

INSA

INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
LYON

NGUYEN Khai
LUONG Tung
ZHANG Hélène
NGUYEN Hang

Document Explicatif du Sujet Chess

Filière ASINSA
Groupe 97
Promotion n°64
2021-2022

Projet
d'informatique





Sommaire

I. Cahier des Charges.....	1
1.1 Objectif.....	1
1.2 Fonctionnement du jeu	1
1.3 Interface Graphique	1
II. Description du Problème Posé	5
III. Principe de l’algorithme.....	6
IV. Bibliographie + Médiagraphie	7
V. Structuration des données	8
VI. Suggestions d’améliorations de votre projet, bugs connus ..	8
VII. Carnet de Route	9
VIII. Pourcentage d’implication	10

I) Cahier des Charges

1.1) Objectif

Le jeu d'échecs est l'un des jeux les plus populaires et les plus intéressants à jouer dans de nombreux pays et cultures. Jouer aux échecs favorise le développement des capacités de réflexion et de la perspective, améliore la mémoire, approfondit la concentration, stimule les capacités de planification et élève même la pensée créative. Il est donc étroitement lié à la croissance exponentielle de l'« informatique » en générale dont l'homme qui en est la source a réussi à bâtir. Nous avons alors pensé que la programmation de ce jeu était une excellente opportunité de mettre en pratique nos connaissances sur l'IHM et l'Orienté Objet sur l'Interface de l'échiquier mais également de découvrir l'utilisation de l'IA, Apprentissage Automatique. Pour cela, nous utiliserons des «réseaux neuronaux», qui agissent comme des cerveaux humains, pour permettre à l'ordinateur de jouer, par lui-même, les échecs.

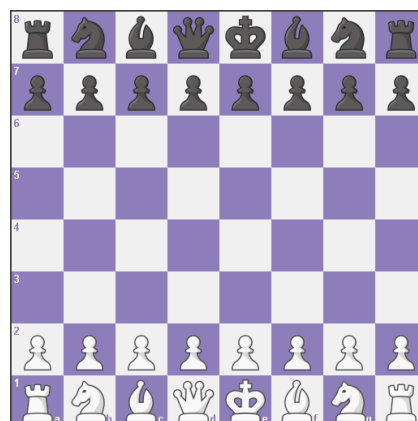
1.2) Fonctionnement du jeu

Le jeu dispose de deux modes : jeu contre l'ordinateur et jeu à 2 personnes. Pour chaque mode, le temps d'un match est contrôlé par des subdivisions : échecs en blitz en 3 minutes et échecs rapides en 10 minutes... Au départ, le joueur choisit le mode de jeu et la couleur de son jeu (Noir ou Blanc). Les joueurs effectuent les déplacements des pièces en cliquant dans un premier temps sur les pièces elles-mêmes qu'ils souhaitent jouer; et dans un deuxième temps sur une case inoccupée (déplacement simple) ou occupée par une pièce de l'ennemi, il réalise donc la fonction «manger» : la pièce de l'ennemi est alors retirée du jeu. Si le déplacement est possible, la pièce se déplace, sinon l'écran affiche un message demandant au joueur de se déplacer à nouveau. Toutes les pièces mangées disparaissent de l'échiquier et se placent derrière le jeu de l'adversaire. La pièce mangée ne retourne sur l'échiquier que s'il y a promotion (lorsqu'un pion arrive sur la huitième rangée). Un joueur n'a pas le droit de sauter un tour, chaque déplacement est obligatoire dans un tour. Dans le jeu, les Blancs jouent en premier, ensuite les joueurs alternent leurs tours en déplaçant une pièce pour chaque tour (sauf pour le Roque, où deux pièces sont déplacées). Le but du jeu est de mettre l'adversaire en échec et mat, ce qui se produit lorsque le Roi de l'adversaire est en échec (ne peut plus faire de déplacement sinon se fait manger). Si le but est atteint, le joueur gagne et le jeu se termine













1.3) Interface Graphique

Tout se passera sur un plateau de jeu mais nous vous proposons plus de fonctionnalités pour plus de fun ! A découvrir dans les paragraphes qui suivent.

