Projet S4 : Rapport de soutenace n°1

Vj Studio:
LESIEUR Leo
DESCUBES Lucas
TEXIER Thibaud

15 Février 2023

Table des matières

1 Récapitulatif cahier des charges **3**
- 1.1 Présentation du groupe . . . 3
- 1.2 Présentation du projet . . . 3
- 1.3 Présentation des différentes tâches . . . 3
- 1.4 Rappel du tableau prévisionnel d'avancement . . . 4

2 Réalisation des tâches **5**
- 2.1 Mécanique du jeu : Lucas et Léo . . . 5
- 2.2 Intelligence artificielle : l'ensemble du groupe . . . 10
- 2.3 Interface graphique : Léo et Lucas . . . 11
- 2.4 Site web : Thibaud . . . 14

3 Etat d'avancement du projet **16**
- 3.1 Avancement actuel . . . 16
- 3.2 Tableau prévisionnel d'avancement actualisé . . . 16

4 Notre ressenti **17**
- 4.1 Réussites . . . 17
- 4.2 Difficultés . . . 17
- 4.3 Impressions personnelles . . . 17
  - 4.3.1 Léo . . . 17
  - 4.3.2 Lucas . . . 17
  - 4.3.3 Thibaud . . . 18

5 Conclusion **19**

1 Récapitulatif cahier des charges

1.1 Présentation du groupe

Nous sommes un groupe de trois personnes en C1 à l'école d'ingénieur Epita. Le groupe se compose de Léo Lesieur, Lucas Descubes et Thibaud Texier. Léo et Lucas se sont connus lors d'un semestre Erasmus passé ensemble à Prague, tandis que Thibaud était déjà en C1 lors du dernier semestre. Nous sommes ravis de pouvoir travailler ensemble sur ce projet du semestre 4 qui va nous permettre de développer nos connaissances en programmation en C et de nous préparer efficacement pour les prochains semestres, notamment l'ing1 et la piscine qui sont des étapes clés de notre parcours d'ingénieur. Cette collaboration va également nous permettre de renforcer nos compétences en termes de communication et de travail d'équipe, qui sont des compétences essentielles pour réussir dans notre domaine professionnel. Nous sommes donc très enthousiastes à l'idée de débuter ce projet et de voir les résultats que nous allons obtenir ensemble.

1.2 Présentation du projet

Dans un premier temps, lorsque nous avons commencé à discuter de la réalisation d'un jeu, nous étions tous d'accord sur le fait que c'était un sujet extrêmement intéressant et enrichissant. Nous avions tous une passion pour les jeux et nous pensions que c'était une opportunité unique de mettre nos compétences en programmation à l'épreuve en créant notre propre jeu. En réfléchissant davantage sur le projet, nous nous sommes rendu compte que la réalisation d'un jeu nous permettrait de progresser dans plusieurs domaines de l'informatique.

1.3 Présentation des différentes tâches

Nous avons commencé à élaborer un plan détaillé pour le jeu, en décidant des fonctionnalités que nous voulions inclure et de la manière dont nous allions les implémenter.

Il faut aller travailler sur la conception graphique du jeu et développer une interface utilisateur intuitive qui va rendre la navigation des joueurs à travers les différents menus et options plus facile. Il faut

également travailler sur l'implémentation de l'algorithme d'IA qui va permettre aux joueurs de jouer contre l'ordinateur. Et bien une bonne mécanique de jeu à savoir les mouvements, le respect des règles etc..

Nous savions que le développement du jeu allait être un processus en plusieurs étapes, nous avons donc décider de travailler toujours au moins en duo pour résoudre les différents tâches du projet.

1.4 Rappel du tableau prévisionnel d'avancement

	1ère soutenance	2ème soutenance	Soutenance finale
Mécanique du jeu	**40%**	**75%**	**100%**
Intellignce artificielle	**45%**	**80%**	**100%**
Interface graphique	**50%**	**80%**	**100%**
Site web	**30%**	**65%**	**100%**

On peut donc voir juste au dessus nos estimations d'avancement pour cette première soutenance. Nous souhaitions se rapprocher au maximum des 50% dans les 3 tâches principales de ce projet. Nous allons donc vous présenter dans la suite de ce rapport si nous y sommes arrivés et par quels moyens.

