

Projet L1 – CMI
Spécialité Informatique

Jeu de Dames

Etudiants :

GALLERNE Romain
NAVEL Morgne

Année : 2020 – 2021

Soutenu : Début Mai 2021



Sommaire

I – INTRODUCTION

(Page 3)

- I.1 Généralités
- I.2 - Le Sujet
- I.3 - Cahier des Charges

II – ORGANISATION DU PROJET

(Page 4)

- II.1 - Organisation du Travail
- II.2 - Choix des Outils de développement

III – ANALYSE DU PROJET

(Page 5)

- III.1 – Algorithmes
- III.2 - Liste des fonctions et explications de leur utilité

IV – DÉVELOPPEMENT

(Page 7)

- IV.1 – Calcul de déplacement et structures de données
- IV.2 – Intelligence Artificielle
- IV.3 – Interface Utilisateur

V – MANUEL D'UTILISATION

(Page 9)

- V.1 – Le Menu
- V.2 – Guide de Jeu

VI – PERSPECTIVES ET CONCLUSION

(Page 12)

- VI.1 – Perspectives
- VI.2 – Conclusion

I – INTRODUCTION

I.1 – Généralités

Dans le cadre de l'unité d'enseignement HLSE202 ou «Projet Informatique CMI » nous devons réaliser un programme, de préférence un jeu vidéo afin de le soumettre dans ce présent rapport final ainsi que dans divers oraux.

Ce projet a pour objectif de nous habituer à la gestion de projet autant sur le plan programmation que gestion avec des rapports à rédiger. Pour réaliser ce projet nous avons une durée d'environ un mois et demi avec notamment plusieurs séances Zoom le vendredi.

I.2 – Le Sujet

Concernant le sujet nous avons choisi de revisiter le jeu de Dames. Nous l'avons donc sobrement intitulé «Jeu de Dames». Ce jeu se jouera uniquement via le support de la console de commande et sera une version adaptée du jeu de Dames, en effet les règles de ce jeu diffèrent beaucoup d'une personne à l'autre mais nous y reviendrons plus tard.

I.3 – Cahier des Charges

Le Jeu de dames devra pouvoir se jouer de deux façons :
Soit contre un autre humain.
Soit contre l'ordinateur.

Il devra être possible de choisir quel couleur commence. Il faudra aussi afficher une interface intuitive pour l'utilisateur. Le programme ne devra pas s'interrompre tant que l'utilisateur ne l'a pas explicitement demandé, il peut donc rejouer autant qu'il le veut. Le programme devra notifier le joueur d'une éventuelle erreur d'entrée pour l'aider à la corriger.

Le programme tournera sous le support Linux.

Il comprendra une IA relativement limitée essentiellement basée sur l'aléatoire.

Le Jeu devra respecter les règles du Jeu de Dames suivantes :

- Le Damier mesure 10 par 10 et contient 20 pions blancs et 20 pions noirs au commencement de la partie.
- Les pions blancs ne peuvent qu'avancer et les pions noirs que reculer le tout en diagonale
- Lors d'une prise, le pion prenant « saute » par dessus le pion pris et se retrouve derrière lui, à condition que cette case soit vide. Le pion pris disparaît.
- Si le pion prenant peut reprendre tout de suite après une première prise alors il peut enchaîner directement avec ce coup et ce autant de fois qu'il le veut.
- Lorsque un pion enchaîne plusieurs coups il peut le faire en arrière, c'est le seul cas dans lequel un pion peut reculer.
- Si un pion atteint la bordure adverse du plateau il devient une dame, le nombre de dame n'est pas limité
- La dame est le seul pion à pouvoir reculer en plus d'avancer lors d'un simple déplacement mais elle est toujours limitée à une seule case de portée.
- La victoire est déclarée lorsque plus aucune pièce d'une couleur n'est présente sur le terrain
- On est obligé de prendre lorsque c'est possible

II – ORGANISATION DU PROJET

II.1 – Organisation du Travail

Afin de mener à bien le projet nous avons décidé d'organiser chaque vendredi, au début, pendant la réunion zoom hebdomadaire une réunion d'environ 2H pour faire le point sur l'avancement du projet et les difficultés rencontrés.

