterons quelques résultats expérimentaux au chapitre 6.

4.1 Réseaux de neurones

Des réseaux de neurones sont utilisés pour la résolution de jeux de société. Si l'on prend le système TD-Gammon, il propose de bons résultats au jeu de Backgammon. Il a été réalisé en utilisant le principe des réseaux de neurones, le jeu de Backgammon est relativement petit et simple, comparativement au jeu Axis & Allies mais il comporte des lancements de dés comme Axis & Allies.

L'algorithme du réseau de neurones est un algorithme heuristique qui calque le comportement du cerveau humain pour trouver une réponse qui n'est pas nécessairement optimale, mais qui se veut viable et bonne.

Les réseaux de neurones formels sont à l'origine une tentative de modélisation mathématique du cerveau humain. Les premiers travaux datent de 1943 et sont l'œuvre de M.M. Mac Culloch et Pitts. Ils présentent un modèle assez simple pour les neurones

et explorent les possibilités de ce modèle.¹⁷

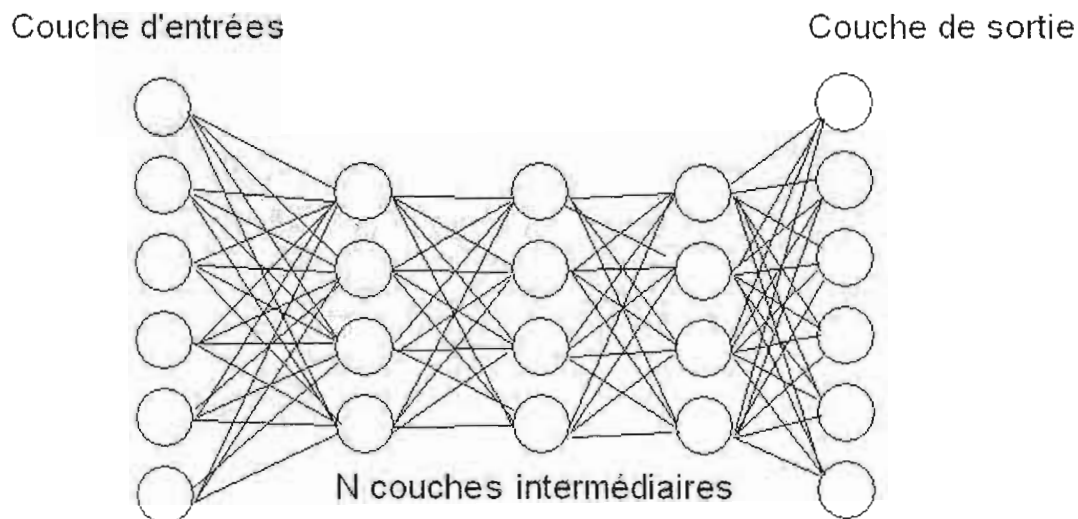


Figure 4.1 : Représentation d'un réseau de neurones.

L'idée principale des réseaux de neurones « modernes » est la suivante.

On se donne une unité simple, un neurone, qui est capable de réaliser quelques calculs élémentaires. On relie ensuite entre elles un nombre important de ces unités et on essaie de déterminer la puissance de calcul du réseau ainsi obtenu. Il est important de noter que ces neurones manipulent des données numériques et non symboliques.

Deux visions s'affrontent donc : d'un côté les tenants de la modélisation biologique, qui veulent respecter un certain nombre de contraintes liées à la nature du cerveau, et de l'autre,

¹⁷ Université de Sherbrooke. s.d. « Introduction aux réseaux de neurones », In GCH445 - Systèmes experts en génie chimique. En ligne. <<http://www.chimie.usherbrooke.ca/cours/gch445/neurones-intro.html>>. Consulté le 21 janvier 2009.

les tenants de la puissance de calcul, qui s'intéressent au modèle en lui-même, sans aucun lien avec la réalité biologique.

Pour représenter un réseau de neurones, il faut donc trouver une représentation des données sous forme de plusieurs nœuds, ce qui formera la couche d'entrée ainsi qu'une représentation des résultats. On peut voir la forme générale d'un tel réseau de neurones à la figure 4.1.

4.1.1 Implémentations

D'abord, afin d'évaluer si un réseau de neurones constitue une bonne option, il faut se rappeler qu'on doit répondre à quatre questions pour pouvoir jouer un tour de jeu.

Serait-il préférable de faire un seul réseau de neurones ou d'en faire deux, trois ou quatre? Avant de se poser cette question, analysons ce que devrait être un réseau de neurones devant répondre à une seule question. Nous choisirons de répondre à la question portant sur l'attaque. Une fois la complexité de cette question examinée, nous passerons à la question suivante.

La grande difficulté consistera à déterminer ce que seront les neurones de la couche d'entrée et celles de la couche de sortie.

4.1.2 La couche d'entrée

Ce que le système doit recevoir pour être en mesure de prendre une décision sur un tour de jeu, c'est la disposition du plateau de jeu. C'est là que le processus devient plus complexe. Le plateau de jeu se compose de 122 cases, et chaque case peut soutenir un nombre indéterminé de chaque unité. Il y a huit sortes d'unités, sans compter qu'il peut y avoir un anti-aérien ou une industrie. On a donc un total possible de 10 types d'unités. De plus, le système doit savoir qui contrôle le terrain.

Pour représenter un terrain, on parle de 11 nœuds d'entrée par pays, le tout représenté comme suit :

- Pour les huit premiers neurones, le chiffre qui sera propagé sera le nombre d'unités sur le plateau de jeu.
- Pour le 9^e et le 10^e neurone, le chiffre propagé sera zéro s'il n'y en a pas, et un s'il y a l'unité en question.
- Pour le 11^e neurone, il y a un chiffre de 1 à 5 représentant la nation qui contrôle le territoire.

La figure 4.2 représente un territoire dans un réseau de neurones.



Figure 4.2 : La couche d'entrée d'un réseau de neurones conçue pour Axis & Allies.

Axis & Allies contenant 122 territoires, on parle alors de 1 342 neurones d'entrée. Avec beaucoup de moyens et de longs processus d'apprentissage, il est encore possible de faire fonctionner un tel réseau de neurones, mais il ne s'agit pas d'une solution simple si l'on dispose de ressources limitées.

4.1.3 La couche de sortie

La couche de sortie doit représenter les mouvements des unités qui attaquent, donc pour chaque unité vers quel territoire elle se déplace et en passant par quel territoire intermédiaire si elle se déplace de plus d'une case. S'il s'agit d'un tank, il faut mentionner s'il attaque un territoire vide. S'il s'agit d'un navire de guerre, il faut dire si le navire bombarde les côtes, et si c'est un avion, il faut définir le territoire d'atterrissage.

La complexité des informations sur la couche de sortie est réelle et présente. Techniquement, pour chaque pièce, il faudrait cinq nœuds. Il suffit de savoir le nombre de pièces pour savoir la grandeur de la couche de sortie. Le problème devient plus compliqué, puisque dans Axis & Allies, il peut techniquement y avoir un nombre non borné de pièces, ce qui pourrait se gérer de deux manières. Pour la première manière, il s'agit de faire un nœud de sortie pour chaque type de déplacement de troupes et y associer un chiffre qui représente le nombre de troupes. Mais ceci n'est pas fonctionnel, étant donné la complexité du plateau de jeu d'Axis & Allies. L'autre solution est un peu de la triche. Le nombre de pièces dans le jeu est géré par le joueur. Il serait donc possible de proposer un nombre de pièces maximum, et le système d'IA joue en fonction de ne jamais dépasser ce nombre de troupes. Évidemment, cette technique pollue le jeu de l'IA par une donnée supplémentaire. Il serait donc possible que le système exécute une attaque suicide, seulement pour pouvoir acheter plus de troupes au prochain tour. Une estimation réaliste d'un nombre limite de troupes serait de l'ordre de 150. La couche de sortie comporterait alors 750 nœuds.

4.1.4 Limitations et faisabilité

Il n'est pas idéal d'avoir à limiter artificiellement le nombre de pièces, mais cela ne constitue pas le principal défaut. Le plus grand défaut de ce réseau de neurones est qu'il faut l'entraîner. Évidemment, l'entraînement d'un réseau de neurones se fait en fonction de la grandeur du réseau de neurones. Le réseau qu'il faudrait créer pour ce jeu est très grand par rapport à la complexité de la prise de décision. En effet, le problème est celui de devoir prendre en compte la totalité du plateau de jeu dans le réseau. C'est ce qui complexifie de beaucoup le système. Si l'on pouvait représenter seulement les pièces et non le plateau, le problème serait différent. Il faut aussi se rappeler que l'analyse faite jusqu'à présent ne porte que sur une seule des quatre questions que le système d'IA doit gérer.

De plus, il n'existe pas de banque de données facilement utilisable qui donnerait des mouvements et des indicateurs de succès, car pour entraîner le réseau de neurones, il faut lui injecter des coups des parties précédentes.

4.1.5 Conclusion

Même s'il est possible de créer un système d'IA utilisant cette méthode, il n'en demeure pas moins qu'il sera coûteux. Ainsi, dans le cas du jeu de Go, dont la complexité est moindre que celle d'Axis & Allies, les systèmes basés sur cette méthode ne se classent pas parmi les premiers.

4.2 Système expert

De manière simple, un système expert est un système dans lequel un expert propose sa connaissance sous forme de règles. Ces règles sont par la suite transformées en un système informatique. Le système met donc en application les règles de l'expert.

Une définition des systèmes experts est la suivante :

Un système expert est constitué par l'alliance d'un ordinateur et d'un logiciel conçu pour simuler le savoir-faire d'un spécialiste dans un domaine précis. L'organisation d'un système expert est caractérisée par une base de connaissances représentant l'expertise proprement dite (cette base ne peut être constituée qu'avec l'aide d'un spécialiste du domaine traité), par les algorithmes gérant cette connaissance (constituant le moteur d'inférences, ou interpréteur, ils sont généraux, c'est-à-dire indépendants du domaine), par une base de faits (ensemble de faits décrivant le problème précis que l'on propose au système de traiter)¹⁸.

Un système expert se compose donc de trois parties :

- Une base de faits ;
- Une base de règles ;
- Un moteur d'inférence.

4.2.1 Implémentations

Pour créer un système expert qui puisse jouer à Axis & Allies, il faudra donc implémenter chacune des parties. Commençons par discuter de la première partie : la base de faits.

- **Une base de faits**

La base de faits sera, dans notre contexte, la description de l'état du plateau de jeu. Il s'agira tout simplement de savoir où se situent les unités militaires sur le plateau de jeu. Pour

¹⁸ Cordier, Marie-Odile, et Rousset, Marie-Christine. "Système expert", in *Encyclopedia Universalis*, éd. 2007.

cela, il faudra construire un graphe des terrains. Chaque terrain sera relié aux autres avec lesquels il possède une frontière. Sur les terrains, il y aura l'information sur les troupes qui s'y trouvent ainsi que sur la nation qui le contrôle.