2 Réalisation des tâches

2.1 Mécanique du jeu : Lucas et Léo

Tout d'abord la première étpae a été de créer un plateau d'échecs avec les pièces placées au bon endroit. Pour cela nous avons crée une fonction create_table() et nous allons vous expliquer son fonctionnement.

La fonction create_table() permet de créer un tableau de jeu d'échecs en initialisant les différentes cases du tableau avec les pièces appropriées. Elle renvoie un pointeur vers cette structure de tableau de jeu.

La fonction alloue dynamiquement de la mémoire pour toutes les différentes structures de pièces nécessaires, y compris les pions, les tours, les cavaliers et les fous. Elle utilise ensuite des boucles pour initialiser les propriétés de chaque pièce, telles que le caractère utilisé pour représenter chaque pièce, l'équipe à laquelle chaque pièce appartient, le type de chaque pièce et sa valeur. Les positions initiales des différentes pièces sur le tableau sont également stockées dans des tableaux d'entiers distincts pour les équipes noire et blanche.

Enfin, la fonction utilise ces informations pour initialiser le tableau de jeu lui-même en affectant chaque pièce à la case appropriée dans le tableau. Les cases initiales des pions, des tours, des cavaliers et des fous sont déterminées et stockées à l'aide des tableaux d'entiers des positions, et les propriétés de la case de la structure de chaque pièce sont mises à jour en conséquence. En effet au début es pièces sont placées sur les cases du plateau de jeu d'échecs en fonction de leur position de départ respective pour chaque joueur. Les pions sont placés sur la rangée devant les autres pièces, tandis que les pièces de plus grande valeur sont placées sur les rangées arrières. Pour chaque joueur, de gauche à droite, la disposition des pièces est la suivante : tour, cavalier, fou, reine, roi, fou, cavalier, tour.

En somme, cette fonction permet de créer un tableau de jeu d'échecs initialisé avec les pièces appropriées aux bonnes positions pour commencer une partie d'échecs.

Ensuite nous avons implémenté des fonctions qui permettent de vérifier si un déplacement est possible.

En effet une de nos fonctions prend en paramètre la position d'un pion sur l'échiquier, la position de la case de destination (destination) et l'équipe du pion. Elle teste si la case de destination est accessible par le pion en prenant en compte les règles spéciales de déplacement des pions. Elle retourne 0 si la case de destination est accessible, 1 sinon. Il y a également une autre fonction qui prend en paramètre l'échiquier, la position de la pièce à déplacer (source) et la position de la case de destination (destination). Elle appelle la fonction de test correspondante à la pièce en question (pion, tour, fou, roi, cavalier, ou reine) et retourne le résultat de ce test. Le résultat est 0 si le déplacement est possible, 1 sinon.

Forcément nous avons du créer des fonctions permettant de savoir si une pièce est éliminée ou même si il y a échec et mat ce qui signifie que la partie est finie. Pour cela une première petite fonction qui prend en paramètre l'échiquier, la position d'une pièce sur l'échiquier (source) et l'équipe de la pièce. Elle teste si la pièce est en échec en vérifiant si elle peut être capturée par une pièce de l'équipe adverse. Elle retourne 1 si la pièce est en échec, 0 sinon.

Et ensuite une autre fonction qui prend en paramètre l'échiquier et l'équipe pour laquelle on teste si elle est en échec et mat. Elle vérifie si le roi de l'équipe est en échec en appelant la fonction que nous avons décrit juste au dessus. Si le roi n'est pas en échec, elle retourne 0. Sinon, elle teste si le roi peut être protégé en déplaçant les autres pièces de l'équipe. Si aucune pièce ne peut protéger le roi, la fonction retourne 1, sinon elle retourne 0.

Ensuite nous devions implémenter peut être une des parties les plus importantes du projet qui est une fonction de validation de mouvement pour chaque pièce différente du jeu d'échecs.

Nous allons commencer par vous expliquer celle des pions. Elle prend en entrée un pointeur vers une structure Table représentant l'état du jeu, ainsi que les index de la case source et de la case destination.

La fonction vérifie si la pièce sur la case source est un pion noir ou blanc en fonction de son équipe. Si la pièce est un pion noir, elle vérifie si le mouvement est valide. Un pion noir peut avancer d'une case diagonalement (source + 7 ou source + 9) si la case destination contient une pièce adverse (team == 1) et n'est pas vide (free == 0). Si la case source est sur la rangée 2, le pion peut avancer de deux cases vers l'avant (source + 16) si les deux cases sont vides (free == 1).

