Jeu d'échec sur Unity

Dossier de projet - TPI

Écrit par : **Arthur Bottemanne**

Chef de projet : **Loïc Viret**

Premier expert : **Roberto Ferrari**

Deuxième expert : **Claude-Albert Muller Theurillat**

cpnv
Centre professionnel
du Nord vaudois

Glossaire :

Pièce clouée : Au échecs, une pièce clouée est définie comme une pièce menacée ne pouvant pas se déplacer sans exposer une autre pièce de plus grande valeur à une capture.

Moteur de jeu : Un moteur de jeu est un logiciel qui facilite le développement de jeux vidéo en mettant à disposition des outils tels que la simulation de physique, le rendu graphique, la gestion des entrées utilisateur, et autres fonctionnalités.

Unity : Unity est un moteur de jeu multiplateforme en deux dimensions et en trois dimensions.

Table des matières

1 Analyse préliminaire

1.1 Introduction

Ceci est le dossier de projet pour mon TPI qui est réalisé dans le cadre du CPNV. Pour ce projet, 90 heures sont mises à disposition pour effectuer ce projet.

L'objectif est de créer un jeu d'échecs représenté en deux dimensions en utilisant Unity comme moteur de jeu. Les règles officielles devront être implémenter et fonctionnel, pour but que deux joueurs s'affrontent dans une partie.

Les parties seront chronométrer ou le temps dépendra du choix des joueurs, et les parties seront enregistrer dans un fichier en notation algébrique

1.2 Objectifs

L'objectif de ce projet est de développer le jeu des échecs en deux dimensions à l'aide du moteur de jeu Unity.

Le jeu permettra de jouer aux échecs à deux sur le même ordinateur, incluant les coups spéciaux, il permettra aussi de fournir le chronomètre pour chaque joueur afin de pouvoir faire des parties « officielles » ainsi que des parties rapides.

Le jeu doit permettre à deux joueurs de s'affronter aux échecs avec les règles officielles incluant :

1. Ce sont toujours les blancs qui commencent. Les joueurs choisiront eux même qui commence entre les deux.
2. Chaque joueur doit être capable de ne bouger que ses propres pièces
3. Il doit être possible d'activer/désactiver la prévision d'où pourront se déplacer les pièces en fonction de leur type.
4. Les coups spéciaux doivent être implémentés (Roque, prise en passant et promotion)
5. Un joueur ne doit pas pouvoir déplacer une pièce clouée
6. Les conditions de victoire doivent être implémentées (échec, échec et mat, pat, 50 coups, triple répétition, accord et temps)
7. Un fichier de récapitulation de la partie en notation algébrique résumera la partie.

Pour les règles du jeu d'échecs, le candidat peut se baser sur cette page :
https://fr.wikipedia.org/wiki/R%C3%A8gles_du_jeu_d%27%C3%A9checs

Pour la notation algébrique, le candidat peut se baser sur cette page :
https://fr.wikipedia.org/wiki/Notation_alg%C3%A9brique

Pour les objectifs, il existe aussi 7 points techniques qui seront évalués pour ce projet.

1. Déplacement des pièces

    - Respect des différents déplacements des pièces
    - Prévision des positions possibles lors d'un déplacement
    - Ergonomie du déplacement

2. Gestion du tour des joueurs

    - Identification du joueur à qui est le tour
    - Possibilité de ne déplacer que ses pièces
    - Pièce clouée

3. Coups spéciaux

    - Le petit roque
    - Le grand roque
    - La prise en passant

4. Promotion

    - Le déclanchement de la promotion
    - Les possibilités de promotion
    - L'ergonomie de la fonctionnalité

5. Conditions de victoire

    - La gestion de la mise en échec
    - La validation de l'échec et mat
    - Le pat

6. Le fichier récapitulatif

    - Les code de pièces
    - Les cases concernées
    - Les prises

7. Setup fonctionnel avec son protocole d'installation

    - Sans erreur sur une machine Windows 10 64bits
    - Pertinence du nom de l'exécutable et de son emplacement
    - Protocole d'installation clair

1.3 Planification initiale

N°	Mode Tâche	Nom de la tâche	Durée	Début	Fin
1		**Sprint 1**	**3 jrs**	**Mar 30.04.24**	**Ven 03.05.24**
2		Planification initiale	1 jr	Mar 30.04.24	Mar 30.04.24
3		Documentation analyse préliminaire	1 jr	Mar 30.04.24	Mar 30.04.24
4		Création du diagramme de classes	1 jr	Jeu 02.05.24	Jeu 02.05.24
5		Création des diagrammes de flux	1 jr	Jeu 02.05.24	Jeu 02.05.24
6		Création des maquettes	1 jr	Ven 03.05.24	Ven 03.05.24
7		Création des diagrammes de séquences	1 jr	Ven 03.05.24	Ven 03.05.24
8		**Sprint 2**	**2 jrs**	**Lun 06.05.24**	**Mar 07.05.24**
9		Écriture des scénarios de tests	1 jr	Lun 06.05.24	Lun 06.05.24
10		Création et configuration du projet Unity	1 jr	Lun 06.05.24	Lun 06.05.24
11		Implémentation des menus et graphiques	2 jrs	Lun 06.05.24	Mar 07.05.24
12		Implémentation de l'échiquier et le mouvement des pièces	1 jr	Mar 07.05.24	Mar 07.05.24
13		**Sprint 3**	**4 jrs**	**Lun 13.05.24**	**Ven 17.05.24**
14		Implémentation de l'échiquier et le mouvement des pièces	1 jr	Lun 13.05.24	Lun 13.05.24
15		Implémentation des coups spéciaux	2 jrs	Mar 14.05.24	Jeu 16.05.24
16		Implémentation des promotions	1 jr	Ven 17.05.24	Ven 17.05.24
17		**Sprint 4**	**3 jrs**	**Mar 21.05.24**	**Ven 24.05.24**
18		Implémentation des promotions	1 jr	Mar 21.05.24	Mar 21.05.24
19		Implémentation de la gestion du tour des joueurs	1 jr	Jeu 23.05.24	Jeu 23.05.24
20		Implémentation des conditions de victoire	1 jr	Ven 24.05.24	Ven 24.05.24
21		**Sprint 5**	**3 jrs**	**Lun 27.05.24**	**Jeu 30.05.24**
22		Implémentation du fichier récapitulatif	1 jr	Lun 27.05.24	Lun 27.05.24
23		Réalisation des tests exhaustif finaux	1 jr	Mar 28.05.24	Mar 28.05.24
24		Finalisation du projet et de la documentation	1 jr	Jeu 30.05.24	Jeu 30.05.24

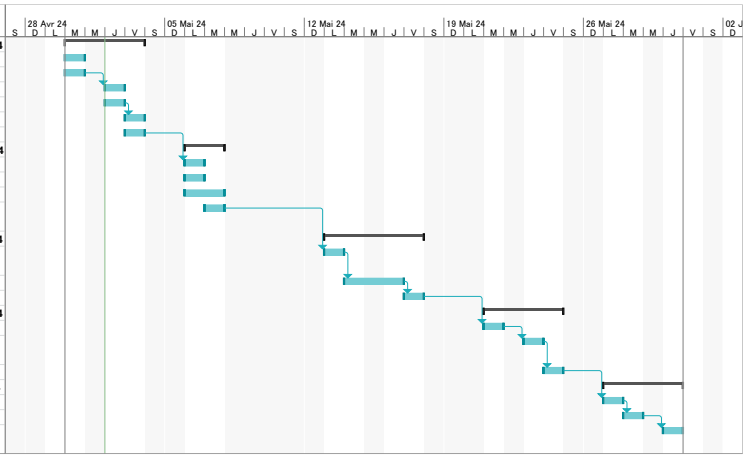


Figure 1: Planification initiale

La planification initiale a été réalisé en planifiant autours du nombre de jours qu'une tâche prendrait à finir pour avoir un aperçu plus général. Cela veut dire qu'une tâche pourrait être indiqué qu'elle prend une journée à compléter, mais soit en réalité plus rapide à faire.

Les tâches de tests et de documentation sont aussi prises en compte lors des tâches d'implémentation, car j'ai jugé que durant l'implémentation, les documentations et les tests nécessaire seront fait simultanément.

Les tâches de tests et de documentation seront planifiées durant la révision de la planification initiale.