Plateau de jeu :

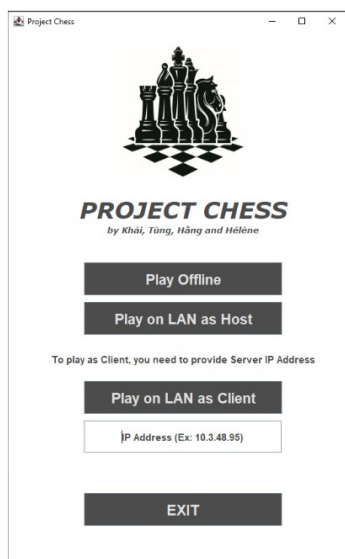


16 pièces au total dont 6 pièces différentes :

					
1 Roi	1 Reine	2 Tours	2 Cavaliers	2 Fous	8 Pions
					
Déplacement sur une case dans toutes les directions et sens du plateau	Déplacement dans tous les sens en diagonale, vertical et horizontal du plateau	Déplacement dans tous les sens dans la direction verticale et horizontale du plateau	Déplacement en L (3 cases verticales puis 2 cases horizontales ou 2 cases verticales et 3 cases horizontales)	Déplacement dans tous les sens dans la direction diagonale du plateau	1 ^{er} déplacement : 1 ou 2 cases dans la direction verticale et dans le sens de l'avancée du jeu Prochain déplacement : 1 case dans la même direction et même sens Manger : il n'élimine une pièce qu'en diagonale et dans le sens de l'avancée du jeu

1) Main Menu :

Pour démarrer le jeu, il faut compiler et exécuter la classe **GameStart**. Il apparaît la fenêtre ci-dessous :



-Pour jouer hors ligne, appuyer sur le bouton **Play Offline**, puis passer à la Section 4.

-Pour jouer en ligne en tant que hôte, appuyer sur le bouton **Play on LAN as Host**, puis passer en Section 2 pour plus d'instructions.

-Pour jouer en ligne en tant qu'invité (un hôte doit nécessairement être en ligne), appuyer sur **Play on LAN as Client**, Entrer un adresse IP, mettre l'adresse IP de l'hôte avec qui vous souhaitez jouer. Puis passer à la Section 3.

-Pour fermer le jeu, appuyer sur le bouton **EXIT**.

2) Jouer en tant que hôte

Après avoir appuyé sur le bouton **Play on LAN as Host**, vous serez dirigés sur une page indiquant que vous êtes en attente d'un autre joueur. Si un joueur rejoint la partie, alors vous serez avertis. Ensuite vous pouvez choisir une limite de temps en cliquant sur la zone de la liste déroulante.



Vous avez également le choix de la couleur des pièces entre White et Black en appuyant sur le bouton correspondant. Une fois que vous êtes prêts à jouer, appuyer sur **Play**. Mais ce n'est seulement qu'après qu'un joueur ait rejoint la partie que la partie commencera. Aller à la Section 5.

Attention : Sur la fenêtre de l'hôte sera indiquée son adresse IP, ce qui sera plus simple pour un autre joueur (invité) de se connecter à la partie (car il devra entrer votre adresse IP avant de rejoindre). Mais soyez prudent, l'adresse IP est personnel et secret, attention à ne pas le fuiter :p.

3) Jouer en tant qu'invité



Lors de l'entrée de l'adresse IP, si la machine trouve le hôte alors une page apparaîtra sur l'écran.

Vous devez attendre que le hôte commence la partie.

Allez à la Section 5 pour continuer.

4) Jouer hors ligne

Dans un premier temps, vous avez la possibilité de customiser votre jeu.

