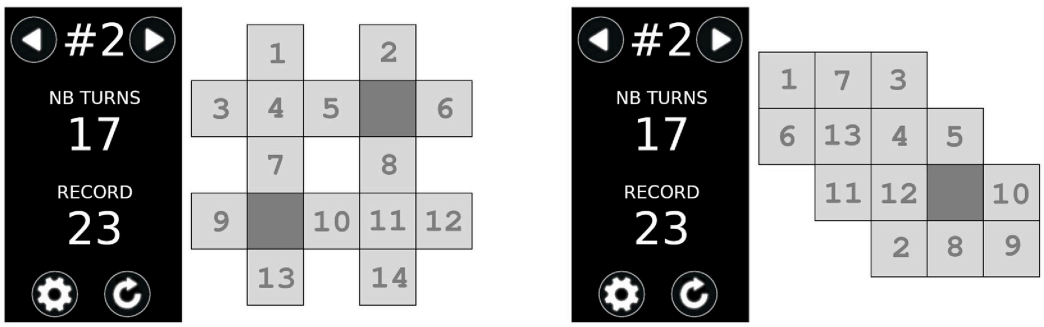
v1.1 Projet CY-SLIDE

CLASSE ING1-GI • 2022-2023
AUTEURS Eva ANSERMIN / Romuald GRIGNON
E-MAIL eva.ansermin@cyu.fr / romuald.grignon@cyu.fr

DESCRIPTION

Ce projet vous propose de coder une application pour jouer/résoudre automatiquement au je u de taquin.
Ce jeu demande de déplacer une par une des cases sur un plan orthogonal, pour les remettre dans l'ordre. Nous proposons ici de remplir chacune des cases avec un nombre, et e but est de les remettre dans l'ordre.
Les règles initiales du jeu de taquin impose une zone rectangulaire avec une case vide pour pouvoir faire "glisser" les pièces. Ici la zone sera libre, elle pourra avoir n'importe quelle forme, avec des bords lisses ou non, et des cases vides et immobiles au centre ou non.
Le projet devra proposer de charger les différentes zones à résoudre (un peu comme différents niveaux de difficulté. Il devra également proposer de résoudre automatiquement le niveau (si cela est possible).
Voici quelques exemples de rendu graphique que votre projet pourra proposer :



INFORMATIONS GENERALES

Taille de l'équipe
Ce projet est un travail d'équipe. Il est autorisé de se réunir en groupe de 4 ou 5 personnes.
Démarrage du projet
Vous obtiendrez de plus amples informations quant aux dates précises, les critères d'évaluation, le format du rapport associé à votre projet, les liens vers la personne chargée de vous suivre pendant ce projet, ..., quand le projet démarrera officiellement.
Jalons
En fonction de votre chargé de projet, et du sujet, vous aurez à fournir des éléments de progression à des dates spécifiques. Ces jalons dépendront du

sujet du projet, de l'organisation de l'équipe, ainsi que du planning de votre chargé de projet. Le but principal est de prouver que votre équipe progresse dans le livrable du projet, et que tous les membres sont impliqués.

Rapport du projet

Un rapport écrit est requis, contenant une brève description de l'équipe et du sujet. Il décrira les différents problèmes rencontrés, les solutions apportées et les résultats. L'idée principale est de montrer comment l'équipe s'est organisée, et quel était le flux de travail appliqué pour atteindre les objectifs du cahier des charges. Le rapport du projet peut être rédigé en français.

Démonstration

Le jour de la présentation de votre projet, vous utiliserez la même version que celle fournie à votre chargé de projet (version finale livrée) même si vous avez ajouté des fonctionnalités ou corrigé des bugs entre temps. L'idéal est de déposer votre application sur un dépôt de code d'un outil de gestion de version.

Vous ferez votre démonstration, en fonction des exigences du cahier des charges de votre projet, et vous aurez à modifier votre application suite aux requêtes du jury.

De plus des questions supplémentaires pourront être posées afin d'évaluer votre connaissance de l'implémentation de votre application.

FONCTIONNALITES PRINCIPALES DU PROJET

- Le but principal du projet est de fournir une application fonctionnelle.
- Votre projet complet **devrait (dans l'idéal)** être stocké sur un dépôt git (ou un outil similaire) pour au moins trois raisons : éviter de perdre du travail tout au long du développement de votre application, être capable de travailler en équipe efficacement, et partager vos progrès de développement facilement avec votre chargé de projet. De plus il est **recommandé** de mettre en place un environnement de travail en équipe en utilisant divers outils pour cela (slack, Trello, Discord, ...).
- Tous les éléments de votre code **seront** écrits en langue anglaise.
- Votre application ne doit **jamais** s'interrompre de manière intempestive (crash), quelle que soit la raison. Toutes les erreurs doivent être gérées correctement. Il est préférable de d'avoir une application stable avec moins de fonctionnalités qu'une application contenant toutes les exigences du cahier des charges mais qui plante trop souvent.
- Vous **devez** fournir une **documentation** de l'ensemble de votre code en utilisant des outils dédiés tels que Doxygen ou Javadoc. Vous modifierez votre code en fonction. Votre livrable pourra être réutilisé pour re-générer la documentation comme vous l'aurez fait pour la livraison.
- Votre projet devra être codé en langage JAVA.
- Votre projet doit implémenter toutes les fonctionnalités listées en début de ce document.

