Stratego
DEV4 - Projet

R. Absil, J. Beleho, P. Bettens, M. Burda, P. Hauweele, N. Vansteenkiste

Année académique 2021 - 2022

Ce document présente l'énoncé de votre projet de `C++` pour l'UE DEV4, consistant à implémenter par groupe de deux étudiants le jeu Stratego. Ce projet est divisé en trois remises.

Table des matières

1 Introduction

Le but de ce projet est de vous faire implémenter en plusieurs étapes, par groupe de deux étudiants (binôme), la version originale du jeu Stratego, incluant une interface graphique. Som-

mairement, le but de ce jeu de plateau est de diriger une armée pour s'emparer du drapeau adverse, tout en protégeant le sien.

Votre implémentation est divisée en trois parties : une modélisation logicielle, une implémentation métier avec interface console et finalement une interface graphique sur base des classes métier précédemment développées. Ces divisions vous permettent ainsi de concevoir votre programme itérativement, par paliers.

Ce document détaille les consignes de ce travail. La section 2 apporte plus de détails quant aux règles du jeu jointes en annexe si elles devaient s'avérer insuffisantes. La section 3 détaille des fonctionnalités additionnelles demandées à titre de facilité d'utilisation du jeu. La section 4 détaille ensuite les différentes étapes de développement de votre projet, et la section 5 conclut ce document.

2 Règles du jeu

Les règles du jeu doivent être implémentées *telles que décrites* dans le manuel officiel du jeu, joint en annexe de ce document.

Au vu de ce manuel, on remarque en particulier :
— que le plateau est une grille rectangulaire de 10×10 cases composée de deux sous-grilles de 10×4 cases, connectées entre elles par trois « passages » de 2×2 cases ;
— que le jeu est dit *à information partielle*, dans le sens où un joueur n'a *a priori* pas connaissance de l'identité des pièces de son adversaire ;
— que le jeu se déroule en deux étapes principales : la mise en place des pièces, et « la bataille » qui s'en suit ;
— que les deux joueurs possèdent chacun initialement 40 pièces, qu'ils placent sans les révéler lors de la première phase du jeu ;
— qu'une pièce n'est révélée que quand un combat survient, et uniquement le temps de résoudre ce combat.

3 Spécificités supplémentaires

Cette section décrit des fonctionnalités additionnelles demandées dans votre implémentation, afin d'en faciliter la prise en main.

Numérotation des cases

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, D6, G5, G6, H5 et H6 ne sont pas des cases valides, tant pour le placement initial des pièces que lors des déplacements ultérieurs.

Ainsi, le joueur « bleu » place ses 40 pièces sur les lignes 1, 2, 3 et 4, et le joueur « rouge » les place sur les lignes 7, 8, 9 et 10.

Mise en place des pièces

Afin de faciliter l'étape de mise en place des pièces, il est demandé de pouvoir les placer soit interactivement, soit en lisant, par joueur, un fichier à la structure suivante :

- chaque fichier est un fichier texte composé de quatre lignes, potentiellement terminé par une ligne vide ;
- chaque ligne du fichier est composée des symboles repris dans la table 1, séparés par des espaces ;
- chaque symbole encode l'identité d'une pièce, tel que décrit dans la table 1 ;
- la position d'un symbole dans une ligne décrit la position (en colonne) de la pièce considérée. Par exemple, un 6 en cinquième position signifie qu'un commandant se trouve en colonne E. Si ce 6 apparaît sur la deuxième ligne du fichier du joueur « bleu », le major correspondant se trouve sur le plateau en position E2. Si ce 7 apparaît sur la deuxième ligne du fichier du joueur « rouge », ce major est positionné en position E9. Ainsi, il convient « d'inverser » le placement (en ligne) pour le joueur rouge, tel qu'illustré à l'exemple 1.

Pièce	**Symbole**	**Quantité**
Maréchal	10	1
Général	9	1
Colonel	8	2
Major	7	3
Commandant	6	4
Lieutenant	5	4
Sergent	4	4
Démineur	3	5
Éclaireur	2	8
Espionne	1	1
Drapeau	D	1
Bombe	B	6

Table 1 – Encodage des pièces

Exemple 1. Si les joueurs bleu et rouge choisissent de charger le *même* fichier

```
1 10 9 D 6 6 7 7 7 8 8
2 1 5 4 4 6 6 4 4 5 5
3 2 2 2 3 3 3 3 2 2 2
4 B B B 2 3 5 2 B B B
```

cela correspond au placement illustré à la figure 1.

