

IUT SENART-FONTAINEBLEAU

RAPPORT DU PROJET SAMEGAME

Année 2017

02/05/2017

Table des matières

| | |
|---|---|
| I-Description du projet | 2 |
| II-Description des fonctionnalités du programme | 3 |
| 1/Le menu | 3 |
| 2/Déroulement d'une partie | 4 |
| | 5 |
| 3/Ecran de fin | 5 |
| III-Présentation de la structure interne du programme | 6 |
| IV-Algorithme de sélection des blocs | 6 |
| V-Conclusion personnelle..... | 7 |
| 1/Ranto RALIJAONA | 7 |
| 2/Jérôme GAUDIN | 7 |
| Annexe 1..... | 8 |

I-Description du projet

Dans le jeu SameGame, une grille composée de 15 colonnes et 10 lignes est remplie de blocs de trois types (de couleur rouge, vert et bleu) et le but du jeu est de la vider entièrement. Lorsque la souris survole un bloc qui fait partie d'un groupe, tout le groupe est visuellement accentué. Si le joueur clique sur un groupe, celui-ci est retiré de la grille. Tous les blocs chutent alors (de haut en bas) pour boucher les trous. Enfin, si une colonne a été entièrement vidée, les colonnes à sa droite sont décalées vers la gauche.

On augmente le score du joueur à chaque groupe éliminé. Plus le groupe compte de blocs, plus le gain est important. La progression n'est pas linéaire afin d'encourager le joueur à éliminer le plus de blocs possible d'un coup.