2 Analyse / Conception

2.1 Concept

2.1.1 Diagrammes de flux

Pour cette conception, cinq diagrammes de flux ont été réalisé pour démontrer la logique des aspects important du programme.

Les diagrammes de flux représentent respectivement :

- La gestion du mouvement des pièces
- La gestion des mouvements spéciaux (roque, en passant)
- La gestion du tour des joueurs
- Les conditions de fin de partie
- La gestion du fichier récapitulatif

Avec ces diagrammes de flux, on peut avoir un aperçu entier sur la logique du programme.

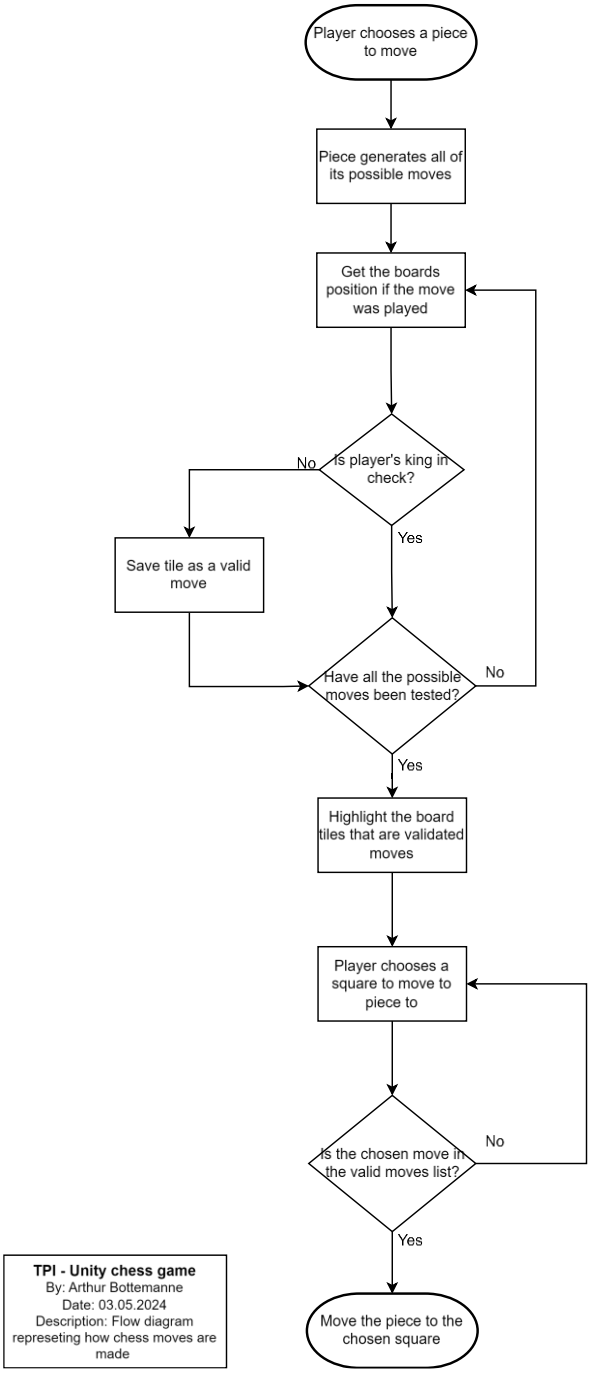


Figure 2: Diagramme de flux de la gestion du mouvement des pièces

Ce diagramme représente la logique de la génération de mouvement d'une pièce et de la validation du mouvement.

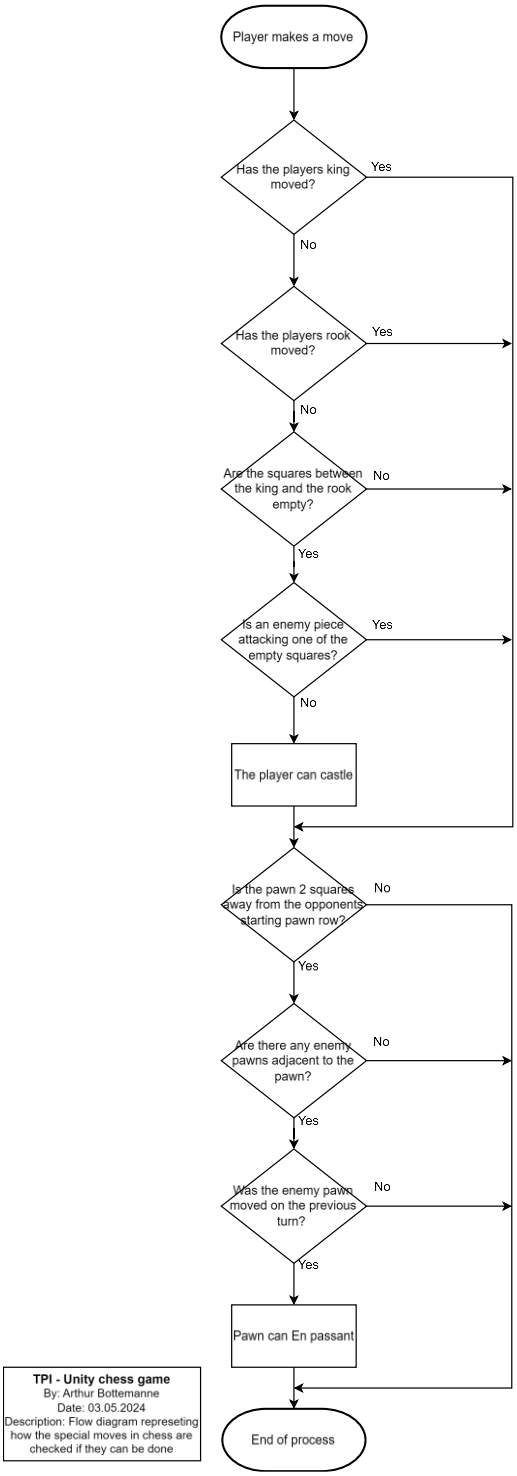


Figure 3: Diagramme de flux de la gestion des mouvements spéciaux

Ce diagramme représente les vérifications des conditions pour le roque et l'en passant aux échecs.

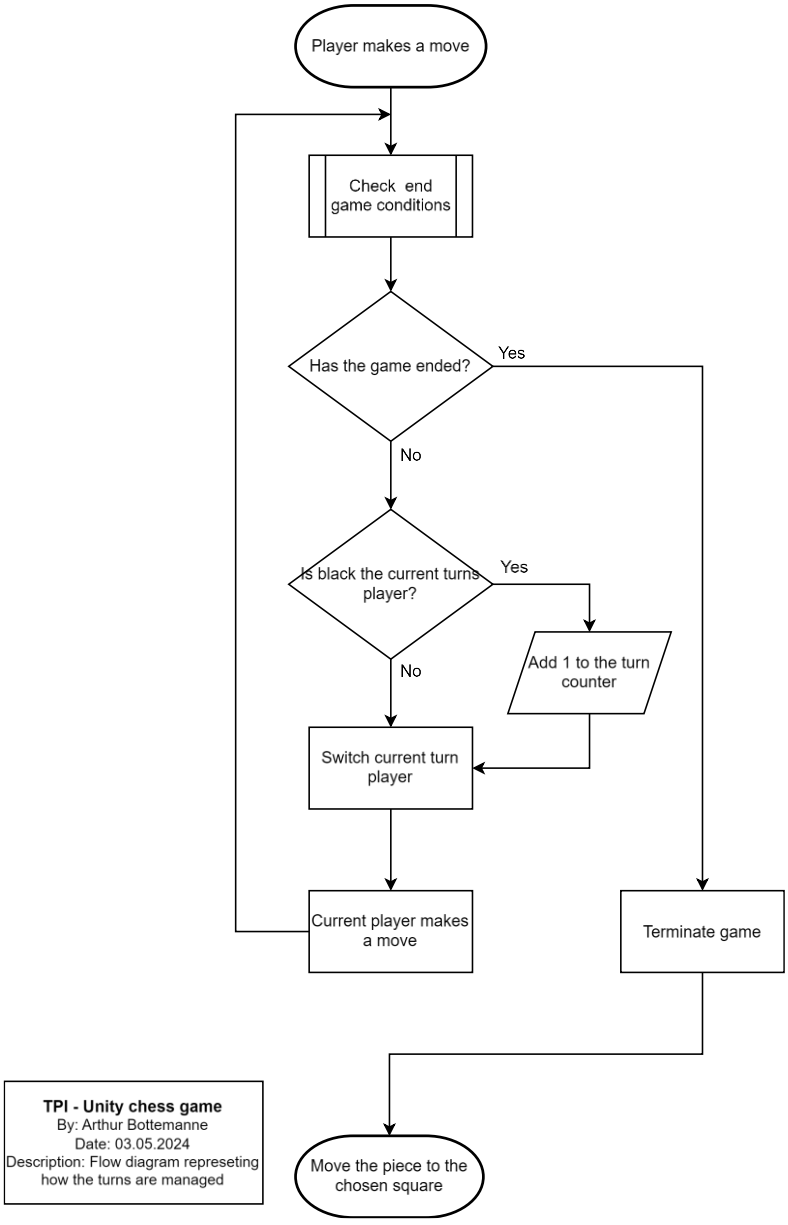


Figure 4: Diagramme de flux de la gestion du tour des joueurs

Ce diagramme représente la logique pour le changement et le compte des tours.

