Programmation orientée objet

Luis L. marques
Enzo Carré

2019/2020

Table des matières

Comment jouer

Le royaume est dans le chaos après que le roi se soit éteint. Vous êtes un duc et devez battre les autres ducs afin d'arriver jusqu'au trône. De plus, il y a également des ducs qui ne convoitent pas le trône et sont donc neutres.

Votre château est celui avec le drapeau, tandis que ceux des autres ducs sont les autres châteaux colorés. Les châteaux neutres sont gris.

Quand vous cliquez sur un château, ses informations sont affichées, qu'il soit allié ou ennemi. Ensuite vous avez des boutons en haut au centre de la fenêtre.

Dans l'ordre, il s'agit des boutons de :

— Recrutement.
— Mouvement (Mouvement si vous cliquez sur un château allié, Attaque sinon).
— Augmenter le niveau du château.
— Transférer de l'argent.
— Construire la muraille du château.
— Améliorer la caserne du château.

Avec ces commandes, vous devriez être capable de jouer et de vous amuser sans plus d'aide nécessaire. Le jeu se termine lorsque vous n'avez plus de châteaux (défaite) ou lorsque vous avez battu tous les ducs qui convoitent le trône (victoire).

Remarque : Lorsque vous transférez de l'argent, vous pouvez écrire le montant dans la case prévue à cet effet, mais il faut également appuyer sur Entrée pour valider la saisie, sinon le transfer s'effectuera avec une autre valeur.

Création du jeu

Outils utilisés

Le cours relatif à ce projet était enseigné en Java, et pour cette raison on a fait le projet en Java, avec JavaFX pour la partie graphique.

Pour les IDE, nous avons utilisé Eclipse et IntelliJ.

La version basique

Le client nous a d'abord commandé une version basique, avec moins de fonctions à implémenter.

Pour cette version, nous avions les règles suivantes :

— Seulement un type de troupe. Nous avons choisi le chevalier.
— Seulement un niveau pour les châteaux.
— Les troupes n'esquivent pas les obstacles lors de leurs déplacements.
— Les troupes ne sortent pas par la porte.
— Par tour, les châteaux génèrent des troupes plutôt que de l'argent.
— Les autres ducs ne font rien.

Ceci était dans l'objectif d'avoir une version fonctionnelle qui pouvait déjà être joué, même s'il n'y avait pas de compétitivité.

La version avancée

Pour la version avancée, nous avions d'autres règles que celles spécifiées au dessus :

— Plusieurs types the troupes. Nous en avons 3 : Chevalier, Onagre, Piquier.
— Plusieurs niveaux pour les châteaux. Nous avons choisi un maximum de 10 niveaux.

— Les troupes évitent les obstacles lorsqu'elles se déplacent.
— Les troupes quittent le château par la porte.
— Par tour, un château produit de l'argent qui peut être utilisé pour faire d'autres troupes ou améliorer le château.
— Les autres ducs font des actions basiques.

Comme bonus, nous avons ajouté ce qui suit :

— Les murailles autour des châteaux.
— Les transfer d'argent.

Les bugs connus

Version basique :

— De manière aléatoire, l'application se lance pas.
— Parfois, les troupes suivent la route de déplacement mais sont légèrement écartées de ce dernier.
— Après la conquête d'un château, renvoyer des troupes les fera apparaître en double, mais elles seront considérées par le jeu une seule fois, comme prévu.
— Barre blanche sur la droite de la fenêtre.

Version avancée :

— De manière aléatoire, l'application se lance pas.
— Barre blanche sur la droite de la fenêtre.
— Le recrutement multiple est imprévisible.
— Crash aléatoire (sûrement à cause de l'IA).

Comment on l'a fait

On était deux sur le projet. Dans la première partie, nous nous sommes lancés directement dans le code pour savoir ce qu'on pouvait faire et ce que nos outils nous proposaient.

Le travail en commun

En commun, nous avons fait le design de l'interface et des classes qu'on utiliserait par la suite. Au début du codage, nous avons également travaillé en simultané le code qui était partagé par les deux versions.

Le travail individuel

A un moment, nous avons divisé le projet pour partir sur les deux version différentes.

J'ai donc fait la version basique, la javadoc et la documentation tandis qu'Enzo s'est grandement occupé de la version avancée. Vers la fin du projet, nous étions tous les deux sur la version avancée.

Ressenti du projet

Luis L. Marques

Personnellement, au début, j'ai trouvé le projet un peu ennuyant, car ce n'est pas mon type de jeux et je n'étais pas familier avec le langage utilisé.

Après m'être familiaré avec le langage et la bibliothèque JavaFX, c'était plûtot sympa comme expérience.

Au niveau du contrat, j'ai trouvé qu'il manquait parfois de précision sur les joueurs et sur le système d'attaque. De plus, j'ai aussi senti qu'il nous manquait un peu de temps. Une ou deux semaines en plus aurait été parfait pour nous.

Au niveau du code, c'était une grande opportunité d'apprendre un nouveau langage et comme pour tout projet, une bonne opportunité d'améliorer son travail d'équipe.

Enzo Carré

Ce projet a été une réelle opportunité pour moi de découvrir lorganisation nécessaire pour un projet de cette ampleur. Mais de mon point de vue, le projet était beaucoup trop long pour le temps que nous avions. Nous avions les idées et les compétences pour y arriver cependant le temps manquait. Je pense que parce que je suis en MATH-INFO me donne un réel désavantage comparé aux classes de CMI et dINFO, parce que nous ne savons pas comment travailler sur ce genre de format de projet. Dans tous les cas, ce fut une expérience enrichissante pour moi et mon binôme, qui ma permis par ailleurs de découvrir une nouvelle façon de programmer (en binôme).