RAPPORT STRATEGO 56149 & 56172

Liste d'écart :

Au niveau des règles du jeu aucun écart n'a été fait durant le développement.

Par rapport à notre analyse, nous avons apporté des changements que vous trouverez ci-dessous.

Ces changements ont été faits pour une question de simplification, car quand nous en sommes arrivés à implémenter le code, nous avions remarqué que nous pouvons faciliter beaucoup de choses.

Au niveau de la classe Soldier

- Transformation de la classe en GameElement
- Tout les attributs sont en std::optional
- Ajout de l'attribut decor
- Retrait de l'attribut movement
- Constructeur GameElement(PIECE,INT,PLAYER)
- Constructeur GameElement(string)
- Constructeur GameElement()
- Ajout des getteur
- Ajout des méthodes
    - isDecor()
    - isWater()

Au niveau de la classe Board :

- Disparition des deux listes soldierBlue, soldierRed
- Le tableau battleField et devenu un std::array<std::array<GameElement>>
- Ajout des méthode private suivante :
    - fillArray
    - mirrorVector
    - attack
    - reveal
    - unReveal
- Changement des attributs de la méthode initializeBattleField(Position,Player,GameElement)
- Ajout des méthodes public suivante :
    - moveEclaireur(Position & p, Position & toGo, Position dir, int distance)
    - fillVectorSoldier(vector<GameElement> & soldier, PLAYER color)
    - canBeTake(const Position &p, PLAYER player)
    - isFull()
- Retrait des méthodes suivantes :
    - getBattleField()
    - operator[](Position)

Au niveau de la classe Game :

- Ajout des attributs suivant :
    - Optional<GameElement *>oldSoldierRED
    - Optional<GameElement *>oldSoldierBLUE
    - Vector<string> movementRed
    - Vector<string> movementBlue
- Ajout de la méthode private checkCanMakeMove()
- Retrait des méthodes private suivantes :
    - canBeBeat(Soldier,Position)
    - caseIsEmpty(Position)
    - sameColorSoldier(Position)
    - fillMovementList(bool & canMakeMove, Position toGo)
    - makeMove(optional<GameElement> & saveSoldier,Position p, Position direction, DIRECTION d, int distance)
    - clearMovementList(const Position p)
- Ajout des méthodes public suivantes
    - canBePlaceStart(const Position & p, Player color)
    - canBeTake(const Position &p, Player player)
    - isFull()
    - nextPlayerPlacement();
    - goodFile(vector<string> const & pieces)
- Changement au niveau des méthodes
    - move(Position & p, DIRECTION & d, int distance = 1)
    - endgame retour -> int
    - getSoldier retour -> std::optional

Au niveau de la classe Position:

- Ajout de la méthode tostring();
- Retrait des setter

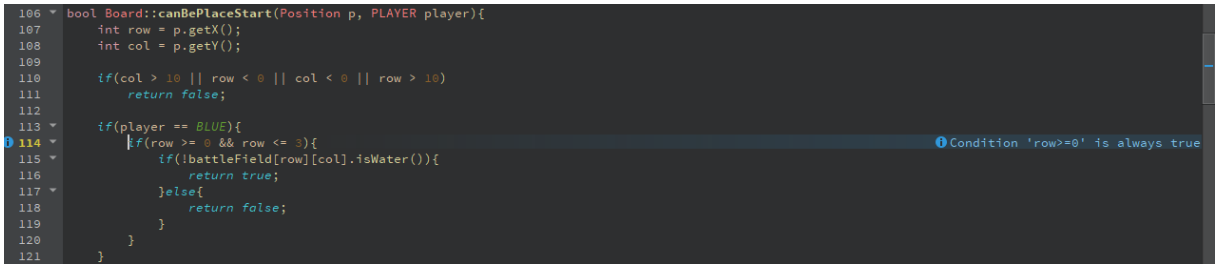
Les problèmes rencontrés ont été majoritairement résolus à l'aide des changements rapportés à notre analyse de départ.

L'une des plus grosses difficultés était de tester le jeu, surtout lors de la phase finale du projet , effectivement nous avions par exemple à placer 80 pièces pour créer une situation de bug et voir si elle était enfin résolue.

Une autre difficulté était l'implémentation de la règle qui spécifie pas plus de trois allers-retours consécutifs, mais après mûre réflexion, nous avons réussi à implémenter un algorithme fonctionnel.

Warning cppcheck non résolu

Celui-ci nous était impossible, car en mettant const Position & p nous avions une erreur.



Celui-ci a été laissé, car nous voulions absolument garder la condition telle quelle afin d'éviter tout bug par la suite.

Pour le développement du mode console, cela nous aura pris environ 35 heures.