Nous avons décidé de nous répartir les tâches selon le planning suivant (les dates indiquent ce pour quand la tâche doit être terminée).

Romain

12 mars : Mise en place des outils

19 mars : Création des données initiale et fonction main

26 mars : Création de plateau de Jeu initiale

2 avril : Création de la fonction Principale : Jeu

9 avril : Fin des fonctions victoires et update

16 avril : Suite et fin de la fonction prenable, menu en caractère ASCII

23 avril : Création de l'IA et débogage

Morgan

12 mars : Planning

19 mars : Création de la fonction Affichage

26 mars : Début de la fonction Victoire

2 avril : Début de la fonction update

9 avril : Fonction DeuxièmeCoup et début de la fonction prenable

16 avril : Suite et fin de la fonction prenable

23 avril : Tests et recherche de problèmes

N'étant que deux sur le projet nous n'avons pas vraiment eu la nécessité de choisir un chef de projet malgré que Romain se soit vite imposé comme tel du fait de ses connaissances plus avancées en programmation.

II.2 – Choix des Outils de Développement

Nous avons eu un débat sur le choix du langage de développement, en effet Romain souhaitait partir sur du Python, langage dans lequel il a un niveau plutôt avancé et qui aurait permis une version graphique du projet avec l'usage de la bibliothèque Tkinter. Le souci étant que Morgan n'avait absolument aucune notion de Python. Nous avons donc finalement fait le choix de partir sur du C/C++ que nous connaissions tous les deux. Malheureusement cela nous a empêché de faire une version graphique.

Concernant le support nous avons fait le choix d'utiliser le site Replit.com pour sa simplicité, la capacité à pouvoir éditer tout deux en même temps un même morceau de code ainsi que pour la capacité à pouvoir accéder à tout moment au code depuis n'importe quel machine. Nous l'utilisons également comme gestionnaire de projet, nous sauvegardons chaque les différentes versions dans des dossiers séparés.

Afin de générer les dessins en caractères ASCII nous avons utilisé le site rakko.tool, qui est libre d'utilisation. Romain a réalisé les images via un éditeur d'image avant de les transformer via cet outil.

III – ANALYSE DU PROJET

III.1 – Algorithme

Avant de nous lancer dans la programmation du jeu nous avons choisit d'écrire nos fonctions principale sous forme algorithmique, notamment l'IA qui a été rédigé ainsi que la fonction Jeu dont voici les algorithmes :

Algorithme de Jeu :

On demande au joueur les coordonnées du pion qui va bouger
On demande au joueur les coordonnées de son point d'arriver
Les cases entrés existent-elles ?
Si oui
On vérifier si il possède des pions prenable
Si oui, la case d'origine qu'il a entré peut-elle prendre ?
Si oui, la case d'arrivé est-elle bien libre ?
Si oui, Y a-t-il bien un pion adverse au milieu ?
Si oui, jouer cette prise
Si non, erreur, on redemande
Si non, erreur, on redemande
Si non, erreur, on redemande
Si non, la case d'arrivé est-elle bien libre ?
Si oui, la case est-elle bien atteignable par ce type de piece ?
Si oui, jouer ce déplacement
Si non, erreur, on redemande
Si non, erreur, on redemande

Algorithme de l'IA :

Puis-je prendre un des pions adverses ?
Si oui, Y en a-t-il plusieurs ?
Si oui, jouer l'un d'eux aléatoirement
Si non, jouer l'unique prise
Si non, Puis-je faire une dame ?
Si oui, Y en a-t-il plusieurs ?
Si oui, jouer l'un de ces coups au hasard
Si non, jouer l'unique formation de dame
Si non, Jouer un déplacement aléatoire