- Votre projet devra pouvoir être jouable, soit au clavier, soit à la souris.
- Votre projet devra être codé en langage JAVA.
- Votre projet devra pouvoir charger différents niveaux stockés dans des fichiers. A vous de définir le format exact de ces fichiers. Vous devez proposer au moins 10 niveaux avec des caractéristiques différents.
- Chaque niveau contient au moins 1 case vide mais vous pouvez ajouter autant de cases vides que vous le souhaitez suivant la configuration du niveau cela pourra simplifier la résolution.
- Chaque niveau peut contenir des cases inexistantes ou indestructibles

suivant comment vous appréhendez le concept. L'idée de ces cases, est de ne pas pouvoir être déplacées ou remplacées. C'est comme si ces cases n'existaient pas dans la zone de jeu. Suivant votre conception du modèle de données (exemple une matrice rectangulaire) vous pourrez avoir à ajouter ce type de case pour reproduire le motif de la zone de jeu (cf. exemples de niveaux en début de document).

- Votre projet devra pouvoir charger un niveau, afficher le niveau résolu en tout premier lieu, puis, mélanger le niveau.
- Le mélange sera fait de 2 manières, soit totalement aléatoire, soit en "déplaçant les cases aléatoirement. Dans le deuxième cas, on est certain que le niveau peut être résolu. Dans le premier cas, ce n'est pas forcément le cas. Ce mélange sera, soit un choix de l'utilisateur, soit un choix aléatoire du programme, dans ce dernier cas, une information doit apparaitre pour indiquer le type de mélange. Dans le cas du mélange par déplacement de cases, le programme devra en déplacer un nombre relativement grand pour garantir un bon mélange des cases.
- Dans tous les cas, au début d'un niveau après mélange, AUCUNE case ne devra être positionnée sur sa position de départ. Votre programme s'assurera de cet état. Si cet état n'est pas possible, votre programme affichera une information à ce sujet (un niveau possédant une case qui n'a qu'une seule case adjacente et qui ne possède qu'une seule case vide, est une topologie qui peut créer cette situation).
- Votre programme devra détecter si le niveau est jouable ou non (si le joueur peut arriver à le terminer) et afficher une information pour l'indiquer.
- Le joueur pourra déplacer les cases une à une, en respectant les règles du jeu de taquin (déplacement seulement si une case adjacente est disponible (vide)). A chaque déplacement, un compteur sera incrémenté pour savoir combien de coups le joueur a effectué.
- A la fin d'un niveau (résolution correcte), le nombre de coups de ce niveau sera sauvegardé dans un fichier pour être affiché comme record à chaque fois qu'un joueur tente le niveau.
- Vous devez mettre en oeuvre un système de déblocage de niveau quand le précédent est réussi. Par défaut seul le niveau #1 est accessible, les autres niveaux ne peuvent pas être tentés. Il faut réussir le niveau #1 pour pouvoir tenter le niveau #2, etc… .
- Les nivreaux accessibles ou non sont des informations stockées dans des fichiers avec le formalisme que vous souhaitez. Cea peut être une information dans les fichiers des niveaux eux-même ou des fichiers à part.
- Lorsque le joueur le décide, il peut demander à résoudre le niveau automatiquement : à ce moment là, si le niveau peut être résolu, le programme affichera chaque étape de la résolution. Une fois la zone de jeu complètement résolue, évidemment que le niveau ne sera pas comptabilisé comme débloqué puisque ce n'est pas le joueur qui l'a fait.
- Pour les étapes de la résolution automatique, vous pouvez faire défiler les étapes automatiquement, ou bien utiliser des boutons pour chaque étape, au choix. Si un défilement automatique est utilisé, la vitesse de défilement doit être réglée de manière à pouvoir visualiser correctement les étapes (ni trop rapide ni trop lent).

RESSOURCES UTILES

Github

- www.github.com
- https://docs.github.com/en/get-started/quickstart/hello-world

La sérialisation d'objets en JAVA

- https://docs.oracle.com/javase/8/docs/technotes/guides/serialization/index.html

Jeu de taquin : règles de base

- https://fr.wikipedia.org/wiki/Taquin

Algorithmes de recherche de chemins

- https://fr.wikipedia.org/wiki/Recherche_de_chemin