

RAPPORT MINI PROJET PROGRAMMATION C++ « JEUX DE DAMES »

ENSEIGNANTS

Mme Chiraz JLASSI

Mme Imène ELLOUMI

ETUDIANTS

Dhkar Mortadha

Bayoudh Rahma

-2014-

Remerciement

On tient d'abord à remercier nos encadreurs Mme Chiraz JLASSI et Mme Imène ELLOUMI pour avoir bien voulu encadrer ce travail ainsi que pour sa riche contribution et ses précieux conseils

TABLES DES MATIERES

I -INTRODUCTION	3
II -CAHIER DE CHARGE.....	4
2.1 Introduction.....	4
2.2 Liste de toutes les classes créées.....	4
2.3 Les classes représentées par un graphisme UML	5
2.3.1 Diagramme de classe.....	5
2.2.2 Cardinalité entre les classes.....	7
2.2.3 Test.....	8
III -LES DIFFICULTÉS RENCONTRES LORS DU PROJET.....	9
3.1 La recherche du coup obligatoire.....	9
3.2 Le choix de la prise maximale.....	9
3.3 La réalisation de l'interface graphique...9	
3.3 La contrainte de temps.....	9
IV - CONCLUSION.....	11

Tables des figures :

Figure 1 : Diagramme de classe UML (1/2).....	6
Figure 2 : Menu	8
Figure 3 : Entrer les coordonnées des joueurs.....	9
Figure4 : Déroulement du jeu.....	9

Chapitre 1

INTRODUCTION

Le projet qui nous a été confié est de réaliser un jeu de dames en C++. Le but de ce projet est d'utiliser habilement les concepts de la programmation objet, c'est à dire l'héritage entre les classes, les classes abstraites, etc... Nous avons déjà réalisé un projet en langage C au semestre dernier et il a été donc très intéressant de pouvoir remarquer les différences de programmation entre une programmation séquentielle et une programmation objet.

Pour bien répondre au problème, nous allons d'abord présenter les différentes classes qui sont été nécessaires à la réalisation du jeu de dames.

Dans une autre partie, nous parlerons plus précisément du polymorphisme propre aux langages objets. Nous le verrons ici plus particulièrement en action avec le C++.

Enfin, nous exposerons les différents problèmes que nous avons rencontré lors de la programmation et comment nous y avons remédié.

Chapitre 2

Cahier de charge

2.1 Introduction

Les classes sont la base de la programmation objet. Nous avons eu besoin d'utiliser plusieurs classes dans ce projet avec par exemple des relations d'héritage entre elles (nous verrons ceci plus précisément plus tard). Nous allons tout d'abord exposer les différentes classes créées pour la conception du jeu. Ensuite, nous illustrerons nos explications par un graphisme réalisé à l'aide du logiciel PowerAmc pour bien comprendre la composition des classes ainsi que leurs différentes relations.

2.2 Liste de toutes les classes créées

- **la classe Arbitre** : La classe Arbitre est la classe qui contrôle le déroulement du jeu. C'est l'Arbitre qui vérifie que la partie n'est pas terminée, si un coup est valable. Il détermine aussi les coups obligatoires, etc.
- **la classe Plateau** : La classe Plateau décrit le plateau du jeu de dames.
- **la classe Case** : La classe Case est une classe abstraite décrivant une case du plateau du jeu de dames.
- **la classe CaseBlanche** : La classe CaseBlanche décrit une case blanche du jeu de dames. Cette case est bien sûr inactive car les pions ne se déplacent que sur les cases noires.
- **la classe CaseNoire** : La classe CaseNoire décrit une case noire du jeu. Contrairement aux cases blanches, elles sont actives.
- **la classe Rafle** : La classe Rafle est une des classes importantes du jeu. En effet, elle permet de calculer la rafle maximale et de la signaler au joueur. Une rafle au jeu de dames est donc symbolisée par une liste de Rafle. Chaque élément (objet) contient la Case du début du coup, la case prise (c'est à dire la case du dernier pion pris) du coup précédent et un vecteur contenant la liste des cases suivantes possibles. En effet, il se peut que pour une case donnée, on ait la possibilité d'obtenir plusieurs autres coups possibles.
- **La classe Historique Coup** : Cette classe sert bien sûr à gérer l'historique des coups.
- **la classe Pièce** : La classe Pièce est une autre classe abstraite. Elle symbolise une pièce du jeu. En l'occurrence ici, il s'agira soit d'un pion, soit d'une dame.
- **la classe Pion** : La classe Pion symbolise un pion du jeu de dames.
- **la classe Dames** : La classe Dame représente une dame. Elle est un peu plus complexe que la classe pion car le déplacement d'une dame est plus difficile à étudier que celui

d'un pion.