III.2 - Liste des fonctions et explication de leur utilité

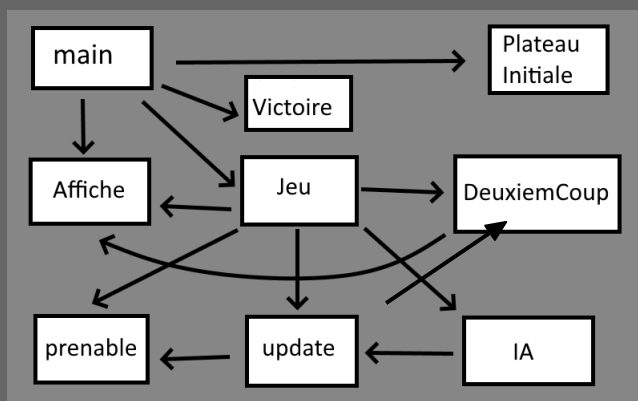


Figure 3.0 – Schéma des appels de fonction

III.2.a – Fonction main

Il s'agit de la fonction principale, c'est elle qui rythme le programme et gère une partie de l'affichage console. C'est aussi elle qui prévoit l'affichage de la victoire, qui génère le menu et ses différents sous-menu excepté le jeu qui est délégué à la fonction Jeu()

III.2.b – Fonction Jeu

Ceci est la fonction principal du jeu qui ne prend aucun paramètre en entrée. C'est elle qui lancera presque toutes les autres fonctions et qui va gérer l'exécution de l'action en cours. Elle est relancé par main au début de chaque tours et gères la répartition des deux joueurs et de l'IA. C'est aussi elle qui effectue toutes les vérifications comme détaillé dans son algorithme ci-dessus.

III.2.c – Fonction DeuxiemeCoup

Cette fonction a pour objectif de gérer les cas dans lesquels le joueur peut faire un coup multiples (prendre plusieurs pion d'un coup). Elle est lancé par la fonction update qui vérifie si l'arrivé après une prise permet une double prise. Elle prend donc en paramètre les coordonnées d'origine de la double prise qui ne sont donc pas libre.

III.2.d – Fonction prenable

Cette fonction a pour objectif de savoir si un pion (ou dame) peut prendre un pion (ou dame) adverse. Elle prend en paramètre les coordonnées de ce pion sous la forme (ligne, colonne).

III.2.e – Fonction update

Cette fonction a pour objectif de mettre à jour la plateau de jeu en fonction des informations transmises par la fonction jeu.

Il s'agit d'une fonction aveugle : elle ne fait aucune vérification et se contente de faire les changements.

Elle prend en paramètre les coordonnées de la case d'origine (x0,y0) et celle de la case d'arrivé (x1,y1) mais aussi un code symbolisé par l'entier n. En effet, n=1 => déplacement simple alors que n=2 => prise d'un pion.

III.2.f – Fonction Plateau Initiale

Cette fonction a pour objectif de générer le plateau de jeu initiale. Elle prend en paramètre l'adresse du premier élément du tableau.

III.2.g – Fonction Affiche

Ceci est la fonction d'affichage. Elle prend en entrée la Matrice du Damier et le formate pour l'afficher joliment en console

Il s'agit uniquement de l'affichage, on retrouve tout ces éléments lorsque on joue.

III.2.h – Fonction Victoire

Cette fonction a pour but de vérifier si un joueur a gagner. Pour cela elle prend en paramètre player qui indique true pour blanc et false pour noir

III.2.i – Fonction AI

Ceci est l'intelligence artificiel qui peut être activé ou non dans le menu Option. Son algorithme est détaillé ci-dessus.