- **Une base de règles**

La base des règles forme le cœur du système. Dans ce cas, il s'agira de mettre en des termes logiques les règles qui permettent de jouer au jeu.

La première étape consiste à trouver un expert qui sache jouer au jeu et il faut analyser son comportement lors des parties. Par la suite, il sera possible de déterminer les règles qui régissent son comportement.

Dans ces règles, on trouvera évidemment des règles qui dicteront les déplacements des troupes pour jouer, mais également des règles d'ouverture classique ainsi que des règles de fin de partie, puisqu'il y a habituellement une différence dans le style de jeu.

- **Un moteur d'inférence**

Une fois que les règles seront déterminées, il faudra mettre sur pied un moteur d'inférence capable de les ordonner et de les mettre en application.

Le système expert semble un processus simple, mais en fait, il s'agit de copier le comportement d'un expert. Cependant, cela ne nécessite pas de calculer toutes les possibilités comme pour d'autres algorithmes, ni de jongler avec plusieurs réponses comme dans les autres systèmes.

4.2.2 Limitations et faisabilité

Un des avantages du système expert est de répliquer le comportement de l'expert. Si notre expert est un bon joueur, le système qui en découlera a de bonnes chances d'être un bon

système. De plus, contrairement à d'autres systèmes, le système expert ne nécessite pas d'entraînement. Une fois terminé, il fonctionne.

Le principal avantage du système expert est qu'il ne nécessite pas une grande puissance de calcul. En fait, il est relativement économique, étant donné que l'algorithme, comme un humain le ferait, ne calcule pas toutes les possibilités. On voit d'ailleurs que les systèmes experts jouant au jeu de Go roulent sur des machines modestes en comparaison à leurs rivaux utilisant des algorithmes basés sur la puissance de calcul des superordinateurs.

C'est d'ailleurs le principal avantage de cette approche en ce qui concerne Axis & Allies. Étant donné la complexité du plateau de jeu, les algorithmes traditionnels qui calculent toutes les possibilités sont ralentis par la complexité du plateau de jeu. La complexité combinatoire d'Axis & Allies étant beaucoup plus complexe que celle du jeu de Go, il faudra attendre de meilleurs ordinateurs pour obtenir de meilleurs résultats avec les autres algorithmes.

Le concept de « système expert » peut sembler simple quand il est présenté de cette manière. La plus grande difficulté du système expert est de présenter les connaissances de l'expert en règles informatiques. D'abord, il faudra avoir un expert qui connaît bien le domaine et il faut qu'il soit capable de les expliquer au programmeur. C'est d'ailleurs le principal problème de cette approche. Il faut avoir un expert sous la main pour être capable d'utiliser cette approche.

4.2.3 Conclusion

Étant donné la complexité combinatoire du jeu, cette approche a plus de chances d'être performante. En effet, les systèmes basés sur les connaissances ont un meilleur rendement que les autres méthodes quand elles sont limitées par la puissance de calcul¹⁹. Un des grands défauts de cette approche apparaît quand il faut trouver un expert qui peut donner les règles à implémenter. Il faudra que l'expert ait une facilité de communiquer ses

¹⁹ Sharma, Karan. 2007. "Designing Knowledge Based Systems as Complex Adaptive Systems", Athens, Artificial Intelligence Center, University of Georgia.

connaissances. Dans le cas de plusieurs programmes, comme gnu go, l'expert est le programmeur ce qui simplifie de beaucoup la solution.

Il est également important de mentionner que jouer au jeu d'Axis & Allies n'est pas aussi complexe que jouer au jeu de Go. Le jeu d'Axis & Allies a une complexité combinatoire énorme, mais les bons coups qu'il faut faire pour gagner sont souvent les mêmes. Ainsi, en créant un système basé sur les connaissances, il est possible, comme le font les joueurs humains, d'éviter d'avoir à explorer tout l'espace de jeu.

4.3 Algorithme génétique

Les algorithmes génétiques sont une méta-heuristique qui se basent sur le principe de l'évolution. On propose des individus, donc des réponses. En leur faisant subir des modifications, on ne garde que les meilleurs afin de simuler ainsi un processus de sélection naturelle. Cette méta-heuristique se situe dans la famille des solutions qu'on nomme « méthode Monté Carlo ». En voici une définition formelle :

Les algorithmes génétiques (A. G.) constituent un exemple représentatif d'un ensemble de méthodes connues sous le nom d'algorithmes évolutionnaires (A. E.). Ces méthodes servent à implanter sur des systèmes artificiels les mécanismes néo-darwiniens de l'évolution naturelle. Cette approche initiée dans les années 1970 par John Holland connaît depuis le début des années 1990 une forte croissance. Les A. E. sont des algorithmes itératifs fondés sur la notion de génération ; mais ils sont, également, par nature, hautement parallèles dans la mesure où ils simulent l'évolution de tout un ensemble de solutions. Concepts de base, un A. G. manipule des chaînes de symboles. Chaque chaîne représente une solution potentielle complète à un problème donné. De façon métaphorique, une chaîne est assimilée à un chromosome, la valeur d'un symbole à un allèle et l'ensemble des chaînes manipulées à une population. Le plus souvent, la population est de taille constante, les chromosomes sont de longueur fixe et l'alphabet des symboles est binaire. D'autres représentations sont également utilisées : structure arborescente, taille de chromosome variable, ou encore codage fondé sur les nombres réels.²⁰

²⁰

Collard, Phillipe. "Algorithmes Génétiques", in *Encyclopedia Universalis*, éd. 2007.

4.3.1 Implémentations

Le défi d'utiliser la méta-heuristique des algorithmes génétiques consiste à pouvoir trouver une représentation appropriée des données permettant d'exploiter cette approche.

Pour qu'un algorithme génétique fonctionne correctement, il doit pouvoir réaliser les étapes suivantes :

- Sélection (avec la fonction « Fitness ») ;
- Permutation ou « Cross Over » ;
- Mutation.

Dans le cas d'un système qui joue à Axis & Allies, les items qui seront manipulés par l'algorithme génétique seront :

- Des achats de nouvelles troupes ;
- Des déplacements d'unités ;
- Des déploiements de nouvelles unités.

On doit se rappeler que le système doit être capable de répondre à quatre questions. Devrait-on proposer d'exécuter la méta-heuristique quatre fois ou de l'exécuter seulement une fois, mais que la représentation des données propose les quatre réponses ? Puisque ces trois options sont intimement liées, il serait avantageux que le système prenne toutes les décisions en même temps, afin que les options soient cohérentes entre elles.

Une manière qui serait envisageable consisterait à segmenter les individus en trois parties distinctes. Chaque partie représenterait une partie des réponses qu'on attend pour pouvoir jouer au jeu. Ainsi, comme on peut le voir sur la figure 4.3, la première partie porterait sur les troupes à acheter, la deuxième partie sur les mouvements des troupes et la troisième partie sur le placement des troupes nouvelles.

Information sur les troupes à acheter.	Information sur les mouvements des troupes, offensifs et défensifs.	Information sur le placement des nouvelles troupes.
--	---	---

Figure 4.3 : Un individu dans un algorithme génétique conçu pour Axis & Allies.

Il est à noter que la troisième partie porterait sur les mouvements agressifs et défensifs. Étant donné le fait qu'une troupe ne peut se déplacer plus d'une fois dans un tour, il est clair que ces deux options peuvent être combinées dans cet algorithme.

Les phases de l'algorithme génétique

Sélections (avec la fonction « Fitness »)

En premier lieu, il faudra une fonction « Fitness ». Cette fonction ressemble beaucoup à la fonction d'évaluation de l'algorithme Min-Max que nous verrons plus loin. Le but de cette fonction est de sélectionner les éléments qui sont les plus prometteurs et qui devraient offrir les meilleurs résultats.

La fonction « Fitness » doit donc évaluer ces réponses et en donner une estimation. Pour ce faire, si l'on choisit d'englober tous les éléments d'un tour de jeu dans un individu, la fonction « Fitness » devra évaluer si l'individu réagit selon les critères suivants :

- Les mouvements défensifs permettent-ils une bonne défense contre l'adversaire (en incluant l'achat et le positionnement des nouvelles troupes comme un déplacement) ?
- Les mouvements offensifs ont-ils fait avancer le joueur ?

Les candidats ayant les meilleures valeurs de « fitness » seront retenus, tandis que les autres seront rejetés.

Permutation ou « Cross Over »

Pour que la recherche progresse, il faudra générer de nouvelles réponses. Les candidats les plus prometteurs seront combinés entre eux. Ainsi, de nouvelles réponses seront créées. Si ces réponses s'avèrent meilleures que les anciennes, elles seront conservées dans la première phase.

Étant donné le format des données, il est impossible d'échanger une partie de l'information qui se trouve dans la partie « achat des nouvelles troupes » contre une partie de l'information qui se trouve dans la partie « déplacements ». Lors des permutations d'un élément contre un autre, il faudra faire des permutations seulement d'une partie de l'information avec la même partie de l'autre élément.

Mutation

Pour permettre les recherches dans diverses directions, on fera muter certains individus. Un léger changement sera apporté à certains d'entre eux. Il est question de changer un « bit » à un des individus.

4.3.2 Limitations et faisabilité

L'algorithme génétique permet d'encoder seulement les unités militaires, sans avoir à explorer l'énorme espace du plateau de jeu. Par exemple, si au début du tour il y a x unités à déplacer, la section « troupes à déplacer » de l'individu contiendra l'information sur x troupes. Cette section peut être extensible, elle s'adaptera en fonction du nombre de troupes. Les données sur le plateau de jeu, qui elles ne changent pas, ne feront par contre jamais partie de l'individu qui représente le mouvement des troupes. C'est ce qui fait la force de cet algorithme, qui fonctionne de mieux en mieux dans les jeux complexes comme le jeu de Go.

Un problème majeur demeure les mouvements spéciaux de certaines pièces, étant donné la possibilité de certaines pièces d'effectuer des mouvements plus complexes, comme les soldats qui réalisent un débarquement via un traversier ou un tank qui conquiert deux

territoires à la fois. Ces mouvements spéciaux requièrent une complexification de la représentation des données.

Cependant, le problème majeur, et une des différences entre le jeu de Go et Axis & Allies est la complexité combinatoire d'un tour de jeu. La complexité du plateau de jeu d'Axis est peut-être un peu moindre que celle du jeu de Go, mais la complexité du nombre de mouvements possibles n'est tout simplement pas du même ordre. Il a été mentionné ci-dessus que les échecs ont un ordre de 40 possibilités de coups jouables, tandis que pour le jeu de Go, il y a 300 possibilités de coups. Dans Axis & Allies, seulement pour la phase de combat, donc pour une des quatre questions, il y a, au premier tour de jeu, une possibilité de plus de 2000 coups. Le problème vient du fait qu'il est possible de jouer toutes les pièces à chaque tour.

4.3.3 Conclusion

Étant donné l'ampleur du nombre de décisions à prendre, l'efficacité des algorithmes génétique risque de n'être pas appropriée. Le temps de calcul nécessaire pour parvenir à une bonne solution peut être long. Par contre, si on restreignait le temps de calcul, la réponse pourrait être de piètre qualité.