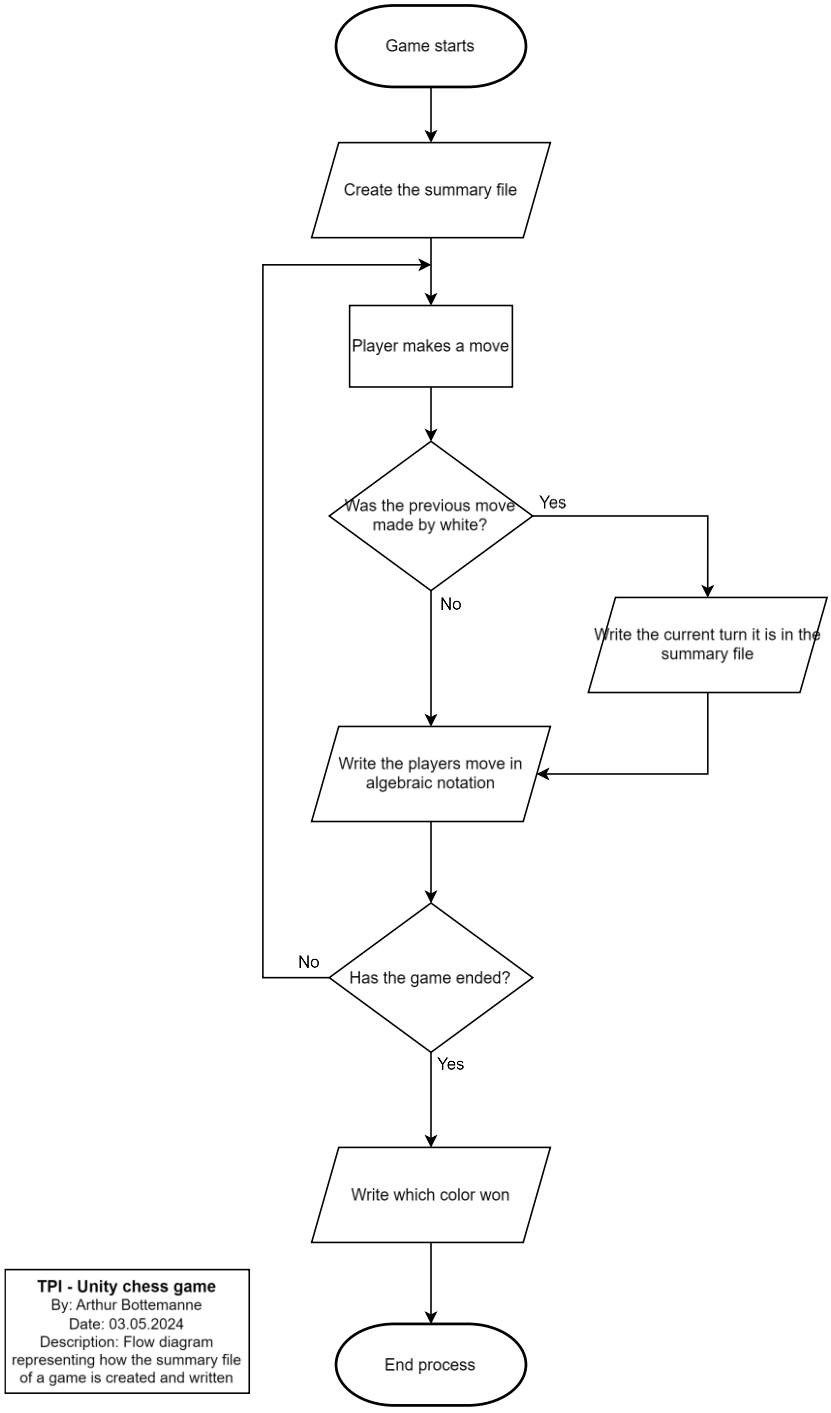


Figure 5: Diagramme de flux de la gestion du fichier récapitulatif

Ce diagramme représente la logique du fichier récapitulatif, quand il écrit les coups et note les tours.

2.1.2 Maquettes

Cinq maquettes ont été réaliser pour ce jeu :

- Le menu principal
- La sélection du chronomètre
- La partie en cours
- La proposition d'un match nulle
- Le message de fin de partie

Voici ci-dessous les maquettes :