blocs gain

2 0

3 1

4 4

5 9

6 16

7 25

8 36

9 49

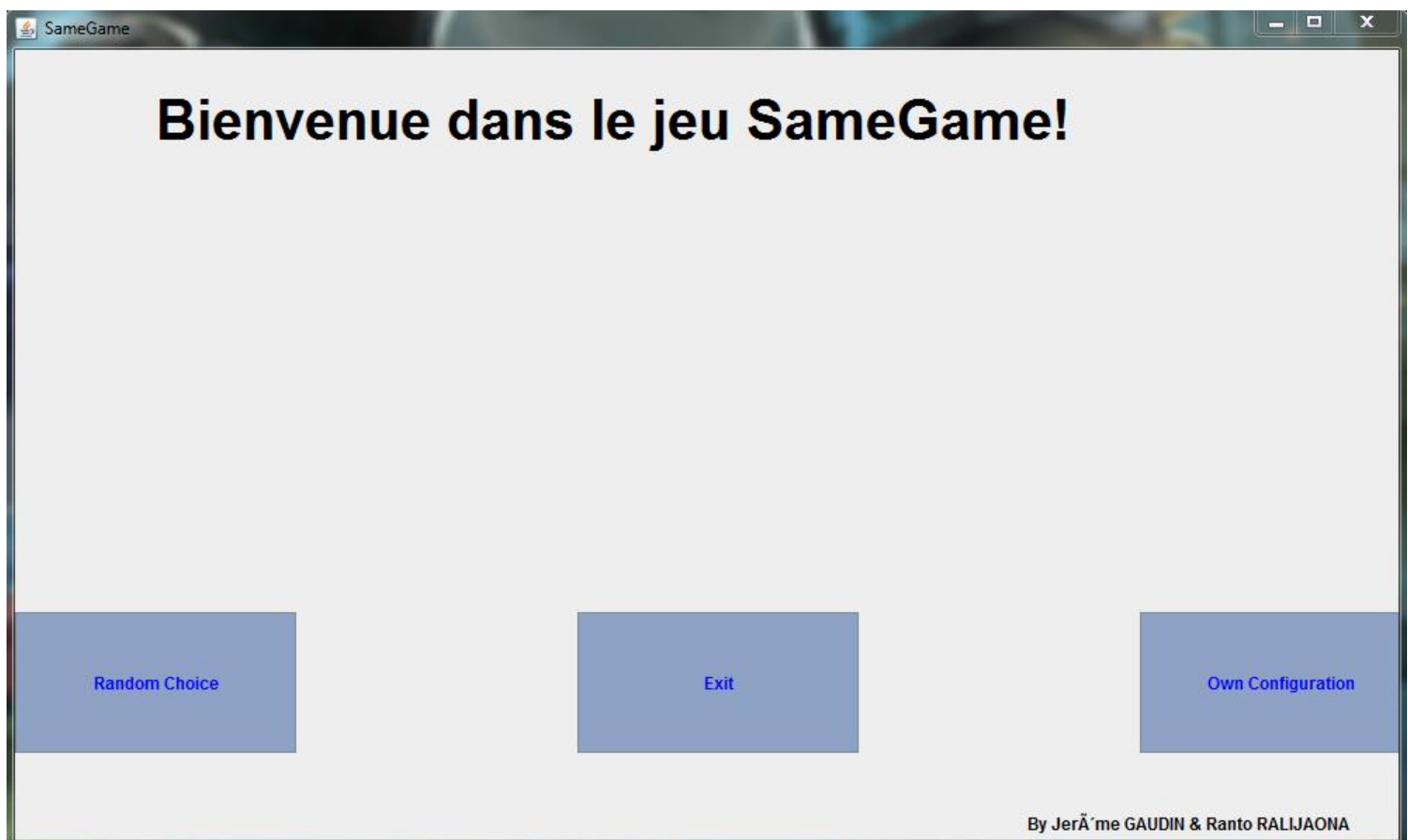
n $(n - 2)^2$

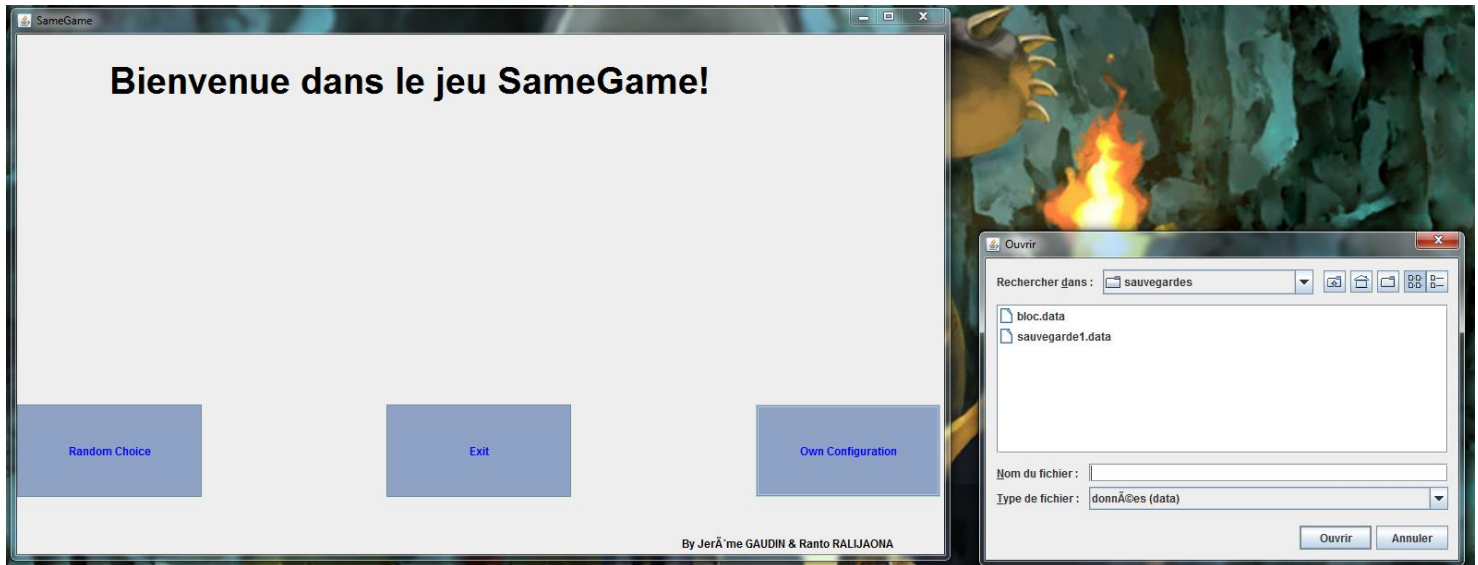
La partie se termine quand il n'y a plus aucun groupe (il peut rester des blocs isolés).