4.4 Algorithme Min-Max

L'algorithme Min-Max est un inévitable lorsqu'on s'attaque à un jeu de société. C'est un algorithme simple mais efficace si l'on possède la puissance de calcul nécessaire. L'algorithme Min-Max simule l'affrontement de deux joueurs sur plusieurs tours. Le fonctionnement est assez simple, puisqu'on examine toutes les options possibles. En voici une définition formelle:

« The Minimax algorithm computes the Minimax decision from the current state. It uses a simple recursive computation of the Minimax values of each successor state, directly implementing the defining equations. The recursion proceeds all the way down to the

leaves of the tree, and then Minimax values are backed up through the tree as the recursion unwinds²¹. »

Une représentation graphique de l'arbre de décisions est montrée à la figure 4.4.

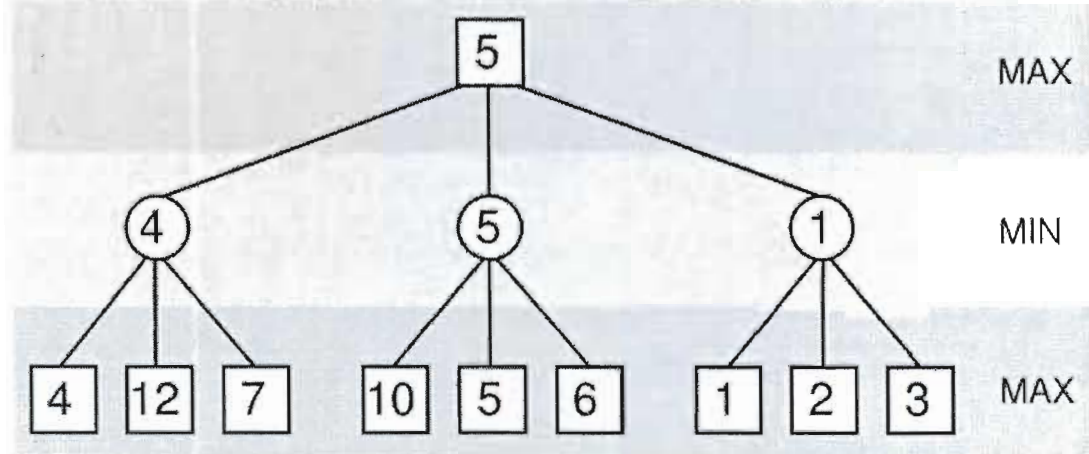


Figure 4.4 : Représentation de l'algorithme Min-Max.

4.4.1 Implémentations

Comme pour d'autres options, nous commencerons par analyser la complexité nécessaire pour répondre à la question du déplacement des troupes lors de la phase d'attaque.

Tel que mentionné ci-dessus, une étape cruciale au bon fonctionnement de l'algorithme Min-Max est la création d'une fonction d'évaluation. Cette fonction évalue l'état du jeu, à savoir si l'on s'est approché ou éloigné de la solution.

Pour la phase de mouvements de combats, il faudra agir comme suit. Puisque la manière de gagner à Axis & Allies consiste à conquérir une puissance de production industrielle en envahissant une grande quantité de pays, cette fonction heuristique, qui détermine l'état du jeu, devrait compter l'indice de production industrielle totale du joueur. Plus l'indice de production industrielle est grand, plus la partie va bon train.

²¹ Stuart Jonathan Russell, Peter Norvig, John F. Canny, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003, p. 165, 1080 p.

Il serait par contre trop simpliste de négliger la valeur des unités militaires présentes. Il faut acquérir des territoires afin d'augmenter son indice de production industrielle, soit, mais il faut également se garder des troupes pour se défendre. Il est donc préférable de calculer aussi la valeur totale des unités militaires d'un joueur après un coup.

La fonction heuristique devrait se baser sur ces deux facteurs. Il est difficile de dire lequel de ces deux facteurs est le plus important. Les troupes ont un coût en IPC, et les terrains ont un revenu en IPC. Donc, théoriquement, puisqu'il s'agit de la même mesure, on pourrait les mettre sur le même niveau. Mais, en pratique, il n'est pas évident que ce serait une bonne idée. Il faudrait pondérer les deux facteurs et effectuer des tests avec les pondérations pour pouvoir trouver la meilleure fonction heuristique. Il est important de comprendre que l'algorithme Min-Max repose sur la fonction heuristique, et que si celle-ci est erronée, l'algorithme ne pourra pas fonctionner correctement ou efficacement.

Une fois la fonction heuristique créée, de plus graves problèmes commencent. Traditionnellement, Min-Max a été créé pour un jeu à deux joueurs où Max est un des joueurs et Min l'autre. Dans le cas d'Axis & Allies, puisqu'il y a plusieurs joueurs présents, l'optique sera différente. Le jeu se joue à cinq joueurs, même s'il y a quand même deux alliances qui s'opposent.

Si l'IA joue seulement l'un des joueurs, disons l'Allemagne, par exemple, il faudra adapter l'algorithme Min-Max. Habituellement, Min-Max calcule en analysant quel serait le meilleur coup possible, supposant que son adversaire jouera pour gagner lui aussi. Dans ce cas, il devra jouer en fonction du fait que ses alliés joueront pour le mieux et que ses adversaires joueront pour lui gêner la vie. La figure 4.5 démontre visuellement ce concept.

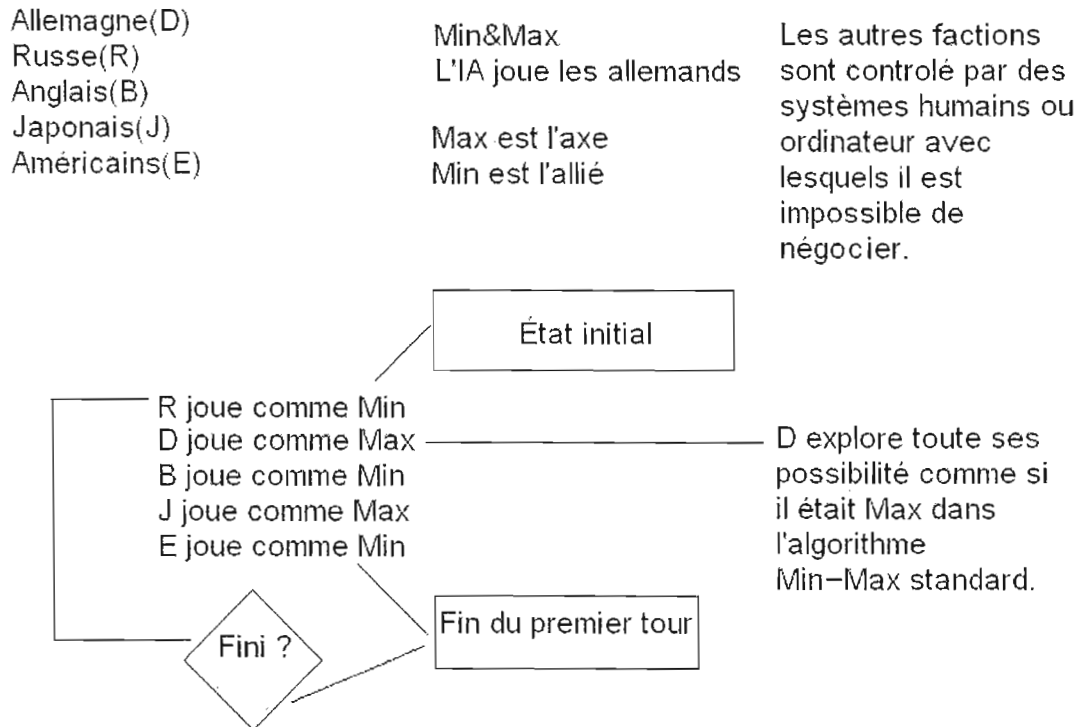


Figure 4.5 : Un tour de jeu pour l'algorithme Min-Max conçu pour Axis & Allies.

4.4.2 Limitations et faisabilité

Cet algorithme exige une puissance de calcul imposante, puisqu'il doit calculer les coups de tous les joueurs. Les algorithmes utilisant l'algorithme Min & Max doivent utiliser de superordinateurs pour obtenir un bon rendement. Mais si l'on doit modifier l'algorithme pour qu'il puisse fonctionner à cinq joueurs, celui-ci doit calculer les déplacements des cinq joueurs. Il se heurte donc à la fois à la complexité du plateau de jeu ainsi qu'à la complexité de la prise de décisions, et ce, cinq fois plutôt qu'une, dans un contexte où les décisions doivent être prises rapidement et avec des moyens limités. Cette approche ne semble donc pas réalisable.

4.4.3 Conclusion

Le problème majeur dans ce cas est l'explosion de l'arbre de prise de décisions. Si le jeu se jouait à deux, cette approche serait lourde, mais une fois cette approche adaptée à une réalité à cinq joueurs, elle n'est plus possible.

4.5 Choix du système

En se basant sur l'analyse de différentes solutions possibles, notre choix s'est arrêté sur le système expert. Ce qui motive le choix de cette méthode est la simplicité face aux autres méthodes. Puisque le jeu en question est un jeu avec une représentation complexe, il nous faudra une méthode légère. Dans le cas de cette mise en œuvre avec un système expert, une des difficultés est de faire communiquer l'expert et le concepteur du système. Cette difficulté n'en sera pas une dans le cas présent puisque le concepteur et l'expert seront la même personne.

Un autre facteur qui justifie ce choix est l'analyse des systèmes d'intelligence artificielle conçu pour le jeu de go ces quatre dernières années. Comme mentionné auparavant, sur les quatre derniers tournois de ICGA pour le jeu de Go, deux des quatre gagnants utilisaient un système à base de connaissances. Le jeu de Go étant le jeu classique avec la complexité la plus près de celle d'Axis & Allies, ceci confirme d'autant plus notre choix.

CHAPITRE V – GRANDES LIGNES DE LA MISE EN OEUVRE

Dans ce chapitre, il sera question de la mise en oeuvre du système. Les thèmes abordés seront l'architecture du programme, une vue d'ensemble de la mise en oeuvre du système expert, la vue d'ensemble de la représentation des données et la liste des règles de la base de connaissances.

5.1 Architecture du programme

L'algorithme choisi est celui du « système à base de connaissances ». La mise en oeuvre de ce « *Knowledge Based System* » a été réalisée sous la forme d'un système expert.

Peu importe la solution choisie, la création d'un système d'intelligence artificielle n'est pas un processus simple. Il a été choisi dans ce travail de focaliser sur un seul des joueurs. Donc, faute de plus de moyens, il a été décidé que le système ne jouerait que l'Allemagne pour cette phase. Suite à ce, nous verrons si le système donne des résultats intéressants et si nous devons continuer avec cette approche.

Nous avons choisi d'effectuer l'interface de communication avec le client en mode Web pour faciliter la distribution du programme. Cette partie de la solution utilise une architecture Web basée sur la technologie Struts. Le reste de la solution est codée en langage Java et peut être interprétée sur n'importe quelle machine.

