

Projet de Langage Objet Avancé - C++

I) Généralités

Ce projet a pour but la construction d'une plateforme, ou framework, permettant d'implémenter plusieurs jeux proches : il s'agit de jeux de courses, où l'objectif est de progresser le long d'un trajet pour atteindre un but.

Un framework est une sorte de bibliothèque logicielle dont la caractéristique principale est l'usage important de différentes formes de généricité et qui définit un cadre encourageant certaines architectures logicielles. L'obtention de ce framework devra se faire à l'aide d'une analyse soignée des différents jeux. Pour chaque jeu, il devra être possible de mélanger des joueurs humains et des robots. Les joueurs humains interagiront à l'aide du clavier. Les robots devront pouvoir jouer automatiquement sans pour autant posséder une quelconque intelligence (ils doivent toutefois respecter les règles du jeu). Le premier jeu à implanter sera le jeu du serpent et des échelles.

Il devra vous servir comme un test de votre framework. Vous noterez que les jeux diffèrent assez peu les uns des autres, si votre framework est correctement architecturé, l'implémentation de chaque nouveau jeu s'en trouvera grandement simplifiée.

II) Réalisation

Votre code doit être proprement conçu, et documenté. Les réalisations de tests, de phases de démonstrations sont fortement encouragés. Ne vous lancez pas tout de suite dans l'écriture du code ! Commencez par prendre du temps pour analyser comment ces jeux fonctionnent et quelles sont les briques fonctionnelles utilisées, etc. Il n'est pas indispensable d'avoir une interface graphique très évoluée (c'est même très fortement déconseillé). Une interface graphique avec un code mal conçu et des implémentations douteuses pourrait être source d'incompréhensions dans la notation. Pour vous éviter de partir dans tout les sens, la seule bibliothèque graphique que vous pourrez éventuellement utiliser est SFML, reportez vous à l'annexe en fin de sujet.

Cela ne signifie nullement qu'il ne faut pas employer un motif conceptuel de type MVC. L'accent doit être mis sur l'architecture logicielle, de sorte que le code soit aisé à lire et modifier ; ce point est très important car il permettra de vérifier vos compétences dans la maîtrise d'un développement orienté objet. Le framework devra avoir été pensé de sorte qu'on puisse sans trop de difficultés imaginer comment implanter d'autres jeux du même type.

III) Coté pratique

Le projet doit être réalisé en **trînômes**, nous vous en demanderons la constitution très rapidement. Vous devriez être capable de réaliser 3 ou 4 de ces jeux.

Vous soumettrez vos travaux sur Didel, sous forme d'une archive nommée *nom1-nom2-nom3*. Elle devra contenir :

- les sources de votre programme
- un fichier nommé **README** qui indique comment on se sert de votre programme ;
- un rapport au format *PDF* de quelques pages expliquant concisément les parties traitées, les problèmes connus, et les pistes d'extensions que vous n'auriez pas encore implémentées. Naturellement il contiendra un diagramme de classes. Vous imprimerez ce rapport pour le jour de la soutenance.

IV) Présentation des jeux

a) Serpents et échelles

Si besoin vous trouverez une description détaillée de ce jeu sur http://www.ludosbxl.irisnet.be/docnotice/echelles_et_serpents_2010_02_8_14_04_35.pdf



L'idée est la suivante : les joueurs chacun à leur tours lancent un dé et font progresser leur pion d'autant de cases. Si sur la case destination se trouve le bas d'une échelle, le pion progresse encore gratuitement jusqu'en haut de l'échelle ; si par contre il tombe sur une extrémité d'un serpent, il dégringole jusqu'à l'autre extrémité.

Ce jeu très simple dispose d'une variante où sur une case orange le joueur rejoue immédiatement, et sur une case verte il passe son prochain tour.

b) Variantes : jeu pédagogique

Lorsqu'un joueur se pose sur une case, une question lui est posée. Une version du jeu pourrait compter le nombre de bonnes réponses et établir un score à la fin du jeu. Une autre version pourrait consister à dire que seule une bonne réponse permet de valider le mouvement, une mauvaise réponse l'annulerait.