- **la classe Joueur** : A nouveau une classe abstraite. Elle décrit un joueur du jeu. Ce joueur peut aussi bien être un humain que l'ordinateur.
- **la classe Humain** : La classe Humain décrit un joueur du type Humain.
- **la classe Ordinateur** : La classe Ordinateur représente bien sûr un joueur contrôlé par l'ordinateur pendant la partie.
- **la classe JeuDeDamesWindow** : Cette classe hérite de la classe JComponent comme la classe Case vu précédemment. Elle gère la fenêtre du jeu de dames et l'affichage des composants de la fenêtre.
- **la classe NouvellePartieWindow** : Comme son nom l'indique, la classe NouvellePartie décrit la fenêtre permettant de commencer une nouvelle partie.
- **la classe APropos** : C'est la classe qui décrit la fenêtre APropos. Cette fenêtre donne des renseignements sur le jeu ; ses concepteurs par exemple ou d'autres indications.

2.3 Les classes représentées par un graphisme UML

Cette idée d'utiliser le logiciel PowerAmc nous a permis d'obtenir une programmation très structurée. En effet, nous avons eu la possibilité d'illustrer toutes les classes avec leurs attributs et leurs méthodes. Nous avons pu facilement remarquer les différents liens que nous devons instaurer entre les classes.

