Travail 1 (8%)

Programmation Objet 1

Modalités

Ce travail sera individuel. Les règles départementales habituelles s'appliquent concernant le français écrit, l'intégrité intellectuelle et les retards.

Travail demandé

Pour ce travail, vous devrez programmer un jeu de Tic-tac-toe. Le jeu se joue normalement dans une grille 3x3 où deux joueurs s'affrontent, les « X » et les « O ». Pour gagner, un des joueurs devra réussir à placer une ligne, une colonne ou une diagonale complète de « X » ou de « O ». Pour rendre le jeu plus intéressant, les joueurs veulent pouvoir jouer dans une grille 4x4. Puis, lorsque la partie sera terminée, les joueurs doivent pouvoir recommencer sans quitter le programme. Ensuite, il devra pouvoir être possible d'alterner entre la grille 3x3 et 4x4 en cachant les cellules superflues s'il y a lieu.

L'utilisation d'une ou de classe(s) est obligatoire. Vous devrez vous assurez que l'utilisation des attributs, méthodes, constructeurs et de l'accessibilité est cohérente avec ce que nous avons vu jusqu'à présent.

De plus, vous devrez utiliser des boucles pour déterminer le gagnant. Vous devrez utiliser les fonctions et les méthodes judicieusement pour éviter d'avoir de la répétition inutile de code.

La qualité du code sera aussi évaluée. Jusqu'à maintenant nous avons vu que l'importance de choisir de bons noms pour nos variables, fonctions, méthodes, attributs, etc. Il sera donc important de bien les choisir. Dans le même ordre d'idées, vous devrez choisir d'utiliser ou non les commentaires dans votre code et leur utilisation devra être juste.

Finalement, vous devrez faire utilisation de Git pour votre travail. Vous devrez utiliser judicieusement les « Commits » et faire usage des bonnes pratiques concernant ceux-ci.

Critères de correction

1. **Fonctionnalités** (30 pts)
    a. Pouvoir jouer dans une grille 3x3 (15 pts)
    b. Pouvoir recommencer une partie (5 pts)
    c. Pouvoir jouer dans une grille 4x4 (5 pts)
    d. Pouvoir redimensionner la grille en cachant les cellules superflu (5 pts)
2. **Utilisation correcte des classes** (25 pts)
    a. L'utilisation des attributs est cohérente (7.5 pts)
    b. La ou les classes ont des comportements cohérents (10 pts)
    c. L'utilisation de constructeur(s) (2.5 pts)
    d. Les attributs et méthodes ne sont pas plus accessibles que nécessaire (5 pts)
3. **Algorithmes efficaces et flexibles** (20 pts)
    a. Utilisation correcte des boucles (12.5 pts)
    b. Pas de répétition inutile (7.5 pts)
4. **Qualité du code** (15 pts)
    a. Noms significatifs (10 pts)
    b. Commentaires aux bons endroits, si applicable (5 pts)
5. **Utilisation de git** (10 pts)
    a. Avoir au moins un commit contenant tout le code (5 pts)
    b. Avoir une suite de commit cohérente (5 pts)

Modalités de remise (8 mars 2020)

La remise de votre travail s'effectuera dans un dossier .zip sur LÉA. *Ne pas oublier le dossier .git à la base de votre répertoire git. La date limite pour la remise sera le 8 mars 2020 à 23h59.