

# Rapport Stratego 54147\_52075

## Remise Console

[Introduction](#)

[Readme](#)

[Ecart Consigne](#)

[Révélation des pièces](#)

[Placement interactif](#)

[Lecture fichier](#)

[Numérotation des cases](#)

[Sélection des pièce](#)

[Déplacement des pièces](#)

[Aller retour](#)

[Ajout consigne](#)

[Précision consigne](#)

[Bugs](#)

[Bugs Classes métier](#)

[Bugs Console](#)

[Relevé des Challenges rencontrés](#)

[La sécurité du code](#)

[Justifications pour chaque Warning](#)

[Produit par gcc](#)

[Produit par cppcheck](#)

[Estimation du temps passé](#)

[Conclusion](#)

## Introduction

Après un peu près un mois de codage enrichissant en challenge et apprentissage, nous voici arrivé pour la remise Console.

Cette remise comporte Un projet du jeu “Stratego” jouable en version console. Etant habitué et après consultation de notre maître assistant, nous avons décidé de suivre le Patron Modèle - Vue - Controller où Le Controller gérera les interactions console avec l’user, le modèle toute la logique du jeu et enfin la vue **uniquement** des affichages (comme recommandé par Mr NVS).

Tout au long du codage, nous avons fait de notre mieux pour être le plus proche possible des consignes et d'implémenter toutes les règles liées au jeu.

Au travers de ce Rapport vous retrouver notamment nos écarts face aux consignes, nos bugs et challenge rencontré et enfin nos justifications face au Warning.

## Readme

Attention à bien supprimer tout .pro en doublon afin de pouvoir run le programme sans soucis !

Les fichiers de configuration des pièces se trouvent dans src/model/red.txt & blue.txt

Bonne lecture.

## Ecart Consigne

### Révélation des pièces

- Pendant un combat aucune pièce n'est révélée, étant donné que le combat se fait "instantanément", en quelques millisecondes alors que "une pièce n'est révélée que quand un combat survient et uniquement le temps de résoudre ce combat" (Point 2 des consignes générales)

### Placement interactif

- Le joueur RED & BLUE dépose 1 chacun à leur tour alors que "les deux joueurs possèdent chacun initialement 40 pièces, qu'ils placent sans les révéler lors de la première phase du jeu" (Point 2 des consignes générales).
- En plus de cela les joueurs RED et BLUE déposent leur pion avec 2 valeurs int (une pour la ligne et une pour la colonne allant de 0 à 9) alors que "Les cases du plateau de jeu doivent être numérotées : -les colonnes, de la gauche vers la droite, sont numérotées de A à J. - les lignes, du haut vers le bas, sont numérotées de 1 à 10. On remarque, en conséquence, que les positions C5, C6, D5... ne sont pas des cases valides, tant pour le **placement initial** que lors des déplacements ultérieurs. Ainsi le **joueur <<bleu>>** place ses 40 pièces **sur les lignes 1,2,3,4** & le **joueur <<rouge>>** les place **sur les lignes 7,8,9,10**"

## Lecture fichier

- Pour le joueur rouge les pions sont placés de la colonne 6 à 9 en lisant le début du fichier (1er symbole = en haut à gauche) alors que “si ce 7 apparait sur la deuxième ligne du fichier du joueur <<rouge>>, ce major est positionné en E9” alors qu’avec notre code il sera positionné en E7 (E8 si les cases sont numéroté de 1 à 10) “... Ainsi il convient <<d’inverser>>**le placement (en ligne)** pour le joueur rouge”

## Numérotation des cases

- Notre plateau est numéroté de 0 & 9 pour les lignes (row) et de 0 à 9 pour les colonnes (columns) alors que “Les cases du plateau de jeu doivent être numérotées : -les colonnes, de la gauche vers la droite, sont numérotées de A à J. - les lignes, du haut vers le bas, sont numérotées de 1 à 10”

## Sélection des pièce

- Notre sélection se fait avec 2 int : 1 pour la ligne & 1 pour la colonne allant de 0 à 9 alors que “-**les colonnes**, de la gauche vers la droite, sont numérotées de **A à J**. - **les lignes**, du haut vers le bas, sont numérotées de **1 à 10**”

## Déplacement des pièces

- Nous déplaçons en notant 2 valeurs int, une pour la ligne et une pour la colonne alors que Les cases du plateau de jeu doivent être numérotées : -les colonnes, de la gauche vers la droite, sont numérotées de A à J. - les lignes, du haut vers le bas, sont numérotées de 1 à 10. On remarque, en conséquence, que les position C5,C6,D5... ne sont pas des cases valides, tant pour le placement initial que **lors des déplacements ultérieurs.**”

## Aller retour

- Dans le Manuel Stratego Original::La règle rapide::Déplacement “Il est interdit de faire effectuer à une pièce plus de trois allers-retours consécutifs entre deux cases identique” n’as pas été implémentée.