IV – Développement

IV.1 – Calcul de déplacement et Structure de donnée

Afin de pouvoir vérifier les déplacement du joueur et de calculer ceux de l'IA il a fallut définir les relations entre les différentes cases du Damier. Le Damier est en effet une Matrice de taille 10x10. En terme de programmation c'est un tableau de 10 tableau de 10 éléments. Cela nous a permis de mettre en place le système de coordonnée suivant : Damier[y][x] renvoyant la valeur à la case (x,y). Nous avons ainsi deux méthode de calcul : L'une via les coordonnées et l'une via les adresses. Les deux nous seront utile dans le programme, certaine plus que d'autres selon la situation. En effet pour saisir les données entrées par l'utilisateur il est bien préférable d'utiliser le format coordonnée. En revanche lorsque on a besoin de calculer le pion moyen entre l'origine et l'arrivé le calcul par adresse est bien plus pratique.

IV.2 – Intelligence Artificielle

L'intelligence artificielle utilisé dans la fonction IA a été conçu de 0. Cependant elle est relativement simple et repose énormément sur l'aléatoire. En effet cette IA rudimentaire connaît les règles de base des dames et les appliques dans l'ordre énoncé ci-dessus. Pour ce qui est des choix aléatoire elle fait appelle aux méthodes rand() et srand(). Pour une raison inconnu ces deux méthodes, notamment la méthode srand() cause beaucoup de latence mais nous n'avons pas le choix pour avoir un pseudo aléatoire crédible. L'IA n'est donc pas très intelligente, c'est simplement un algorithme qui applique les règles des échecs et effectuent tout ces choix aléatoirement avec les méthodes rand() et srand(). En effet elle ne calcule pas quel coup serait le plus avantageux pour elle.

IV.3 – Interface Utilisateur

Lors du lancement du programme, l'utilisateur arrive sur un menu qui lui permet de choisir entre quatre sections :

- Jouer
- Option
- Aide
- Quitter

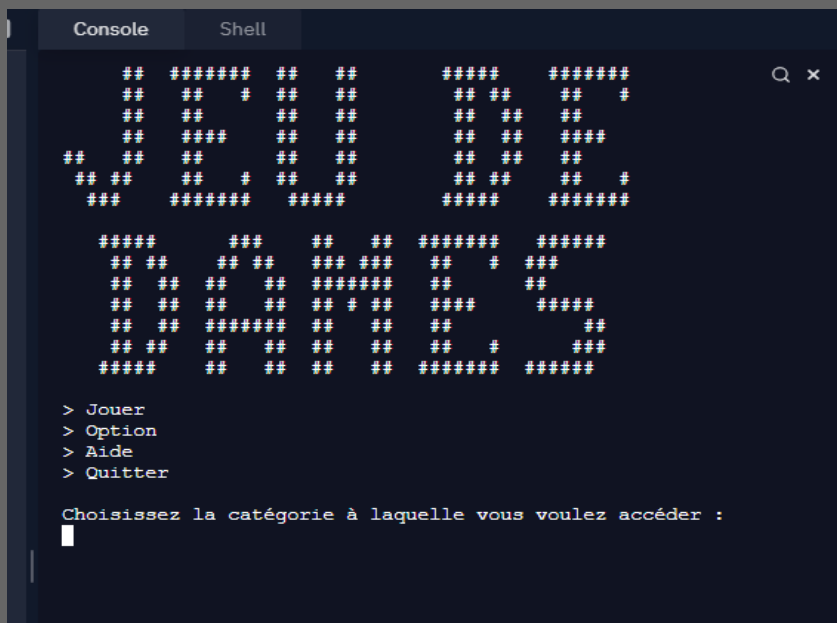


Figure 4.0 – Menu affiché au lancement du jeu

Comme cité précédemment, l'affichage en caractère ASCII a été généré par le site rakko.tool qui est libre de droit.

On dispose ici de quatre section accessible depuis le menu. Nous nous sommes limité à ces quatre sections afin de simplifier au maximum l'interface. L'utilisateur n'a alors qu'à entrer la catégorie dans laquelle il souhaite aller.

Nous détaillerons la majorité des catégorie dans la partie « Manuel d'utilisation » mais nous allons ici revenir sur l'aspect technique des catégories « Option » et « Quitter ».

