

**3iL ingénieurs**

L'Institut d'ingénierie informatique de Limoges

---

**Jeu de dames**

*Développement en C*

*Par*

**François Proust-Juvin**

**Baptiste Piegelin**

**Nathan Herault**

Mémoire présenté en vue du projet en C

En CPI 2- Algorithmique et Structures de Données - C

Samedi 10 décembre 2022

© François Proust-Juvin CPI2

© Baptiste Piegelin CPI2

© Nathan HERAULT CPI2

**3iL ingénieurs**

L'Institut d'ingénierie informatique de Limoges

---

**Jeu de dames**

*Développement en C*

*Présenté par*

**François Proust-Juvin**

**Baptiste Piegelin**

**Nathan Herault**

*A été évalué par un jury composé de :*

**Mohamed Hamroun**

Enseignant CPI 2 - Algorithmique et Structures de Données - C

# **Introduction**

Notre objectif lors de ce projet était de concevoir un jeu de dames à l'aide du langage de programmation « C ». Implémenter une interface graphique pour donner un aspect visuel attirant au projet. Ajouter d'un adversaire selon le mode de jeu désiré (1vs1) ou une intelligence artificielle pour joueur seul (You vs IA). Offrir au joueur un historique de coup pour revisionner ses parties et améliorer son expérience de jeu.

## **Table des matières**

Chapitre 1 – Les règles du jeu

Chapitre 2 – L'interface graphique

Chapitre 3 – La création du jeu

Chapitre 4 – Les éléments graphiques

# Chapitre 1 – Les règles du jeu

## Le matériel

---

Le jeu de dames international se joue sur un damier carré divisé en 100 cases égales, alternativement claires et foncées.

Le jeu se joue sur les cases foncées du damier. Il y a donc 50 cases actives. La plus longue diagonale, joignant deux coins du damier et comprenant 10 cases foncées, se dénomme la grande diagonale.

Le damier doit être placé de sorte que la première case de gauche, pour chaque joueur, soit une case foncée.

Le jeu de dames international se joue avec 20 pions blancs (clairs) et 20 pions noirs (foncés). Avant de débiter une partie, les 20 pions noirs et les 20 pions blancs sont disposés sur les 4 premières rangées de chaque joueur.

## La marche des pièces

---

Il existe deux types de pièces : les pions et les dames.

Le premier coup est toujours joué par les blancs. Les adversaires jouent un coup chacun à tour de rôle avec leurs pièces.

Un pion se déplace obligatoirement vers l'avant, en diagonale, d'une case sur une case libre de la rangée suivante.

Lorsqu'il atteint la dernière rangée, le pion devient dame. Pour cela, on couronne le pion en plaçant dessus un deuxième pion de la même couleur.

Une dame doit attendre que l'adversaire ait joué au moins une fois avant d'entrer en action.

Une dame se déplace en arrière ou en avant sur les cases libres successives de la diagonale qu'elle occupe.

Elle peut donc se poser, au-delà de cases libres, sur une case libre éloignée.

Pour un joueur à qui c'est le tour de jouer, le fait de toucher une de ses pièces jouables implique l'obligation de jouer cette pièce, pour autant que cela soit possible.

Tant qu'une pièce touchée ou en cours de déplacement n'a pas été lâchée, il est permis de la poser sur une autre case, pour autant que cela soit possible.

Le joueur qui a le trait et qui désire replacer correctement une ou plusieurs pièces doit le notifier d'avance et distinctement à l'adversaire par la formule "j'adoube".

# La prise

---

La prise des pièces adverse est obligatoire et s'effectue aussi bien en avant qu'en arrière.

Lorsqu'un pion se trouve en présence, diagonalement, d'une pièce adverse derrière laquelle se trouve une case libre, il doit obligatoirement sauter par-dessus cette pièce et occuper la case libre.

Cette pièce adverse est alors enlevée du damier. Cette opération complète est la prise par un pion.

Lorsqu'une dame se trouve en présence sur la même diagonale, directement ou à distance, d'une pièce adverse derrière laquelle se trouvent une ou plusieurs cases libres, elle doit obligatoirement passer par-dessus cette pièce et occuper, au choix, une des cases libres. Cette pièce est alors enlevée du damier. Cette opération complète est la prise par une dame.

Lorsqu'au cours d'une prise par un pion, celui-ci se trouve à nouveau en présence, diagonalement, d'une pièce adverse derrière laquelle se trouve une case libre, il doit obligatoirement sauter par-dessus cette seconde pièce, voire d'une troisième et ainsi de suite, et occuper la case libre se trouvant derrière la dernière pièce capturée. Les pièces adverses ainsi capturées sont ensuite enlevées du damier dans l'ordre de la prise. Cette opération complète est une rafle par un pion.