2.2.1 Le diagramme des classes

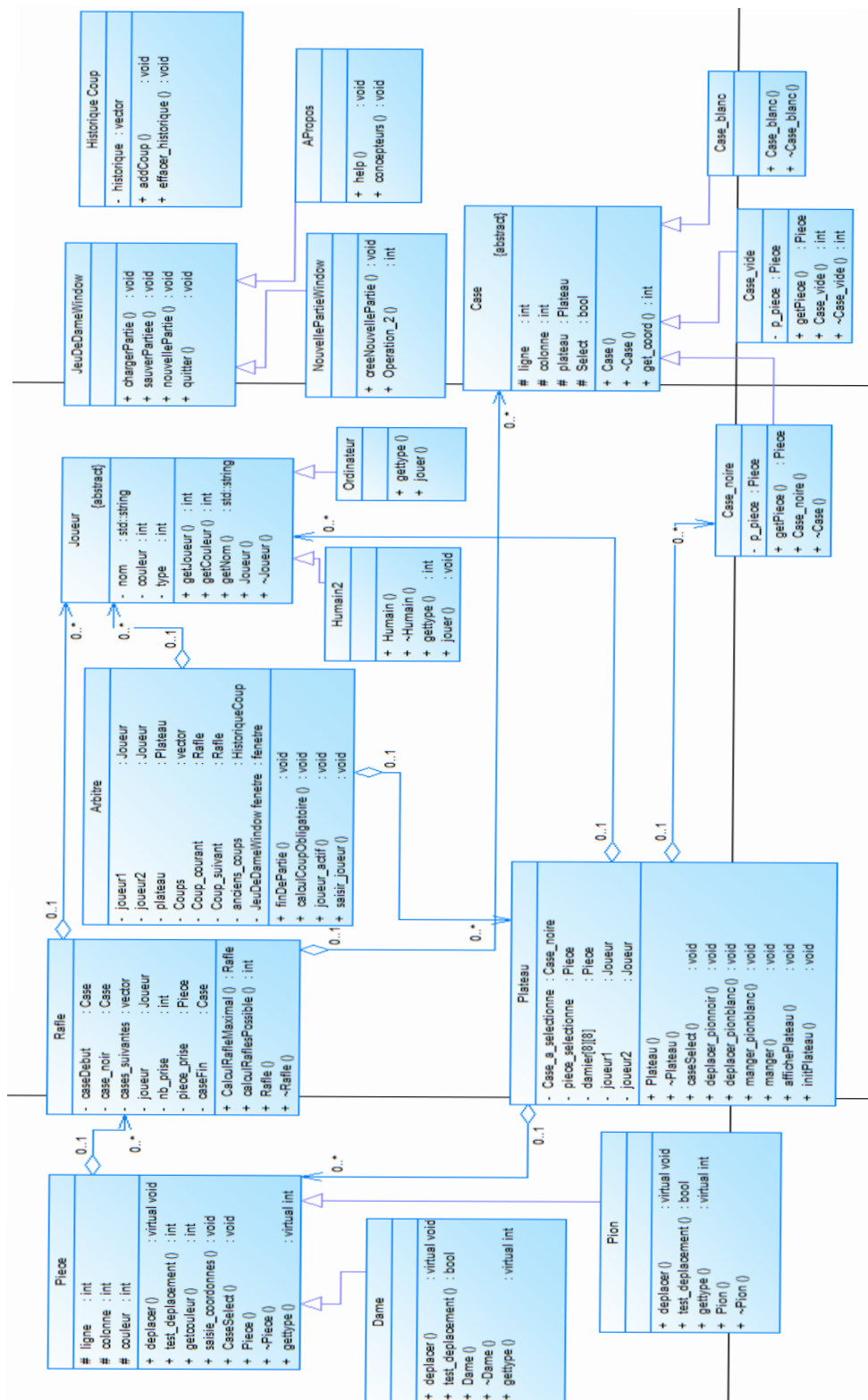


Figure1 : Diagramme de classe

2.2.2 Cardinalité entre les classes

Chaque partie possède deux joueurs et un échiquier

Chaque damier possède 64 cases

Chaque joueur participe dans une partie

Chaque pièce est posée sur une case

2.2.3 Test



Figure 2 : Menu

```

1
entrez le nom de joueur:mortada
choix de couleur : taper 0 pour Noir [X] ou 1 pour Blanc [O]
0
vous avez choisie les pions Noirs [X]
entrez le nom de joueur:rahma
choix de couleur : taper 0 pour Noir [X] ou 1 pour Blanc [O]
1
vous avez choisie les pions Blanc [O]

  0   1   2   3   4   5   6   7
-----
|   | X |   | X |   | X |   | X | 0
| X |   | X |   | X |   | X |   | 1
|   | X |   | X |   | X |   | X | 2
|   |   |   |   |   |   |   |   | 3
|   |   |   |   |   |   |   |   | 4
| 0 |   | 0 |   | 0 |   | 0 |   | 5
-----

```

Figure 3 : Entrer les coordonnées des joueurs

```

  0   1   2   3   4   5   6   7
-----
|   | X |   | X |   | X |   | X | 0
| X |   | X |   | X |   | X |   | 1
|   |   |   | X |   | X |   | X | 2
|   |   |   |   |   | 0 |   |   | 3
|   | X |   |   |   |   |   |   | 4
| 0 |   |   |   | 0 |   | 0 |   | 5
|   | 0 |   | 0 |   | 0 |   | 0 | 6
| 0 |   | 0 |   | 0 |   | 0 |   | 7
-----
piece 0 a déplacer <ligne><colonne> =

```

Figure 4: Déroulement du jeu

Chapitre 3

LES DIFFICULTÉS RENCONTRÉES LORS DU PROJET

Dans ce chapitre, nous allons aborder les difficultés auxquelles nous nous sommes heurtés lors de la réalisation de ce projet.

3.1 La recherche du coup obligatoire

Cette partie a été la plus dure à réaliser. La recherche de la prise maximale est complexe à obtenir. En effet, il n'est déjà pas évident de trouver la prise maximale, mais il est encore plus difficile d'enregistrer toutes les rafles équivalentes (même nombre de prises). Nous avons tout d'abord fait un algorithme qui recherche la rafle maximale sans se préoccuper des égalités. C'est une fois celui-ci effectué qu'on l'a amélioré pour qu'il puisse gérer les diverses possibilités de rafles, mais malheureusement il n'a pas nous donné le résultat attendu.

3.2 Le choix de la prise maximale

On sait que dans le jeu de dames, la rafle maximale, si elle existe doit être jouée. Par contre, si il existe plusieurs rafles effectuant le même nombre de prises, le joueur a le choix d'effectuer celle qui semble la plus pertinente.

3.3 La réalisation de l'interface graphique

On voulu réaliser une interface graphique pour notre mini-projet dans le sens manipuler un nouveau logiciel et de reconnaître de nouveaux syntaxe autre que celui de la programmation mais la contrainte du temps nous a empêché d'exprimer ceux-ci soit avec SDL ou Qt. Donc, vu tout cela nous avons juste essayé de créer une petite fenêtre de menu avec quelques boutons mais qui ne permet pas à joueur de lancer une partie.

3.3 La contrainte de temps

Que le temps passe vite ! Dire que nous avons déjà trop de travail à faire...et le jour de l'exposition se rapproche à grands pas ! Les décisions sont à peine prises qu'il faut déjà avoir terminé le projet. Le temps passe, laissant derrière lui de longues nuits, mais il y a tellement de décisions à prendre, tellement de détails qu'il faut organiser, la pression monte, le jour J est bientôt là. Et toujours, le temps passe...

Chapitre 4

CONCLUSION

Ce jeu de dames en langage C++ fut très intéressant à concevoir et à programmer. Ce projet nous a permis de bien comprendre les qualités de la programmation objet. Nous avons essayé d'utiliser au mieux ses caractéristiques telles que l'héritage ou les classes abstraites.

Ce jeu de dames fût très intéressant à réaliser, car il nous a permis de bien comprendre le fonctionnement des pointeurs. On a pu remarquer que une programmation structurée était vraiment efficace lorsque la longueur et la complexité du code devient conséquente.

Au début du projet, nous avons trouvé les différentes classes et méthodes à créer grâce à l'utilisation de scénarios et de diagrammes de classes. Cette démarche permet d'avoir une réflexion structurée tout au long de la programmation.

Au niveau des améliorations possibles du projet, il serait également plus agréable de faire une interface graphique, il serait également très intéressant de pouvoir mettre le jeu en réseau.