Tout d'abord, la catégorie « Quitter » qui est plutôt simple. En effet la fonction main est construite dans une boucle infini puisque le programme ne doit pas s'arrêter tant que l'utilisateur ne l'a pas demander. À l'unique condition que le joueur entre « Quitter », la fonction va effectuer un `return 0 ;` ce qui aura pour effet de mettre fin à la boucle et d'achever l'exécution du programme.

Enfin, la catégorie « Option » va proposer deux choix au joueur : Le premier est de choisir quel couleur commence : Les blancs ou les noirs. Dans tout les cas, si l'IA est activé c'est toujours le joueur qui commencera, qu'il soit blanc ou noir. Pour faire cela on modifie avant même l'exécution du programme la variable booléenne « player » qui est de base à 1 pour blanc. En effet à la fin de chaque coup cette variable prend la valeur opposé à celle qu'elle avait précédemment. La majorité des fonctions et notamment la fonction `Jeu()` sont sensible à cette variable pour savoir quelle joueur doit jouer. En la modifiant dès le début on change sa valeur de base au début de l'exécution et on permet ainsi de commencer en tant que noir (car par défaut les blancs commencent comme dans le vrai jeu de dames). On applique le même procédé sur la variable « IA » qui indique si l'IA est activé.

V – MANUEL D'UTILISATION

V.1 – Le Menu

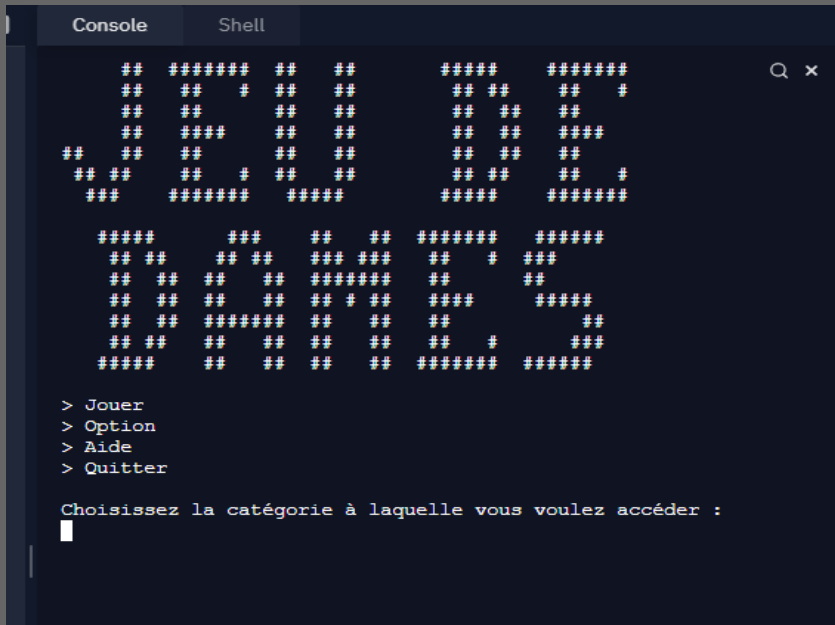


Figure 5.0 – Menu affiché au lancement du jeu

Comme dit précédemment quatre choix s'offre à nous au lancement du jeu. Nous allons détaillé ces quatre choix.

V.1.a – Jouer

La catégorie « Jouer » est celle qui va lancer le jeu. Par défaut, si les paramètres n'ont pas été modifié les blancs commenceront, puis viendra le tours des noirs.

