

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, ...). Nous souhaitons également que votre programme soit bien conçu, et idéalement qu'il s'inspire de patterns que nous verrons ensemble.

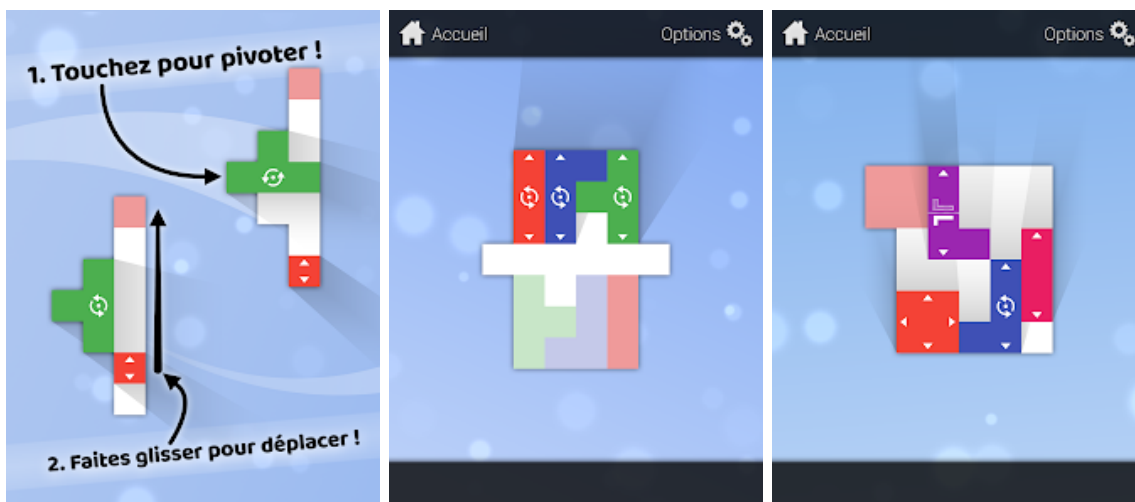
Normalement le sujet est suffisamment important pour que les IA génératives ne vous donnent pas de solution globale. Vous pouvez vous aider de ce qu'elles vous proposent, mais la moindre des choses est d'adapter ce qu'elles vous proposent et surtout d'être capable de défendre chaque partie de votre code à l'oral, à la fois en détail et dans la structure générale.

La soutenance sera longue, et donnera lieu à toutes sortes de questions.

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.

Présentation du jeu PieceOut



La figure de gauche est un petit tutorial qui vous permet de comprendre rapidement. Il s'agit de manipuler des pièces inspirées de celles de téttris. L'indication 1 vous invite à faire pivoter la pièce verte sur son centre dans le sens indiqué par l'icône (sens trigonométrique ici), cela vous amènera à la situation 2. La petite pièce rouge est déplaçable dans 2 directions : celles qui sont indiquées par les flèches à ses extrémités, elle n'est plus bloquée par la pièce verte et on peut à présent l'amener à sa destination indiquée en rouge pale. Le niveau sera alors terminé.

Plus généralement, le but est d'atteindre une configuration alors que les mouvements des pièces sont limités. C'est un casse tête. La rotation est orientée, la translation limitée à certaines directions, et la surface que peuvent occuper les pièces est limitée. Vous en voyez d'autres exemples dans les deux autres figures. Le blanc est utilisé pour indiquer la surface libre. Les couleurs pales viennent en superposition des cases blanches pour indiquer les zones où il faut amener les pièces (elles sont en couleur vives).

Le niveau présenté sur la figure du milieu est déjà plus compliqué. Notez d'abord les symboles : chaque pièce peut être tournée dans le sens anti-horaire, et dans leur position initiale elles peuvent être déplacées verticalement (mais la bleue est bloquée par la verte). Si vous pouviez tourner une pièce d'un quart de tour, le déplacement autorisé deviendrait alors horizontal.

Voici une séquence de mouvement possible : descendre le bloc rouge de 2 cases, le tourner, le déplacer vers la droite... pas sûr que ce soit la solution ...

Dans la troisième illustration vous rencontrez un nouveau symbole, il s'agit d'une symétrie par rapport à l'axe indiqué.

Une démonstration vidéo est disponible ici : <https://youtu.be/08M9ZJqARP0> et nous vous invitons à télécharger le jeu sur votre téléphone pour l'essayer.

