

(R1.01) Le sujet de SAE1.02 - qui sera à compléter

Le 29 novembre 2023 – durée : 4 heures

Alain Casali alain.casali@univ-amu.fr

Aix Marseille Université

I.U.T. d'Aix en Provence - Département Informatique



Preamble

Documents autorisés : une feuille simple A4 manuscrite recto / verso.

Remarques :

- Lire attentivement chaque question du sujet. OSEF du barème ;

Le but de cet exercice est d'écrire un clone du célèbre jeu "Candy Crush". Dans cette version, vous n'allez pas manipuler des bonbons (*i.e.* des candies), mais des nombres.

Pour ce faire, il vous faudra :

- créer un projet,
- remplacer le main par celui-ci : <https://amubox.univ-amu.fr/s/XrA6e9p6wfLTpW>
- compléter le code ;

Prenez 5 minutes pour voir les déclarations des types et des constantes utilisées dans ce TP.

EXERCICE 1.

Question 1.1 (2 points) :

Le but de cet exercice est d'écrire le corps de la procédure `initMat ()` dont le profil est le suivant :

```
void initMat (CMatrice & mat, const size_t & nbLignes = 10,
              const size_t & nbColonnes = 10,
              const unsigned & nbMax= KPlusGrandNombreDansLaMatrice);
```

Cette fonction doit initialiser une matrice à `[nbLignes][nbColonnes]`, chaque case de la matrice étant un `entier_naturel` compris en 1 et `KPlusGrandNombreDansLaMatrice`.

Question 1.2 (2 points) :

Le but de cet exercice est d'écrire le corps de la procédure `afficheMatriceV0 ()` dont le profil est le suivant :

```
void afficheMatriceV0 (const CMatrice & Mat);
```

Cette procédure affiche toutes les cases de la matrice avec la même couleur pour les nombres identiques (sans les numéros de lignes ou colonnes en haut ou à gauche). Par exemple tous les 1 sont affichés en rouge, tous les 2 en bleu, ... (vous avez le choix dans les couleurs).

EXERCICE 2.

Normalement, vous devez pouvoir identifier certains groupes de (au moins) 3 chiffres identiques sur la même ligne. Ce sont ces groupes qu'il va falloir identifier.

Question 2.1 (4 points) :

Compléter le code de la fonction `detectionExplosionUneBombeHorizontale ()` selon les spécifications demandées dans les commentaires de cette fonction.

NB : ne vous préoccupez pas, pour le moment, de la valeur de retour de ce sous-programme.

EXERCICE 3.

Une fois un motif d'au moins 3 nombres identiques identifié, il va falloir supprimer ces nombres, et faire descendre tous les chiffres qui sont dans lignes d'indice plus petit. On affectera la valeur `KIgnorer` les cases qui ne sont plus utilisées.

Question 3.1 (4 points) :

Compléter le code de la fonction `explosionUneBombeHorizontale ()` selon les spécifications demandées dans les commentaires de cette fonction.

Question 3.2 (4 points) :

Ecrire le corps de la fonction `showMatrixV1` qui remplace les cases ayant pour valeur `KAIgnorer` par un fond différent de celui de votre terminal, tout en n'affichant pas le chiffre `KAIgnorer`.

EXERCICE 4.

Comme vous pouvez le deviner, lorsqu'on supprime des cases, cela peut avoir une influence sur d'autres groupes de chiffres identiques. C'est pourquoi la fonction `detectionExplosionUneBombeHorizontale ()` est un prédicat. Il renvoie `vrai` si au moins un groupe de 3 chiffres identiques est détecté dans la matrice, `faux` sinon. L'utilisation de ce prédicat vous est montré dans la fonction `ppalExo03 ()`.

Question 4.1 (4 points) :

Modifier le code de la fonction `ppalExo03 ()`, afin d'obtenir des explosions en cascade.

EXERCICE 5.

On souhaite demander à l'utilisateur quelle pièce du jeu il souhaite bouger.

Question 5.1 (4 points) :

Modifier l'affichage de la matrice en indiquant les numéros de ligne et de colonne à leur bon emplacement via la procédure `afficheMatriceV2 ()`.

Question 5.2 (4 points) :

Gérer les déplacements des joueurs via la fonction `faitUnMouvement ()`.

EXERCICE 6.

Pour la SAE 1.02, il va falloir, à minima :

1. finir ce TP - sans chat GPT (ou autre LLM) et l'aide de dev externes ;
2. faire exploser les bombes verticalement ;
3. permettre à l'utilisateur de modifier les touches via un fichier au format YAML, la taille de la grille, et le nombre maximum (cela fera l'objet d'un futur TP).
4. rendre une documentation technique via un site web avec le graphe des fonctions appelées et celui des fonctions appelantes (cela fera l'objet d'un futur TP).
5. fournir une documentation sommaire de ce que vous avez ajouté.
6. fournir un guide d'utilisateur.

En plus, vous pouvez :

1. faire des niveaux ;
2. faire un éditeur de niveaux ;
3. établir d'autres règles du jeu - mais il faut les définir à l'avance ;
4. utiliser minGL.
5. ...