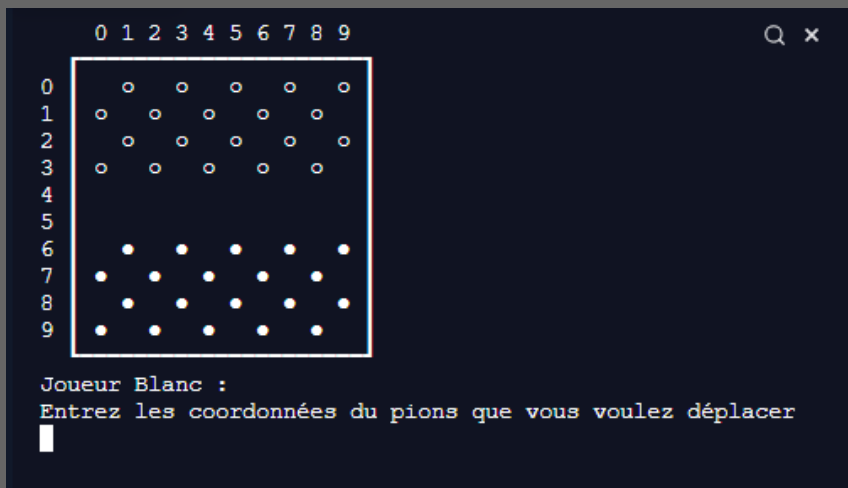


Figure 5.2 – Début du premier tours, les blancs commencent

V.1.b – Option

Le menu option permet de répondre à deux questions qui vous seront posé :

Quel couleur commence ?

Vous pourrez alors répondre « Blanc » ou « Noir »

Joue-t-on contre l'IA ?

Vous pourrez répondre « Oui » ou « Non » si vous souhaitez jouer contre un autre humain

V.1.a – Aide

La catégorie aide énonce les règles du jeu de Dames ainsi qu'un exemple de coordonnées pouvant être entrée.

```
Le Jeu de Dame si présent utilise les règles modernisé du jeu de Dame : L'obj
ectif est d'être le dernier joueur à avoir des pions sur le plateau. Le Jeu d
evra respecter les règles du Jeu de Dames suivantes :
- Le Damier mesure 10 par 10 et contient 20 pions blancs et 20 pions noir au
commencement de la partie.
- Les pions blancs ne peuvent qu'avancer et les pions noirs que reculer le to
ut en diagonale
- Lors d'une prise, le pion prenant « saute » par dessus le pion pris et se r
etrouve derrière lui, à condition que cette case soit vide. Le pion prit disp
araît.
- Si le pion prenant peut reprendre tout de suite après une première prise al
ors il peut enchaîner directement avec ce coup et ce autant de fois qu'il le
veut.
- Lorsque un pion enchaîne plusieurs coup il peut le faire en arrière, c'est
le seul cas dans lequel un pion peut reculer.
- Si un pion atteint la bordure adverse du plateau il devient une dame, le no
mbre de dame n'est pas limité
- La dame est le seul pion à pouvoir reculer en plus d'avancer lors d'un simp
le déplacement mais elle est toujours limité à une seule case de portée.
- La victoire est déclaré lorsque plus aucune pièce d'une couleur n'est prése
nte sur le terrain
- On est obligé de prendre lorsque c'est possible Pour cela vous devrez entre
r la case de derrière (qui doit être libre).
Pour vous déplacer, vous devez entrer les coordonnées de la case sous la form
e 'ligne colonne', exemple : '5 4'.
```

```
Entrez 'Retour' pour quitter
```

Figure 5.3 – Affichage d'Aide

V.1.a – Quitter

La catégorie « Quitter » permet simplement de fermer le jeu.

V.2 – Guide du Jeu

V.2.a – Déplacement

Pour se déplacer le Joueur doit rentrer en premier les coordonnées du pion à déplacer et en second son point d'arrivée. Par exemple ici, on a rentré en premier « 6 1 » puis « 5 2 ».

