

Contrôle Interfaces Graphiques

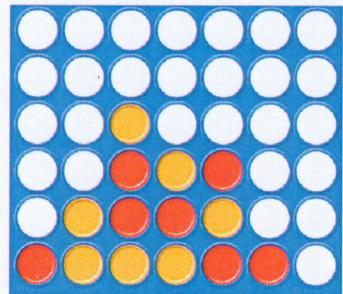
Date de l'épreuve : 4 juin de 9h à 10h30

Aucun document autorisé.

Responsable du sujet : M. Gautier

Exercice I : Puissance* (14 points)

Puissance4 est un jeu à deux joueurs qui se présente sous la forme d'un plateau dans lequel on peut insérer des pions par le haut uniquement ; un pion inséré en haut tombe (naturellement) vers le bas du plateau et repose sur le dernier pion placé dans cette colonne (s'il y en a un). Chaque joueur dispose de pions en nombre suffisant, d'une couleur différente pour chacun. Au départ, le plateau est vide. Chacun son tour, chaque joueur prend un de ses pions et le dépose dans la colonne de son choix (sous réserve qu'elle ne soit pas pleine, bien sûr!). Pour gagner, un joueur doit réussir à aligner des pions de la même couleur, sur une ligne verticale, horizontale ou diagonale.



Un début de partie de Puissance 4.

Dans cet exercice, on vise le développement d'une application graphique permettant de jouer à ce jeu. Plusieurs parties successives sont possibles ; chaque partie gagnée donne droit à un point.

Les tailles du plateau et de l'alignement à réaliser ainsi que les noms des deux joueurs et les couleurs des jetons sont fixées par défaut, mais peuvent être changées par l'utilisateur, au lancement ou entre deux parties.

Les scores sont tous enregistrés dans un (seul) fichier, chargé lors du lancement de l'application ; ce fichier mémorise un score par couple de joueurs ; il est sauvegardé à la demande ou lorsque l'application se termine. A tout moment, le score courant des deux joueurs est affiché (0-0 s'il s'agit d'un nouveau couple de joueurs).

Question a) Proposez une maquette de l'interface graphique de ce jeu. Complétez par une description textuelle correctement rédigée, permettant de comprendre les interactions avec l'utilisateur.

Question b) Sur la base de la maquette, dessinez l'arborescence des composants graphiques.

Question c) On décide de développer cette application avec JavaFX en mode procédural. Dessinez le diagramme de classes UML, incluant toutes les classes nécessaires à la réalisation de cette application. Hormis pour les classes de la bibliothèque, indiquez toutes les fonctions des classes.

Question d) On veut donner à l'utilisateur la possibilité de voir l'historique des couples de joueurs enregistrés, avec pour chacun d'eux le nom des joueurs et le score. Quels composants peut-on utiliser pour afficher ce nouvel élément dans l'interface ? Proposez deux solutions différentes et comparez-les.

Exercice II : JavaFX (6 points)

Utilisation des fonctionnalités JavaFX

Question a) Quels sont les deux modes de développement possibles en JavaFX ? Expliquez ces deux modes dans les grandes lignes en soulignant les différences. Le mélange des deux modes est-il possible ? Si oui, donnez un exemple.

Quels avantages/inconvénients voyez-vous dans l'utilisation de ces deux modes ?

Question b) Vous avez à étudier le diagramme de classes conçu par un norvégien pour une application graphique. Les noms de classe sont écrits en norvégien, langue que vous ne connaissez pas du tout. Quels sont les moyens que vous pouvez mettre en oeuvre pour savoir si le concepteur a appliqué le modèle d'architecture MVW ?

Le diagramme de classes du norvégien est le suivant :

```
graph TD; A[Classe A] --> B[Classe B]; A --> C[Classe C]; A --> D[Classe D]; B --> E[Classe E]; C --> F[Classe F]; D --> G[Classe G]; E --> H[Classe H]; F --> I[Classe I]; G --> J[Classe J]; H --> K[Classe K]; I --> L[Classe L]; J --> M[Classe M]; K --> N[Classe N]; L --> O[Classe O]; M --> P[Classe P]; N --> Q[Classe Q]; O --> R[Classe R]; P --> S[Classe S]; Q --> T[Classe T];
```

Les classes A, B, C et D sont toutes déclarées avec la même signature : `new A()`, `new B()`, `new C()` et `new D()`. Les classes E, F, G, H, I, J, K, L, M, N, O, P, Q, R et S sont toutes déclarées avec la même signature : `new E()`, `new F()`, `new G()`, `new H()`, `new I()`, `new J()`, `new K()`, `new L()`, `new M()`, `new N()`, `new O()`, `new P()`, `new Q()`, `new R()` et `new S()`.



Dans cet exercice, on vous présente une application JavaFX développée par un autre élève. Votre mission est de déterminer si l'application a été réalisée avec le modèle MVW. Pour cela, vous devez étudier le code source de l'application et identifier les classes qui sont déclarées avec la signature `new A()` (ou `new B()`, etc.). Ces classes sont généralement nommées avec un préfixe comme `Controller` ou `View`. Les autres classes sont généralement nommées avec un préfixe comme `Model` ou `Service`. Il est également possible que certaines classes soient déclarées avec une signature différente, mais qui correspond au modèle MVW.

Question a) Proposez une méthode de l'interface `Shape` qui permet de déterminer si une forme est une ellipse.

Question b) Si je passe de la méthode `lancer()` à la méthode `lancer2()`, quelles sont les modifications nécessaires ?

Question c) On décide de développer cette application avec JavaFX au lieu de Java Swing. Pour cela, nous devons utiliser des classes UI. Expliquez toutes les options nécessaires pour cette application.

Question d) On doit donner à l'utilisateur la possibilité de voir l'historique des coups de joueur. Un moyen simple, mais bon à faire, est de faire que si l'utilisateur fait un clic droit sur un bouton, il affiche un menu contextuel avec les fonctions de classe.