Si le mouvement n'est pas valide, la fonction retourne 1 (false). Si le mouvement est valide, elle retourne 0 (true).

Si la pièce sur la case source est un pion blanc, la fonction effectue des vérifications similaires pour un mouvement vers l'arrière et vers le haut (source - 7 ou source - 9) et un mouvement vers l'avant (source - 8). Les vérifications pour les mouvements de deux cases vers l'avant et le cas des pions blancs sur la rangée 7 sont également effectuées.

La fonction retourne 1 si le mouvement n'est pas valide et 0 s'il est valide.

Maintenant voyons le fonctionnement pour les tours. Elle prend en entrée un pointeur vers une structure de tableau qui contient les informations sur l'état actuel du plateau d'échecs, ainsi que les index de la case source et de la case destination du mouvement.

La première instruction vérifie si la case source et la case destination ont la même équipe et si la case destination n'est pas vide. Si c'est le cas, cela signifie que la tour ne peut pas se déplacer sur cette case et la fonction renvoie 1 (true).

Sinon, la fonction vérifie si la tour se déplace vers le bas, la droite, le haut ou la gauche. Si elle se déplace vers le bas ou la droite, la fonction vérifie si les cases entre la case source et la case destination sont vides. Si elles ne le sont pas, la fonction renvoie 1 (true).

Si la tour se déplace vers le haut ou la gauche, la fonction vérifie également si les cases entre la case source et la case destination sont vides. Si elles ne le sont pas, la fonction renvoie 1 (true).

Si toutes les vérifications sont réussies, la fonction renvoie 0 (false), ce qui signifie que le mouvement de la tour est valide.

Parlons maintenant des fous. Cette fonction prend en entrée un pointeur vers une structure Table qui représente un plateau de jeu

d'échecs, ainsi que deux positions source et destination sur le plateau. La fonction renvoie un entier : 1 si le mouvement du fou de la case source vers la case destination est illégal, et 0 s'il est légal.

La fonction commence par vérifier si la pièce à la case source et celle à la case destination appartiennent à la même équipe et si la case destination est occupée par une pièce. Si tel est le cas, elle renvoie 1 car le mouvement est illégal.

Ensuite, la fonction vérifie si la case destination est sur la diagonale de la case source. Si ce n'est pas le cas, la fonction renvoie 1 car le mouvement est illégal.

Si la case destination est sur la diagonale de la case source, la fonction vérifie si le mouvement est valide en fonction de la direction de la diagonale. Elle considère quatre cas : mouvement vers le bas à droite, mouvement vers le haut à droite, mouvement vers le haut à gauche et mouvement vers le bas à gauche.

Si le mouvement est vers le bas à droite ou vers le haut à gauche, la fonction parcourt les cases intermédiaires pour s'assurer qu'elles sont libres et que la case destination est atteinte en parcourant les cases de la diagonale. Si l'une des cases intermédiaires n'est pas libre, la fonction renvoie 1 car le mouvement est illégal. Si toutes les cases intermédiaires sont libres, la fonction renvoie 0 car le mouvement est légal.

Si le mouvement est vers le haut à droite ou vers le bas à gauche, la fonction vérifie également que toutes les cases intermédiaires sont libres. Si toutes les cases intermédiaires sont libres, la fonction renvoie 0 car le mouvement est légal. Sinon, elle renvoie 1 car le mouvement est illégal. Si le mouvement ne se fait pas sur une diagonale, la fonction renvoie 1.

Venons en maintenant aux cavaliers. La fonction prend en entrée un pointeur vers une structure "Table" , une position source et une position destination. Elle renvoie 0 si le déplacement d'un cavalier depuis la position source à la position destination est valide et 1 sinon.

La fonction commence par vérifier si la case de départ et la case d'arrivée ont la même équipe et si la case d'arrivée est occupée, si c'est le cas, elle renvoie 1 pour indiquer que le mouvement n'est pas valide.

Ensuite, la fonction vérifie toutes les cases vers lesquelles un ca-

valier peut se déplacer en utilisant les règles de déplacement du cavalier aux échecs. Si la case de destination correspond à l'une de ces cases, la fonction renvoie 0 pour indiquer que le mouvement est valide, sinon elle renvoie 1.

