Romain SAVARY
Antoine MARCHAL DOMBRAT

Projet Tutoré

2020-2021
IUT de Sénart-Fontainebleau

Sommaire

Introduction

Ce document est un rapport décrivant le projet tutoré réalisé durant le second semestre de la formation en DUT informatique. Il s'agit de la réalisation du jeu Same Game en java.
Le jeu se présente en 3 écrans. Le premier est le menu où l'utilisateur choisi s' il veut lancer la partie à partir d'une grille générée aléatoirement ou à partir d'un fichier. Le deuxième est l'écran de jeu. À l'écran est affiché une grille de 15 * 10 éléments. Le joueur peut éliminer un groupe en cliquant sur un élément. Un groupe est un ensemble d'au moins deux blocs, tous de même type et chacun adjacent à au moins un autre membre du groupe.
A chaque retrait d'un groupe, le joueur du score augmente. Cette augmentation n'est pas linéaire pour inciter le joueur à retirer les plus gros groupes possibles.
La partie se termine lorsqu'il ne reste plus aucun groupe.

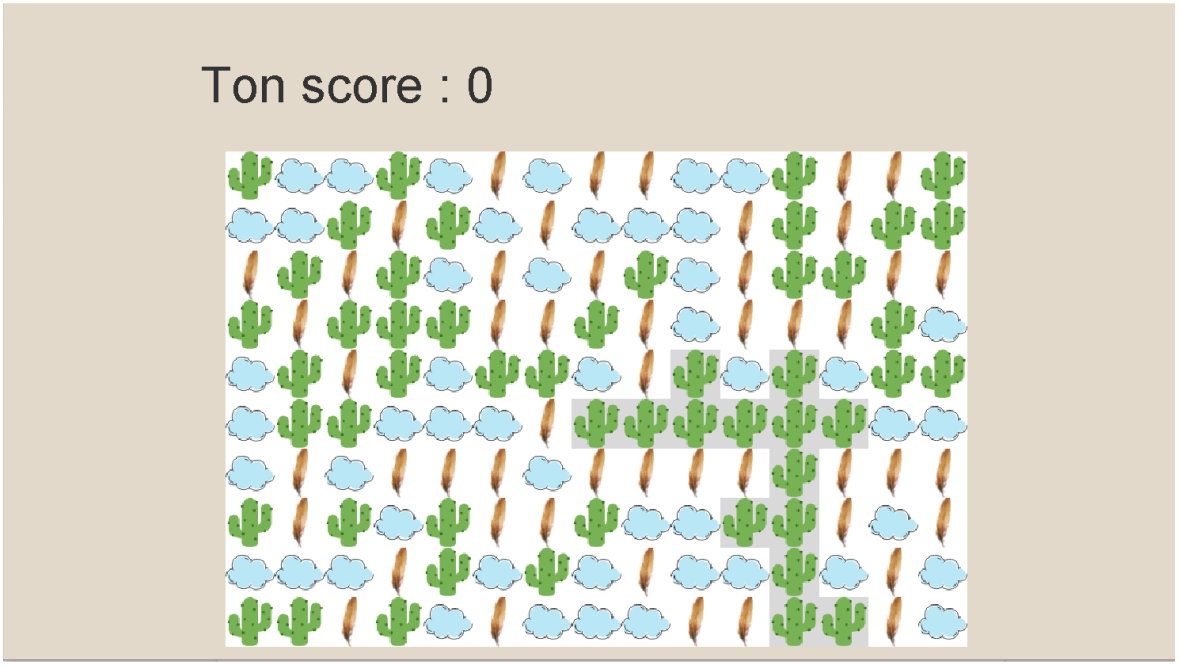
I. Les différentes fonctionnalités du programme.

1) Le Menu

Lorsque le programme est lancé, une fenêtre s'ouvre pour afficher le menu du jeu.
On y aperçoit le titre du jeu "Same Game", ainsi que 3 boutons nommés :
"Grille aléatoire", "Grille à partir d'un fichier", "Quitter".
L'utilisateur a alors le choix entre ces différentes options. Le premier bouton permet de lancer le jeu en générant la grille de manière aléatoire. Le second permet de créer une partie en générant la grille à partir d'un fichier que l'utilisateur pourra sélectionner grâce à une petite fenêtre. Enfin, le troisième bouton met fin au programme.