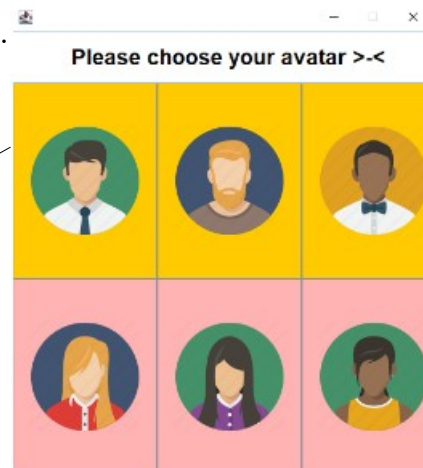
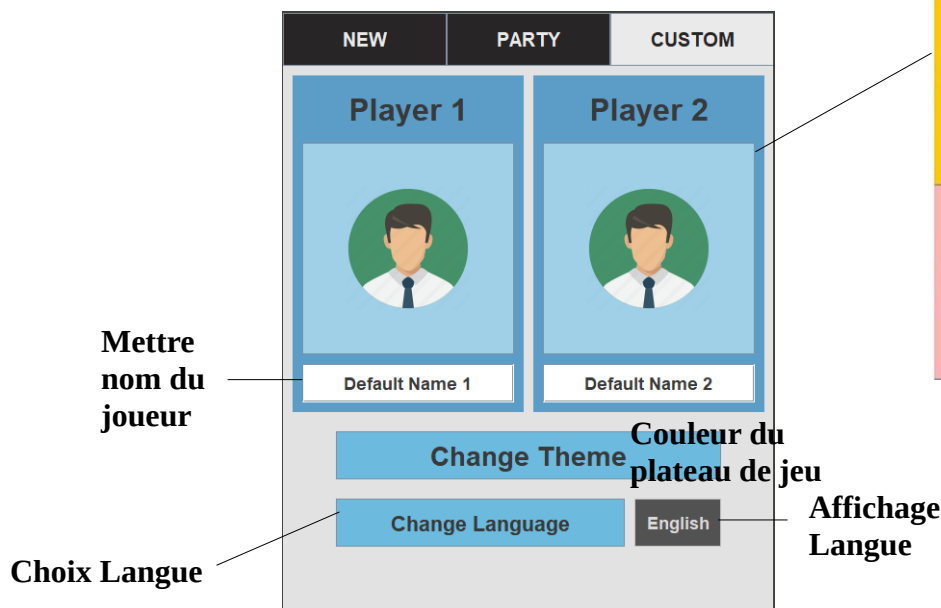
Vous pouvez cliquer sur l'avatar de votre choix parmi les 6 proposés.

Vous pouvez changer votre pseudo.

Ainsi apparaîtra vos modifications dans la page de jeu une fois que la partie sera lancée.

Vous pouvez également modifier la couleur du plateau de jeu (thème du jeu). Appuyer sur le bouton **Change Theme** et soyez impressionnés !

Vous pouvez avoir le choix entre 3 langues : Français, Anglais et Japonais.



Une fois terminée, appuyer sur **New**, il apparaîtra ainsi la page suivante. Mais ne vous inquiétez pas, vous pouvez retourner sur la page de customisation si vous souhaitez modifier une nouvelle fois.

Il y a 2 modes de jeu qui sont proposés.

Si vous souhaitez jouer avec un ami, appuyer sur **Player vs Computer** pour changer de mode. Puis choisissez votre temps de jeu, ensuite appuyer sur **PLAY** et amusez-vous !!!

Si vous êtes seul ☹ , vous avez toutefois de la chance parce qu'on vous propose un mode spécial pour vous, votre ami l'ordinateur est prêt à jouer avec vous ☺ . Dans ce cas vous avez la possibilité de choisir votre couleur de jeu en appuyant sur le bouton à côté de **Player vs Computer**. Ne vous inquiétez pas le mec n'est pas si intelligent que ça. Il est juste capable de voir jusqu'à 3 coups à l'avance... Essayez de le battre !!!!

Poursuivez en Section 5.



5) Dans la partie...

