

Projet de C++

Introduction

Ce projet est un exercice dans lequel vous devez faire la démonstration que vous savez utiliser en c++ les éléments que nous avons vus ce semestre (la construction/destruction/copie, l'héritage, la généricité, les exceptions, les énumérations, la généricité, ...) qui doivent venir enrichir votre style et vous permettre d'écrire des programmes clairs, avec une analyse qui invite à une réutilisation simple lorsqu'il vous faut résoudre des problèmes conceptuellement proches.

Concrètement, nous allons vous décrire des jeux “*conceptuellement proches*”, où il vous faudra identifier des points communs, formaliser des concepts. Les prototypes que vous réaliserez vont organiser et réutiliser autant que possible ces concepts.¹

Le jeu le plus connu est celui des Dames ... Il se joue sur un damier, comme les autres jeux de cette famille. Les variations que vous aurez identifiées d'un jeu à l'autre vous guideront dans vos choix d'abstraction/spécialisation. Avant de commencer, lisez les règles, cela vous permettra d'avoir en tête ce qu'il faudra factoriser tôt ou tard.²

Généralités

- Le projet est à faire en binôme. Les monômes ne seront pas acceptés sauf pour des questions de parité dans les groupes de TD ou des cas très exceptionnels. Un forum Moodle sera ouvert pour mettre en contact ceux qui cherchent un partenaire. Il n'y aura absolument pas de trinômes, et il est entendu que si des travaux se ressemblent trop nous prendrons les sanctions qui s'imposent.
- La note de soutenance pourra être individualisée, chacun doit donc maîtriser l'ensemble du travail présenté, y compris la partie développée par son camarade.
- Il vous faut déclarer la constitution de votre binôme dans les 15 jours à venir, en choisissant un groupe **binôme_xy** dans l'activité d'inscription mise en place sur moodle. Ceci nous permettra de nous assurer que le travail a bien commencé, et d'organiser la suite.
- La soutenance aura lieu durant la période des examens, début janvier. Votre travail sera à rendre quelques jours auparavant. Nous vous donnerons les dates exactes lorsque les réservations seront confirmées.

Jeu de dames et variantes

Les différents jeux que nous vous proposons se déploient sur un damier qui visuellement alterne des cases sombres et claires. Ils se jouent en général à deux joueurs qui se font face. Les types de pièces avec leurs déplacements et leurs méthodes de prise varient légèrement d'un jeu à l'autre, mais sont sans grandes surprises.

L'important est de reconnaître pourquoi ils peuvent être considérés comme étant de la même famille : il y a des constantes, des ressemblances. Elles doivent être exploitées : il faudra réfléchir à une architecture qui permette de mutualiser le code correspondant à ces similitudes.

Voici une présentation sommaire de 6 jeux. Nous les avons regroupés par 2 et vous laissons la liberté d'en choisir un dans chaque section.

1. Réutiliser n'est pas copier-coller ! On veut utiliser les *même* classes de base dans le contexte de jeux différents
2. Rappel : un code ne s'écrit pas d'une seule traite, vous serez amenés à le réorganiser.

a) Deux jeux simples où les mouvements sont uniformes

Butin

Le jeu se joue à au moins deux joueurs sur un damier rempli de pièces de trois couleurs qui n'appartiennent à personne au départ. Lors de son tour, un joueur va en capturer : la prise de pièces s'effectue en faisant sauter une pièce par dessus des pièces voisines, éventuellement en cascade. Chaque couleur correspond à une certaine valeur, et les joueurs voudront maximiser la valeur totale de leurs prises.

Voir http://jeuxstrategieter.free.fr/Butin_complet.php

Incognito

La nature d'une des pièces est secrète, et il faut la protéger de l'adversaire. Ce petit mystère introduit un aspect plus psychologique dans le ressenti du joueur, mais formellement vous y retrouvez des mouvements, une notion de prise, ainsi qu'une zone du damier qui a un rôle particulier.

Voir http://jeuxstrategieter.free.fr/Incognito_complet.php

b) Deux jeux plus tactiques, avec des pièces différenciées

Les Dames

Le plus classique de cette sélection, beaucoup d'entre vous le connaissent déjà. Le jeu est au départ une phase d'affrontement frontal avec une progression des pièces vers celles de l'adversaire qui lui font obstacle. Si un joueur arrive à traverser la zone des troupes adverses, une de ses pièces "pion" sera promue en pièce "dame" et ses mouvements seront alors plus dangereux.

C'était l'inspiration initiale de ce sujet, il est naturel que vous le programmiez, mais vous pouvez le remplacer par Gounki.

Voir https://infolib.re/storage/files/documents/jeux_de_societe/Les%20dames%20-%20R%C3%A8gles%20du%20jeu.pdf

Gounki

Des pièces rondes se déplacent comme le font les pions du jeu de dames et des pièces carrées se déplacent sur ses cases voisines verticales et horizontales. Sous certaines conditions elles peuvent s'associer et combiner leurs caractéristiques de déplacement. Elles peuvent aussi se dissocier.