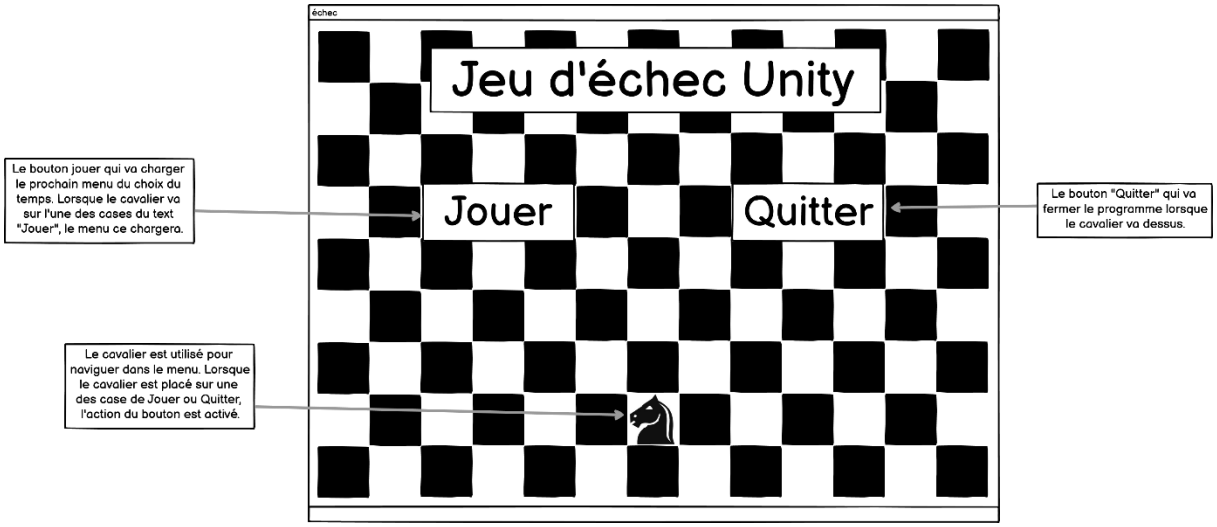


Figure 6: Maquette du menu principal

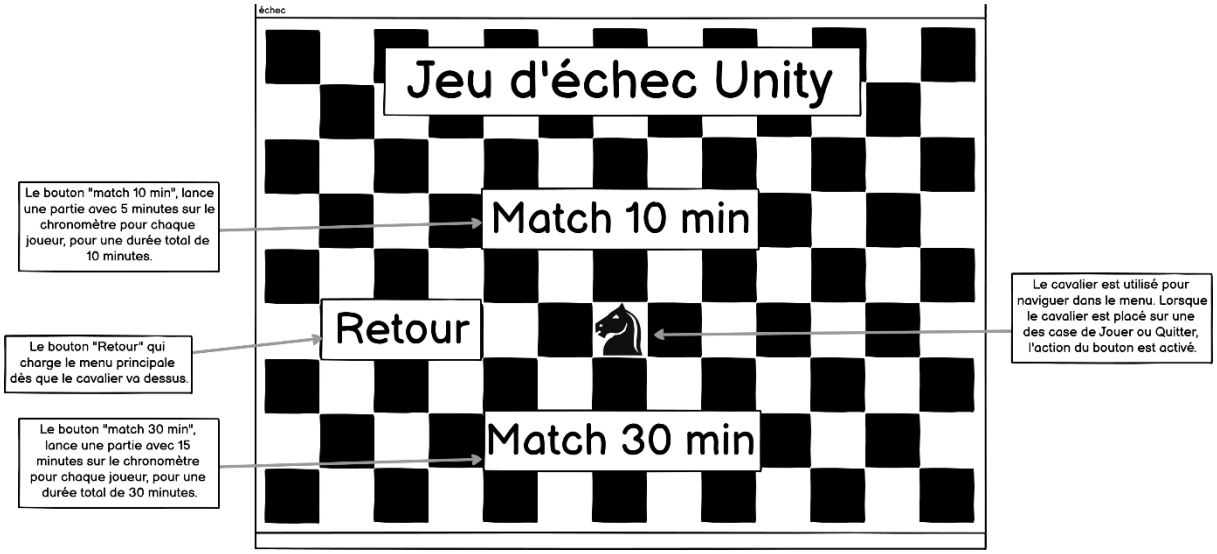


Figure 7: Maquette de la sélection du chronomètre

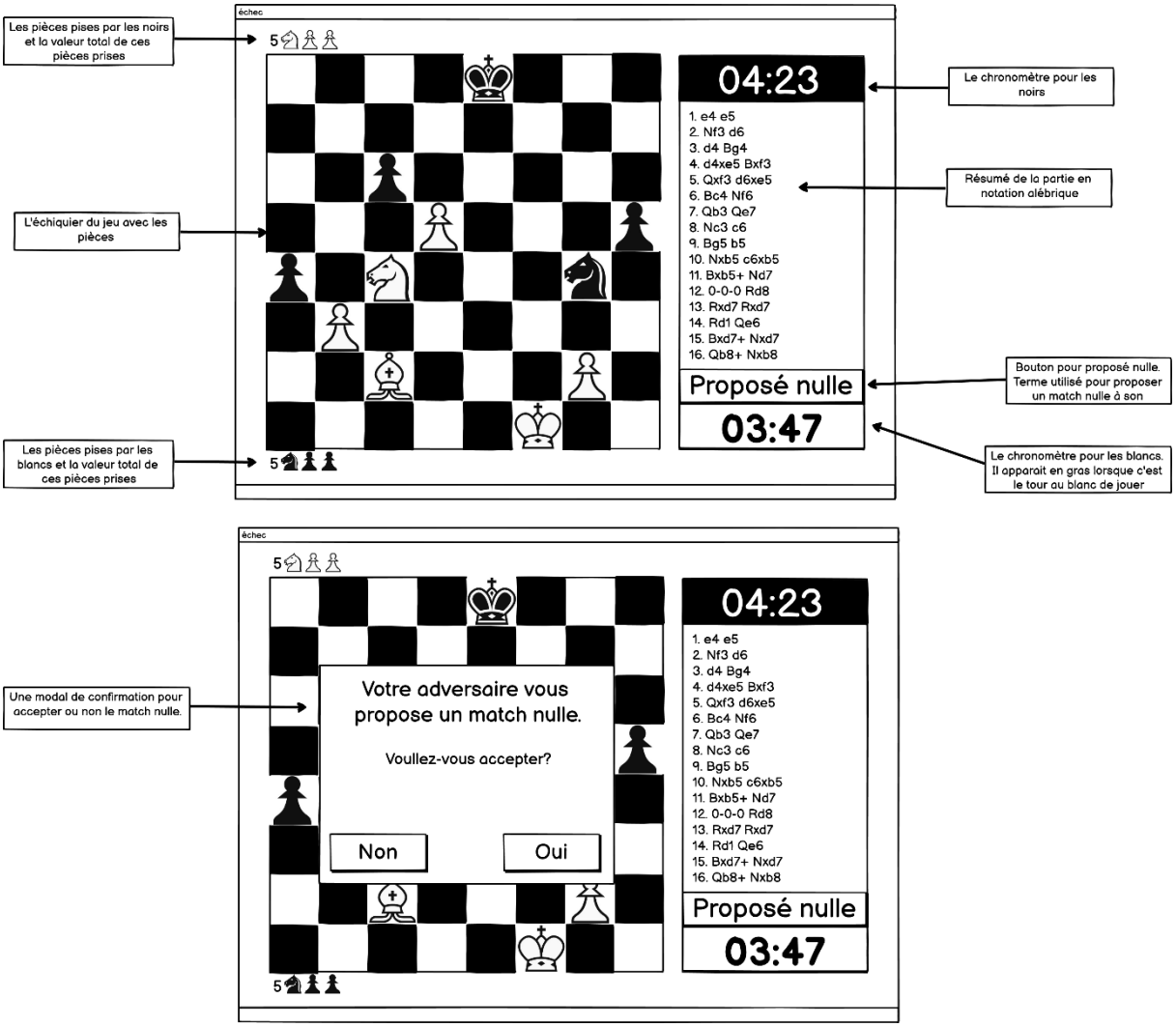


Figure 8: Maquette d'une partie en cours

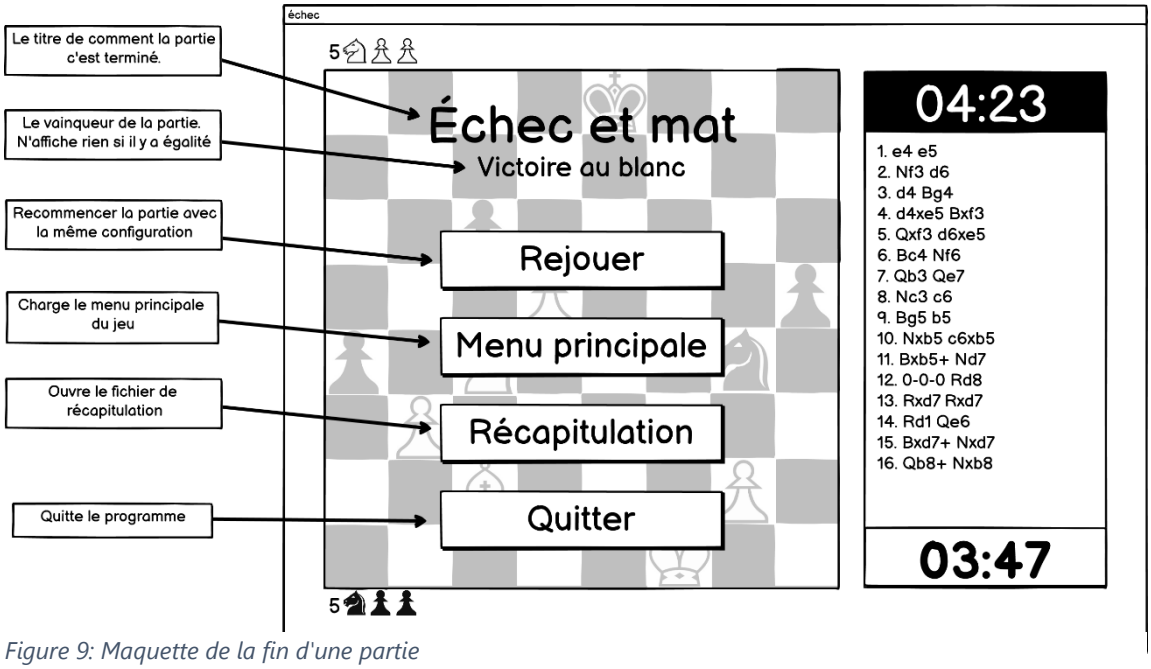


Figure 9: Maquette de la fin d'une partie

2.2 Stratégie de test

La stratégie de test comporte deux parties, les tests unitaires automatisés et les tests de bout en bout manuels.

Les tests unitaires automatisés :

Pour les tests unitaires, le « Test Runner » de Unity sera utilisé en « Play mode ». Les tests couveront les fonctionnalités essentielles du programme. La raison étant le temps alloué pour la réalisation des tests et mon inexpérience avec les tests automatisés avec Unity. Donc une grande majorité des tests pour assurer le bon fonctionnement du programme seront des tests de bout en bout manuels. Ces tests seront écrits et exécuté tout du long de la réalisation du projet.