2) La partie

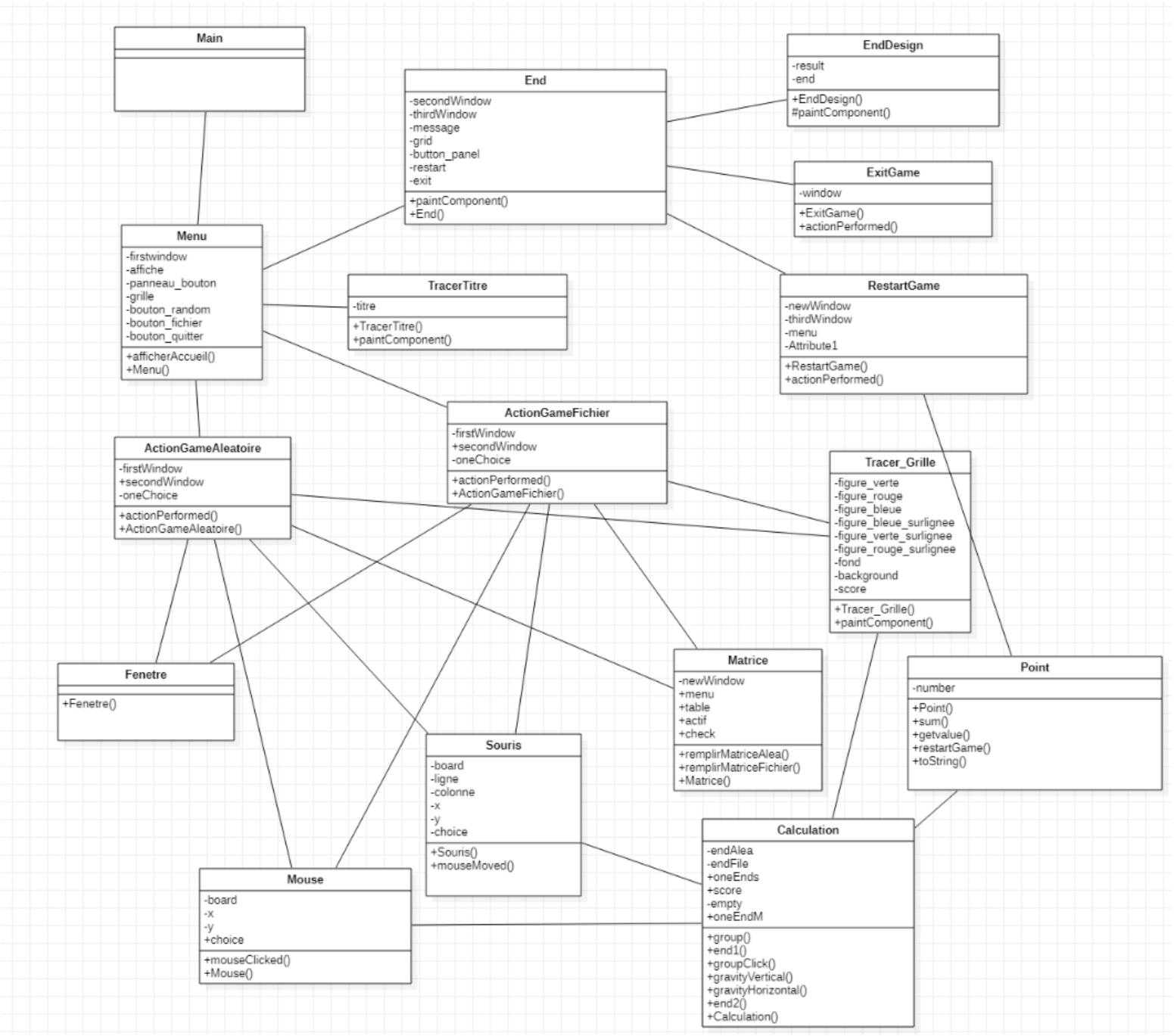
Lorsque la partie commence, la fenêtre du menu se ferme et une nouvelle fenêtre s'ouvre pour contenir le jeu. La grille est affichée au milieu de l'écran. Elle se compose de 3 éléments différents : un rouge, un vert et un bleu. En haut de la fenêtre le score actuel du joueur est affiché. La partie va se dérouler en fonction des actions effectuées par l'utilisateur à l'aide de la souris. En survolant les éléments de la grille avec la souris, le groupe contenant cet élément va être mis en évidence par un changement de couleur de fond. Lorsque l'utilisateur clique sur un élément de la grille, tout son groupe est retiré. Ensuite, tous les éléments sont attirés en bas, de telle sorte qu'il n'y ait plus de case vide sous un élément. Puis, si une colonne est vide, tous les éléments situés à sa droite sont déplacés vers la gauche pour venir combler cette colonne afin qu'elle se retrouve sur le côté droit de la grille. Enfin, le score du joueur augmente d'un montant équivalent à : ((nb d'éléments retirés)- 2)². La partie prend fin lorsqu'il n'y a plus de groupe au sein de la grille.