Après avoir appuyé sur le bouton **PLAY**, vous serez dirigés vers la page **PARTY**, où apparaîtra vos mouvements effectués au cours de la partie. Savez-vous que dans les compétitions d'échecs, les joueurs écrivent toujours les mouvements réalisés pour en faire une étude profonde sur leur partie ? Ici on le fait pour vous !

Voici un exemple de ce que vous aurez :

NEW	PARTY	CUSTOM
1	g4	Nh6
2	Nh3	Nxg4
3	f3	Ne5
4	Bg2	d6
5	f4	Nc4
6	Nc3	Bxh3
7	Bxh3	a5
8	d3	Nb6
9	Nb5	Nd5
Historique du jeu		

Attention le temps s'écoule pendant la partie, donc gérez bien votre temps !

+ 3 fin possibles (à vous de découvrir)

Nous vous souhaitons un bon jeu !!!

II) La description du problème posé

L'objectif ultime des échecs est de mettre en *échec et mat* (to checkmate) l'adversaire. C'est un jeu stratégique, souvent de longue durée d'où la contrainte du temps proposée dans de nombreux jeux d'échecs. Nous proposerons cette contrainte en plus du jeu de base dans notre projet pour corser. Commençons par les explications de base, le principe reste le même que tous les jeux d'échec et nous apporteront en plus quelques fonctionnalités pour améliorer le jeu. C'est un jeu entre 2 joueurs : Blanc et Noir. Les blancs commencent toujours la partie puis ce sera au tour des noirs et ainsi de suite. Pour chaque tour, chaque équipe déplace une pièce de son choix. Ce mouvement est possible si et seulement si :

- le temps n'est pas écoulé
- la case visée est vide
- la pièce déplacée mange la pièce adverse, cela entraîne par la même occasion la suppression de cette pièce du jeu
- le déplacement n'entraîne pas un échec de son roi

Un tour ne peut pas être sauté. Chaque mouvement est obligatoire dans un tour.

Chaque pièce a sa façon de se déplacer.

Il y a possibilités de faire revenir une pièce dans le jeu avec une promotion c'est-à-dire lorsqu'un pion atteint le 8^{ème} rang ou le 1^{er} rang de l'équipe adverse.

Le jeu se termine si le jeu est nul ou si l'un des 2 couleurs a mis en *échec et mat* l'autre joueur.

Le jeu est nul si l'adversaire n'est pas en *échec et mat* mais aucun mouvement n'est possible de sa part non plus.



Ici le Roi est bloqué, il ne peut plus bouger au sinon il sera mangé par l'une des pièces blanches. Il n'est pas en situation d'*échec et mat* puisqu'il n'est pas attaqué par une des pièces adverses. En effet un *échec et mat* est effectif uniquement si le Roi ne peut plus faire de mouvement et s'il est attaqué c'est-à-dire en situation d'être mangé, autrement dit en échec.