Les différentes conditions correspondent aux 8 cases vers lesquelles un cavalier peut se déplacer. Si la case source est sur le bord gauche ou droit de l'échiquier, alors certaines cases ne sont pas disponibles pour le cavalier, d'où les vérifications pour les positions 15, 17, 6 et 10.

Nous allons voir le fonctionnement pour le roi maintenant. Elle prend en entrée un pointeur vers une structure Table qui contient la configuration actuelle de l'échiquier, l'indice de la case source (source) et l'indice de la case destination (destination).

La fonction commence par vérifier si la pièce dans la case source et la pièce dans la case destination ont la même équipe et si la case destination est libre. Si c'est le cas, elle retourne 1, indiquant que le déplacement est invalide.

Ensuite, la fonction teste différents déplacements possibles pour le roi, en vérifiant si la case destination est dans la même rangée, colonne ou diagonale que la case source. Si la case destination est valide, la fonction utilise une fonction auxiliaire expliquée un peu plus haut pour vérifier si le roi serait en échec après le déplacement. Si le déplacement est valide et ne met pas le roi en échec, la fonction retourne 0 pour indiquer que le déplacement est valide. Sinon, la fonction retourne 1 pour indiquer que le déplacement est invalide.

Et enfin l'explication pour la dame. La fonction prend en entrée un pointeur vers une structure Table, la position de la pièce source et la position de la pièce de destination, et renvoie un entier qui vaut 1 si le déplacement est valide et 0 sinon.

La fonction vérifie si le déplacement de la reine est valide en appelant la fonction de test sur la tour et la fonction de test sur le fou. Si les deux fonctions retournent 0, cela signifie que le déplacement est valide et la fonction renvoie 0. Si l'une des fonctions retourne 1, cela signifie que le déplacement n'est pas valide et la fonction renvoie 1.

En d'autres termes, la fonction vérifie si la reine peut se déplacer horizontalement, verticalement ou en diagonale sur la grille d'échecs

en appelant les fonctions de test tour et fou, qui vérifient si la reine peut se déplacer comme une tour ou comme un fou, respectivement. Si la reine peut se déplacer comme une tour ou un fou, alors elle peut se déplacer comme une reine, et la fonction retourne 0.

Et enfin après avoir défini tout ça il fallait bien évidemment créer une fonction qui fait bouger une pièce d'une case source vers une case destination. Nous avons réalisé cela de la manière suivante.Elle prend en entrée un pointeur vers une structure Table représentant l'état du jeu d'échecs, ainsi que deux positions source et destination du tableau représentant la table. La fonction effectue un déplacement d'une pièce de la position source vers la position destination, en vérifiant si le déplacement est valide et en mettant à jour les différentes variables de l'état du jeu.

La première partie de la fonction vérifie si la case de destination est occupée par une pièce, et si c'est le cas, elle vérifie si cette pièce appartient à la même équipe que la pièce qui est déplacée. Si c'est le cas, la fonction retourne EXIT_FAILURE. Sinon, elle met à jour le nombre de pièces restantes de l'équipe adverse.

Ensuite, la fonction parcourt les tableaux coord_black et coord_white de la structure Table pour mettre à jour la position de la pièce qui vient d'être déplacée.

Enfin, la fonction copie la pièce de la case source vers la case destination, met la case source à vide, et retourne EXIT_SUCCESS si le déplacement a été effectué avec succès.

2.2 Intelligence artificielle : l'ensemble du groupe

Le tableau d'avancement initial incluait une bonne avancée pour la conception d'une intelligence artificielle pour le jeu, mais peu de progrès ont été réalisés dans cette direction pour le moment. Les efforts se sont plutôt concentrés sur la mécanique du jeu, considérée comme l'élément fondamental du projet, ainsi que sur l'interface graphique pour un aspect plus concret.

Malgré tout, une approche simple pour coder une IA a été choisie, dans le but de mieux comprendre son fonctionnement, avec l'objectif ultérieur de la rendre plus compétitive. Cependant voici les

différentes étapes du code actuel :

- La représentation du plateau de jeu à l'aide d'une matrice 2D.