II-Description des fonctionnalités du programme

1/Le menu

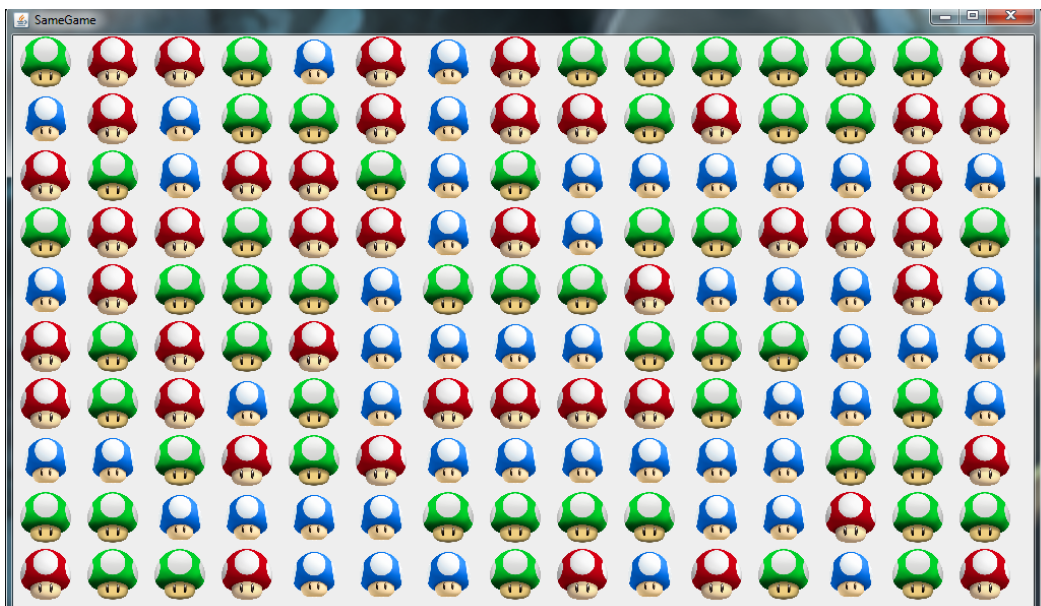
Le menu se présente comme un écran comportant le choix de configuration (donc à partir d'un fichier généré précédemment ou configuré aléatoirement), ainsi que le choix de quitter le jeu. Lorsque l'on choisit la configuration de la grille, le jeu se lance immédiatement et on arrive sur l'écran principal, celui du jeu.