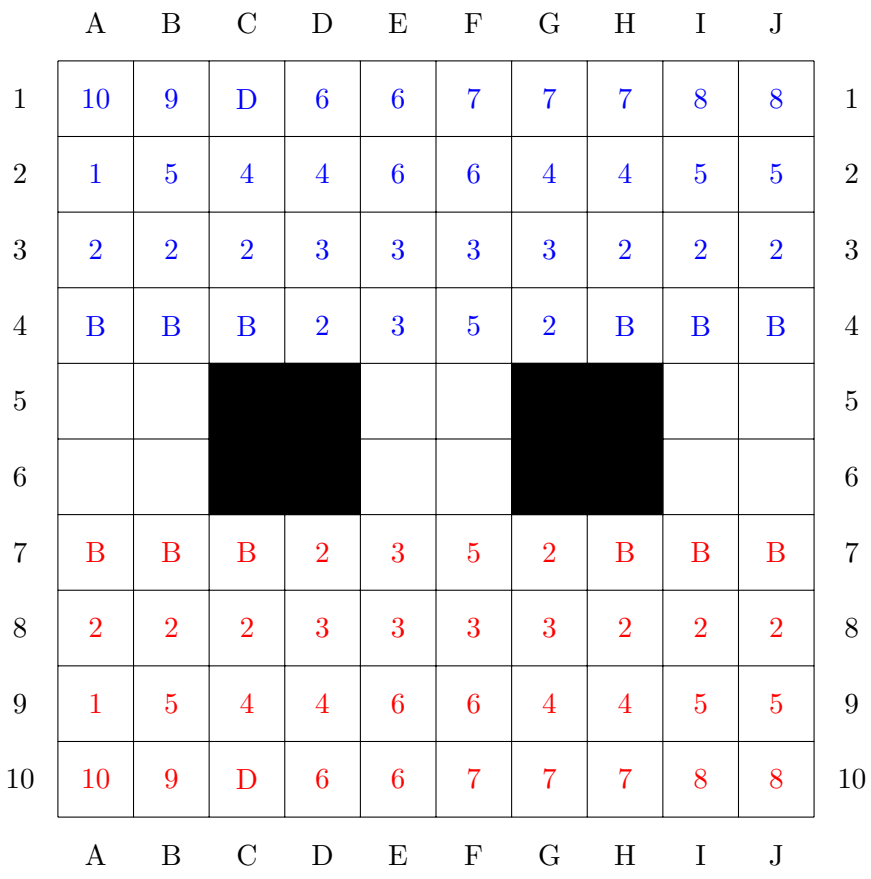

FIGURE 1 – Illustration du chargement du fichier de l'exemple 1.

Écarts par rapports aux règles

En dépit des règles de base du jeu,
— vous n'êtes pas tenus d'implémenter la règle vous interdisant de poursuivre indéfiniment une pièce ;
— vous n'êtes pas tenus d'implémenter la règle vous forçant à attaquer une pièce adverse en prise ;
— vous devez permettre, à titre de débugage, un mode de jeu permettant de laisser visible l'identité d'une pièce préalablement révélée lors d'un combat.

4 Remises

Dans la mesure où ce projet est réalisé sur un quadrimestre, plusieurs remises intermédiaires sont demandées, une par groupe de deux étudiants. Notez que dépendant de la contribution de chaque membre et des explications qu'ils fournissent, rien ne garantit que les étudiants au sein du groupe auront la même note.

Le but de ces remises n'est pas seulement de forcer l'étudiant à travailler régulièrement, mais à pouvoir le corriger rapidement. Ainsi, une prise de décision significativement mauvaise au début de la conception du projet sera corrigée par le professeur et aura un impact limité sur la suite du développement.

Ne perdez jamais de vue, lors de l'analyse et, plus tard, du développement des classes métier, que vos applications console puis graphique du jeu ne sont pas leurs uniques finalités. Considérez à tout moment que *vos* classes métier constituent une *bibliothèque* qui pourrait être reprise par un *autre* développeur que vous dans le but de produire une nouvelle application à interface textuelle ou graphique du jeu Stratego. En ce sens, à travers l'interface publique de la classe métier qui gère le jeu [1], il doit être impossible de tricher, volontairement ou involontairement, c'est-à-dire de mettre le jeu dans un état interdit par ses règles [2].

Remarquez qu'il vous est *imposé* d'implémenter le design pattern « Observateur / Observé », ou « Modèle / Vue / Contrôleur », notamment (mais pas exclusivement) pour les interactions avec l'utilisateur. Ainsi, il est *inadmissible* d'avoir des dépendances de la partie métier de votre projet vers la partie « contrôleur », console ou graphique [3] !

Notez également que, lors de l'implémentation de la partie console ou graphique, si vous deviez vous écarter de la modélisation prévue initialement, il convient de justifier judicieusement ce changement.

Par ailleurs, si vous utilisez des librairies tierces (autres que Qt) pour certaines parties de votre travail (par exemple pour effectuer des tests unitaires), vous devez faire en sorte que le travail remis soit « autosuffisant ». Ce n'est pas à votre professeur de régler les problèmes

1. De la *façade* du jeu, *pas* d'un contrôleur console ou graphique, *ni* d'une vue console ou graphique également !
2. Par exemple, il doit être interdit à un joueur de joueur deux fois d'affilée par le biais d'un appel à des fonctions membres publiques.
3. À l'évidence, il y aura des dépendances depuis les parties graphique et console vers la partie métier.

de dépendances externes (entre autres) ! Ceci exclut l'utilisation de librairies non portables (`windows.h`, etc.), ainsi que des chemins absolus vers des ressources, etc.