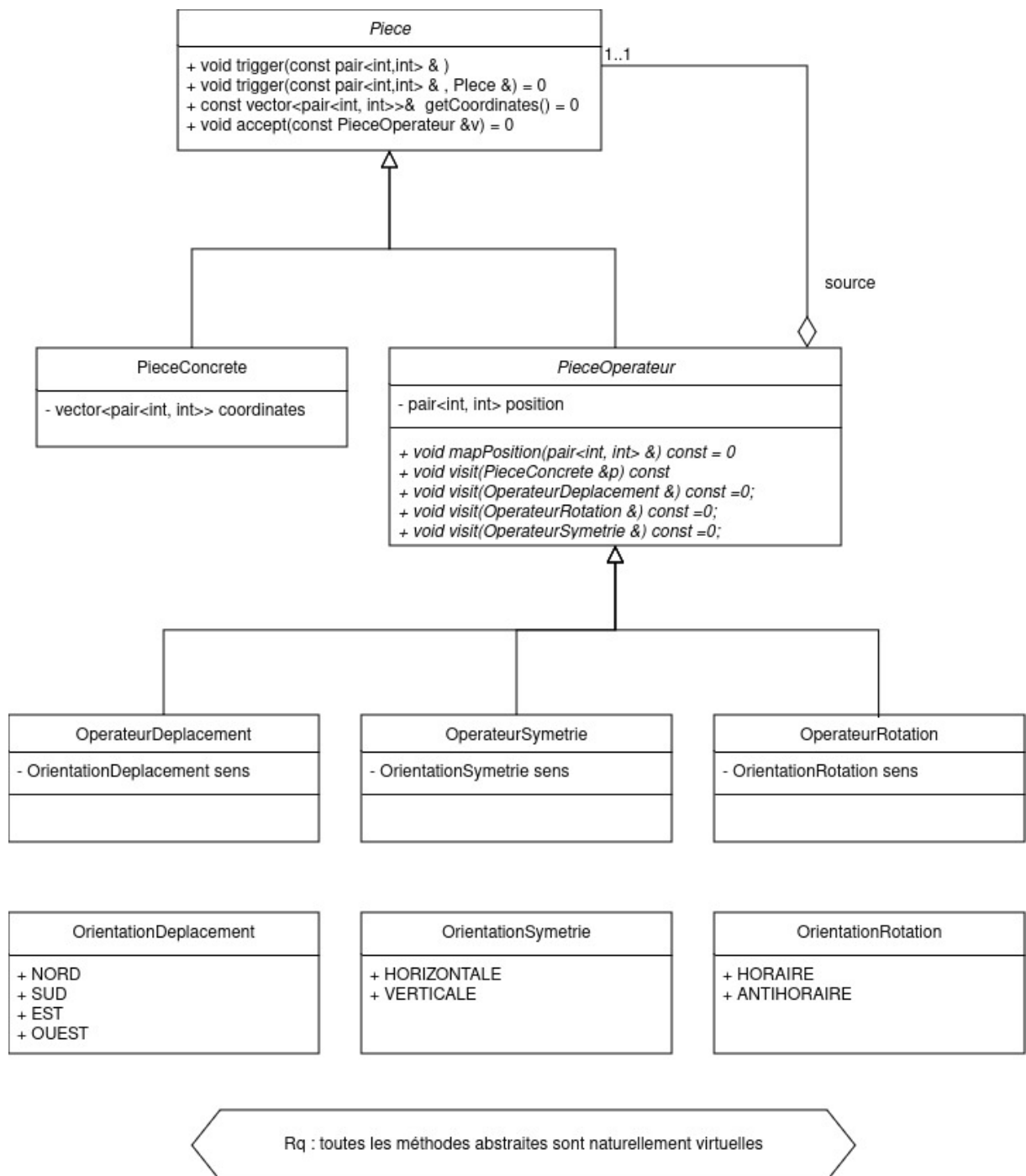
Un point de départ pour la visualisation

Vous utiliserez la bibliothèque SFML pour le rendu graphique. Il vous faudra installer cette bibliothèque (voir cours). Pour vous faire gagner un peu de temps, et pour l'illustrer nous vous fournissons un fichier `ebaucheVue.cpp`.

Un point de départ pour le modèle

Dans une première approximation les pièces peuvent être vue comme un assemblage de cases vierges. Puis nous les "décorons" en positionnant des actions (rotation, déplacement, symétrie). Les effets de ces opérations se concrétisent lors d'une "visite" complète de la pièce décorée.

Ce vocabulaire signifie que nous avons réfléchi au problème en nous inspirant de deux patrons : Decorator et Visitor. Vous pouvez les comprendre assez facilement à l'aide du diagramme UML suivant qui mixe ces deux patrons, et du fichier `ebaucheModel.cpp` qui est juste une ébauche de faisabilité à retravailler.



Travail à faire

L'objectif est de réaliser le plus de choses possible du jeu, d'y ajouter une touche personnelle. Au minimum, voilà ce que votre code devra proposer :

- L'ajout de décorations,
- La définition du modèle (espace de jeu, detection de gain)
- Un environnement d'accueil (qui peut être textuel) permettant de procéder à une démonstration. Vous penserez à un paramétrage rapide, l'accès à des situation particulières

- ...
- Nous vous encourageons à développer votre code en utilisant d'autres patterns. Par exemple si vous traitez la résolution de puzzle, si vous introduisez la possibilité de revenir en arrière, de jouer à deux, etc cela doit être fait de manière élégante.

Conseils habituels

- 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)
- 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.
- Séparer autant que possible le modèle et sa vue, c'est à dire ne mélangez pas trop les parties du code qui correspondent à ce qui est propre au jeu, et celles qui correspondent à l'affichage.
- Illustrez ce que nous avons vu ensemble : ce qui est créé doit être détruit ; ne copiez que ce qui est nécessaire ; utilisez des références et pas seulement des pointeurs ; ajoutez des const où c'est le cas ; séparez bien votre code ; utilisez la stl à bon escient etc ...

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)¹. Elle s'extraira dans un répertoire *nom1-nom2*, et contiendra :

- les sources et ressources (images ...) de votre programme ;
- un Makefile compréhensible
- 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 faire² :
 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 ;

1. `tar cvf binomeNum.tar fichiers...` crée l'archive `mon_fichier.tar`
2. Testez vous même votre rendu sur une autre machine

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 pour nous convaincre de vos acquis.