

Rapport

Sommaire

- 1) Introduction
- 2) Description des fonctionnalités
- 3) Algorithme de selection
- 4) Diagramme de classe
- 5) Conclusion

Rapport

1) Introduction

Notre travail consiste en un jeu vidéo. Il est connu sous le nom de *SameGame*.

Le but du jeu est de détruire des regroupements de cases de même couleur (ici des bonbons) dans une grille.

Plus il y a de cases dans le regroupement détruit, plus le nombre de points gagnés est grand.

Une case seule n'est pas considérée comme un groupe.

Le jeu s'arrête quand il n'y a plus de groupe dans la grille.

Lors du lancement du jeu, le joueur se retrouve sur un menu. Ce menu lui permet de créer sa grille aléatoirement ou de la choisir dans un fichier pré-construit.

Quand le jeu est terminé, le joueur a la possibilité de retourner au menu pour rejouer si il le souhaite.

Rapport

2) Description des fonctionnalités

1) Le menu

Le menu est simple, il y a deux boutons (figure 1.1). Le premier permet de débiter une partie avec une grille aléatoire, le deuxième permet de débiter une partie avec une grille pré-existante que le joueur peut choisir via un sélecteur de fichiers (figure 1.2).

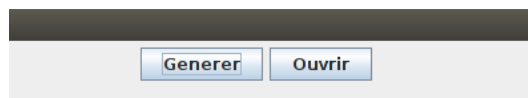


Figure 1.1

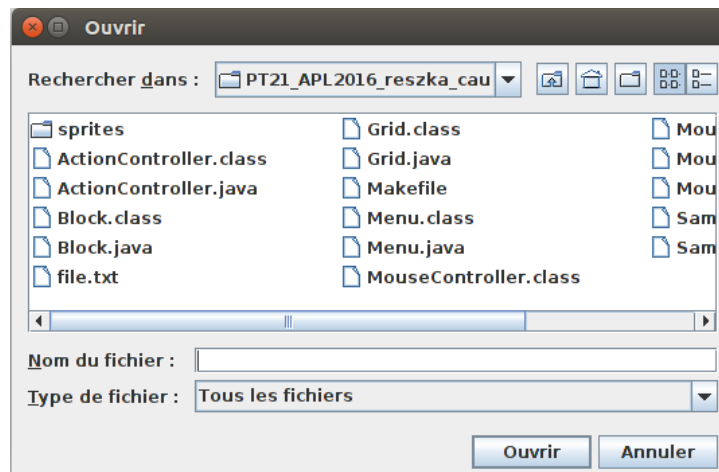


Figure 1.2

Rapport

2) La grille

Après avoir cliqué sur l'un des deux boutons, le joueur se retrouve sur une grille avec des bonbons de différentes couleurs. (figure 2.1)

Le but du jeu est de supprimer les groupes de couleurs identiques les uns après les autres.

Les groupements destructibles sont surlignés en jaune (figure 2.2) tandis que les autres sont surlignés en rose. (figure 2.3)

Pour détruire un groupe, le joueur doit cliquer dessus. Les espaces vides sont ensuite comblés de haut en bas et de droite à gauche par les bonbons qui se retrouvent isolés.

Le score est automatiquement mis à jour. (figure 2.4) Plus le groupe contient de bonbons plus il rapporte des points.

Le jeu est fini quand il n'y a plus de groupe à détruire. Le but étant de faire le meilleur score possible.