4.1 Dates et modalités de remise

Votre projet devra être remis en trois fois, une fois pour chaque étape charnière de votre développement :

1. remise modélisation le lundi 14 février 2022 à 18h00 : modélisation des classes métiers ;
2. remise console le vendredi 25 mars 2022 à 18h00 : implémentation des classes métiers et d'une interface en console ;
3. remise graphique le vendredi 6 mai 2022 à 18h00 : implémentation d'une interface graphique sur base des classes métiers de la deuxième remise.

Toutes ces remises seront effectuées via un dépôt git [4] (une remise par binôme). À ce titre, vous devez également utiliser le fichier `.gitignore` fourni en annexe de ce document, en l'ajoutant à la racine de votre projet.

Pour chaque remise, vous devez effectuer *en temps et en heure* un `commit` et un `tag` spécial, à l'aides des commandes suivantes :

— remise modélisation

```
1 git commit -a -m "modelization release"
2 git tag -a modelization -m "modelization release"
3 git push --follow-tags
```

— remise console

```
1 git commit -a -m "console release"
2 git tag -a console -m "console release"
3 git push --follow-tags
```

— remise gui

```
1 git commit -a -m "gui release"
2 git tag -a gui -m "gui release"
3 git push --follow-tags
```

L'avantage des tags est qu'ils vont vous permettre de spécifiquement désigner le commit que vous souhaitez faire corriger par votre maître-assistant. À ce titre, notez qu'il est primordial de bien respecter la nomenclature des tags (`modelization`, `console` et `gui`).

Notez que, pour la composition de votre binôme, vous pouvez vous associer à un étudiant n'étant pas de votre groupe, sous réserve que votre maître-assistant vous donne son accord. Le cas échéant, un maître-assistant vous sera assigné comme correcteur [5], et vous devrez vous plier à ses exigences personnelles.

4. Votre maître-assistant vous donnera plus de détail à ce sujet concernant la création des dépôts.
5. Vous ne pouvez donc *en aucun cas* choisir qui sera votre correcteur.

4.2 Modélisation métier

Ce travail est à remettre pour le lundi 14 février 2022 à 18h00. Vous devez, à ce stade, remettre une modélisation documentée de la partie métier (c'est-à-dire hors interaction clavier / souris utilisateur) de votre projet. Dépendant des exigences de votre maître-assistant, cette remise peut prendre diverses formes, telles que des headers documentés, d'un rapport, de diagrammes de classes, de séquence, de cas d'utilisations, etc.

Notez que même si le projet est divisé en trois remises, la modélisation doit inclure *tous* les concepts permettant de jouer, hors entrées et sorties à destination de l'utilisateur.

En aucun cas, en l'occurrence, il ne vous est demandé de remettre du code fonctionnel. Votre travail doit, à ce stade, bien représenter les différents composants de votre projet, leurs interactions, etc.

4.3 Implémentation console

Ce travail est à remettre pour le vendredi 25 mars 2022 à 18h00. Le but de ce travail est de remettre un projet complètement implémenté, avec interface console, qui réponde aux fonctionnalités décrites en section 2.

Lors de la première phase de jeu, il doit être possible pour chacun des joueurs de placer ses pions, soit de manière interactive, soit via un fichier.

Lors de la seconde phase de jeu, vous devez, à chaque tour, afficher le plateau sur la sortie standard, et demander à l'utilisateur ce qu'il souhaite faire. À ce titre, il est probablement utile de demander à un joueur le déplacement ou l'attaque qu'il souhaite faire par le biais de la numérotation des cases.

Notez que cette version doit être implémentée exclusivement avec la librairie standard, sans librairie tierce quelle qu'elle soit, si ce n'est pour les tests.

4.4 Implémentation graphique

Ce travail est à remettre pour le vendredi 6 mai 2022 à 18h00. Le but de ce travail est de remettre un projet complètement implémenté, avec interface graphique, qui réponde aux fonctionnalités décrites en section 2. Vous devez réaliser votre interface graphique à l'aide du framework `Qt`.

Les classes métier sont celles de la remise 2, à d'éventuelles corrections de bugs ou d'incohérences près. Notez cependant que seule l'interface graphique fait l'objet d'une évaluation à ce stade.

5 Conclusion

L'énoncé du projet, ses consignes, dates de remises et exigences ont été décrites. Pour rappel, il faut créer la version classique du jeu Stratego par groupe de deux étudiants. Notez que bien que ces travaux soient en groupe, rien ne garantit que les étudiants au sein d'un même groupe auront la même note.

Plusieurs remises intermédiaires sont demandées afin de vous suivre dans votre travail, respectivement le lundi 14 février 2022 à 18h00 pour l'analyse métier de votre travail, le vendredi 25 mars 2022 à 18h00 pour l'implémentation de l'interface console et le vendredi 6 mai 2022 à 18h00 pour l'implémentation de l'interface graphique.

Notez également que les exigences exposées ici sont celles communes aux maîtres-assistants en charge de l'unité d'enseignement DEV4. Il convient donc de vous renseigner des contraintes supplémentaires spécifiques de votre maître-assistant.

Références