- L'écriture de fonctions pour afficher le plateau et récupérer les coups possibles pour une pièce donnée.

- La mise en place d'une boucle de jeu alternant entre le joueur humain et l'IA.

- La sélection d'un coup aléatoire parmi les coups possibles pour l'IA.- L'évaluation de la fin de la partie pour déterminer si l'IA a gagné, perdu ou si la partie est nulle.

Les travaux sont encore en cours pour finaliser la conception de l'IA pour ensuite la rendre davantage compétitive et enfin l'intégrer pleinement dans le projet.

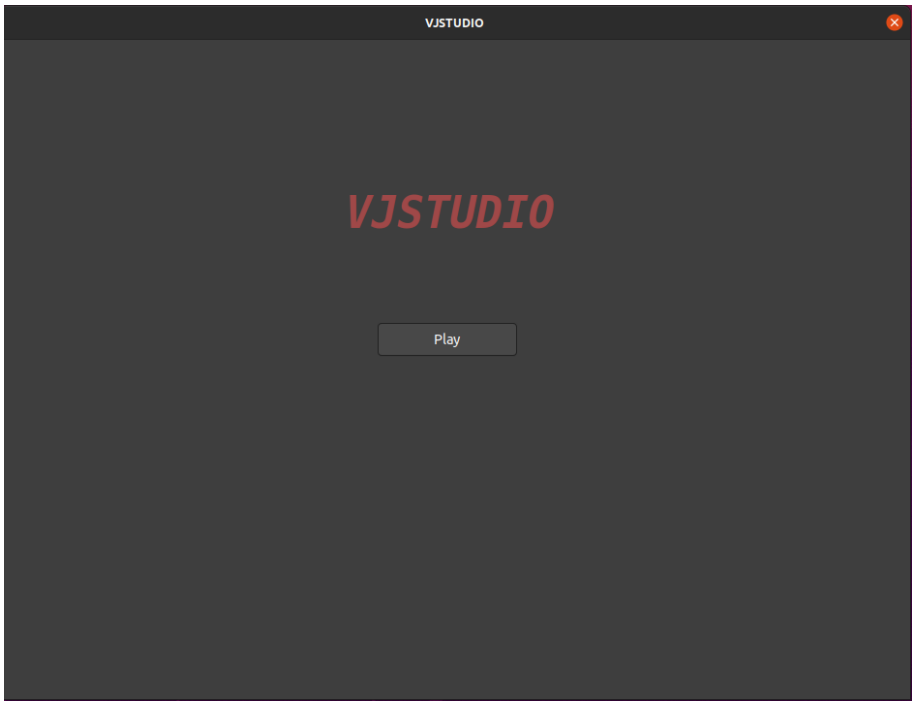
2.3 Interface graphique : Léo et Lucas

L'interface graphique est une partie essentielle du projet car lorsque le jeu sera finalisé, le premier aspect du jeu que l'on voit est le visuel et l'ergonomie du jeu.

Pour réaliser la notre nous avons décidé d'utiliser la bibliothèque GTK ainsi que Glade qui est un environnement de développement intégré compatible avec GTK.

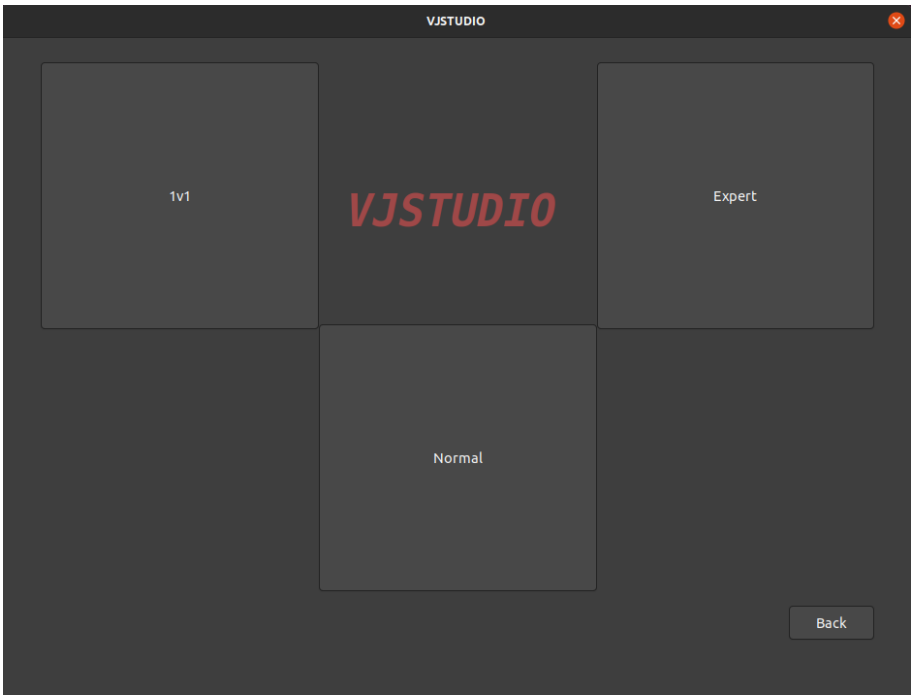
Tout d'abord nous avons créer une première page qui est donc notre page d'accueil.