Le système est donc un système qui, d'un point de vue structure, est plutôt simple. Comme nous pouvons le voir sur la figure 5.1, l'utilisateur peut effectuer deux actions, il peut demander au système de jouer l'Allemagne ou de modifier les valeurs d'un territoire.

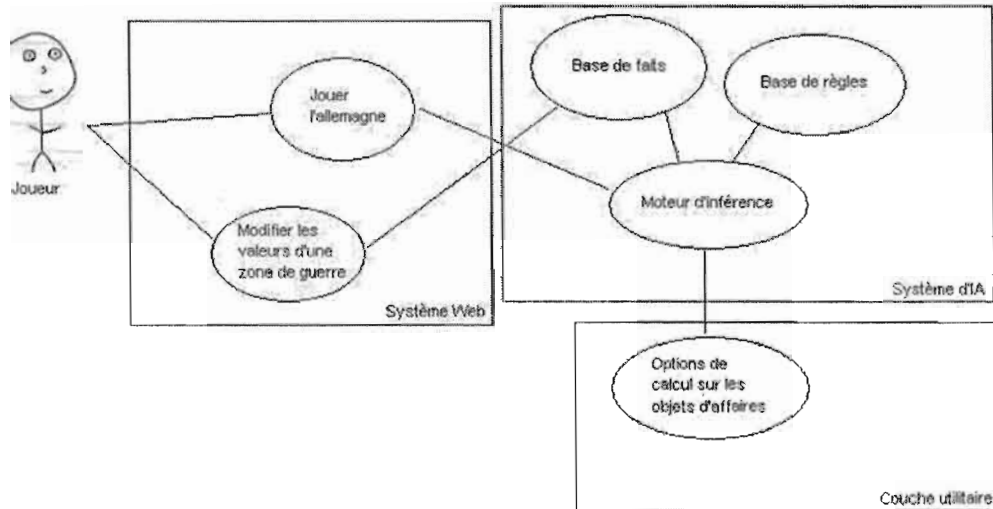


Figure 5.1 : Représentation de haut niveau du système d'IA.

Le système est séparé en quatre parties ayant des complexités bien différentes.

5.1.1 Le système Web

Le système Web gère les choix de l'utilisateur. Il permet au joueur de mettre à jour les données du plateau de jeu pour représenter les coups joués par les autres joueurs. Les données sont stockées dans la base de faits.

Il s'agit de fichiers Jsp et de fichiers Java qui sont exécutés sur le serveur au moment de l'exécution du programme. Cette partie fonctionne avec la bibliothèque Struts. Il est question de tâches communes pour des systèmes transactionnels Web. L'interface web gère aussi la langue d'affichage en fonction de la langue du navigateur. La langue par défaut est l'anglais.

5.1.2 Le système d'IA

Le système étant créé sur le modèle d'un système expert, il faudra y retrouver la base de faits, la base de règles et le moteur d'inférence.

Base de faits

La base de faits est la représentation du plateau de jeu. Dans le cas de ce système d'intelligence artificielle, il est question de représenter la position des pièces sur le plateau de jeu. Cette partie sera discutée plus longuement dans la section de la représentation des données.

Base de règles

La base de règles contient les règles que le système exécute pour trouver le coup à jouer. La liste de règles a été déduite des comportements de l'expert dans diverses parties. Dans ce cas, l'expert étant le programmeur, il a joué plusieurs parties contre le système d'IA de Hasbro ainsi que contre des adversaires réels. Il a aussi observé d'autres joueurs humains afin d'obtenir une information plus juste. L'expert a pris des notes sur les tours de jeu qui ont été les plus efficaces en termes de croissance vers la victoire pour étudier les stratégies qui étaient utilisées.

Suite à l'observation des résultats, plusieurs mouvements clés se sont montrés importants pour mener à la victoire. Ces mouvements ont donc été encodés sous forme de règles. Des stratégies plus vagues, comme des objectifs à réaliser dans certaines circonstances, ont également ressorti des analyses. Par exemple, l'Allemagne a avantage à couler les transports de la flotte anglaise, mais sans plus, et seulement si elle ne perd pas d'avion dans le processus. Ce schème de pensée a aussi été encodé dans une règle. Pour terminer, un comportement général des pièces a aussi été encodé sous forme de règles. Le but consiste à gérer les pièces qui n'ont pas été touchées par les règles des situations.

Une liste plus détaillée des règles est disponible un peu plus bas dans la section sur les règles.

Moteur d'inférence

Le moteur d'inférence va chercher les données dans la base de faits et il exécute les règles sur la plaque de jeu. Le moteur d'inférence utilise un système de priorités de règles pour qu'elles soient exécutées dans l'ordre requis. Chaque fois qu'une règle confirme un mouvement, le moteur d'inférence enlève la pièce afin d'éviter de faire bouger la pièce plus d'une fois.

Pour évaluer les règles, le moteur d'inférence utilise des objets spécialisés qui servent aux différents calculs nécessaires pour évaluer si les conditions des règles sont vérifiées. Ces objets spécialisés se trouvent dans la couche des objets utilitaires.

5.1.3 La couche des objets utilitaires

Les objets utilitaires permettent au moteur d'inférence d'évaluer les règles en fonction de la base de faits. Ces classes ont été séparées du système d'IA parce qu'elles exécutent des calculs qui pourraient être utiles dans n'importe quel type de système d'IA. Nous expliquerons les deux classes les plus importantes, qui permettent de calculer des réalités sur la base de faits.

- Combat Calculator

Cet objet permet d'évaluer le résultat statistique d'un combat. Le résultat reçu est retourné sous une forme qualitative. Les mêmes mots ont été utilisés pour représenter les réalités de l'expert lors de la création des règles. Les résultats possibles sont les suivants :

- « Blitz » ;
- « Over Kill » ;
- « Sans Perte » ;
- « Bien » ;
- « Correct » ;

- « Risqué » ;
- « Bien sans Prendre le Territoire » ;
- « Mal ».

L'objectif est de représenter les données dans les termes qu'utilise l'expert lors de sa prise de décision, ce qui permettra aussi à l'expert d'être plus indépendant et de pouvoir les utiliser dans un jeu connexe, comme Axis & Allies Europe, qui utilise les mêmes règles mais avec d'autres pièces.

- Distance Calculator

Distance calculator est l'objet qui évalue la distance ainsi que les chemins possibles entre les différents terrains. Ainsi, pour que l'algorithme puisse jouer à un jeu semblable mais avec un territoire différent, il suffit de changer le comportement de cet objet.

5.1.4 Les objets d'affaires

Les objets d'affaires sont les objets qui servent à représenter les données nécessaires au bon fonctionnement du jeu. On pense à l'objet « Zone De Guerre » qui représente un terrain et ce qui se trouve sur lui. On pense aussi à la classe « AxisBoardData », qui représente un plateau de jeu d'Axis & Allies. Ce sont les objets que les trois parties du diagramme 5.1 s'échangent entre elles pour se communiquer de l'information.

5.2 Les écrans

L'écran de base du jeu est tel que démontré à la figure 5.2.



Figure 5.2 : Écran de base du jeu d'axis & allies (version française).

Pour modifier la valeur d'un des pays, le joueur doit cliquer une fois sur le pays de son choix. Comme exemple, en cliquant sur l'Égypte, la fenêtre de la figure 5.3 apparaît.

http://localhost:8080/these/l - Google

http://ar...3002.pdf Roderick...CKENZIE Apple

Égypte

Le terrain Egy à été sélectionné : AKA Egy.
Ce territoire est contrôlé par :

C'est une zone terrestre

Le nombre d'infanterie

Le nombre de tank

Le nombre de Chasseur

Le nombre de bombardier

Figure 5.3 : Écran de modification d'une valeur (version française).

Le joueur peut changer les valeurs pour refléter la réalité du plateau de jeu. Quand il pèse sur « soumettre », les valeurs sont enregistrées dans la base de faits. Ensuite, le joueur retourne à la page principale (voir la figure 5.2).

Pour connaître les mouvements générés par le système d'IA, le joueur appuie sur le bouton « Jouer IA ». Une fenêtre avec les mouvements du système d'IA apparaît. La liste des mouvements étant normalement assez longue, nous présenterons trois écrans différents.

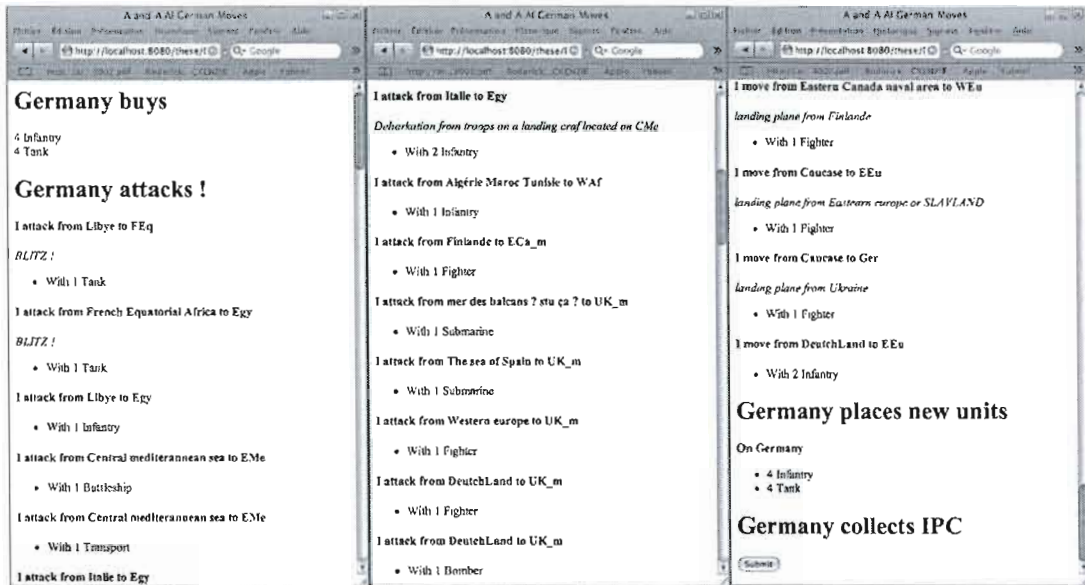


Figure 5.4 : Action exécutée par le système d'IA lors d'un tour de jeu (version anglaise).

Il s'agit donc d'une liste en mode texte des mouvements réalisés par le système d'IA. Quand un mouvement utilise une capacité spéciale propre à une pièce, elle est mentionnée avant le mouvement, par exemple, dans la figure 5.4, on voit le mot « Blitz », qui réfère à la capacité d'un tank d'attaquer plus d'un territoire lors d'un tour donné.

5.3 La représentation des données

Les données qui sont conservées sont les données du plateau de jeu, donc les pièces et les territoires. Le jeu d'Axis & Allies comporte 122 cases qui sont reliées les unes avec les autres en formant un graphe complexe. Les pièces, quant à elles, se trouvent sur le plateau de jeu. Pour représenter la réalité du plateau de jeu, il suffit de créer un graphe orienté, dans lequel les sommets seront les territoires du jeu, et les arêtes sont tracées entre deux sommets quand ses deux territoires sont adjacents sur le plateau de jeu.