Lorsqu'au cours d'une prise par une dame, celle-ci se trouve à nouveau en présence, sur une même diagonale, d'une pièce adverse derrière laquelle se trouve une ou plusieurs cases libres, elle doit obligatoirement sauter par-dessus cette seconde pièce, voire d'une troisième et ainsi de suite, et occuper au choix une case libre se trouvant derrière et sur la même diagonale que la dernière pièce capturée. Les pièces adverses ainsi capturées sont ensuite enlevées du damier dans l'ordre de la prise. Cette opération complète est une rafle par une dame.

Au cours d'une rafle, il est interdit de passer au-dessus de ses propres pièces.

Au cours d'une rafle, il est permis de passer plusieurs fois sur une même case libre mais il est interdit de passer plus d'une fois au-dessus d'une même pièce adverse.

Une rafle doit être clairement indiquée, pièce par pièce, en posant la pièce preneuse sur la case de saut, le temps d'y passer, et en déposant cette pièce sur la case terminale.

Les pièces prises ne peuvent être enlevées du damier qu'à l'issue de l'exécution complète de la rafle.

La prise du plus grand nombre de pièces est prioritaire, donc obligatoire. Dans ce cas, une dame compte pour une pièce, tout comme un pion. Elle ne confère nulle priorité et n'impose aucune obligation.

Si les pièces à prendre sont en nombre égal dans deux ou plusieurs possibilités de prise ou de rafle, le joueur est libre de choisir une de ces possibilités, soit avec un pion, soit avec une dame.

Un pion qui au cours d'une rafle, ne fait que passer sur une des 5 cases de la rangée de base adverse reste un pion lorsque la rafle est terminée.

# Les irrégularités

---

Si un des joueurs a commis une irrégularité, son adversaire a seul le droit de décider si l'irrégularité doit être rectifiée ou si, au contraire, elle doit être maintenue. Quelques exemples d'irrégularité :

*Jouer deux fois de suite*

*Jouer une pièce de l'adversaire*

*Jouer une pièce alors qu'une prise ou une rafle est possible*

*Enlever sans raison des pièces de l'adversaire ou ses propres pièces*

*Prendre un nombre inférieur ou supérieur de pièces possibles à prendre*

*S'arrêter dans l'exécution d'une rafle*

*Enlever irrégulièrement une pièce, au cours d'une rafle non terminée*

Si un joueur refuse de se soumettre au règlement officiel du jeu, son adversaire est en droit de l'y obliger.

## La fin de partie égale

---

La fin de partie est considérée comme égale lorsque la même position se représente pour la troisième fois, le même joueur ayant le trait.

Si, durant 25 coups, il n'y a ni déplacement de pion ni prise, la fin de partie est considérée comme égale.

S'il n'y a plus que trois dames, deux dames et un pion, ou une dame et deux pions contre une dame, la fin de partie sera considérée comme égale lorsque les deux joueurs auront encore joué chacun 16 coups au maximum.

Pour autant qu'il n'y ait pas de phase de jeu en cours, la fin de partie de deux dames contre une dame, et a fortiori, d'une dame contre une dame, sera considérée égale.

Le résultat

Le gain est obtenu par un joueur lorsque son adversaire :

*Abandonne la partie*

*Se trouve dans l'impossibilité de jouer alors qu'il a le trait*

*N'a plus de pièces*

L'égalité, dénommée aussi remise ou nulle, est obtenue pour chacun des adversaires :

*Qui la déclare, de commun accord*

*Lorsqu'aucun des deux ne parvient au gain*

## Chapitre 2 – L'interface graphique

Pour alimenter l'aspect graphique, nous avons décidé d'utiliser SDL2 une librairie graphique populaire et disponible en C.

Simple DirectMedia Layer (SDL) est une bibliothèque logicielle libre. Son API est utilisée pour créer des applications multimédias en deux dimensions pouvant comprendre du son comme les jeux vidéo, les démos graphiques, les émulateurs, etc. Sa portabilité sur la plupart des plateformes et sa licence Zlib, très permissive, contribuent à son succès.

Nous avons donc dû adapter notre jeu à cette interface en associant à chaque entier de notre matrice de « plateau.c » une image selon sa valeur. Le jeu a été créé en fonction de l'interface graphique et non adapté par la suite à l'interface pour une optimisation maximale de la jouabilité.

Dans le cas de notre projet nous avons décidé de ne pas utiliser SDL\_image. Nous avons donc dû pour chaque image l'importer puis l'imprimer sur une texture pour ensuite l'affecter à un rectangle. Cette longue manipulation aurait pu être évitée avec SDL\_image mais, par choix, nous avons décidé de pas l'utiliser.