Les tests d'intégrations ne sont pas prévus d'être réalisé vu mon inexpérience avec les tests de Unity et le temps alloué pour les réaliser. Le but étant de garder les tests unitaires pour les fonctionnalités nécessaires, cependant, si le temps le permet, il est possible que certains tests d'intégrations soient réalisés.

Les tests de bout en bout manuels :

Ces tests garantissent de manière exhaustif le bon fonctionnement du programme. Les scénarios de tests ainsi que les cas de tests seront définis au préalable avant que les tests soient effectués. Ces tests seront réalisés moins fréquemment que les tests unitaires, mais sont impératif pour assurer que le programme fonctionne correctement vu les tests unitaires non-exhaustif.

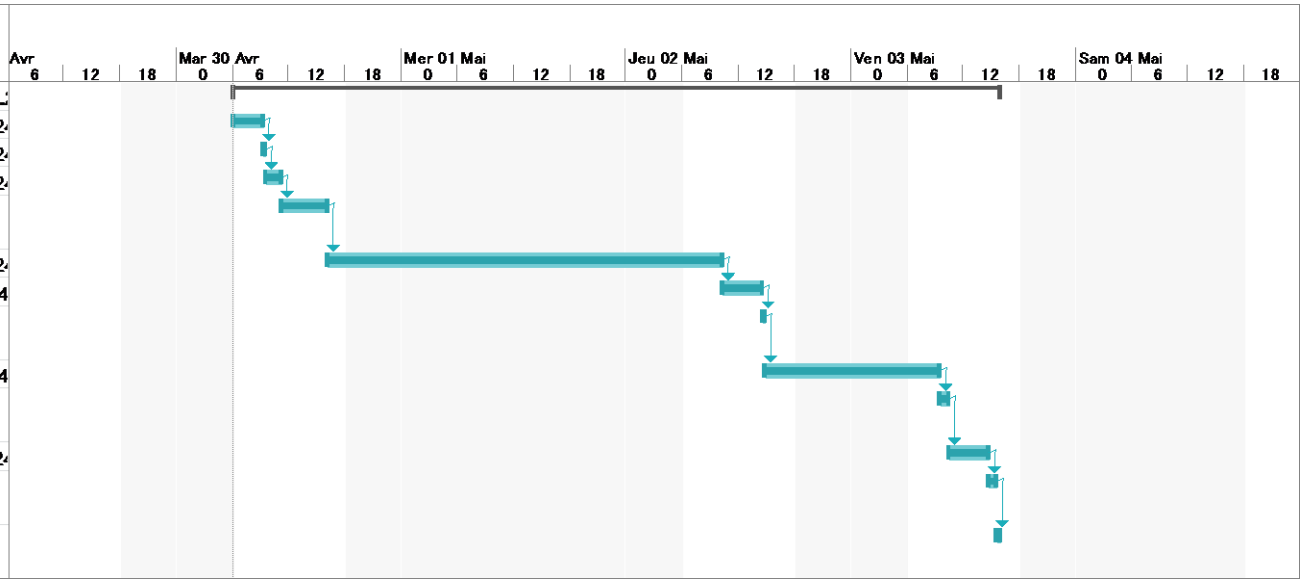
2.3 Risques techniques

Le premier risque majeur serait mes compétences de Unity. Bien que j'aie utilisé Unity quelques fois avant ce projet, il me manque tout de même des connaissances. Pour certaines implémentations que je n'ai pas faites précédemment, cela pourrait se montrer difficile à réaliser. Pour remédier à ce risque, des recherches approfondies au préalables sur Unity seront nécessaire pour bien comprendre comment implémenter la fonction désirée dans Unity, les différentes manières possibles d'implémenter et la meilleure manière de le savoir.

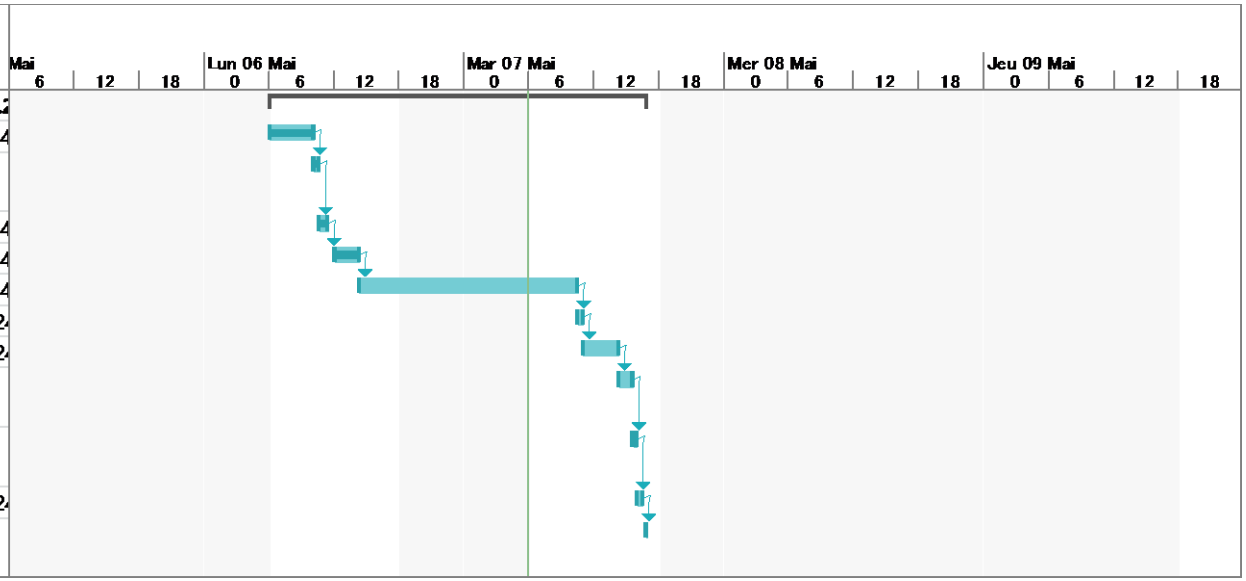
Le deuxième risque que j'anticipe est comment m'assurer que mon programme correspond à 100% avec les règles des échecs. Ceci est compliqué car il y a environ 10^{40} positions légales aux échecs. Dans ces positions, chaque pièce à ses propres règles de déplacement et de capture ce qui peux créer des configurations complexes et avec de nombreuses interactions. Et dans ces configurations, tester et assurer le bon fonctionnement du programme nécessite une analyse minutieuse et exhaustive. Pour éviter ces erreurs, je vais devoir tester le programme avec une variété de positions et de scénarios pour m'assurer qu'il fonctionne correctement dans toutes les situations possibles.