Sur cette page d'accueil on peut donc voir notre nom de groupe ainsi qu'un simple bouton play. Dans la programmation GTK, on peut associer des fonctions à des boutons en utilisant la fonction "g_signal_connect". Donc lorsque l'utilisateur appuie sur le bouton play une fonction est lancée et cette fonction permet d'arriver sur une nouvelle page que nous allons vous présenter par la suite.



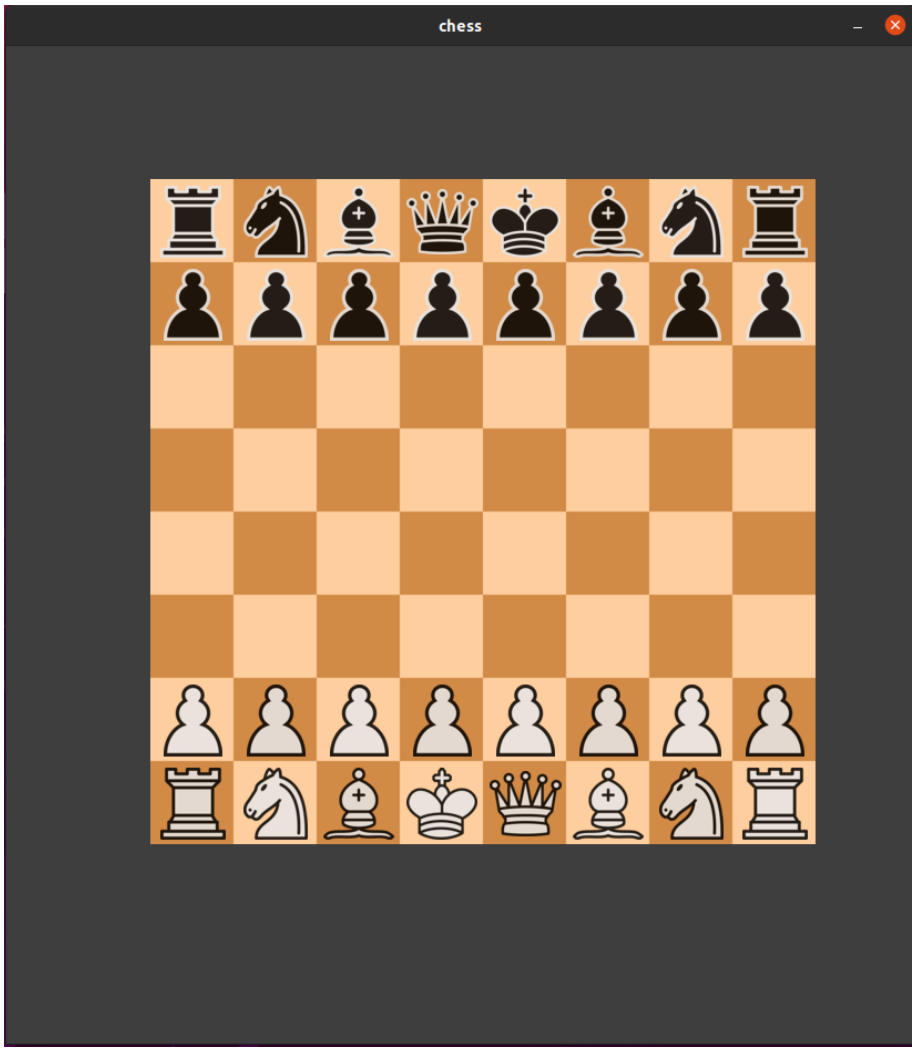
Page d'accueil

Nous arrivons donc maintenant à la deuxième page qui est celle qui va permettre au joueur de choisir le mode de jeu qu'il souhaite. En effet on peut voir graviter autour de notre nom de groupe trois modes de jeux différents. Dans un premier temps il y a mode 1V1 qui ne fait donc pas intervenir l'intelligence artificielle, mais qui permet juste de jouer avec un ami par exemple depuis le même ordinateur. Ensuite il y a deux modes qui font intervenir l'intelligence artificielle qui sont le mode Normal et le mode Expert.En effet nous avons pour objectif d'être capable d'adapter notre intelligence artificielle en fonction de la difficulté choisie par l'utilisateur. Cependant pour l'instant lorsque nous appuyons sur ces boutons il ne se passe rien car n'ayant pas une intelligence artificielle finie ces modes de jeu n'ont pour le moment aucun intérêt. En bas à droite il y a également un bouton Back qui permet de revenir sur notre page d'accueil. Ensuite par exemple lorsque l'utilisateur une nouvelle et dernière page s'affiche et nous allons vous la présenter tout de suite.



Choix du mode de jeu

Et voici la dernière page qui est la page la plus importante car c'est celle du jeu en lui même. On peut donc y voir le plateau avec toutes les pièces blanches et noires placées sur leur positions initiales. On peut donc voir tous les pions sur la deuxième ligne de chaque couleur, les tours dans les coins, les cavaliers à côté des tours, la reine et le roi au milieu et les fous à côté de la reine et du roi. Pour faire bouger une pièce il suffit d'appuyer sur la pièce et ensuite d'appuyer sur la case ou on veut la mettre. Si le mouvement est possible alors le plateau sera immédiatement à jour avec la position de la place actualisée et si le mouvement n'est pas possible rien ne se passera et il faudra alors choisir une nouvelle case.