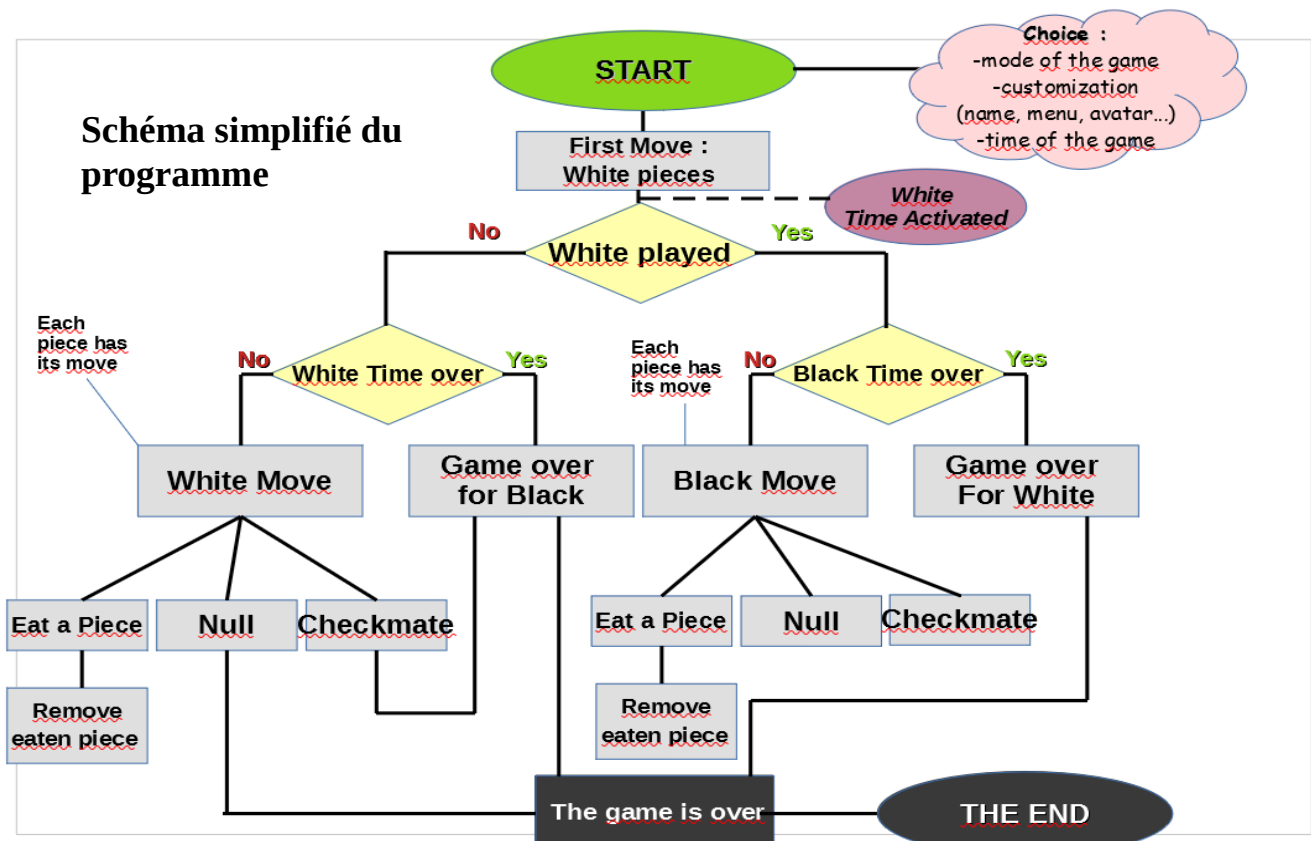
Comme dit précédemment, il y a *échec et mat* si et seulement s'il répond aux conditions suivantes :

- le Roi de l'adversaire n'a plus aucun mouvement possible
- le Roi est attaqué c'est-à-dire en échec

Deux modes de jeu vous seront proposés : un jeu à 2 joueurs et un jeu contre une machine réalisée par nos soins.

De plus, on associera un temps au jeu soit une limite de 1, 3, 5 et 10 minutes par joueur. Pour chaque tour, le joueur concerné déduit son temps de jeu. Ainsi une autre fin possible est ajoutée, si le temps est écoulé pour le joueur concerné alors il perd sa partie et le jeu se termine à la suite. La base attendue et voulue pour le jeu est dans le programme.

III) Principe de l'algorithme



IV) Bibliographie + Médiagraphie

Sites :

- « Le vocabulaire des échecs en anglais ». [Ecole Apprendre-les-echecs. GeneratePress, 17 mars 2021. Web : Le vocabulaire des échecs en anglais - Ecole Apprendre-les-echecs](#)
- « How To Set A Timer In Java With Examples ». Software Testing Help. Vijay, 3 avril 2022. Web : [Java Timer - How To Set A Timer In Java With Examples \(softwaretestinghelp.com\)](#)
- « Every Game Developer has a tab open to Stack Overflow ». stackoverflow. cc by-sa, 21 avril 2022. Web : <https://stackoverflow.com/>
- « Chess Fen forsyth edwards notation ». GGBud. 2021. Blog : [Forsyth-Edwards Notation FEN in Chess \(gbud.in\)](#)
- « ChessProgramming – Main Page ». Wikipedia. 30 juin 2021. Web : <https://www.chessprogramming.org/>