2.4 Planification détaillée

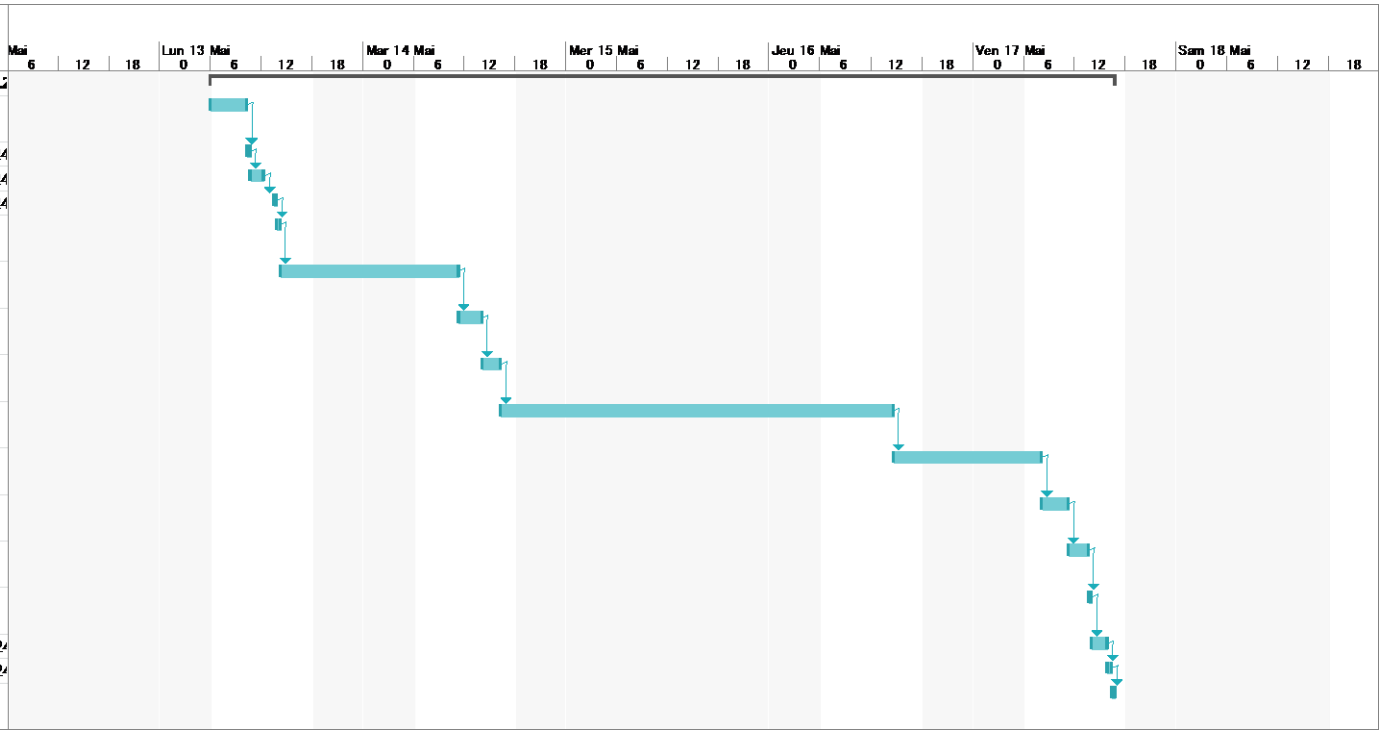
N°	Nom de la tâche	Durée	Début
1	**Sprint 1**	**18.83 heur**	**Mar 30.04.2**
2	Rendez-vous avec le premier exp	70 min	Mar 30.04.2
3	Création du journal de travail	20 min	Mar 30.04.2
4	Réalisation de la planification init	85 min	Mar 30.04.2
5	Rédaction de l'analyse préliminaire du dossier de projet	205 min	Mar 30.04.24
6	Mise en place du GitHub	120 min	Mar 30.04.2
7	Création du diagramme de classe	180 min	Jeu 02.05.24
8	Rédaction du diagramme de classes dans le dossier de projet	15 min	Jeu 02.05.24
9	Création des diagrammes de flux	150 min	Jeu 02.05.24
10	Rédaction des diagrammes de flux dans le dossier de projet	50 min	Ven 03.05.24
11	Création des maquettes	180 min	Ven 03.05.2
12	Rédaction des maquettes dans le dossier de projet	30 min	Ven 03.05.24
13	Début de la création des diagrammes de séquences	25 min	Ven 03.05.24



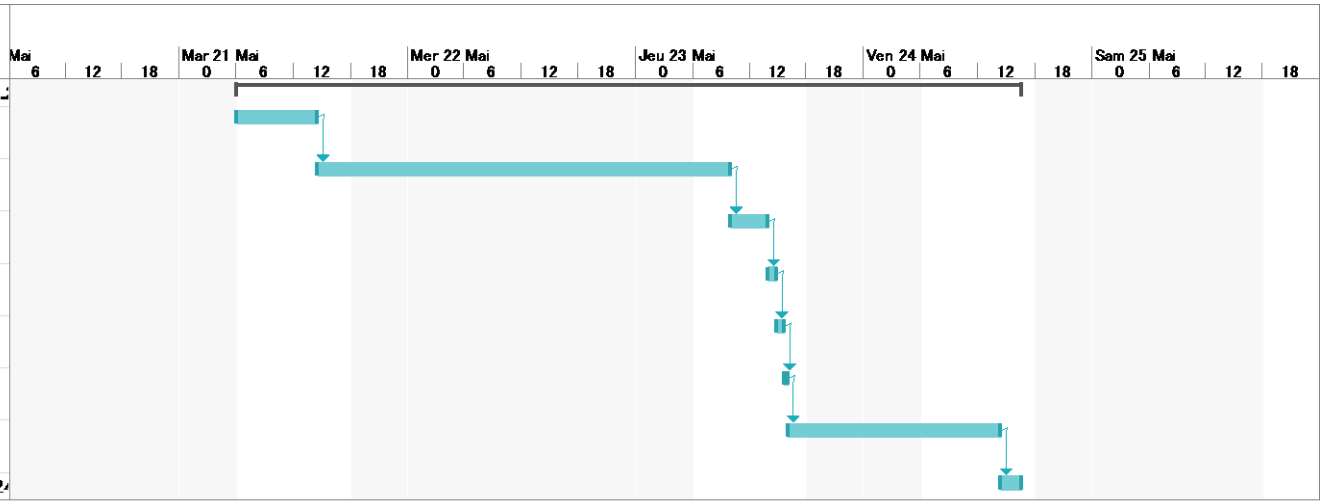
N°		Nom de la tâche	Durée	Début
14		**Sprint 2**	**12.75 heur**	**Lun 06.05.2**
15	✓	Création des diagrammes de séqu	115 min	Lun 06.05.24
16	✓	Rédaction des diagrammes de séquences dans le dossier de	30 min	Lun 06.05.24
17	✓	Rédaction des stratégies de tests	50 min	Lun 06.05.24
18	✓	Rédactions des risques technique	65 min	Lun 06.05.24
19		Création de la planification détail	180 min	Lun 06.05.24
20		Rédaction de la planification déta	30 min	Mar 07.05.2
21		Création des scénarios de tests	120 min	Mar 07.05.2
22		Création et configuration basique du projet Unity	60 min	Mar 07.05.24
23		Rédaction des spécification de la configuration du projet	30 min	Mar 07.05.24
24		Ajout des assets dans le projet	30 min	Mar 07.05.2
25		Rédaction des assets ajoutés dans le dossier de projet	15 min	Mar 07.05.24



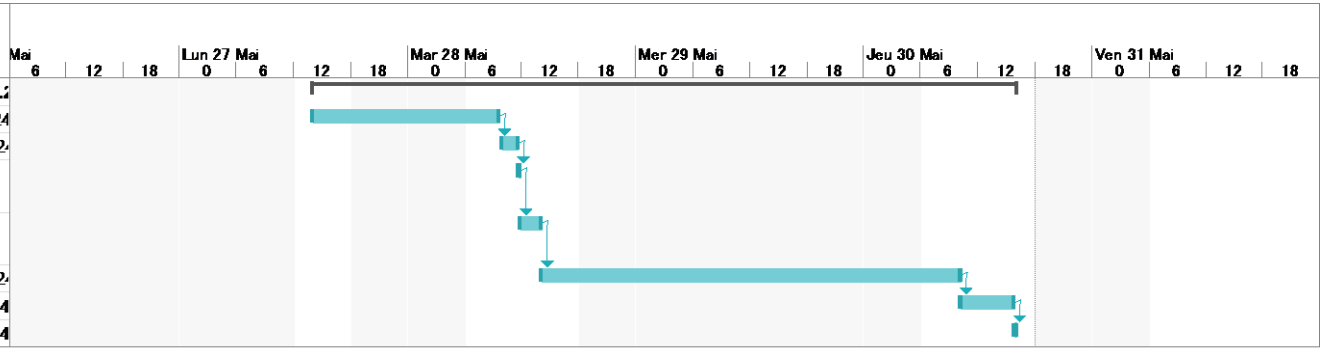
N°	Nom de la tâche	Durée	Début
26	**Sprint 3**	**25.25 heur**	**Lun 13.05.2**
27	Implémentation d'une version simple du menu dans le projet	120 min	Lun 13.05.24
28	Implémenter les tests du menu	30 min	Lun 13.05.24
29	Implémentation de l'échiquier da	90 min	Lun 13.05.24
30	Implémentation des tests de l'éch	20 min	Lun 13.05.24
31	Rédaction de l'implémentation de l'échiquier dans le dossier de	25 min	Lun 13.05.24
32	Implémentation du mouvements des pièces dans Unity	230 min	Lun 13.05.24
33	Implémentation des tests du mouvements des pièces	90 min	Mar 14.05.24
34	Rédaction de l'implémentation du mouvements des pièces	120 min	Mar 14.05.24
35	Implémentation du menu avec les déplacements du cavalier	300 min	Mar 14.05.24
36	Rédactoin du menu dans le dossier de projet	75 min	Jeu 16.05.24
37	Implémentation des coups spéciaux dans Unity	180 min	Ven 17.05.24
38	Implémentation des tests des coups spéciaux	60 min	Ven 17.05.24
39	Rédaction des coups spéciaux dans le dossier de projet	30 min	Ven 17.05.24
40	Implémentation des promotions d	90 min	Ven 17.05.24
41	Implémentation des tests des pro	30 min	Ven 17.05.24
42	Rédaction des promotions dans le dossier de projet	25 min	Ven 17.05.24



