SameGame

DUT INFORMATIQUE 2021 IUT FONTAINEBLEAU (77)

**Arthur DECORBEZ & Shana LEFEVRE**

**Groupe 4 et 3**

**Table des matières**

[​ I] Introduction 3](#__RefHeading___Toc210_2238476167)

[​ II] Description et fonctionnalités 3](#__RefHeading___Toc212_2238476167)

[​ III] Présentation de la structure du programme (avec diagramme de classe) 3](#__RefHeading___Toc214_2238476167)

[​ IV] Algorithme 3](#__RefHeading___Toc634_2238476167)

[1er algorithme : Détection des groupes au survole de la souris : 4](#__RefHeading___Toc88_503968669)

[2e algorithme : Détection des lignes vides 4](#__RefHeading___Toc100_503968669)

[​ V] Conclusion 4](#__RefHeading___Toc636_2238476167)

# I] Introduction

Pour ce deuxième semestre, nous avons comme projet tutoré le «Samegame» à coder en java. Le but de ce jeu est de vider la grilles des ses blocs. Il y a trois blocs possibles: bleu, rouge et vert. Pour éliminer les blocs, il faut qu’il y ait un groupe d’au minimum de deux mêmes blocs adjacents. Le système de points dépend du nombre de blocs par groupe. On utilisera la formule suivante:

**( n - 2 ) ²**

*Exemple: Groupe de 5 → ( 5 – 2 ) ² = 9 points*

Dès qu’un groupe est éliminer, les autres blocs se déplacent afin de supprimer les «trous».

# II] Description et fonctionnalités

La grg

# III] Présentation de la structure du programme (avec diagramme de classe)

# IV] Algorithme

## 1er algorithme : Détection des groupes au survole de la souris :

Nous utilisons la méthode Radar qui prend en paramètre un x et un y de type int qui correspondent respectivement à une ligne et à une colonne que le joueur aura survolés avec sa souris.

La méthode va commencer par vérifier si tabTerm[x][y] a la même valeur que les tabTerm[][] du haut, du bas, de la droite et de la gauche soit R, V ou B.

Si le tabTerm[][] découvert est du même type, alors à ses coordonnées , on y place un ‘T’ (true) au tableau bool[][]. Le tableau bool[][] permet de dire que tel bloc appartient ou non au groupe donc, si bool[][]==’F’ (false) alors le tabTerm[][] correspondant n’appartient pas au groupe contrairement à bool[][]=’T’ . Donc, pour lancer une vérification dans les 4 sens, il faut que bool[][]==’F’.

Après chaque appel de la méthode Radar, on réinitialise le bool[][]=’F’. Cependant si bool[][]=’.’, alors cela signifie que tabTerm[][] n’a plus de bloc correspondant (R, V, B), il prend donc la valeur ‘X’.

## 2e algorithme : Détection des lignes vides

# V] Conclusion

Arthur :

Shana :