

Cours 420-520-SF Approfondissement de la programmation objet Cégep Sainte-Foy Département d'informatique François Gagnon Automne 2014	TP 1 20 %
--	--

CONSIGNES

- Ce travail se fait individuellement ou en équipe de 2 ou 3.
- Vous devez remettre, par Omnivox, une archive zip de votre projet Eclipse. Votre projet Eclipse et son répertoire racine se nomme selon la convention suivante : Nom1_Nom2_Nom3_TP1 (chaque étudiants de l'équipe doit mettre son nom de famille).
- À la racine de votre projet Eclipse vous devriez avoir un jar pouvant s'exécuter avec la commande :

Java -jar connect4 R C V
- Vous devez vérifier que l'archive remise respecte bien les critères et que votre jar s'exécute sans problème avec la commande indiquée.

MISE EN SITUATION

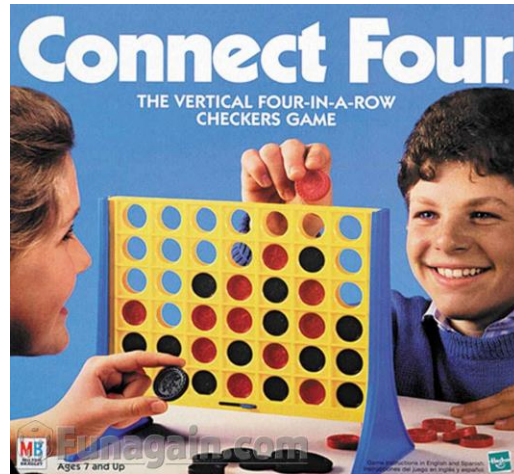
Vous devez programmer un jeu de connect4 selon le modèle MVC. Le GUI doit afficher l'état de la planche de jeu et le joueur à qui c'est le tour (Ex. : joueur 1 ou 2). La planche de jeu (qui est verticale) consiste en un certain nombre R de rangées et un certain nombre C de colonnes (par défaut, 6 rangées et 7 colonnes).

Pour jouer, un joueur choisi la colonne dans laquelle il souhaite placer un jeton. Le jeton tombe alors dans la colonne jusqu'à la rangée la plus basse possible (en bas complètement si personne n'a joué dans cette colonne, par-dessus le dernier jeton joué dans cette colonne sinon). Si toutes les rangées d'une colonne sont occupées, un joueur ne peut alors pas jouer dans cette colonne.

Un joueur gagne la partie lorsqu'il parvient à aligner V de ses jetons (verticalement, horizontalement ou en diagonale). Par défaut V=4. Si toutes les colonnes sont pleines sans conditions gagnantes, la partie est nulle.

En tout temps, un joueur peut choisir d'abandonner la partie (à l'aide d'un élément du menu) ce qui a pour effet de mettre fin à la partie pour ainsi faire gagner son adversaire.

Lorsqu'une partie est terminée, votre application doit indiquer le résultat à l'aide d'un popup. Ce popup doit aussi permettre de recommencer une nouvelle partie.



Src : funagain.com

L'apparence de la vue n'est pas importante, en autant qu'elle soit fonctionnelle.

CRITÈRES D'ÉVALUATION

Fonctionnalités : <ul style="list-style-type: none"> • Abandon • Rejouer • Colonnes pleines • Déterminer la victoire • ... 	5 pts
Qualité du code : <ul style="list-style-type: none"> • Nomenclature • Séparation des fonctions • Duplication de code • Utilisation judicieuse des structures de données • ... 	5 pts
Architecture du code : <ul style="list-style-type: none"> • MVC • Classes/interfaces • ... 	5 pts
Option Avancé : <ul style="list-style-type: none"> • Lancement votre application recevra trois paramètres R, C et W qui détermineront alors de nombre de rangées (R) de colonnes (C) et de jetons consécutif pour une victoire (V). Ne pas mettre trop de temps sur l'interface graphique ici non plus, c'est l'engin qui est important. 	2 pts
Gestion des exceptions : <ul style="list-style-type: none"> • Votre modèle devrait générer des exceptions • Votre contrôleur devrait gérer ces exceptions 	2 pts
Respect de l'énoncé	1 pt

Note : Pour être corrigée, votre application doit être minimalement fonctionnelle. Il faut au moins que les joueurs puissent jouer à tour de rôle.