Outils :

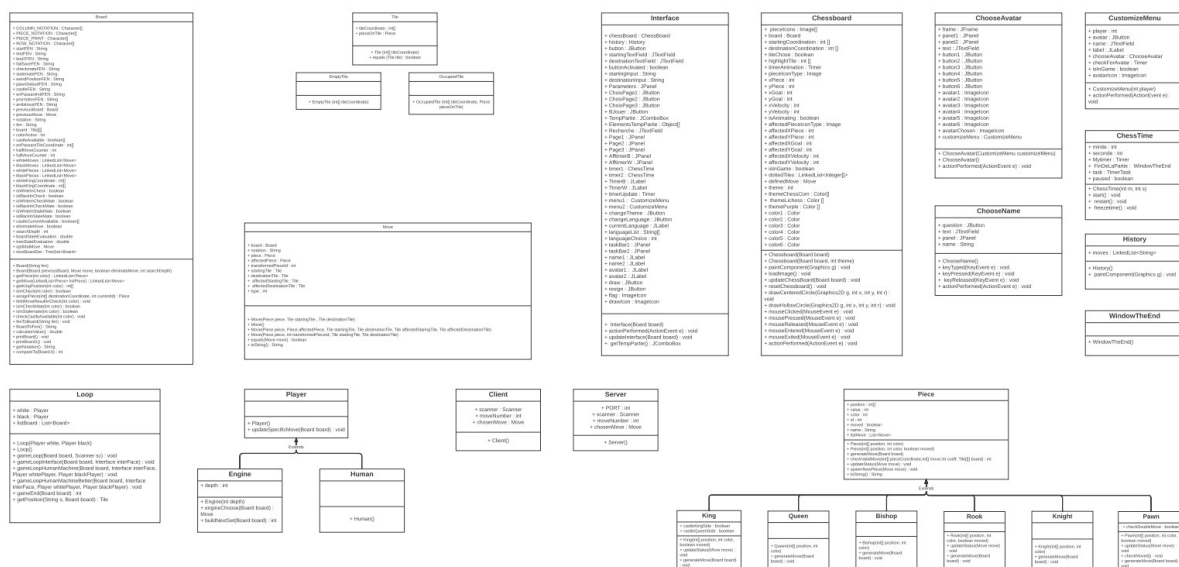
- FEN code generator by Lichess : <https://lichess.org/editor>
- The Stockfish chess engine (Stockfish 15) : <https://stockfishchess.org/>
- Leela chess engine (Leela) : <https://lczero.org/>

Vidéos :