3) Le bilan de la partie

Une fois la partie finie, la fenêtre de la partie se ferme pour laisser place à une nouvelle qui accueille le bilan de la partie. L'utilisateur a alors deux choix qui s'offrent à lui : Il peut soit retourner au menu principal, soit quitter le programme.

II. Présentation de la structure du programme



III. Exposition de l'algorithme qui identifie les groupes

L’algorithme identifiant les groupes se situe dans la classe Calculation et c’est la méthode group. Cette méthode se divise en 3 étapes. La première permet de déterminer l’endroit et la couleur de l’élément survolé. La seconde de déterminer les voisins de cet élément survolé et la troisième de vérifier si l’on a un groupe ou un élément isolé.

Lors de ces étapes, on se sert de deux tableaux qui ressemblent à des grilles. Dans l’un y est rentré les couleurs des éléments (on l’appelle : table) et dans l’autre le statut des éléments (on l’appelle : actif), statut représenté par des chiffres, par défaut c’est 0.

Pour la première étape, une fois qu’on a obtenu les coordonnées de l’élément que le joueur survole, on transmet ces informations aux tableaux. Dans le tableau actif, à l’endroit de l’élément survolé on remplace le 0 par un 1. Dans le tableau table on regarde la couleur de l’élément survolé.

Ce qui nous permet de passer à l’étape 2. Lors de cette étape on fait une lecture du tableau actif que l’on répète 23 fois. Lors du premier passage, on cherche où se trouve l’élément survolé. Une fois trouvé on regarde si ses voisins sont de la même couleur que lui. Si oui alors on place un 1 à leurs endroits sinon on passe notre chemin. On répète la lecture plusieurs fois car si un ou des 1 se sont ajoutés dans le tableau actif alors il faut qu’on vérifie que celui ou ceux-là n’ont pas des voisins qui sont de la même couleur qu’eux et qui n'auraient pas été pris lors du premier passage. Pourquoi faire 23 lectures de tableaux, car c’est ce qu’il faut pour sélectionner une grille remplie d’une seule couleur en survolant l’élément tout en bas à droite.

Et viens ensuite l’étape 3. Celle-ci permet de s’assurer que le joueur survole un groupe et non un élément isolé. Pour le coup c’est une simple lecture de tableau. On se sert d’un compteur pour déterminer si oui ou non il y a plusieurs 1 dans le tableau actif. Si c’est un groupe alors le compteur sera supérieur à 1. Sinon ça veut dire que ce sera un élément isolé, on changera alors son 1 du tableau actif en 0. Cela permettra de ne pas afficher l’élément survolé comme un élément cliquable.

.

IV. Conclusions personnelles

La conclusion personnelle de Romain :

Ce projet était cool à faire. J'ai apprécié me casser la tête avec des boucles imbriquées pour déterminer un groupe. Bon… Ce n'était pas facile au début mais une fois que j'ai été lancé je n'ai pas pu m'arrêter. Ce projet m'a permis de mieux réfléchir en java, avant j'avais parfois des idées de comment faire mais pour les réaliser c'était à la manière du C. Ce qui va m'être pratique car ma sœur m'a demandé de lui créer un jeu et je compte bien réussir son défi (le faire en java permettra qu'elle l'ait sur l'ordinateur de famille plus facilement qu'en C). Enfin, c'est toujours agréable de faire des choses concrètes et pas juste des exercices, je peux présenter du contenu à mes proches quand ils me posent des questions à ce sujet.

La conclusion personnelle d'Antoine :

La réalisation de ce projet m'a permis de grandement m'améliorer en programmation java. En effet, j'ai pu développer ma compréhension de certains éléments tels que : les variables et méthodes statiques, ainsi que la séparation du code en plusieurs fichiers. En effet, c'était notre premier projet en java, jusque là nous n'avions eu que des exercices. De ce fait, nous avons dû nous efforcer de bien séparer le code en différentes classes au fur et à mesure de la réalisation du projet. Enfin, je dirais que je suis fier de l'aboutissement de ce projet, car nous avons réussi à partager la charge de travail et à faire face aux difficultés qui sont apparues.