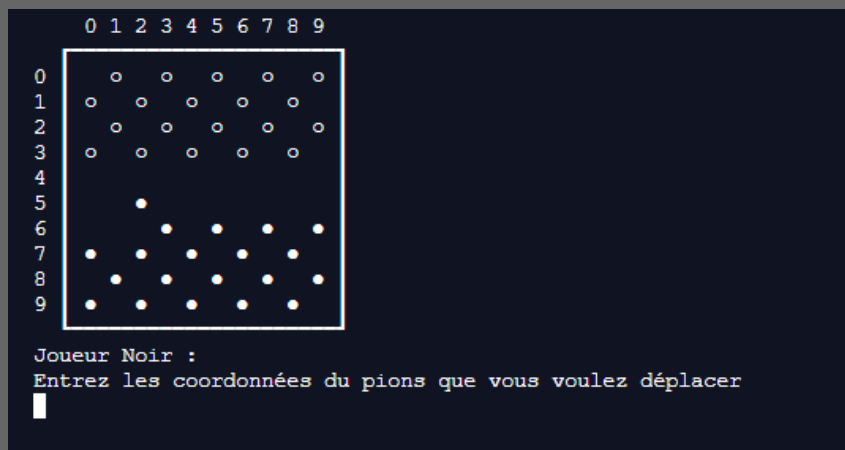


Figure 5.4 – Premier déplacement effectué

V.2.b – Prise

Afin d'effectuer une prise il faut entrer les coordonnées du pion qui doit effectuer cette prise puis celle de son emplacement d'arrivée (Attention, pas celle de la case qu'il prend). Le programme se chargera de vérifier toutes les conditions et si il le faut, affichera un message d'erreur pour vous aider à corriger le problème.

V.2.c – Double Prise

Lors d'une double prise, le programme affichera « Joueur "Couleur", poursuivez l'enchaînement : ». À ce moment, il n'est plus nécessaire d'entrer les coordonnées du pion que se déplace puisque celui-ci est déjà connu du programme, c'est votre destination précédente. Il suffit alors d'entrer votre prochaine destination à partir de ce pion uniquement. Comme pour la prise simple, on entre les coordonnées de la case situé derrière la pièce visée.

V.2.d - Dames

Si vous parvenez à atteindre le bord adverse du terrain avec l'un de vos pions vous formerez automatiquement une dame. La dame est un pion plus puissant car celui-ci peut reculer

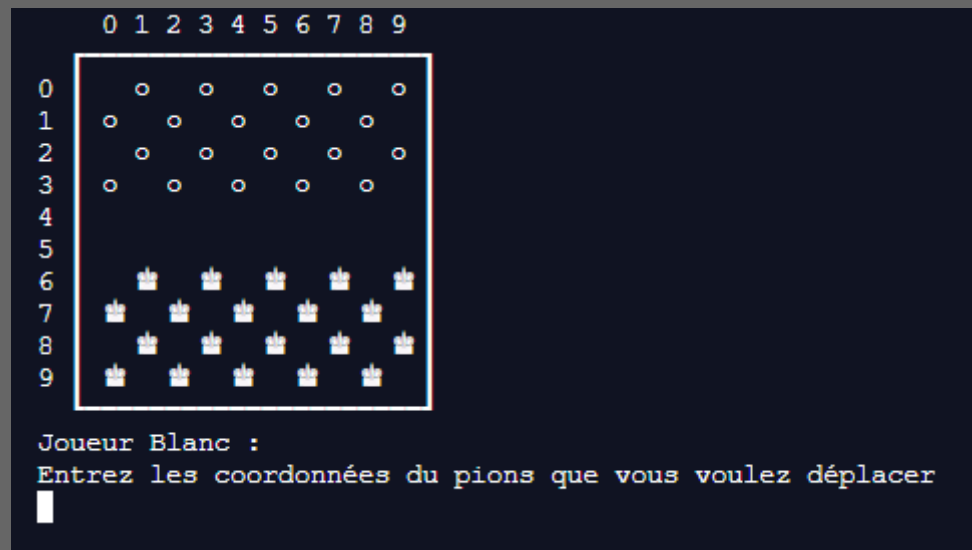


Figure 5.5 – Début d'une partie parfaitement équilibré

VI – PERSPECTIVES ET CONCLUSION

VI.1 – Perspectives

Si il fallait citer quelques points à améliorer ce serait probablement :

En premier améliorer le manque de clarté au niveau de l'entrer des coordonnées ainsi que l'affichage des cases, car on n'a pas de ligne donc on prend un peu de temps avant de trouver la case correspondante, et là où on veut jouer. Sur un damier on représente souvent le damier avec 2 couleurs, pour justement être sûr de là où on est.