N°	Nom de la tâche	Durée	Début
43	**Sprint 4**	**19.5 heures**	**Mar 21.05.2**
44	Implémentation de la validation des mouvements des pièces	300 min	Mar 21.05.24
45	Implémentation des tests de la validation des mouvements des	180 min	Mar 21.05.24
46	Rédaction de la validation des mouvements des pièces	160 min	Jeu 23.05.24
47	Implémentation de la gestion des tours dans Unity	60 min	Jeu 23.05.24
48	Implémentation des tests de la gestion des tours	30 min	Jeu 23.05.24
49	Rédaction de la gestion des tours dans le dossier de projet	20 min	Jeu 23.05.24
50	Implémentation de la fin de partie dans Unity	300 min	Jeu 23.05.24
51	Implémentation des tests de la fir	120 min	Ven 24.05.2



N°	Nom de la tâche	Durée	Début
52	**Sprint 5**	**13.75 heur**	**Lun 27.05.2**
53	Rédaction de la fin de partie	160 min	Lun 27.05.24
54	Implémentation du fichier récapit	100 min	Mar 28.05.2
55	Implémentation des tests du fichier récapitulatif	20 min	Mar 28.05.24
56	Rédaction du fichier récapitulatif dans le dossier de projet	60 min	Mar 28.05.24
57	Rédaction de la réalisation des te	220 min	Mar 28.05.2
58	Rédaction de la conclusion	250 min	Jeu 30.05.24
59	Rendu du projet	15 min	Jeu 30.05.24



2.5 Dossier de conception

2.5.1 Diagramme de classes

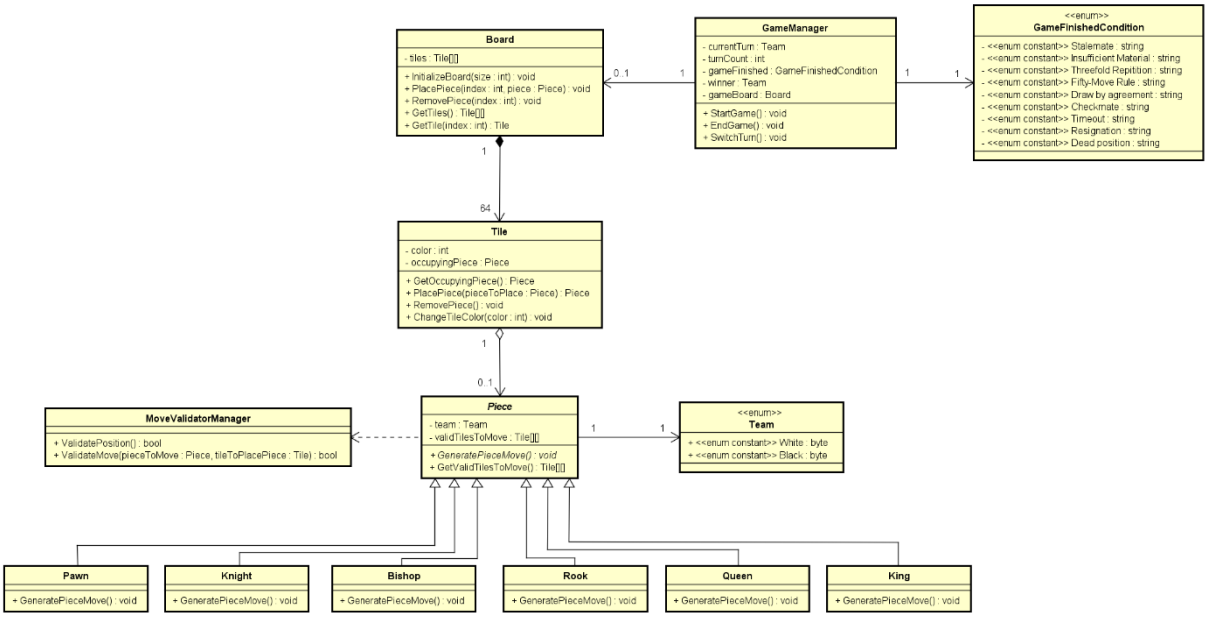


Figure 10: Diagramme de classes

Le diagramme de classes ci-dessus représente la structure des classes que j'envisage d'implémenter. À la fin de la réalisation du projet, un diagramme de classes à jour sera comparé à celui-ci.

Dans ce diagramme, la classe « GameManager » gère les fonctionnalités de condition de victoire et de tour, le « Board » représente l'échiquier et agis sur les mouvements des pièces et les classes des pièces (« Pawn », « Bishop » etc…) utilise des concepts de polymorphisme pour implémenter leur propre méthode de mouvement.

2.5.2 Diagrammes de séquences

Les diagrammes de séquences réalisé sont concentrés sur le fonctionnement d'une partie, plus spécifiquement sur l'initialisation d'une partie, du mouvement d'une pièce et de la fin d'une partie.