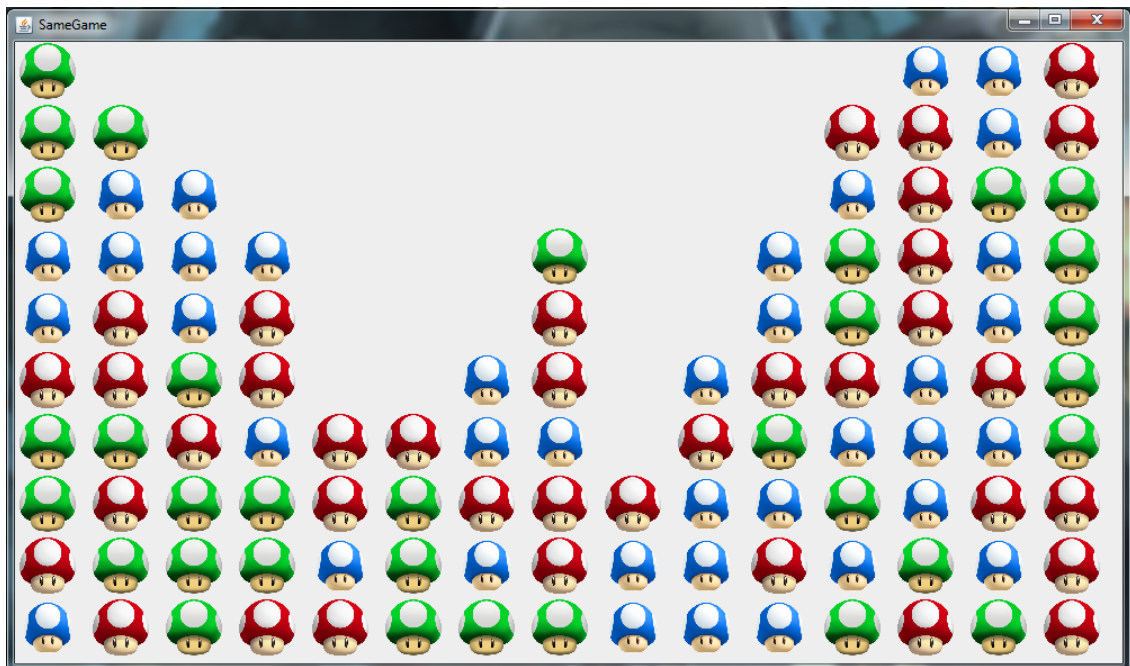




2/Déroulement d'une partie

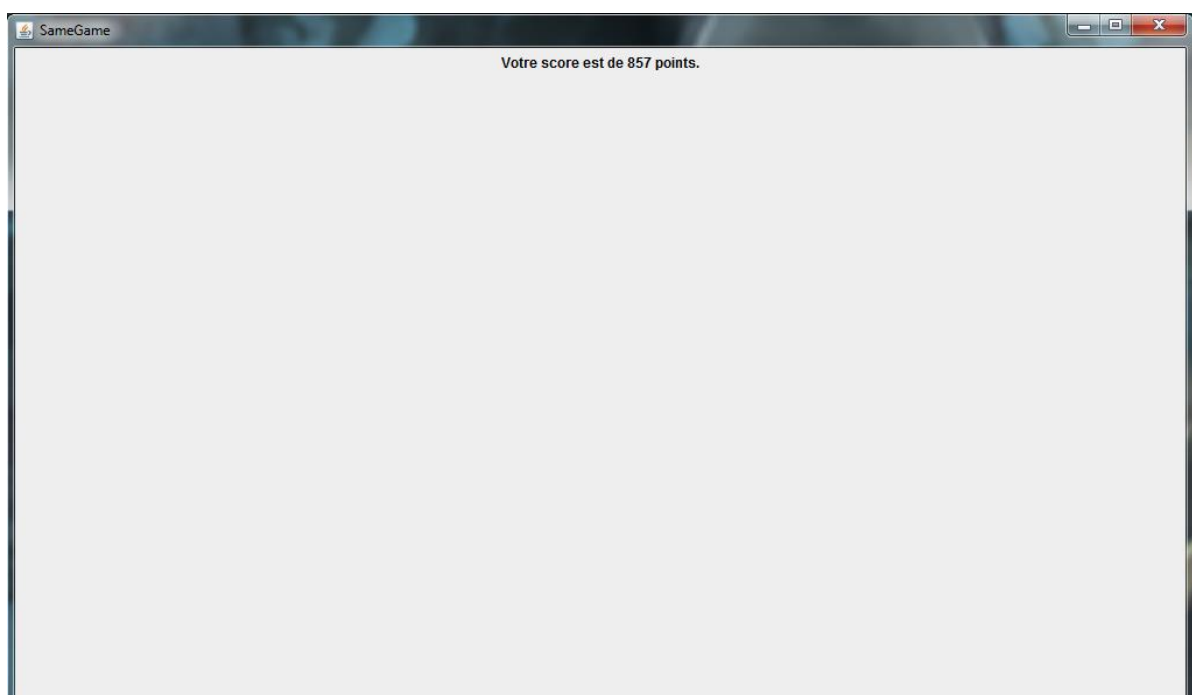
Comme indiqué dans la description du sujet, le joueur doit vider entièrement la grille pour gagner. Il sélectionne un groupe de blocs avec la souris pour le faire tomber et ainsi, les blocs restants descendent automatiquement pour créer de nouveaux groupes de blocs. Le joueur doit être stratégique dans le choix des groupes, il doit en même temps anticiper le mouvement des blocs qui descendent pour ne pas laisser des blocs isolés. Quand la grille est vidée ou qu'il ne reste que des blocs isolés, la partie se termine et le joueur peut voir son score.