- (2014, 31 décembre). How to code a TIMER in java! | Java Tutorial [Vidéo]. Soham Govande. YouTube. (4) [How to code a TIMER in Java! | Java Tutorial - YouTube](#)
- (2021, 21 avril). Comment créer un compte à rebours / timer personnalisé avec Storyline et du JavaScript [Vidéo]. Elearning Impact. YouTube. (4) [Comment créer un compte à rebours / timer personnalisé avec Storyline et du JavaScript. - YouTube](#)
- (2019, 15 août). Practice Java Program – Build your own Reaction Timer [Vidéo]. Alex Lee. YouTube. (4) [Practice Java Program - Build your own Reaction Timer - YouTube](#)
- (2020, 20 juin). Java TimerTask [Vidéo]. YouTube. (4) [Java TimerTask ☺ - YouTube](#)
- (2018, 29 mars). How to get selected value from JComboBox in java : NetBeans JComboBox JAVA Swing Tutorial [Vidéo]. YouTube. (4) [How to get selected value from JComboBox in java : NetBeans JComboBox JAVA Swing Tutorial – YouTube](#)
- (2020, 14 septembre). Java GUI [free] [Vidéo]. YouTube. [Java GUI ☺ **【Free】**](#)
- (2021, 1 novembre). Recursion in Java Full Tutorial – How To Create Recursive Methods [Vidéo]. YouTube. [Recursion in Java Full Tutorial - How to Create Recursive Methods - YouTube](#) (used for engine)
- (2018, 2 novembre). Java socket programming – Simple client server program [Vidéo]. YouTube. [Java socket programming - Simple client server program - YouTube](#) (TCP Server)
- (Dernier ajout : 2020, 25 mars). Java Chess Engine Tutorial [Vidéo]. 52 vidéos. YouTube. (1) [Java Chess Engine Tutorial - YouTube](#)
- (2020, 18 décembre). How To Make a Chess Board In Java – Java Chess Tutorial #1 [Vidéo]. YouTube. (1) [How To Make a Chess Board In Java - Java Chess Tutorial #1 – YouTube](#)
- (2021, 12 février). Coding Adventure: Chess AI [Vidéo]. YouTube. (1) [Coding Adventure: Chess AI - YouTube](#)
- (2021, 13 mai). How AlphaZero Completely CRUSHED Stockfish [Vidéo]. YouTube. (1) [How AlphaZero Completely CRUSHED Stockfish - YouTube](#)
- (2018, 20 avril). Algorithms Explained – minimax and alpha-beta pruning [Vidéo]. YouTube. (1) [Algorithms Explained – minimax and alpha-beta pruning - YouTube](#)

V) Structuration des données UML

Diagramme UML



VI) Suggestions d'amélioration

- Plus de modes : Chess960 et puzzle d'échecs
- Plus de propositions de temps de jeu : Echecs en Bullet 1-0
- Amélioration des fonctionnalités de l'ordinateur jusqu'à dépasser la machine la plus développée en échec : ordinateur sur chess.com
- Autre disposition des pièces au départ

VII) Carnet de route

Semaine	Description du travail réalisé
10/02	<ul style="list-style-type: none"> -Recherche du sujet -Réalisation du cahier des charges -Diagramme UML
17/02	<ul style="list-style-type: none"> -Rendu du cahier des charges -Début de programmation de la classe <i>Board</i> (tout ce qui se passera dans le jeu, fonctionnement du jeu sur le plateau) : Déclaration des attributs (<i>startFEN</i>, <i>testFEN</i>, <i>checkmateFEN</i>, <i>promotionFEN</i>...) Attribution numéro et lettre pour chaque case du plateau, Ecriture du constructeur avec un attribut : <i>String fen</i> -Début de programmation de la classe <i>Piece</i> : Déclaration des attributs (<i>position</i>, <i>color</i>, <i>id</i>, <i>value</i> et <i>name</i>), Ecriture du constructeur avec 2 attributs : <i>position</i> et <i>color</i>
Vacances d'hiver	<ul style="list-style-type: none"> -Avancée des classes précédentes : pour la classe <i>Board</i> les listes de <i>Move</i> et pour la classe <i>Piece</i> les move possibles avec <i>generateMove</i> -Début de programmation de la classe mère <i>Tile</i> : Déclaration des attributs (<i>tileCoordinate</i> et <i>pieceOnTile</i>), Ecriture du constructeur avec 1 attribut : <i>tileCoordinate</i>. + Ecriture des classes fille <i>EmptyTile</i> et <i>OccupiedTile</i> -Début de programmation de la classe <i>Chessboard</i> pour l'interface graphique : Déclaration des attributs (<i>startingCoordination</i>, <i>destinationCoordination</i>, <i>xPiece</i>, <i>yPiece</i>, <i>affectedXPiece</i>, <i>color1</i>...), Dimensionnement du plateau de jeu. -Début de programmation de la classe <i>Interface</i> : Etablissement de l'interface principale du Chess Game dans laquelle il y aura les fonctionnalités proposées et le plateau de jeu - Début de programmation de la classe <i>Move</i> : Déclaration des attributs (<i>board</i>, <i>startingTile</i>, <i>destinationTile</i>, <i>piece</i>...), Ecriture du constructeur avec plusieurs possibilités d'écriture : <i>Move(Piece piece, Tile startingTile, Tile destinationTile)</i>, <i>Move()</i>, <i>Move(Piece piece, Piece affectedPiece, Tile startingTile, Tile destinationTile, Tile affectedStartingTile, Tile affectedDestinationTile)</i>, <i>Move(Piece piece, int transformedPieceId, Tile startingTile, Tile destinationTile)</i> et ses méthodes.
03/03	<ul style="list-style-type: none"> -Avancée des classes précédentes : pour la classe <i>Board</i> ajout des attributs <i>boolean isWhiteInCheck</i>, <i>isWhiteInCheckMate</i>, <i>isWhiteInStaleMate</i> etc et ses méthodes, pour la classe <i>ChessBoard</i> ajout des méthodes -Début de programmation de la classe mère <i>Player</i> et ses méthodes -Début de programmation de la classe de <i>ChooseName</i> pour attribuer un nom aux joueurs sinon nom par défaut -Développement de la machine (jeu contre l'ordinateur)
10/03	<ul style="list-style-type: none"> -Avancée des classes précédentes, ajout de méthodes pour certaines classes -Début de programmation de la classe <i>ChessTime</i> pour ajouter une limite de temps au jeu, Déclaration des attributs (<i>seconde</i> et