## Chapitre 3 – La création du jeu

Pour créer le jeu, nous avons donc eu besoin d'une connaissance parfaite des règles et de l'aspect visuel d'un jeu de dames.

Nous avons donc commencé par créer un plateau qui est une matrice de 10 par 10 entiers, la structure du plateau étant définie dans un header nommé « plateau.h » et la matrice en elle-même étant définie dans « plateau.c ». Les valeurs de la matrice définiront donc le statut des pions se situant dans cette dernière.

Le 0 ne correspondra donc à aucun pion (case vide), le 1 à un pion standard blanc, le 2 à un pion standard noir, le 3 à une dame blanche et le 4 à une dame noire.

```
PS C:\Users\9fran\Documents\JeuxDeDame\src> .\jeux.exe
| 0 | 2 | 0 | 2 | 0 | 2 | 0 | 2 | 0 | 2 |
| 2 | 0 | 2 | 0 | 2 | 0 | 2 | 0 | 2 | 0 |
| 0 | 2 | 0 | 2 | 0 | 2 | 0 | 2 | 0 | 2 |
| 2 | 0 | 2 | 0 | 2 | 0 | 2 | 0 | 2 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
```

*Plateau généré dans l'invité de commande*

Pour interagir avec les pions, il nous a donc fallu la création de fonctions.

Une fonction « bougerCase » permettant de déplacer les pions (qui diffère selon le type de pion concerné par l'exécution de la fonction) cette fonction utilise les coordonnées de la souris pour savoir quel pion nous voulons faire bouger et vers où (diagonale haut gauche ou haut droit).

Nous avons par la suite créé une fonction « MangerCase » qui permet d'éliminer un pion en "sautant par-dessus" si cela est possible et conforme aux règles.

Enfin nous créons une fonction « évoluer » permettant de transformer un pion standard en dame lorsqu'il arrive dans une des cases de la première rangée des pions adverses.

En ce qui concerne le "Main.c", nous l'avons organisé en "chapitre": nous avons créé des rubriques pour ranger nos fonctions selon leurs utilités. Ainsi, au début de notre code, nous avons l'initialisation du plateau de jeu, le lancement de SDL, la création de la fenêtre et du rendu.



Après cela, nous initialisons toutes les textures, nous créons en premier lieu les variables pour les stocker puis à l'aide d'une boucle et de Switch Case nous affectons les images aux variables, si cela est impossible nous appelons la fonction « `SDL_ExitWithError ()` » créée à la fin de notre code pour renvoyer l'erreur dans l'invité de commande.

Sur SDL la création d'image nécessite la création de textures pour les appliquer à des rectangles. Donc à l'aide de deux boucles for nous avons créé des rectangles dans des variables prédéfinies.

Puis viennent les boucles de jeux présentes pour faire interagir les pions créés avec l'utilisateur. Nous avons d'abord fixé une limite de fps (frame per second, soit image par seconde) à l'aide de la méthode « `SDL_LimitFPS ()` » que nous avons créé à la fin du code. Puis nous avons récupéré les coordonnées et le type de clic lorsque l'utilisateur effectue un clic pour pouvoir appeler les fonction « `bougerCase` », « `deselectionnerCase` » et « `selectionnerCase` » afin de sélectionner ou désélectionner un pion avant de le faire bouger selon le clic de l'utilisateur.

À la suite de cela nous avons fait le rendu des pions sur le plateau. Nous avons donc utilisé la même technique que pour le plateau mais cette fois pour chaque pion.

Et tout cela lors de la partie car lorsque qu'elle est terminée, nous arrêtons d'afficher les images à l'aide de la méthode `SDL_DestroyTexture ()` et fermons la fenêtre.

A la fin de notre code, nous avons créé quelques fonctions qui permettent d'initialiser le plateau, la méthode « `SDL_ExitWithError ()` » et la méthode de limite de fps citées auparavant.

## Chapitre 4 – Les éléments graphiques

Nous avons choisi de générer un damier (10 par 10 cases) à l'aide d'une Intelligence artificielle (Dall E 2 en libre accès sur internet et fournis par Open AI).

À la suite de cela nous avons modifier l'image généré par l'IA avec Photoshop afin de faire correspondre au mieux le damier à nos envies et besoins techniques (une dimension de 900 par 900 pixels). Puis nous nous sommes inspirés de modèles de pions trouvé sur internet pour créer les nôtres. Ainsi à l'aide de Photoshop nous avons pu créer respectivement nos 4 pions (pion standard blanc, pion standard noir, dame blanche et dame noire). Pour maximiser la compatibilité avec la librairie graphique SDL2, nous avons donc dû convertir nos images au format .BMP (fichier accepté par la librairie graphique SDL2).