Plateau de jeu

2.4 Site web : Thibaud

La construction de ce site web en HTML et CSS a été une expérience difficile mais enrichissante. Le site Web a été conçu pour présenter notre projet de logiciel d'échecs, y compris sa présentation, l'équipe qui l'a réalisé, le tableau d'avancement et une page de rapport où des fichiers peuvent être téléchargés. Pour créer le site Web, j'ai utilisé le langage HTML pour structurer les pages, notamment les pages d'accueil, de presentation de l'équipe, de chronologie et de

rapport. Sur chacune de ces pages, j'ai utilisé un fichier CSS pour fournir des éléments de style et de conception afin de rendre le site plus attrayant.

Pour créer la page Équipe, il a fallu diviser un long paragraphe en sections plus petites, qui ont ensuite été mises en forme sur la page pour qu'elle soit organisée et esthétiquement agréable. Sur la page Chronologie, j'ai ajouté une image pour visualiser la progression du projet et j'ai écrit une description des différents moments du projet et de leur taux d'achèvement dans quatre catégories : mécanique du jeu, IA, interface utilisateur et site Web. La page du rapport a été conçue de manière à pouvoir télécharger un fichier qui pourrait être ajouté ultérieurement.

L'une des principales difficultés que j'ai rencontrées lors de la création de ce site Web était de m'assurer que le texte de toutes les pages était correctement encodé pour afficher correctement les caractères spéciaux tels que les accents. Bien que j'aie essayé plusieurs solutions, les accents ne s'affichaient toujours pas correctement sur certaines pages. Un autre défi consistait à rendre l'entête et les éléments de texte cohérents sur toutes les pages, car certains éléments apparaissaient plus petits ou plus grands sur certaines pages, bien que les mêmes styles aient été définis dans le fichier CSS.

Malgré ces difficultés, j'ai pu en apprendre beaucoup sur le HTML et le CSS et sur la façon de les utiliser pour construire un site Web fonctionnel. Le design du site n'est pas définitif, et il est principalement là pour la structure pour le moment. Cependant, je pense qu'avec plus de temps et de pratique, je peux continuer à améliorer mes compétences et le site web de ce projet.