Voir <https://www.boardspace.net/gounki/english/Gounki-en-v1.0.pdf>

c) Deux jeux avec un terrain différent

Ces deux jeux exploitent les zones à la frontière entre les cases pour y placer des barrières (fixes ou pas). L'objectif est de prendre au piège l'adversaire.

Safari est plus simple car les barrières sont fixes, sans notion de prise de pièces adverses.

Bulltricker ressemble davantage au jeu de dames complet.

Voir http://www.bulltricker.com/fr/regle_simplifiee.html et <https://escaleajeux.fr/jeu/prene/>

Cahier des charges

On rappelle encore une fois que la bonne utilisation des notions de Programmation Orientée Objet constituera une part importante de l'évaluation. Même s'il était fonctionnel, un code qui

ne comporterait qu'une seule classe serait un cas extrême fortement pénalisé. Essayez d'illustrer au maximum les aspects vus en cours.

Au minimum, voilà ce que votre code devra réaliser :

- Un environnement d'accueil (qui peut être textuel) permettant de procéder à une démonstration. Vous penserez à un paramétrage rapide, l'orientation de l'utilisateur vers un choix de jeu, l'accès à des situations particulières etc ...
- L'implémentation de **trois** jeux, sélectionnés dans **chacune** des catégories suivantes :
 1. Butin ou Incognito
 2. Dames ou Gounki
 3. Bulltricker ou Safari
- Un affichage graphique du plateau en FSML. On pourra utiliser la console et des entrées au clavier.

Vous pouvez naturellement personnaliser votre projet en introduisant une IA, imaginer des options à plusieurs joueurs, soigner l'aspect graphique etc ...

Conseils habituels

Sauvegarde

Sauvegardez régulièrement votre travail. Chaque fois que vous envisagez une modification importante, conservez bien la version antérieure afin d'éviter des catastrophes. (Utilisez git par exemple)

Décomposition du code

Pour pouvoir maîtriser la complexité de votre travail, et être efficace en binôme, il vous faut absolument le décomposer en objets et méthodes qui joueront des rôles que vous aurez bien défini.

On rappelle qu'il est toujours préférable d'avoir plusieurs petites méthodes plutôt qu'une seule qui ferait un travail compliqué à déchiffrer. Si vous avez quelque part dans votre code un bloc qui fait plus d'une vingtaine de lignes, alors il est quasi-certain que vous devriez vous relire pour introduire une phase intermédiaire.

Développement progressif

Faire trop de généralisation dès le départ peut vous conduire en pratique à des impasses. Visez un premier objectif raisonnable³. Vous le retravaillerez ensuite.

Vue et Modèle

Ne mélangez pas trop les parties du code qui correspondent à ce qui est propre au jeu, et celles qui correspondent à l'affichage. Séparer autant que possible le modèle et sa vue.

3. faisable en une quinzaine de jours

Aspects pratiques de l'évaluation

Rendu

Les travaux seront à rendre sur **Moodle** sous la forme d'une archive **binomeNum.tar** (Utilisez impérativement ce format)⁴. Elle s'extraira dans un répertoire *nom1-nom2*, et contiendra :

- les sources et ressources (images ...) de votre programme ;
- un Makefile
- un README qui indique succinctement ce qu'est ce projet, et comment effectuer la prise en main (compilation, exécution et utilisation). Le correcteur ne doit avoir que deux choses à faire⁵ :
 1. décompresser votre dépôt dans une console unix ;
 2. lire votre fichier README et y trouver la ligne de commande qui lance le programme.
- un rapport au format *PDF* **rédigé** explicitant :
 - ce qui a été traité,
 - les aspects les plus significatifs,
 - les quelques problèmes connus,
 - l'état des extensions que vous n'auriez pas eu le temps d'implémenter.
 - un ou plusieurs diagrammes UML avec un niveau de détail suffisant pour comprendre votre architecture.
- toute chose utile pour rendre la soutenance fluide.

Soutenance

Elle se déroulera devant un ou deux enseignants dans un mélange de questions et de tests. Vous commencerez par une démonstration afin que nous ayons rapidement une vue d'ensemble ; nous étudierons votre architecture en nous basant sur votre diagramme ; vous nous guiderez dans votre code lorsque nous chercherons à vérifier la cohérence entre vos explications et votre réalisation. Il pourra vous être demandé d'écrire quelques lignes de code. Votre style sera scruté.

Vous pouvez préparer des supports pour répondre rapidement à des questions que vous anticiperiez. Dans tous les cas, il faudra privilégier les échanges avec le jury.

4. `tar cvf binomeNum.tar fichiers...` crée l'archive `mon_fichier.tar`

5. Testez vous même votre rendu sur une autre machine