5.3.1 Les sommets

Les sommets du graphe, les territoires ou les pays, devront comprendre les informations suivantes :

- Le nom du pays ;
- L'identificateur unique du territoire ;
- La puissance qui contrôle le territoire ;
- Le territoire maritime ou terrestre ;
- La présence d'un anti-aérien ;
- La présence d'une industrie ;
- Le nombre d'infanteries ;
- Le nombre de chars d'assaut ;
- Le nombre de chasseurs ;
- Le nombre de bombardiers ;
- Le nombre de transports ;
- Le nombre de navires de guerre ;
- Le nombre de transports ;
- Le nombre de sous-marins ;
- Le nombre de porte-avions ;

Il faudrait aussi savoir à quels autres sommets ce sommet est adjacent, donc savoir quels autres territoires sont adjacents.

L'identifiant unique de chaque pays se base sur la carte du jeu qu'on peut voir à la figure 5.5. Quand un territoire marin et un territoire terrestre portent le même identifiant, le territoire marin utilise le même identifiant concaténé de « _m » à la fin.

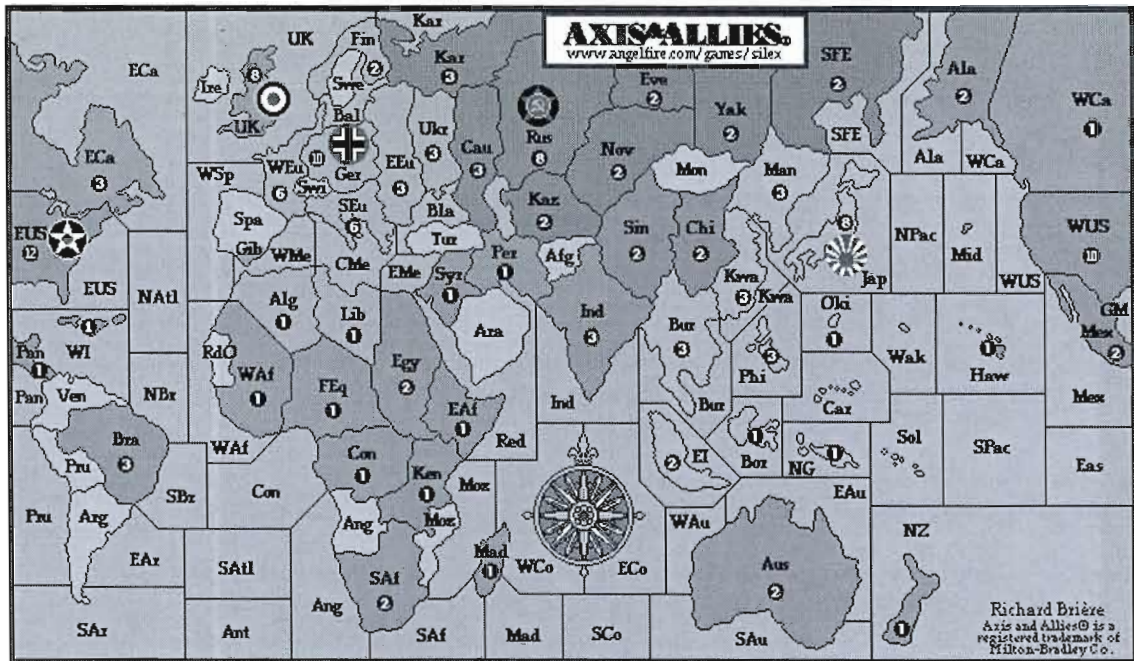


Figure 5.5 : Carte de la nomenclature des terrains.

5.3.2 Les arêtes

Les troupes terrestres ne pouvant se déployer que d'un territoire terrestre à un autre et les troupes marines ne pouvant se déployer que d'un territoire maritime à un autre, pour les déplacements, il est souvent nécessaire de savoir si le terrain adjacent est terrestre. C'est un peu comme si le graphe était pondéré ; les arêtes allant vers un sommet auront la valeur de zéro quand le terrain sera terrestre et de un quand le terrain sera maritime. Cela permet de ne pas avoir à aller chercher les objets seulement pour évaluer si c'est un terrain maritime ou terrestre.

Exemple

Afin de clarifier ce dont il est question voici un exemple comprenant une portion du plateau de jeu. L'exemple représente la portion du plateau de jeu figurant à la figure 5.6.



Figure 5.6 : Exemple d'une portion de la carte.

Il y donc cinq territoires à représenter dans ce graphe. Le graphe qui en résulterait serait donc le graphe représenté à la figure 5.7.

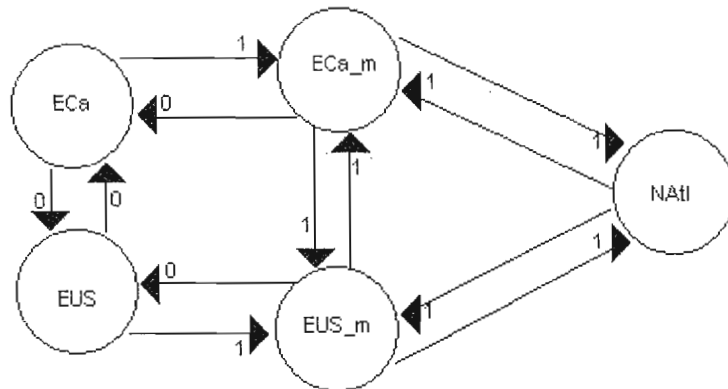


Figure 5.7 : Représentation logique de la figure 5.6.

Pour chaque sommet, les données mentionnées plus haut seront inscrites selon l'état d'avancement du jeu.

5.3.3 Mise en oeuvre

En pratique, pour ne pas avoir à charger tout le graphe en mémoire inutilement, les terrains sont stockés dans une table de hachage. Le hachage se fait sur l'identifiant unique. Donc, chaque sommet contient les identifiants uniques des terrains adjacents.

Les sommets sont représentés par l'objet « Zone De Guerre », qui représente un terrain. Dans cet objet figure les caractéristiques mentionnées plus haut ainsi qu'une liste chaînée d'objets « Type Adjacence ». Chacun de ces objets représente une arête dans le graphe.

5.4 Les règles du système expert

Dans cette section, nous ferons l'énumération des règles qui ont été implémentées pour mettre en oeuvre ce système expert. Nous présenterons les règles en fonction des quatre étapes où le système devra répondre à des questions. Donc, la phase d'achat de nouvelles troupes, la phase de déplacement offensif, la phase de déplacement pacifique et la phase de positionnement de nouvelles unités.

5.4.1 La phase d'achat de nouvelles troupes

Toutes les règles d'achat de troupes sont relativement simples, en voici la liste.

- Règle d'achat de tanks supplémentaires

Si le nombre d'IPC du joueur est plus grand que 40, il faut acheter le nombre de tanks selon la formule présentée ci-dessous avant d'exécuter les autres règles.

$$\frac{\text{IPC_du_joueur} - 40}{\text{Cout_du_tank}}$$

Explication :

Quand le joueur possède plus de 40 IPC, cela signifie qu'il a conquis les territoires qui sont adjacents. Il devient alors nécessaire d'avoir plus de tanks pour pouvoir rapidement attaquer son adversaire.

- Règle d'achat d'avions

Si le nombre d'IPC du joueur est plus grand que 50, 20 % des fois, il lui faudra acheter un avion selon la règle qui suit :

20 % des chances que le joueur achète un bombardier ;

80 % des chances que le joueur achète un chasseur.

Explication :

Les avions sont un luxe qu'un joueur ne peut se payer que s'il possède beaucoup d'argent.

- Règle d'achat d'un transport

Si :

1. le nombre de troupes contrôlées par l'Axe est plus petit que le nombre de troupes contrôlées par les alliés;
2. les Allemands ne contrôlent aucun transport dans la Méditerranée;
3. les alliées ne contrôlent pas la Méditerranée;

il faut acheter un transport.

Explication :

Il faut que l'Allemagne conquiert et conserve le territoire de l'Afrique. Mais il est toujours plus important d'attaquer en Europe en premier. Si le joueur se trouve en bonne posture financière, il achètera un transport si celui-ci peut survivre.

- Règle d'achat des troupes avec le résidu d'argent

Il faut acheter un nombre équivalent d'infanteries et de soldats. La différence entre le nombre de soldats peut-être ajustée de deux, afin de minimiser l'argent inutilisé ce tour-ci.

5.4.2 La phase des mouvements de combat

Les règles de la phase de mouvement des combats sont les plus compliquées. À ce jeu, la meilleure défense est l'attaque. Il est important d'essayer d'utiliser toutes les unités disponibles pour conquérir le plus de territoires possibles en un tour et ainsi obtenir plus d'argent au tour suivant.

Deux des règles du mouvement des combats sont complexes. Elles utilisent les mêmes méthodes de calcul, mais avec un système de priorité différent. Leur algorithme en pseudo-code sera présenté plus loin.

D'ailleurs, cette phase comporte des réponses préfabriquées. Au premier tour de jeu, l'Allemagne jouera avant l'Angleterre. Les troupes anglaises se trouveront toujours au même endroit. Les attaques de l'Angleterre seront toujours les mêmes en Afrique, lieu où se trouvent les troupes anglaises.

- Règles de l'Italie maritime

Lors du premier tour de jeu :

1. Les deux infanteries se trouvant sur le territoire de l'Italie emprunteront le transport dans le centre de la Méditerranée et réaliseront un débarquement en Égypte ;
2. La frégate se trouvant dans le centre de la Méditerranée ira soutenir l'attaque dans l'est de la Méditerranée.

- Règles de la Libye

Lors du premier tour de jeu :

1. Le tank situé sur la Libye attaquera la Guinée Française et ensuite l'Égypte ;
2. L'infanterie sur la Libye attaquera l'Égypte.

- Règles de l'Algérie

Lors du premier tour de jeu :

1. L'infanterie attaquera l'Afrique de l'Ouest.

- Règles des attaques sur les territoires maritimes

Cette règle est l'une des règles dites complexes.

- *L'objectif de la règle*

L'objectif de la règle consiste à détruire le plus de transports ennemis potentiellement dangereux sans subir de pertes aériennes.

- *Seront considérées comme sources :*

Tous les terrains contrôlés par les Allemands qui comportent un avion et tous les terrains maritimes sur lesquels les Allemands possèdent au moins une unité maritime qui n'est pas un transport.

- *Seront considérées comme cibles :*

Tous les terrains maritimes sur lesquels les Américains ou les Anglais possèdent un transport à distance capable d'effectuer un débarquement.

- *Une attaque sera considérée comme bonne si :*

Les forces attaquantes ont une force de frappe comptant pour le double de celle des défenseurs et, statistiquement, le nombre d'unités détruites chez l'attaquant est inférieur ou égal au nombre de bateaux engagés par l'attaquant.