	<i>minute</i>), Ecriture du <i>TimerTask()</i> -Début de programmation de la classe <i>Loop</i> : Déclaration des attributs (<i>white et black</i>), Ecriture du constructeur -Début de programmation des classes filles <i>Engine et Human</i> : Ecriture de leur méthode
17/03	-Avancée des classes précédentes -Début de programmation de la classe <i>ChooseAvatar</i> pour que le joueur puisse choisir un avatar : choix et déclaration des photos d'avatar à importer, création des boutons associés aux icônes (image des avatars), écriture de la méthode <i>actionPerformed(ActionEvent e)</i> -Début de programmation de la classe <i>WindowTheEnd</i> pour la page de fin : importation de l'image <i>Egalite</i> , Déclaration de l'attribut <i>String insert</i> (message de fin)
24/03	-Avancée des classes précédentes -Début de programmation des classes <i>Client et Server</i> : Déclaration des attributs + Ecriture du constructeur
31/03	-Avancée des classes précédentes -Début de programmation de la classe <i>CustomizeMenu</i> : Harmonisation des classes <i>ChooseAvatar et ChooseName</i> en créant une fenêtre contenant chacun des éléments des classes -Début de programmation de la classe <i>History</i> pour afficher chacun des move : Déclaration de l'attribut <i>LinkedList moves</i> et ses méthodes
07/04	-Avancée des classes précédentes : création de la page de fonctionnalité dans l'interface, 3 pages <i>New, Party et Custom</i> + création d'une instance de <i>CustomizeMenu</i> + couleur de fond
14/04	-Avancée des classes précédentes : Ajout de la fonctionnalité couleur du plateau de jeu (peut choisir la couleur du board)
Première Semaine des vacances de Pâques	-Avancée des classes précédentes : Ajout de la fonctionnalité des langues dans l'interface -Début de rédaction du document explicatif
Deuxième Semaine des vacances de Pâques	-Avancée des classes précédentes : Ajout de la fonctionnalité mode de jeu (jeu contre l'ordinateur ou jeu à 2 joueurs), choix de couleur des pièces (<i>white ou black</i>) -Développement du jeu en ligne (LAN, Network Local) Peaufinement

VIII) Pourcentage de chaque membre de l'équipe

Prénom	Pourcentage(%)
Khai	
Tung	
Hélène	
Hang	

Et cela conclue notre présentation !