Il faudrait réécrire le programme avec des programmes qui viennent d'autre page afin d'avoir un code source extrêmement clair, mais cela prend pas mal de temps à faire car on peut avoir des problèmes liés au programme ou au fait qu'ils ne sont pas sur la même page. C'est donc assez compliqué de débogué les fonctions une part une pour savoir si elle fonctionne bien.

Pour que le projets soit esthétiquement plus beau, il faudrait que l'on rajoute une page d'accueil un peu plus élaboré que juste « JEU DE DAME » avec par exemple un dessin en caractère, mais ça demande beaucoup de temps à faire aussi. Dans l'idéal il aurait même fallu faire une véritable interface graphique mais malheureusement le temps a jouer contre nous, il nous aurait fallu apprendre à faire cela car aucun de nous ne savait le faire en C/C++.

VI.2 – Conclusion

VI.2.a – Fonctionnement du programme

On a eu beaucoup de problèmes avec la fonction « prenable » notamment lorsque on avait des pions vides qui mettaient la fonction « prenable » a « true ». On ne parvenait donc pas à prendre des pions, et, de plus, parfois les états prenables n'étaient pas reconnus eux non plus, mais tout cela à était corrigé et réglé. Maintenant le seul problème , par manque de temps, est que l'IA ne peut pas prendre 2 pions pour 1 coup, comme le permet le programme si ce n'est pas l'IA.

Globalement les fonctions de bases pour jouer joueur contre joueur ,au moins, fonctionnent, l'objectif initial du projet étant de faire un jeu en état de marche, cela a été rempli avec succès.

Pour conclure on peut dire que notre projet est en état de fonctionnement, on peut jouer en 1 contre 1 avec une autre personne, de plus esthétiquement c'est relativement joli à regarder, donc notre projet est terminé, et peut être utilisé pour jouer entre amis, ou contre une IA.

VI.2.b - Fonctionnement du groupe de travail

Nous avons, d'après nous, été plutôt efficace car nous sommes parvenu, dès le début, à identifier les fonctions qui seraient incontournable pour faire le jeu. Par la suite nous avons réussi à régler la majorité de nos problèmes en les divisant en sous fonction, c'est par exemple la raison de l'existence de la fonction DeuxiemeCoup qui n'était pas prévu à la base. Lors des créations des fonctions principales nous avons eu quelques problèmes avec les conditions car nous devons faire les conditions une par une car l'utilisation du « ou » ou du « et » ne nous permet pas de distinguer les cas ,quand par exemple un pion peut prendre un autre pion pour savoir sur quelle diagonale le pion est et si il n'y a pas de pion derrière cette case, donc ça nous a pris beaucoup de temps pour tout copier car il y a vraiment beaucoup de conditions un peu partout où il est très facile de perdre le fil.

On a eu beaucoup de problèmes au niveau de certaines fonctions ,où on devait se mettre à deux pour essayer de trouver ce qui n'allait pas, ça a été réglé mais ça nous a pris au moins 2/3 séances.

Au niveau de la gestion du temps, on a été assez bon car malgré les problèmes que l'on a rencontré on a quand même essayé de continuer à créer quelques chose et nous ne sommes pas rester 3 séances à resté bloquer sur la même fonction, quelques choses était toujours fait en parallèle. Cela nous a permis de rattraper le retard que nous avions prit.

Finalement, malgré quelques aléas, nous sommes parvenu à mener à bien ce projet, fonctionnel, que nous soumettons aujourd'hui dans ce rapport. Il va sans dire que les connaissances acquises et l'expérience accumulé ici seront des atouts précieux pour de futurs projets.

Notes et références

- Dessins ASCII : [Rakko.tool](https://www.rakko.net/)
- Interface de code : [Replit.com](https://replit.com/)
- Image de fond de ce présent document : [Pixabay.com](https://www.pixabay.com/)