## Ajout consigne

# Précision consigne

/

## Bugs

### Bugs Classes métier

- Placement interactif non fonctionnel
- La gestion des 3 allers retours non fonctionnels
- Stratego.cpp::ligne24: nous avons fait un notifyObservers() **après** un break !
- Le mode interactif n'est pas fini

### Bugs Console

## Relevé des Challenges rencontrés

La méthode `void movePiece (Position pos);` nous a posé pas mal de complications, nous devons

1. gérer le move de l'éclaireur qui peut se déplacer aussi loin qu'il le souhaite pour autant que la voie est libre. Ensuite nous avons du gérer

**Solution** : nous avons checker que le move est bien perpendiculaire, récupérer la direction du move, simuler le move afin de vérifier que la voie est bel est bien libre, gérer le cas de l'attaque direct suite au move et du déplacement sur une case alliée.

2. gérer tous les cas de move possible et imaginables (hors du tableau, case occupé, case eau, l'attaque, la taille du move, la possibilité de se déplacer)

**Solution** : Implémenter tous les cas possible.

## La sécurité du code

Nous avons eu du mal à comprendre le système de triche avec sur le jeu,

**Solution** : cependant après explications au labo nous avons décidé d'implémenter en private toutes les méthodes **sensibles** d'être utilisé par un tricher et quant aux méthodes public notamment de la classe model Stratego, nous avons utilisé le système de "State" et donc par exemple pour pouvoir faire un move avec la méthode public

`void movePiece (Position pos);` il faut d'abord que le jeu soit dans l'état

PIECE\_SELECTED, (qu'une pièce soit sélectionné) car il doit être impossible de pouvoir move un pion sans au préalable le sélectionner. Donc si cette méthode est appelé en dehors de l'état PIECE\_SELECTED, elle n'aura aucun effet.

## Justifications pour chaque Warning

### Produit par gcc

- file\_reader.cpp::ligne 284 : fonction pas encore implémenté (mauvais timing de notre part)
- file\_reader.cpp::ligne 287 : fonction pas encore implémenté (mauvais timing de notre part)
- file\_reader.cpp::ligne 291 : fonction pas encore implémenté (mauvais timing de notre part)
- file\_reader.cpp::ligne 294 : fonction pas encore implémenté (mauvais timing de notre part)
- 2x view.cpp::ligne 178 : étant donné que nous avons qu'une seule vue, nous n'avons pas jugé pertinent d'utiliser le paramètre `const nvs::Subject *subject`

### Produit par cppcheck

- view.h::ligne 22 c'est vrai qu'il est pertinent d'ajouter `explicit` car l'attribut `Stratego` ne peut être implicitement converti par le compilateur cependant nous avons omis de le faire.
- Stratego.cpp::ligne 5 l'attribut `difficulty_` n'a pas été initialisé car dans notre jeu, la valeur de celui-ci est **forcément** choisi par le joueur et ce avant le début du jeu.

Mais effectivement c'est mieux de toujours donner une valeur d'initialisation.

- `stratego.h::ligne 25` effectivement nous avons fait une erreur d'inattention suite au stress le jour de la remise
- `Stratego.cpp::ligne 166` effectivement cette variable n'est jamais utilisé et donc cette ligne est en trop
- `file_reader.cpp::ligne 284` : fonction pas encore implémenté (mauvais timing de notre part)
- `file_reader.cpp::ligne 287` : fonction pas encore implémenté (mauvais timing de notre part)
- `file_reader.cpp::ligne 291` : fonction pas encore implémenté (mauvais timing de notre part)
- `file_reader.cpp::ligne 294` : fonction pas encore implémenté (mauvais timing de notre part)
- `file_reader.cpp::ligne 19` effectivement `cppcheck` a vu juste, on aurait très bien pu diminuer le scope de cette variable, ceci serait plus "propre", cependant dans le cas de notre code, le résultat est le même
- `file_reader.cpp::ligne 20` effectivement `cppcheck` a vu juste, on aurait très bien pu diminuer le scope de cette variable, ceci serait plus "propre", cependant dans le cas de notre code, le résultat est le même
- `View.cpp::ligne 61` remarque très pertinente de `cppcheck`, nous aurions du faire passer l'attribut `piece` par référence, cependant comme il s'agissait d'un `std::string` et non d'une `Piece` nous n'avons pas pensé à le faire

## Estimation du temps passé

En moyenne 6H/ semaine, un total de 36H chacun.

# Conclusion

Merci du temps consacré !

Si ceci n'est pas le 1er rapport que vous corriger, nous vous souhaitons bon courage avec une bonne canette de Pepsi pour vous accompagner durant la correction !