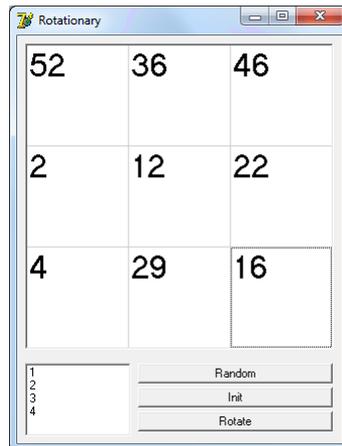


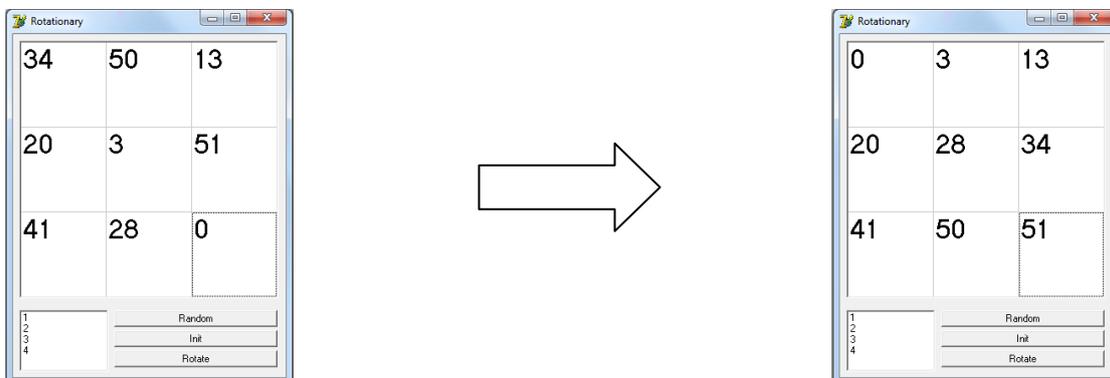
Devoir en classe d'Informatique II,2

70 minutes – 30 points

Vous venez d'inventer un jeu de stratégie appelé « Rotatory » et vous souhaitez programmer une application qui permet d'y jouer. Le jeu se déroule sur une grille de 3x3 cases et vous pensez que l'interface le mieux adapté est le suivant :



La grille du jeu est remplie aléatoirement avec des nombres et le but du jeu est d'appliquer une série de rotations pour mettre les nombres dans le bon ordre.



Remplir la grille (Bouton « Random »)

Les cases de la grille sont remplies aléatoirement, mais avec quelques restrictions :

1. Les cases sont remplies ligne par ligne, colonne par colonne. La première case à remplir est donc la case (0,0) (tout en haut à gauche), ensuite la case (1,0) (tout en haut au milieu), ensuite les cases (2,0), (0,1), (1,1) ... (2,2).
2. La première case (la case (0,0)) est remplie avec un nombre aléatoire entre 0 et 9. La deuxième case est remplie avec un nombre aléatoire, tel que la différence entre la valeur de la deuxième case et la valeur de la première case $\in [1, 10]$. Ce processus est appliqué à toutes les cases successives, c.-à-d. si c_i et c_{i+1} sont deux cases consécutives, alors :

$$(c_{i+1} - c_i) \in [1, 10]$$

Jouer un tour (Bouton « Rotate »)

Pour jouer un tour, le joueur peut choisir dans la listbox entre quatre rotations possibles en sélectionnant le numéro correspondant.

- La rotation « 1 » fait une rotation dans le sens des aiguilles d'une montre des quatre cases supérieur gauche
- La rotation « 2 » fait une rotation dans le sens des aiguilles d'une montre des quatre cases supérieur droite
- La rotation « 3 » fait une rotation dans le sens des aiguilles d'une montre des quatre cases inférieur gauche
- La rotation « 4 » fait une rotation dans le sens des aiguilles d'une montre des quatre cases inférieur droite

Initialiser le jeu (Bouton « Init »)

Les nombres générés par le bouton « Random » se trouvent déjà dans le bon ordre et forment une situation victorieuse pour le joueur. Le bouton « Init » applique 100 rotations choisies aléatoirement depuis les quatre rotations décrites ci-dessus et les applique à la grille. De cette manière, il est garanti que le jeu ainsi généré est résoluble par le joueur.

Fin du jeu (Bouton « Rotate »)

L'utilisateur a gagné lorsqu'il a remis les nombres dans le bon ordre, c.-à-d. lorsqu'il a récréé la situation de départ (la situation existante avant le clic sur le bouton « Init »).

Démarche :

1. L'application à réaliser devra comporter les éléments représentés sur la figure donnée. [6 p.]
2. La listbox doit contenir les nombres de 1 à 4 lors du lancement de l'application. [2 p.]
3. Le bouton « Random » aura pour effet d'initialiser le stringgrid de la manière expliquée dans la partie « remplir la grille ». [6 p.]
4. Ecrivez une procédure rotate qui prend en paramètre un nombre représentant une rotation et qui applique la rotation respective au stringgrid. [6 p.]
5. Le bouton « Init » aura pour effet de générer une nouvelle partie intéressante en appliquant 100 rotations aléatoires au stringgrid. Utilisez la procédure rotate ! [2 p.]
6. Le bouton « Rotate » aura pour effet d'appliquer la rotation sélectionnée par l'utilisateur au stringgrid et de vérifier si la partie a été résolue. L'application affiche un message d'erreur si le joueur n'a pas choisie de sélection. [8 p.]