- Le pseudo-code de haut niveau est le suivant :

Les lettres majuscules sont des ensembles, et les lettres minuscules sont des variables qui font partie de l'ensemble de la même lettre, s'il existe. Le pseudo-code va comme suit :

Pour toutes les sources Y (*les bateaux en priorité, ensuite les avions*)

Pour toutes les unités U de la source y

Pour toutes les cibles Z (*par ordre de cote de danger*)

Créer une attaque possible q vide.

Pour toutes les distances possibles X (*de un à la distance maximale*)

Pour chaque unité u où :

- u peut attaquer z ;
ET
- u a une distance de x avec la cible z .

Ajouter l'unité u à l'attaque possible q.

Évaluer l'attaque q sur la cible z.

Si l'attaque sur q est bonne, confirmer l'attaque.

- Règles des attaques terrestres

- *L'objectif de la règle*

L'objectif de la règle consiste à conquérir le plus de territoires possible en optimisant les chances statistiques de gagner dans chaque combat.

- *Seront considérées comme sources :*

Tous les terrains contrôlés par les Allemands qui comportent au moins une troupe terrestre ainsi que tous les avions qui n'auront pas bougé selon la règle des attaques dans les territoires maritimes.

- *Seront considérées comme cibles :*

Tous les terrains contrôlés par l'Axe qui sont adjacents à un terrain considéré comme source.

- *Une attaque sera considérée comme bonne si :*

Statistiquement parlant, les attaquants détruisent la totalité des troupes défensives, peu importe le nombre de tours. Au moins une troupe terrestre à l'attaque survit, et plus de 30 % des troupes à l'attaque survivent.

- *Le pseudo-code de haut niveau est le suivant :*

Encore une fois, les lettres majuscules constituent des ensembles, et les lettres minuscules sont des variables qui font partie de l'ensemble de la même lettre, s'il existe.

NombreEndroitTroupePeutAttaquer sera une variable qui, sans aucune surprise, désignera le nombre d'endroits où une troupe peut attaquer.

AttaquePossibles[z] désigne une attaque possible sur la cible *z* faisant partie des cibles *Z*. Au début de l'algorithme, chaque position du tableau contient une attaque vide. Le pseudo code va comme suit :

```

Pour tous les NombreEndroitTroupePeutAttaquer possible X
  (de un au nombre d'endroits maximum)
  Pour toutes les cibles Z (par ordre décroissant d'IPC)
    Pour toutes les sources Y
      Pour toutes les unités U de la source y
        Pour chaque unité u où :
          • u peut attaquer z ;
            ET
          • u.NombreEndroitTroupePeutAttaquer= x.

          Ajouter l'unité u à AttaquePossibles[z].
          Évaluer l'attaque AttaquePossibles[z].

          Si l'attaque AttaquePossibles[z] est bonne,
            confirmer l'attaque.

```

5.4.3 La phase des mouvements pacifiques

Cette phase contient des règles applicables au premier tour de jeu ainsi que des règles générales.

- Règles de l'Europe de l'Ouest

Lors du premier tour de jeu :

- Les deux tanks se déplacent vers l'Europe de l'Est s'ils sont toujours sous le contrôle des Allemands, sinon ils se déplacent en Allemagne.
- Deux infanteries sur le territoire de l'Allemagne se déplacent vers l'Europe de l'Ouest.

- Règles de la défense atlantique

À partir du deuxième tour de jeu :

Si l'Europe de l'Ouest peut-être attaquée par des débarquements ennemis, ne pas déplacer les unités.

- Règle de la défense de la capitale

Si l'Allemagne peut être envahie, ne pas déplacer les unités.

- Règle de la défense maritime de la Méditerranée

Si la Méditerranée est contrôlée par les alliés et qu'il y a des transports, ne pas déplacer les unités de l'Europe du Sud.

- Règle générale des mouvements pacifiques

Toutes les unités bougent sur le territoire le plus près de la Russie en fonction de leurs positions et de leurs règles de déplacement.

5.4.4 La phase des placements des nouvelles unités

- Règle de la défense de la capitale

Si l'Allemagne a peur d'être envahie, tout placer sur l'Allemagne.

- Règle de l'usurpation de la Karélie

Si le territoire de la Karélie se trouve sous contrôle allemand, utiliser l'usine qui s'y trouve pour y placer des infanteries et des tanks.

- Règle de l'usurpation de la Russie

Si le territoire de la Russie est sous contrôle allemand, utiliser l'usine qui s'y trouve pour y placer des infanteries et des tanks.

CHAPITRE VI – RÉSULTATS OBTENUS

Le but de ce chapitre est de présenter les résultats des tests qui ont été effectués. Les résultats des différents tests seront analysés.

6.1 Résultats obtenus

Le but est d'estimer si la méthode utilisée pour mettre en œuvre notre système pour jouer à Axis & Allies semble offrir de bons résultats. Un baromètre intéressant est la dernière version du système d'IA développée par Hasbro pour ce jeu qui se nomme Iron Blitz. Il s'agit donc de confronter Iron Blitz à ce système. L'idéal aurait été d'avoir accès au code source de l'application de Hasbro pour pouvoir automatiser les tests. Malheureusement, le code ainsi que les détails de son application restent inconnus. Puisqu'il s'agit d'un jeu de table, il y a toujours une façon de faire affronter les systèmes, qui est de faire l'interface en passant par l'humain. Pour effectuer les tests, une vraie partie d'Axis & Allies est jouée sur le plateau de jeu. Les mouvements d'un système seront joués sur le plateau, puis les mouvements de l'autre système seront joués à leur tour.

Le système créé ne modélise le comportement que d'un seul joueur, l'Allemagne, il faudra donc évaluer ce système avec les performances de ce joueur. Pour toutes les parties, le reste des joueurs est joué par le système d'Hasbro. Le système d'Hasbro possède un réglage pour chacun des joueurs, de trois à cinq étoiles. Le réglage est censé représenter l'efficacité et donc la performance du système. Nous laisserons ce réglage à cinq étoiles, pour donner toutes les chances au système d'Hasbro, Iron Blitz.

Les tests sont longs à opérer, puisqu'il faut faire jouer les ordinateurs manuellement. Nous avons choisi de ne faire jouer que les cinq premiers tours de jeu, afin de ne pas alourdir les données. Étant donné la nature relativement prévisible du jeu, il est possible de voir la progression du jeu aux cinq premiers tours de jeu. Il faut se rappeler qu'une partie se termine souvent en moins de cinq tours et que cinq tours de jeu représentent entre une et trois heures

de jeu selon l'expérience des joueurs. Cet arrêt au cinquième tour du jeu est surtout nécessaire pour les parties où Iron Blitz est le seul joueur puisque les joueurs étant de force plus ou moins égale, la partie peut durer un nombre de tours anormalement élevé.

Le jeu d'Axis & Allies n'est pas un jeu équilibré, c'est-à-dire que statistiquement, tous les joueurs n'ont pas la même chance de gagner au début du jeu. Pour être certain d'obtenir un test viable, la première étape est de voir comment se comporte le système d'Iron Blitz quand il joue contre lui-même. Nous avons exécuté un premier test où tous les joueurs sont exécutés par le système d'Iron Blitz. Le but est de voir comment se comporte l'Allemagne. Les résultats sont disponibles aux figures 6.1, 6.2, 6.3, 6.4 et 6.5. Des prises d'écrans sur les deux premières parties sont disponibles aux figures 6.11 et 6.12. Elles offrent plus de détails sur les résultats.

Ce qu'on constate, c'est que sur les cinq tours, l'Allemagne perd du territoire dans chaque partie. Le but du jeu étant d'envahir les territoires de l'ennemi, il est clair que l'Allemagne se dirige vers un échec dans chacune de ses parties. Nous avons déjà mentionné que le jeu est débalancé naturellement, ce débalancement se fait dans le même sens que la Deuxième Guerre mondiale s'est déroulée, c'est-à-dire que l'Allemagne se trouve sous pression et qu'elle a plus de chances de craquer et ensuite d'entraîner le Japon dans sa perte. Donc, il semble normal que si le système d'Iron Blitz joue contre lui-même, l'Allemagne perdra. Afin de comparer notre système à celui d'Iron Blitz, nous avons réalisé cinq autres tests dans lesquels notre AI joue l'Allemagne, et les autres forces sont jouées par le système d'Iron Blitz. Les résultats sont disponibles dans les figures 6.6, 6.7, 6.8, 6.9 et 6.10. Les figures 6.13 et 6.14 sont les prises d'écrans de deux parties.

Nous constatons que, dans tous les cas, l'Allemagne gagne du terrain, ce qui fait une différence notable, puisqu'elle s'approche de la victoire. Dans certaines parties, l'Allemagne a même conquis la Russie. Le pire rendement de notre système reste meilleur que le meilleur d'Iron Blitz. Étant donné la quantité réduite de tests, il n'est pas possible d'affirmer que notre système est supérieur en tout temps à Iron Blitz, mais nous pouvons conclure que selon les tests effectués, il semble offrir de meilleurs résultats que son prédécesseur.

	URSS	Allemagne	UK	Japon	USA
Joué par	Iron Blitz	Iron Blitz	Iron Blitz	Iron Blitz	Iron Blitz
IPC tour 1	22	32	27	30	36
IPC tour 2	20	32	27	33	36
IPC tour 3	17	30	28	35	35
IPC tour 4	19	30	29	38	34
IPC tour 5	22	26	29	37	37
progression	-2	-6	-1	12	1

Figure 6.1 : Résultats de la première partie où tous les joueurs sont joués par le système Iron Blitz.

Partie 1	URSS	Allemagne	UK	Japon	USA
Joué par	Iron Blitz	Iron Blitz	Iron Blitz	Iron Blitz	Iron Blitz
IPC tour 1	22	32	28	31	36
IPC tour 2	22	31	29	33	31
IPC tour 3	20	25	29	37	37
IPC tour 4	25	22	34	32	37
IPC tour 5	28	17	39	32	31
progression	4	-15	9	7	1

Figure 6.2 : Résultats de la deuxième partie où tous les joueurs sont joués par le système Iron Blitz.

Partie 1	URSS	Allemagne	UK	Japon	USA
Joué par	Iron Blitz	Iron Blitz	Iron Blitz	Iron Blitz	Iron Blitz
IPC tour 1	22	33	29	27	36
IPC tour 2	24	34	26	24	39
IPC tour 3	24	36	24	25	38
IPC tour 4	24	35	24	29	35
IPC tour 5	27	29	26	32	33
progression	3	-3	-4	7	-3

Figure 6.3 : Résultats de la troisième partie où tous les joueurs sont joués par le système Iron Blitz.

Partie 1	URSS	Allemagne	UK	Japon	USA
Joué par	Iron Blitz	Iron Blitz	Iron Blitz	Iron Blitz	Iron Blitz
IPC tour 1	22	32	27	30	36
IPC tour 2	22	31	28	34	32
IPC tour 3	20	25	28	37	37
IPC tour 4	25	20	33	32	37
IPC tour 5	28	17	39	32	31
progression	4	-15	9	7	-5

Figure 6.4 : Résultats de la quatrième partie où tous les joueurs sont joués par le système Iron Blitz.