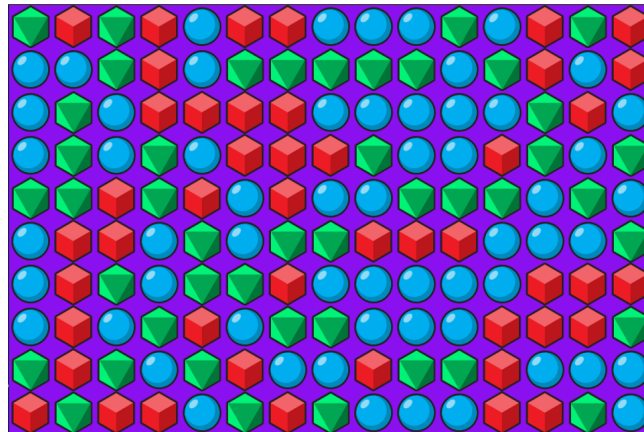


Figure 2.1

Rapport

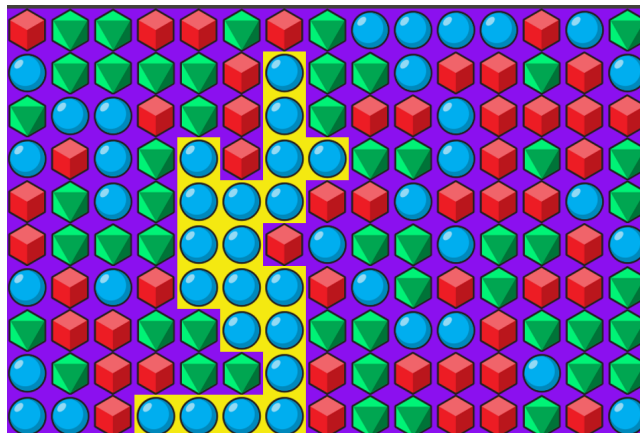


Figure 2.2

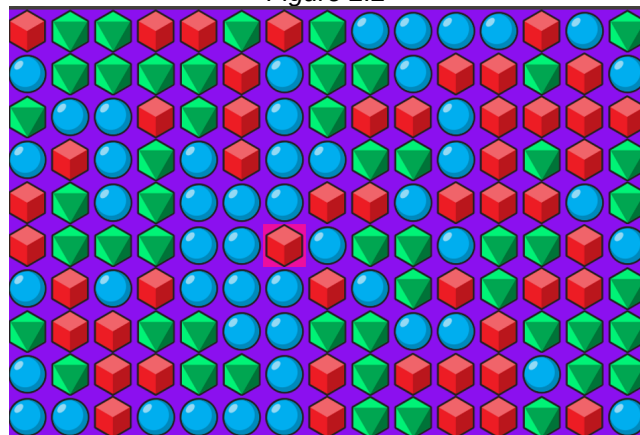


Figure 2.3

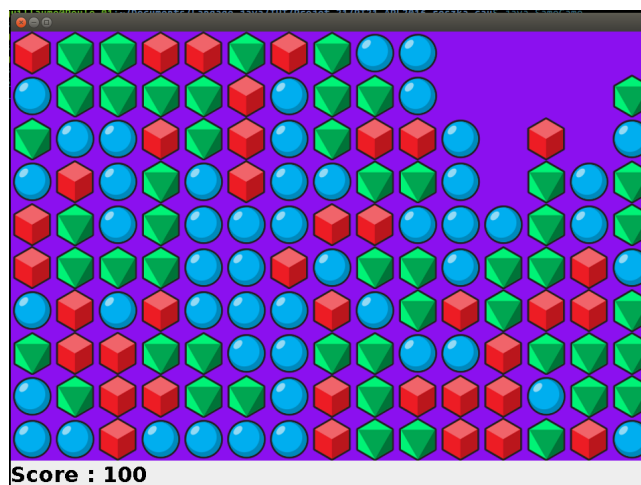


Figure 2.4

Rapport

3) Le Game Over

Lorsque il n'y a plus de groupe destructible, le menu du Game Over apparaît avec le score du joueur.

Il est possible de revenir au menu pour rejouer soit en cliquant sur le bouton du même nom soit en fermant la fenêtre. (figure 3.1)