Voici le lien pour accéder a notre site web :
https ://parmeti.github.io/VJStudioProject/

3 Etat d'avancement du projet

3.1 Avancement actuel

Tout d'abords nos plans ont changés.En effet nous étions partis sur un avancement de toutes les tâches assez équilibré. Cependant nous avons décidé d'avancer le plus possible la mécanique du jeu ce qui a été fait puisque tout est presque fini. Concernant l'interface graphique nous avons globalement respecté les prévisions car elle fonctionne, nous avons toutes les pages que nous voulions mais elle n'est pas finie car par la suite nous souhaitons rajouter des fonctionnalités telles que le temps de la partie, le nom du vainqueur, un historique des dernières parties etc. A propos de l'intelligence artificielle elle est très peu avancée, nous nous sommes beaucoup renseignés et avons commencé à coder mais rien n'est fini pour le moment.C'est donc sur ça que nous allons nous concentrer en majorité pour la deuxième soutenance. Et enfin le site web respecte les prévisions.

3.2 Tableau prévisionnel d'avancement actualisé

	1ère Soutenance	2ème Soutenance	Soutenance finale
Mécanique du jeu	90%	100%	100%
Intelligence artificielle	15%	55%	100%
Interface graphique	60%	70%	100%
Site Web	30%	50%	100%

4 Notre ressenti

4.1 Réussites

Cette première partie de projet est très satisfaisante selon nous car certes nous n'avons pas forcément respecté notre tableau prévisionnel d'avancement mais cela ne veut pas dire que nous sommes en retard.Au contraire une grande partie du projet est presque finalisée et nous allons pouvoir nous concentrer en grande partie sur l'intelligence artificielle dorénavant.

Nous sommes donc content du déroulé du projet et espérons que cela va continuer ainsi.

4.2 Difficultés

Nous n'avons pas rencontré de réelles difficultés lors de cette première partie du projet même si le fait d'avoir modifié notre tableau d'avancement prévisionnel aurait pu nous perturber. La partie mécanique du jeu était tout de même très longue car les pièces d'échecs n'ayant pas toutes les mêmes déplacements nous obligeait à traiter de nombreux cas différents. Nous avons également réalisés qu'avoir une intelligence artificielle compétitive serait un réel défi mais que nous allons bien évidemment essayer de surmonter.

4.3 Impressions personnelles

4.3.1 Léo

D'un point de vue personnel je suis très content de cette première partie de projet car nous sommes largement dans les temps. Je suis ravi de m'être occupé de la partie interface graphique car cela m'a permis d'approfondir mes connaissances en GTK ainsi que d'appréhender Glade. J'ai hâte de poursuivre ce projet et me lancer davantage dans l'intelligence artificielle. Je suis pressé d'arriver au bout de ce projet et voir le rendu final.

4.3.2 Lucas

Je suis très content d'avoir pu travailler avec mes camarades en ce début de semestre sur ce projet. J'ai également pu approfondir mes connaissances en C en programmation orientée objet, ainsi qu'en

Glade. J'ai hâte de pouvoir continuer pour obtenir un jeu fini avec une intelligence artificielle fonctionnelle qui nous permettra de jouer réellement. Je pense que nous avons bien avancé et je suis rassuré que nous soyons dans les temps pour nous permettre d'avoir un résultat final de bonne qualité.

4.3.3 Thibaud

Jusqu'à présent, je suis très satisfait de ce projet. Cela m'a donné l'occasion de mettre en pratique mes connaissances en HTML et CSS pour créer un site web simple qui présente notre projet de développement d'un jeu d'échecs. J'ai également été en mesure de résoudre des problèmes techniques tout au long du processus, ce qui m'a permis d'améliorer mes compétences. Bien que la conception actuelle ne soit probablement qu'une structure temporaire, j'ai hâte de continuer à travailler sur ce projet et de le voir évoluer à mesure que nous nous rapprochons de sa finalisation.

5 Conclusion

En conclusion nous sommes tous les 3 contents de l'avancée de notre projet. Il est vraiment très enrichissant car créer toute la mécanique d'un jeu ne s'avérait pas être une tâche facile mais nous avons réussi. De plus grâce à la bonne base de notre interface graphique nous pouvons concrètement voir le projet prendre forme et c'est un réel plaisir. Nous espérons que la suite du projet va bien se passer afin de vous proposer un jeu le plus complet et attractif possible.