

## LIFAP4 : Conception

Vous avez la charge du développement d'applications autour du jeu d'échecs. Au-delà du jeu, l'objectif est également d'avoir une entité qui va stocker un très grand nombre de parties jouées par des champions. Ces parties seront stockées dans une grande structure de données utilisées comme un dictionnaire de partie. A partir d'une configuration de jeu, cette entité proposera des coups à jouer en cherchant dans le dictionnaire de parties, soit pour aider le joueur, soit pour que l'ordinateur joue contre un humain. Il y aura plusieurs applications : l'application de jeu texte, graphique, et l'application de gestion et de visualisation de toutes les parties du dictionnaire de parties.

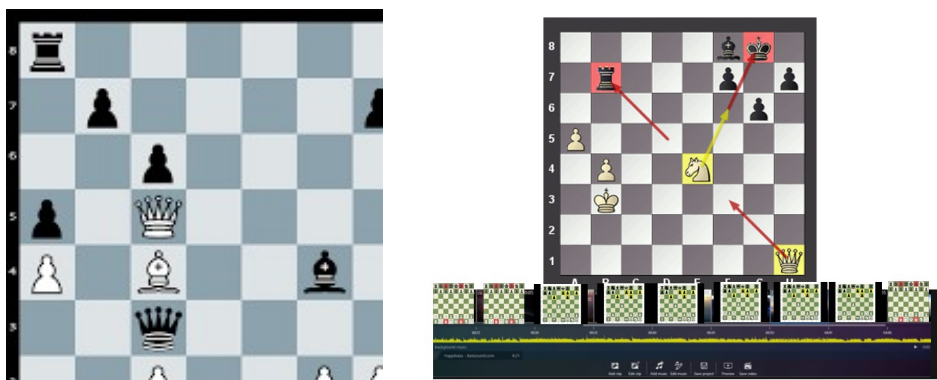
La 1<sup>ère</sup> application permettra de jouer aux échecs avec une visualisation au moins en mode texte et en mode graphique (2D). Les pièces ne pourront bouger qu'en respectant les règles du jeu. Chaque pièce stockera au moins un type de pièce, une position, une couleur, un indice d'importance et un ensemble de déplacements possibles (indépendamment des autres pièces). Une position et un déplacement possible peuvent être représentés par un vecteur 2D. Il faut gérer le fait qu'une pièce puisse être prise ou en jeu. Une partie est une succession de configurations de jeu. Une configuration de jeu représente toutes les pièces, la couleur du joueur qui doit jouer, etc. Par joueur, il y a 16 pièces. Il y aura au moins les fonctions suivantes : une fonction qui indique si un mouvement (coup) est valide ou non à partir d'une configuration de jeu et du mouvement d'une pièce ; une fonction qui joue un coup ; une fonction qui indique si la partie est terminée ou non ; des fonctions qui font les correspondances entre la case du damier et la pièce ; des fonctions qui donnent des informations sur le rapport de force permettant de savoir si un joueur est en bonne voie pour gagner ou non (par exemple en fonction des pièces prises), etc.

La 2<sup>e</sup> applications permettra de gérer le dictionnaire de parties : proposer des coups à un joueur en cherchant dans le dictionnaire ; importer des parties depuis un fichier de parties ; supprimer des parties non pertinentes ; sauver toutes les parties dans un fichier ; visualiser la partie étape par étape un peu à la façon d'une vidéo ; sélectionner des morceaux de parties intéressantes pour ne garder que certaines étapes à la manière du montage de vidéos, etc.

Concevez le diagramme des classes modélisant le noyau, ainsi que les applications. Indiquez si la classe fait partie du noyau ou de l'application (affichage). Vous êtes libre de constituer les classes de votre choix, avec les données de votre choix, voire d'ajouter toutes données qui vous sembleraient pertinentes.

Pour chaque module du logiciel, vous donnerez les classes : données et fonctions membres avec leurs paramètres (mode et type). Soyez pertinent sur les mutateurs et accesseurs, réfléchissez plutôt en « actions » de haut niveau. Pour des raisons pratiques de présentation sur votre feuille, il vous est possible d'écrire les fonctions membres en dehors du diagramme. Précisez alors clairement à quel module/classe elles appartiennent.

Remarque : on ne demande pas d'algorithme détaillant chaque partie mais plutôt que chaque classe prévoit les fonctions nécessaires.



*Un exemple des deux applications :*

*A gauche le jeu, à droite l'application montre une étape d'une partie avec les différentes options présentes dans le dictionnaire.*