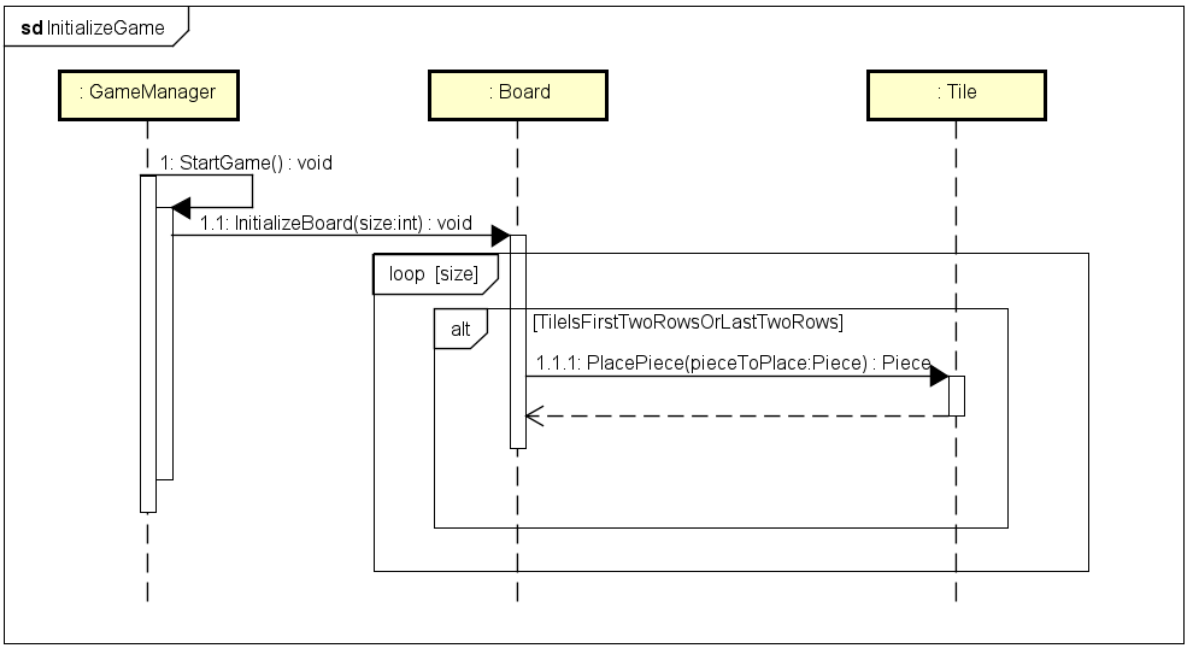


Figure 11: Diagramme de séquence de l'initialisation d'une partie

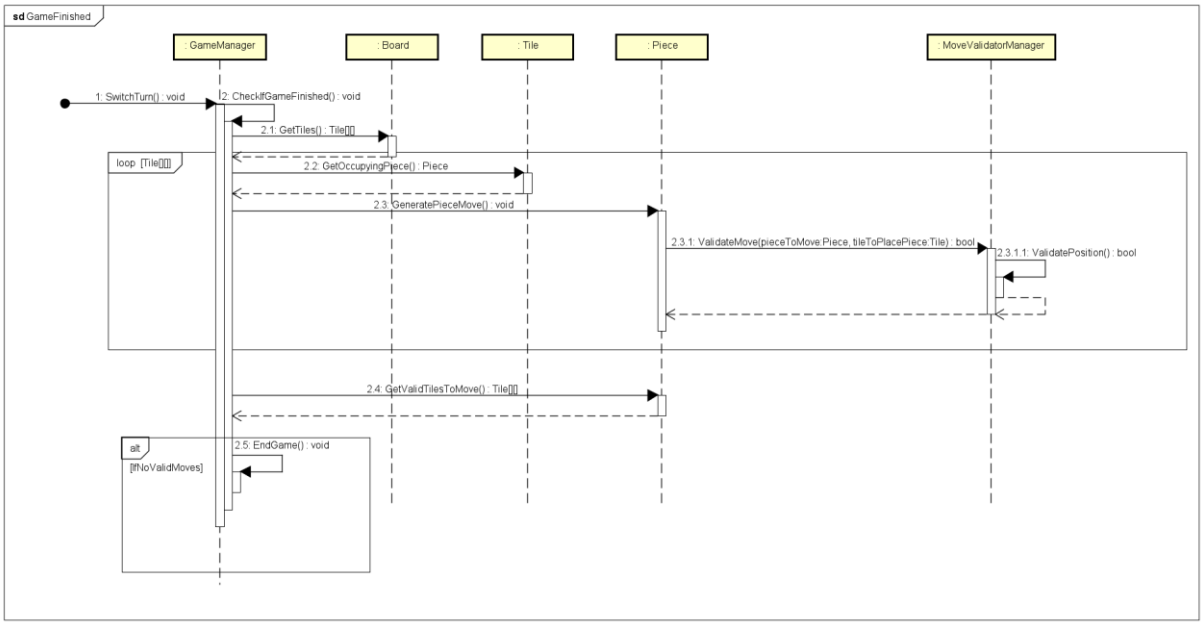


Figure 7: Diagramme de séquence de la fin d'une partie

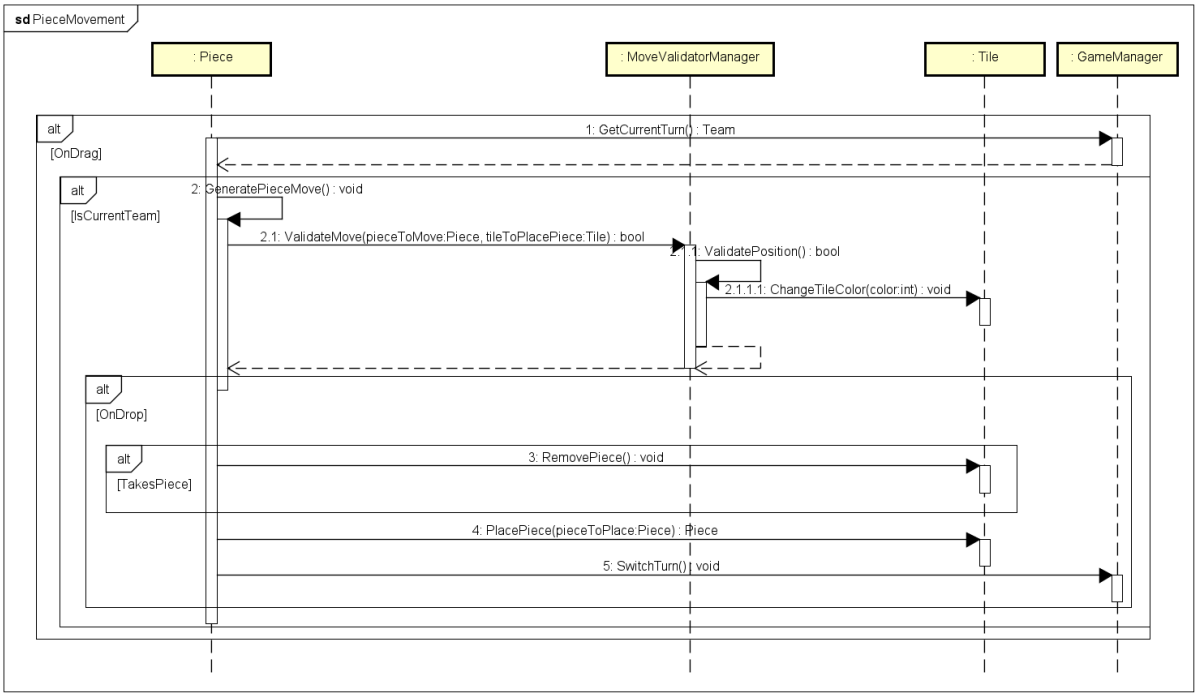


Figure 6: Diagramme de séquence du mouvement des pièces

3 Réalisation

3.1 Dossier de réalisation

3.1.1 Mise en place

Le projet Unity a été créer avec la version LTS (Long term support) 2022.3.19f1.

3.1.2 L'échiquier

L'initialisation de l'échiquier est réalisée en créant les cases à partir du prefab « Tile » en tant qu'enfant du GameObject « Board ».

Elles sont placées avec un décalage initial pour center l'échiquier sur l'écran, puis la distance est multipliée par le nombre de case déjà initié dans la rangé pour la position X et dans la colonne pour la position Y.

La couleur de la case est basée sur le nombre de rangé et de colonnes déjà existante. S'il est divisible par 2, la case sera foncée, dans le cas inverse, la case sera clair.

```
1 reference | ArthurCPNV, 28 minutes ago | 1 author, 1 change
public void InitializeBoard()
{
    for (int rank = 0; rank < boardSize; rank++)
    {
        for (int file = 0; file < boardSize; file++)
        {
            // Get the position in which the tile should be placed.
            RectTransform tilePrefabRectTransform = _tilePrefab.GetComponent<RectTransform>();
            float tileWidth = tilePrefabRectTransform.sizeDelta.x;
            float tileHeight = tilePrefabRectTransform.sizeDelta.y;
            Vector3 tilePosition = new Vector3(_xOffset + (tileWidth * rank), _yOffset + (tileHeight * file), 0);

            // Initialise the tile
            GameObject tile = Instantiate(_tilePrefab, tilePosition, Quaternion.identity);
            tile.transform.parent = transform;
            tile.name = (char)('a' + rank) + (file + 1).ToString();

            // Change the color of the tile alternating from dark to light squares
            if ((rank + file) % 2 == 0)
            {
                tile.GetComponent<Image>().color = darkSquareColor;
            }
            else
            {
                tile.GetComponent<Image>().color = lightSquareColor;

            }
        }
    }
}
```

Figure 8: Méthode de l'initialisation de l'échiquier

Figure 9: Arborescence des GameObjects dans la scène

Décrire la réalisation "physique" de votre projet

- *les répertoires où le logiciel est installé*
- *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
- *les versions des systèmes d'exploitation et des outils logiciels*
- *la description exacte du matériel*
- *le numéro de version de votre produit !*
- *programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.*

3.2 Description des tests effectués

Pour chaque partie testée de votre projet, il faut décrire:

- *les conditions exactes de chaque test*
- *les preuves de test (papier ou fichier)*
- *tests sans preuve: fournir au moins une description*

3.3 Erreurs restantes

S'il reste encore des erreurs:

- *Description détaillée*
- *Conséquences sur l'utilisation du produit*
- *Actions envisagées ou possibles*

3.4 Liste des documents fournis

Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions

- *le rapport de projet*
- *le manuel d'Installation (en annexe)*
- *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
- *autres…*

4 Conclusions

Développez en tous cas les points suivants:

- *Objectifs atteints / non-atteints*
- *Points positifs / négatifs*
- *Difficultés particulières*
- *Suites possibles pour le projet (évolutions & améliorations)*

5 Annexes

5.1 Résumé du rapport du TPI / version succincte de la documentation

5.2 Sources – Bibliographie

Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)… Et de toutes les aides externes (noms)

5.3 Journal de travail

Date	Durée	Activité	Remarques

5.4 Manuel d'Installation

5.5 Manuel d'Utilisation

5.6 Archives du projet

Media, … dans une fourre en plastique