|  |
| --- |
| Langage C++  Projet : Réalisation du jeu d’échec en C++  **IRAQI HOUSSAINI Mehdi**  **Ecole Supérieure d’Ingénierie en Sciences Appliquées**  **Année Universitaire**  **2020-2021** |

**Projet Langage C++ : Jeux d’échecs**

1. **Objectif**

Le but du projet est de réaliser un programme permettant à 2 joueurs humains de s'affronter aux échecs sur un plateau de jeu classique.

1. **Description et cahier des charges**

Les échecs intéressent depuis longtemps le monde de la programmation. D'une part dans un volet pédagogique, car il se prête naturellement à l'apprentissage de la programmation modulaire et l'étude de structures de données et d'algorithmes relativement simples. D'autre part, dans un volet plus avancé, car le développement d'un joueur artificiel compétitif représente un défi algorithmique.

Dans le cadre de ce projet, cette dernière partie ne sera pas obligatoire, mais pourra cependant être abordée par les binômes qui le souhaiteront. Les règles complètes du jeu d'échecs qui nous serviront de référence pour ce projet sont fournies dans un document supplémentaire.

Nous distinguerons pour ce projet 3 types de règles :

\_ Règles de base : toutes les règles du document hors réglés citées ci-dessous

\_ Règles avancées : roque, clouage, échange, prise en passant

\_ Fin de partie : mat et parties nulles

1. **Le développement**

Le développement se déroulera en 3 phases :

**Phase 1 :**

Conception, développement de la logique du programme et d'une interface utilisateur console/clavier.

Au terme de cette phase, il devra être possible pour 2 joueurs humains de jouer aux échecs avec les règles de base (donc sans fin de partie).

Une interface console/clavier sera implémentée et devra inclure : le démarrage de la partie ; la saisie à tour de rôle des coups par les joueurs ; la visualisation du plateau à chaque tour.

**Phase 2 :**

Conception et développement des règles avancées.

Au terme de cette phase, il devra être possible pour 2 joueurs humains de jouer aux échecs avec les règles de base ET les règles avancées (toujours sans fin de partie !).

**Phase 3 :**

Conception et développement d'une interface graphique.

Les mêmes fonctionnalités que pour l'interface console sont requises.

**Phase 4 :**

Les fonctionnalités suivantes pourront être implémentées par les étudiants (bonus à l'évaluation) :

\_ implémenter les fins de partie (mat, et autres fins de partie)

\_ implémenter une intelligence artificielle (jouer contre l'ordinateur)

\_ permettre à tout moment d'enregistrer l'état de la partie dans un fichier afin de la poursuivre ultérieurement lors d'une autre exécution du programme

1. **Organisation**

L'interface graphique devra être dissociée dans le code final. Cela signifie que le code produit pour l'interface graphique devra, autant que possible, directement réutiliser celui produit pour la phase 1 (i.e. sans le dupliquer).

Le projet s'effectuera en binôme et sera organisé de la manière suivante :

\_ Dans un premier temps, il est demandé aux étudiants de fournir une analyse descendante du projet.

Une date limite de rendu sera fixée. Fichier à déposer au format PDF uniquement.

\_ Dans un deuxième temps, une soutenance/présentation du projet aura lieu (les modalités seront expliqués par la suite par l’enseignant), sur machine. Le projet sera à déposer au plus tard une semaine avant la soutenance, accompagné d'un rapport de projet. Le rapport contiendra l'analyse (éventuellement mise à jour), une description du programme et de ses spécificités, ainsi qu'une explication détaillé des choix effectués et des difficultés rencontrées.

1. **Evaluation**

Les points suivants seront évalues prioritairement :

\_ Qualité du code du point de vue de la dissociation logique/interface graphique

\_ Qualité du code du point de vue de sa décomposition en fonctions élémentaires et/ou en classes

\_ Qualité des algorithmes et types de données proposés (i.e. adaptés aux sous-problèmes rencontrés)

\_ Commentaires en ligne et lisibilité du code

\_ Respect du cahier des charges

\_ Fonctionnement global du programme