Image de nos pions



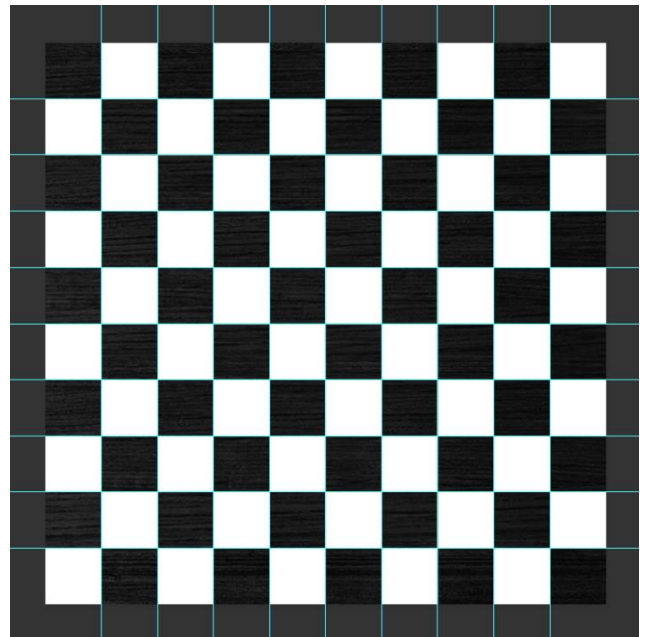
Plateau généré par l'IA [Dall E 2](#)

Malheureusement le plateau que nous avait généré l'IA comportait plusieurs défauts: premièrement le plateau avait 11 lignes or on en avait besoin que de 10 (comme dit précédemment un plateau de dame fait 10 par 10 cases). Secondement il n'était aussi pas parfait, en effet les cases avaient en moyenne 3 ou 4 pixels de plus ou de moins que la taille demandé (90 pixels par 90 pixels par case pour faire un tableau de 900 par 900 pixels finalement).

Nous avons donc dû créer notre propre tableau sur Photoshop via des règles pour que celui-ci soit le plus précis possible. Puis nous avons ajouté une texture bois pour y ajouter un peu de "charme" et coller au plus au plateau originel que nous avait fournis l'IA. Puis nous avons suivi la même démarche de conversion pour l'adapter au mieux à SDL2 soit le convertir en fichier Bitmap (.BMP).



Ancien plateau



Interface Photoshop avec règles pour  
la précision

# Conclusion

Lors de ce projet, nous avons appris à créer un jeu vidéo en C, mais plus globalement à perfectionner notre niveau de connaissance dans ce langage ainsi que dans l'organisation générale d'un code. L'ajout de la librairie SDL2 nous a également apporté des connaissances supplémentaires dans l'adaptation d'un code selon ses contraintes (librairies, langage). L'utilisation d'un cahier des charges nous a permis de maximiser notre efficacité en nous répartissant les tâches de manière optimale pour obtenir un gain d'efficacité conséquent.

Au cours de notre conception du jeu, nous avons fait face à de nombreux problèmes. En effet, la connaissance du C pour ce type de projet nécessite un niveau de maîtrise avancé qui était au-delà de nos capacités initiales. Nous avons donc également dû nous renseigner davantage sur ce langage ce qui a contribué à une augmentation nette de nos capacités de programmations en C. La création de notre plateau devait correspondre à des critères bien précis pour le bon fonctionnement du programme. Là où l'intelligence artificielle aurait pu nous aider sur sa conception, nous nous sommes rapidement rendu compte de l'importance de la précision apportée par le cerveau humain dans ce genre de situation, ne reposant que sur de très fins détails (quelques pixels dans notre cas).

Des améliorations possibles auraient été l'ajout du score de chaque joueur sur le côté de la fenêtre, l'implémentation d'un historique de coups pour permettre aux joueurs de gagner en expérience grâce à l'analyse de leurs parties. Par ailleurs, l'implémentation d'une intelligence artificielle aurait également fait l'objet d'un bon ajout, mais cette dernière nécessiterait une implication plus longue car plus coûteuse sur le plan de la complexité de mise en œuvre.

# Sources

<https://www.libsdl.org/>

<https://openai.com/dall-e-2/>

<http://www.ffjd.fr/Web/index.php?page=reglesdujeu>

[https://www.youtube.com/watch?v=90hGCMC3Chc&list=PLrSOXFDHBtfEh6PCE39HERGgbbaIHhy4j&ab\\_channel=FormationVid%C3%A9o](https://www.youtube.com/watch?v=90hGCMC3Chc&list=PLrSOXFDHBtfEh6PCE39HERGgbbaIHhy4j&ab_channel=FormationVid%C3%A9o)

Support de cours

# Remerciements

**Mohamed Hamroun**

Enseignant CPI 2 – Algorithmique et Structures de Données – C

# Collègues

Élèves de CPI2 des autres groupes