3/Ecran de fin

L'écran de fin indique si le joueur a gagné ou a perdu. Il peut soit revenir à l'écran du menu, soit quitter le jeu.



III-Présentation de la structure interne du programme

Pour mener à bien le projet, 3 fichiers principaux étaient nécessaires pour représenter le menu, le jeu et l'écran de fin, ils sont tous les trois des enfants de la classe Ecran. Après, on a généré d'autres fichiers sources annexes pour nous aider à peaufiner le jeu (création de classes pour que le menu réagisse quand on clique sur un bouton, pour le déplacement des blocs, etc)).

Voir diagramme en annexe (page 8)

IV-Algorithmme de sélection des blocs

La sélection de blocs se fait avec deux fonctions : une fonction récursive et une fonction pour initialiser les paramètres de la réclusion, les deux fonctions ont le même nom : `estDansLeGroupe` (ce sont des surcharges).

Le principe est que : on a un tableau de boolean qui est initialisé à false, ce tableau permet de modéliser notre groupe. On sauvegarde la couleur du bloc cliqué et on met true aux coordonnées du bloc dans le tableau de boolean pour signifier que ce bloc appartient au groupe.

On regarde le bloc du dessus, à droite, à gauche et en bas. S'il est de la même couleur et qu'il n'est pas déjà dans le groupe alors on le signale dans le tableau et on recommence à vérifier dessus, à droite, à gauche et en bas si les blocs autour du nouveau bloc trouvé sont de la même couleur.

V-Conclusion personnelle

On a eu un problème il faut faire « make test » deux fois de suite pour que le menu s'affiche correctement.

1/Ranto RALIJAONA

Pour ce projet, j'étais chargé de la partie graphique du jeu ainsi que du menu dans son ensemble (boutons, gestion des événements avec l'ActionListener par exemple, les blocs pour le jeu). Comparé au premier projet en C (projet Blocus), celui-ci était plus conséquent et avait besoin de plus d'attention de notre part (il y avait beaucoup de fichiers sources, donc il fallait les gérer avec prudence. J'ai vraiment pu remarquer la différence avec le C pendant ce projet ainsi que les avantages de la programmation orienté objet.

Côté programmation, ce projet m'a permis de réutiliser plusieurs connaissances que j'ai apprises pendant mon apprentissage du Java. Il en résulte qu'il me reste beaucoup de choses à voir et à revoir en Java, ce qui m'encourage encore plus à persévérer dans l'apprentissage de l'informatique.

2/Jérôme GAUDIN

Le seul regret que j'ai est de m'y être pris un peu trop tard donc on n'a pas pu avoir l'aspect graphique voulu. Je me suis concentré sur la partie algorithmique et logique du projet, pour ma part je trouve que la programmation orientée objet permet de plus facilement travailler en groupe mais en contrepartie la coordination des fichiers entre eux est plus problématique.

J'ai trouvé ce projet intéressant à faire il m'a permis d'avoir plus d'aisance avec JAVA. J'ai pu apprendre à utiliser des nouvelles méthodes et classes. J'ai aussi appris à diviser le code en plusieurs objets et à coordonner les objets entre eux, ce que l'on fait peut en TP car on a au maximum deux trois classes en même temps.

Annexe 1