Partie 1	URSS	Allemagne	UK	Japon	USA
Joué par	Iron Blitz	Iron Blitz	Iron Blitz	Iron Blitz	Iron Blitz
IPC tour 1	22	32	27	30	36
IPC tour 2	18	32	27	34	36
IPC tour 3	21	28	28	36	34
IPC tour 4	20	28	29	38	32
IPC tour 5	22	22	29	37	37
progression	-2	-10	-1	12	1

Figure 6.5 : Résultats de la cinquième partie où tous les joueurs sont joués par le système Iron Blitz.

Partie 1	URSS	Allemagne	UK	Japon	USA
Joué par	Iron Blitz	Ce système	Iron Blitz	Iron Blitz	Iron Blitz
IPC tour 1	20	38	27	27	36
IPC tour 2	20	39	26	27	36
IPC tour 3	14	44	25	32	34
IPC tour 4	14	45	22	34	32
IPC tour 5	14	48	19	36	30
Victoire économique de l'Axe					
progression	-10	16	-11	11	-6

Figure 6.6 : Résultats de la première partie où l'Allemagne est jouée par mon système et tous les autres joueurs sont joués par le système Iron Blitz.

Partie 1	URSS	Allemagne	UK	Japon	USA
Joué par	Iron Blitz	Ce système	Iron Blitz	Iron Blitz	Iron Blitz
IPC tour 1	20	38	26	26	36
IPC tour 2	0	47	26	30	34
IPC tour 3	0	51	26	35	35
Victoire économique de l'Axe					
progression	-24	19	-4	10	-1

Figure 6.7 : Résultats de la deuxième partie où l'Allemagne est jouée par mon système et tous les autres joueurs sont joués par le système Iron Blitz.

Partie 1	URSS	Allemagne	UK	Japon	USA
Joué par	Iron Blitz	Ce système	Iron Blitz	Iron Blitz	Iron Blitz
IPC tour 1	21	36	26	25	36
IPC tour 2	20	37	25	31	34
IPC tour 3	17	40	21	34	34
IPC tour 4	15	42	20	38	32
IPC tour 5	13	44	16	40	32
Victoire économique de l'Axe					
progression	-11	12	-14	15	-4

Figure 6.8 : Résultats de la troisième partie où l'Allemagne est jouée par mon système et tous les autres joueurs sont joués par le système Iron Blitz.

Partie 1	URSS	Allemagne	UK	Japon	USA
Joué par	Iron Blitz	Ce système	Iron Blitz	Iron Blitz	Iron Blitz
IPC tour 1	19	39	26	24	39
IPC tour 2	21	42	23	22	39
IPC tour 3	0	51	23	29	36
Victoire économique de l'Axe					
progression en %	-24	19	-7	4	0

Figure 6.9 : Résultats de la quatrième partie où l'Allemagne est jouée par mon système et tous les autres joueurs sont joués par le système Iron Blitz.

Partie 1	URSS	Allemagne	UK	Japon	USA
Joué par	Iron Blitz	Ce système	Iron Blitz	Iron Blitz	Iron Blitz
IPC tour 1	21	36	26	25	36
IPC tour 2	20	37	25	31	34
IPC tour 3	23	35	21	34	34
IPC tour 4	20	37	20	38	32
IPC tour 5	14	43	20	40	30
progression	-10	11	-4	15	-6

Figure 6.10 : Résultats de la cinquième partie où l'Allemagne est jouée par mon système et tous les autres joueurs sont joués par le système Iron Blitz.

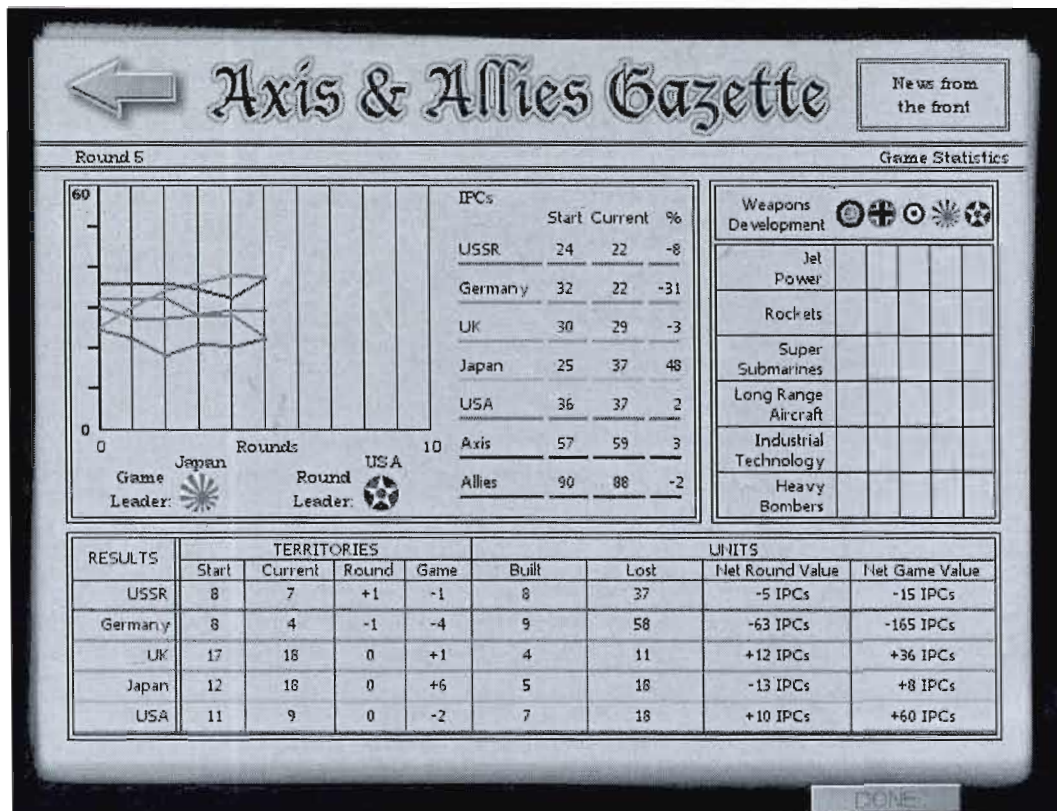


Figure 6.11 : Capture d'écran de la première partie où tous les joueurs sont joués par le système Iron Blitz.

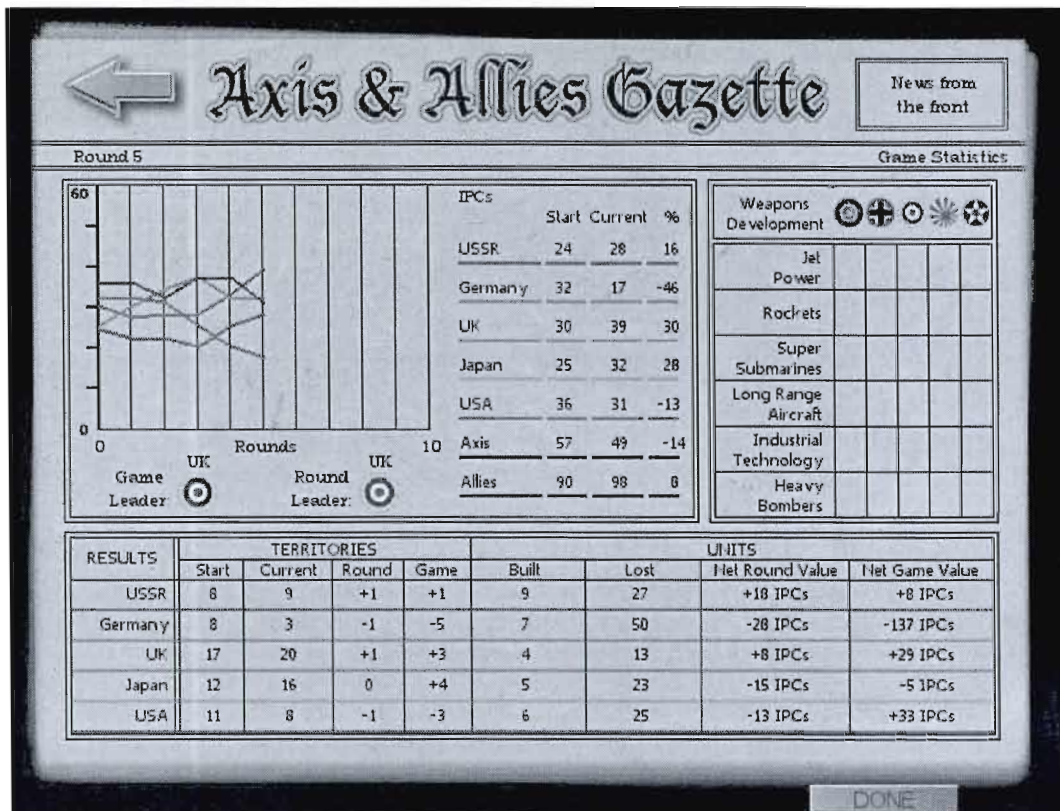


Figure 6.12 : Capture d'écran de la deuxième partie où tous les joueurs sont joués par le système Iron Blitz.

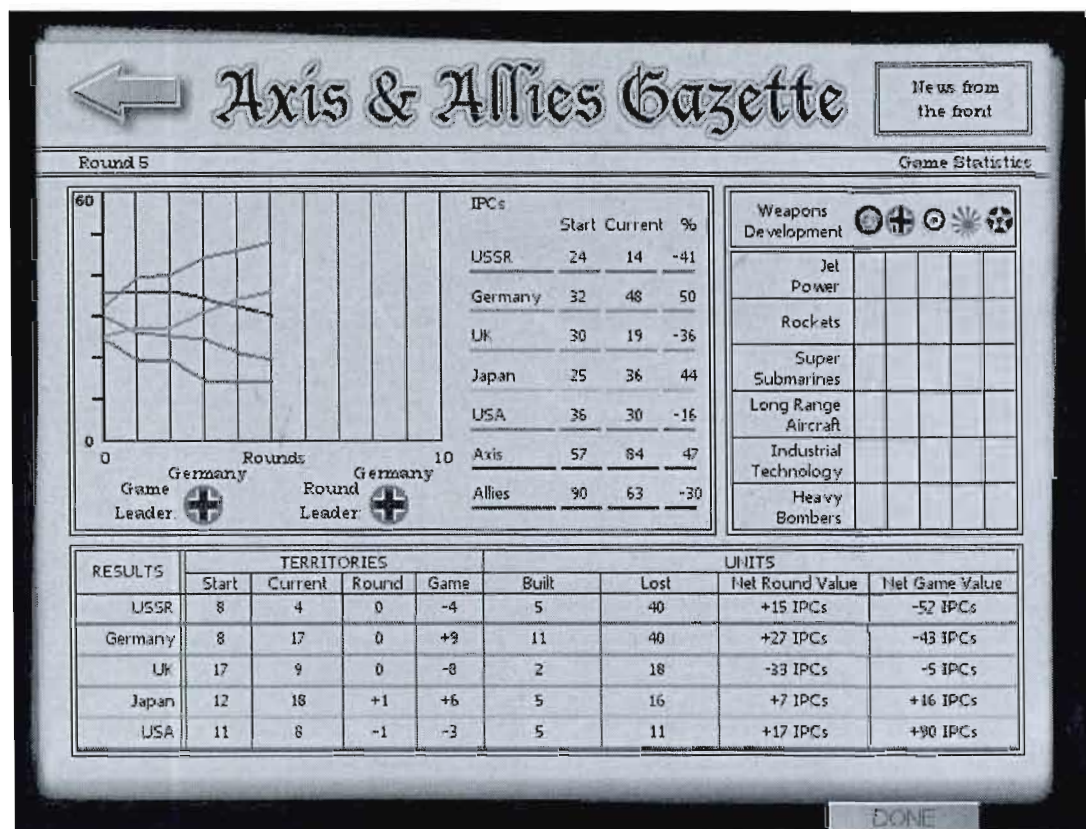


Figure 6.13 : Capture d'écran de la première partie où l'Allemagne est jouée par mon système et tous les autres joueurs sont joués par le système Iron Blitz.

