

Évaluation_Groupe 6

Évaluation du groupe 6 (Almyre, Petureau, Chambres, Mahazoasy) réalisée par Rizki Nabila, Dedieu Thomas et Sabourin Jérémy.

Question 1: L'archive est-elle propre ?

L'archive est propre, elle contient tous les fichiers attendus.

Question 2: La compilation se fait-elle correctement ?

Aucun problème de compilation.

Question 3: Le programme se lance-t-il sans difficulté ?

Aucun problème de lancement. L'archive ne présente pas de fichier README.txt mais le lancement du jeu et la manière de jouer sont intuitifs.

Question 4: Le programme présente-t-il des bogues ?

Le jeu ne présente pas de bogues ni sur le déplacement des tuiles ni sur la fusion.

L'affichage du score est bien présent. Le programme gère également la situation de game over.

Question 5: La bibliothèque libgrid.a peut-elle remplacer votre propre implémentation de libgrid.a sans générer de problème ?

Non, lorsque nous utilisons la bibliothèque libgrid.a de ce groupe, nous obtenons l'erreur suivante:
/usr/lib/gcc/x86_64-linux-gnu/4.9/../../../../x86_64-linux-gnu/crt1.o: dans la fonction « _start »:
/build/glibc-Y1PEyp/glibc-2.19/csu/../sysdeps/x86_64/start.S:118: référence indéfinie vers « main »
collect2: erreur: ld a retourné 1 code d'état d'exécution.

Question 6: Le programme contient-il des fuites de mémoire ?

Il y a visiblement des fuites mémoires : le nombre d'allocations est beaucoup plus élevé que le nombre de free.

Résultat de : valgrind ./jeu

==34306== HEAP SUMMARY:

==34306== in use at exit: 65,030 bytes in 328 blocks

==34306== total heap usage: 336 allocs, 8 frees, 70,047 bytes allocated

==34306==

==34306== LEAK SUMMARY:

==34306== definitely lost: 16 bytes in 1 blocks

==34306== indirectly lost: 96 bytes in 5 blocks

==34306== possibly lost: 0 bytes in 0 blocks

==34306== still reachable: 64,918 bytes in 322 blocks

==34306== suppressed: 0 bytes in 0 blocks

Question 7: Le code est-il correctement indenté ?

Programme correctement indenté.

Question 8: Le nommage des variables et fonctions est-il cohérent ?

Le nommage semble relativement cohérent mise à part la fonction d'affichage dont le nom est en français alors que le reste du code est en anglais, ainsi que quelques variables dont le nom est également en français alors que la majorité des variables sont en anglais.

Question 9: Le code est-il correctement commenté ?

Oui, les commentaires présents permettent de comprendre le rôle de chaque fonction ainsi que leur fonctionnement.

Question 10: Le code contient-il des constantes non abstraites ?

Il n'y a pas de constantes non abstraites.

Question 11: Le code contient-il des parties dupliquées ?

Les seules fonctions où nous pourrions considérer que des parties sont dupliquées sont la `can_move` et la `do_move`, où le code se répète en fonction du déplacement. Seule la valeur d'initialisation des variables change. Cependant, dans ce cas, cela ne semble pas affecter la stabilité du code.

Question 12: Quelle proportion (en gros) du `grid.c` est testée (automatiquement) ?

L'archive contient un fichier `test-grid.c`, mais celui-ci n'étant accompagné d'aucune documentation, nous ne savons pas s'il s'agit d'un fichier permettant de tester le `grid.c` ou autre chose.

Cependant, après lecture des commentaires, ce fichier semble présenter les tests réalisables sur environ 90% du code.

Nous ne comprenons pas si nous sommes censés faire les tests ou si ce fichier montre simplement comment le groupe a testé ses fonctions.

Question 13: Que se passe-t-il si on change uniquement la constante `GRID_SIDE` ?

Le programme s'adapte parfaitement aux changements de `GRID_SIDE`, nous sommes allés jusqu'à `GRID_SIDE=12`.

Conclusion: Le programme fourni par ce groupe répond aux attentes exigées, à savoir: jouer au 2048 dans le terminal.

Le code est propre et lisible et il semble stable.

Les commentaires sont d'une excellente qualité.

Le plus de cette version: la possibilité de jouer avec les flèches sur une seule grille.