Vous pouvez trouver de tels jeux sur <http://www.jf-noblet.fr/jeux/atomes/atomes1.htm> ou sur <http://www.bonjourdefrance.com/n12/jeux/oiebd7.html>.

Pour faire au plus simple vous pouvez vous contenter de poser des questions de calcul mental.

c) Variantes à plusieurs pions

Les joueurs disposent à présent de plusieurs pions, et ils peuvent choisir celui qu'ils vont déplacer avant ou après avoir effectué leur lancer de dés. Le but est de tous les faire parvenir à l'arrivée, en premier.

d) Variantes sans dés, avec des cartes

Cette variante est inspirée du jeu Cartagena
(http://jeuxstrategie.free.fr/Cartagena_complet.php)

Pour résumer, voici ce que nous retenons de la règle pour notre projet :

- Les joueurs disposent de plusieurs pions
- Le plateau est toujours une suite linéaire de cases qui portent des symboles. Il n'y a que 4 ou 5 symboles différents.
- Il n'y a pas de dés, mais les joueurs disposent de cartes sur lesquelles se retrouvent un symbole, les mêmes symboles que l'on retrouve sur les cases.
- A son tour le joueur choisit à la fois une carte et l'un des ses pions. Le mouvement associé consiste à amener ce pion à la prochaine case libre qui a le même symbole que celui de la carte jouée. C'est ainsi que les pions progressent.
- Puisqu'au départ le joueur a un nombre limité de carte, il y a une règle qui lui permet d'en piocher : au lieu d'avancer, il peut désigner l'un de ses pions qui reculera jusqu'à la première case précédente qui ne sera occupée que par seulement 1 ou 2 autres pions. Selon le cas il piochera une ou deux nouvelles cartes. S'il doit reculer jusqu'au départ il piochera deux cartes.

e) Le jeu *Numeri*

Chaque joueur dispose de 6 pions, numérotés de 1 à 6, qui sont tous sur la case départ au début du jeu. Lorsque c'est son tour le joueur lance un dé et obtient une valeur x . Il a la possibilité de jouer les pions de son choix dès lors que leur somme vaut x . (Ainsi s'il fait 4, il peut sélectionner le pion 4 seul, ou bien les pions 1 et 3, mais il ne peut rien faire avec les pions 2, 5, 6). Le joueur prend alors son/ses pions un à un et les fait parvenir à la prochaine case libre (comme au saute mouton).

Si à la fin de son tour il obtient un nouvel alignement de trois de ses pions, il rejoue.

Et ainsi de suite, chacun à son tour. Le jeu s'arrête lorsque les 3 dernières cases sont occupées.

Pour déterminer le gain de la partie, on évalue la position sur le plateau au moment où le jeu s'arrête : chaque case porte une valeur. La figure suivante présente le plateau du jeu standard, le départ étant juste avant les cases de valeur -3. Les cases sans valeurs particulières sont nulles. Chacun des pions d'un joueur multiplie la valeur de la case sur laquelle il se situe par son numéro. La somme obtenue détermine le score d'un joueur, et le vainqueur.



V) Interface graphique

Vous vous limiterez à un usage très simple de la librairie SFML et aucune autre. Le site <http://www.sfml-dev.org/tutorials/2.3/index-fr.php> décrit la procédure d'installation de SFML

Sur didel nous vous fournissons un exemple qui ouvre une fenêtre et affiche deux images qui se superposent : vous vous en inspirerez en utilisant l'image d'un plateau de jeu comme image de fond, et des images de pions de votre choix.

SFML est installé sur lucien. Voici la marche à suivre pour effectuer votre premier test :

- Importez les fichiers de DIDEL
- effectuez `g++ -std=c++11 -c All_In_One.cpp`
- puis `g++ All_In_One.o -o sfml-app -lsfml-graphics -lsfml-window -lsfml-system`
- lancez `./sfml-app`
- dans la console indiquez une direction 'g' ou 'd' pour faire déplacer un pion

Ce genre de rendu graphique sera largement suffisant.