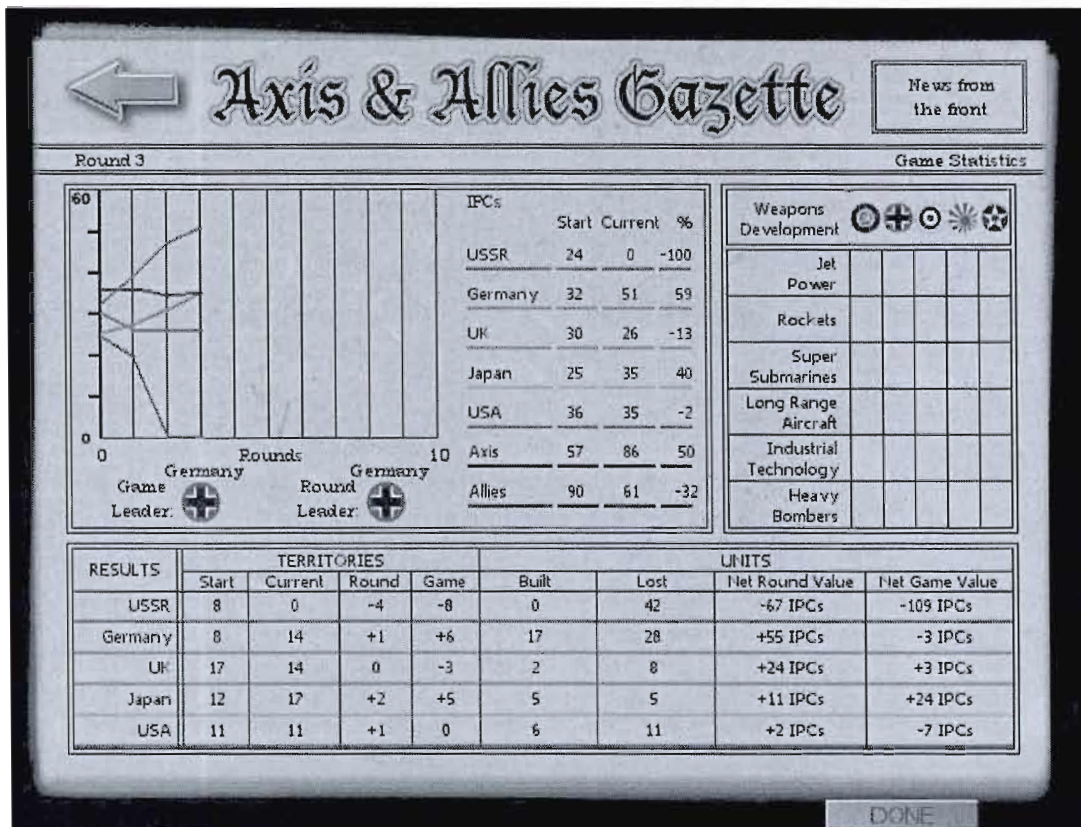


Figure 6.14 : Capture d'écran de la deuxième partie où l'Allemagne est jouée par mon système et tous les autres joueurs sont joués par le système Iron Blitz.

6.2 Test contre des humains

L'intérêt d'un système d'intelligence artificielle est de simuler un comportement humain. Dans ce contexte, il serait intéressant de pouvoir évaluer les performances de ce système d'intelligence artificielle contre des adversaires humains. Toutefois, un problème est de pouvoir évaluer les différents joueurs entre eux. Il n'existe pas de système de classification des joueurs d'Axis & Allies. Donc, il est difficile de donner un sens précis à un test de ce système contre un humain.

Malgré cela, deux tests distincts avec des humains ont été effectués. Ils nous ont permis de tirer quelques conclusions.

Un premier test a été effectué avec un joueur chevronné. Ce joueur jouait les alliés, donc l'empire britannique, les russes et les américains. Le système jouait l'Allemagne et je jouais le Japon. Comme pour les tests entre Iron Blitz et ce système, les parties ont été arrêtées au cinquième tour.

Les résultats sont intéressants : notre système a opposé une bonne résistance en gagnant du territoire au profit du joueur russe et britannique. Par contre, le nombre de troupes des alliés dépassait celui des allemands. Les deux joueurs humains présents à la table ont donc conclu que les alliés finiraient par gagner étant donné la quantité de troupes en réserve des alliés.

Un deuxième test a été effectué contre un joueur n'ayant pas joué à ce jeu depuis plusieurs années. La partie s'est déroulée selon les mêmes règles que la précédente. Les résultats sont satisfaisants: après cinq tours de jeu, l'axe a gagné. La Russie est tombée aux mains des allemands et les japonais ont pris beaucoup de territoire, ce qui a mené à la victoire économique de l'axe.

Tel que mentionné précédemment, ces tests ne permettent pas tirer des conclusions définitives ou statistiques. La seule conclusion possible, suite à ces deux tests supplémentaires, est que le système peut gagner contre des humains qui ne sont pas des débutants.

6.3 Conclusion sur les tests

Lors de la majorité des tests, le système a pris des décisions qui lui ont permis de gagner. Il était donc ardu de trouver les faiblesses du système. Par contre, dans la partie contre un humain expérimenté, il est apparu évident que quelques ajustements mineurs étaient nécessaires. Notamment, l'utilisation des avions pour couler les bateaux anglais. La règle d'évaluation est peut-être trop risquée. Le problème est qu'il existe dans le système d'IA un algorithme d'évaluation des attaques et que, selon l'expert, les avions étant des pièces plus valables que d'autres, il faudrait que le succès d'une attaque soit pondéré par le nombre de pièces aériennes dans les troupes attaquantes ou qu'il existe deux fonctions d'évaluations, une pour les attaques terrestres et l'autre pour les attaques navales. Cette modification aurait rendu le système plus compétitif. Notamment au tour quatre de la partie contre le joueur chevronné où, selon l'avis des humains, l'IA aurait dû faire une attaque massive en Russie et ignorer les bateaux anglais, ce qui lui aurait peut-être permis de gagner la partie.

Évidemment, avec un système de classement des joueurs, on aurait pu donner un sens à ces tests et il aurait été intéressant d'en effectuer plusieurs contre différents adversaires. Suite à quoi, il aurait été possible de classer le système d'IA parmi les joueurs. Étant donné l'absence d'un tel système de classement, les tests contre les humains ne pouvant rien prouver, il n'est pas pertinent d'effectuer une batterie de tests.

CONCLUSION

À cause du nombre de tests restreints, nous ne pouvons affirmer avec certitude que la méthode utilisée par notre système est supérieure à la méthode utilisée par Hasbro même si dans la série de tests effectués, le rendement de ce système s'est avéré supérieur à celui du système de Hasbro, Iron Blitz, et ce à chaque essai. Nous savons toutefois que nous sommes sur la bonne voie. Pour remédier au nombre de tests, il faudrait créer une interface web plus simple à utiliser afin que les usagers du web puissent tester l'efficacité de ce nouveau système. Ainsi, la collecte de résultats pourrait être automatisée.

Une option à évaluer pour améliorer le produit serait de créer un système hybride qui utiliserait plusieurs technologies. Par exemple, les règles d'un système expert choisissent l'endroit à analyser, afin de réduire la complexité et, une fois que les zones ont été choisies, un réseau de neurones ou un algorithme génétique choisit les coups. Peu importe le système hybride, étant donné la complexité du plateau de jeu, il est souhaitable que le premier niveau soit un système expert.

Il aurait été intéressant de créer notre système pour qu'il puisse jouer n'importe quel joueur, mais faute de temps, le système restera tel quel. Il aurait aussi été intéressant de pousser le projet plus loin sur la voie des jeux de société semblables. Plusieurs jeux de société utilisent des concepts de déplacements semblables avec des pièces pouvant se modéliser de la même manière. Suite à nos recherches sur le sujet, nous constatons que le domaine de l'intelligence artificielle sur les jeux de société est très avancé en ce qui concerne les jeux classiques, mais pas sur les jeux de plateau. Il serait intéressant, dans un projet futur, de créer des techniques générales, des patrons de conceptions et une documentation sur les jeux de plateau. Il y aurait assurément beaucoup à dire et à faire sur le sujet.

BIBLIOGRAPHIE

- Anderson, James A. 1995. *An Introduction to Neural Networks*. MIT Press.
- Bourg, David M., et Glenn Seemann. 2004. *AI for Game Developers: Creating Intelligent Behavior in Games*, O'Reilly.
- Coppin, Ben. 2004. *Artificial Intelligence Illuminated*. Jones & Bartlett Publishers.
- Firebaugh, Morris W. 1988. *Artificial Intelligence: A Knowledge-based Approach*. Boyd & Fraser, 1988.
- Ganascia, Jean-Gabriel. 2007. *L'intelligence artificielle*, Le Cavalier Bleu.
- Goldberg, David Edward, Vincent Corruble, Jean-Gabriel Ganascia et John Holland. 1994. *Algorithmes génétiques: exploration, optimisation et apprentissage automatique*. Addison-Wesley.
- Negnevitsky, Michael. 2005. *Artificial Intelligence: A Guide to Intelligent Systems*, Addison-Wesley, 2005.
- Papineau, Élisabeth. 2000. *Le jeu dans la Chine contemporaine : mah-jong, jeu de go et autres loisirs*. Paris : L'Harmattan.
- Russell, Stuart Jonathan, Peter Norvig et John F. Canny. 2003. *Artificial Intelligence: A Modern Approach*. Collaborateur Peter Norvig, John F. Canny, Prentice Hall.
- Stinson, Douglas Robert. 2004. *Combinatorial Designs: Constructions and Analysis*, Springer.

ANNEXE

Afin de ne pas alourdir le document avec du code imprimé, le code est disponible à l'adresse suivante :

www.labunix.uqam.ca/~bustros/code