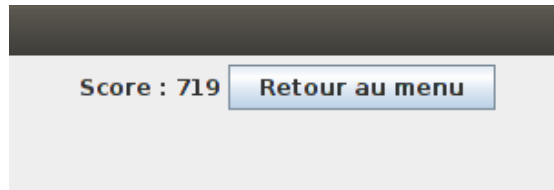


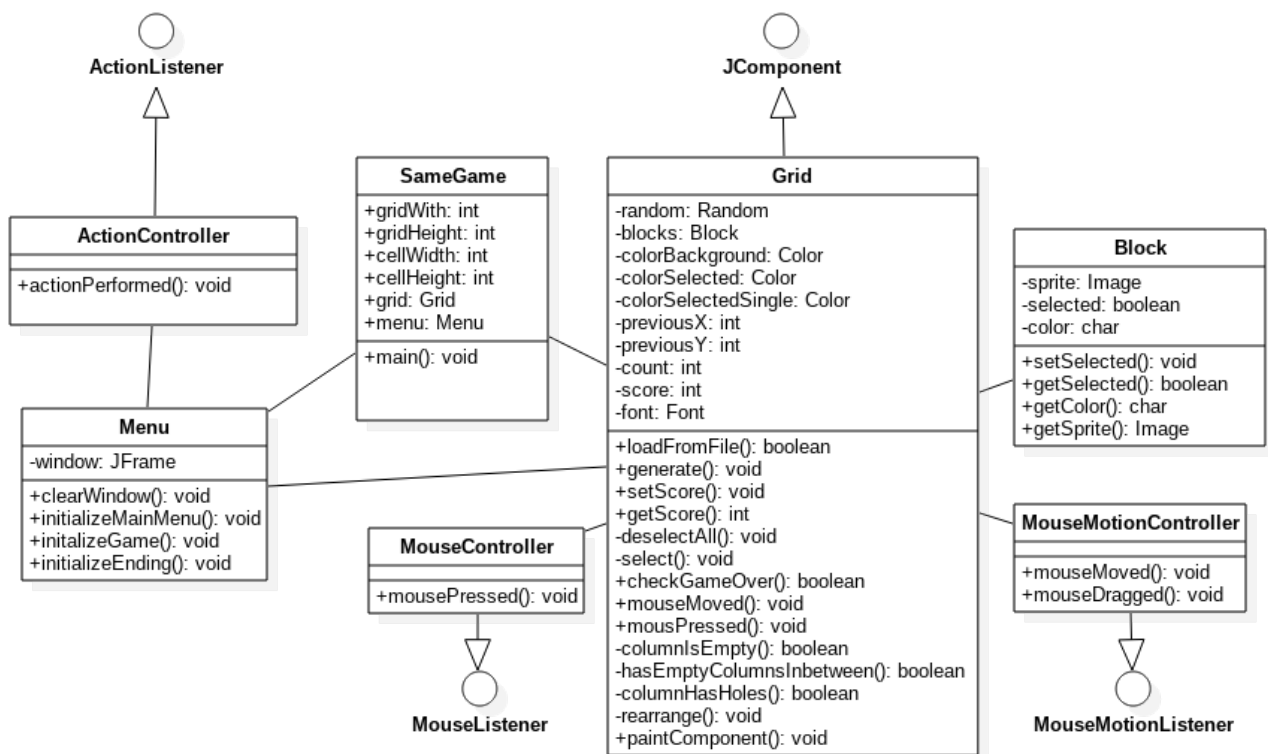
Figure 3.1

Rapport

3) Algorithme de selection

La fonction de sélection utilise un algorithme de remplissage par diffusion. Chaque case possède un flag « selected » qui est vrai lorsque la case est sélectionnée et faux dans le cas contraire. C'est une fonction récursive qui vérifie chaque case adjacente et la marque puis lance la fonction depuis cette même case.

4) Diagramme de classe



Rapport

5) Conclusions

Guillaume Cau : J'ai appris à faire beaucoup de chose lors de ce projet car mes connaissances en java étaient vague. Surtout avec les méthodes de la classe ActionListener. Mais désormais ces méthodes sont plus clair.

Je n'ai cependant pas vraiment amélioré mon travail d'équipe étant donné que j'reprenait là où Nicolas s'arrêtait et ainsi de suite.

Je devrais donc attendre le prochain projet pour essayer de mieux m'organiser.

Nicolas Reszka : J'ai trouvé que ce projet était intéressant à réaliser car il m'a permis de parcourir une très grande partie des notions du langage Java.