02 MAI 2017

FLAVIEN RIZOULIERES
THIBAULT HERVE

RAPPORT PROJET APL 2.1

SameGame

Table des matières

INTRODUCTION 3

LES FONCTIONNALITES DU PROGRAMME 3

LE MENU 3
L'AFFICHAGE DU MENU 3
LA GRILLE ALEATOIRE 4
LA GRILLE GENEREE PAR UN FICHIER 5
LES REGLES DU JEU 6
LA GRILLE 7
LE SCORE 7
SCORE EN COURS DE PARTIE 7
SCORE EN FIN DE PARTIE 7
LE SURVOL DES BLOCS 7
LA REORGANISATION DES BLOCS 8

LA DECOUPE DU PROGRAMME 9

DIAGRAMME DE CLASSE 9

L'ALGORITHME DE RECHERCHE DES GROUPES 10

CONCLUSION PERSONNELLE 11

CONCLUSION DE FLAVIEN 11
CONCLUSION DE THIBAULT 11

Introduction :

Le projet que nous avons dû réaliser pour le semestre 2 de notre première année de DUT Informatique, consiste à programmer en langage Java, un jeu nommé **SameGame**. Ce jeu est un divertissement qui se joue seul, il n'y a pas la possibilité de jouer à deux ou à plusieurs. Dans ce jeu, une grille est remplie de blocs de trois types différenciés par leurs formes et leurs couleurs, le but étant de vider cette grille. Un groupe de blocs adjacents ayant le même type peut être retiré en cliquant dessus. Les blocs restants se réarrangent afin de boucher les trous, et on recommence jusqu'à avoir vidé la grille ou n'avoir plus que des blocs isolés.
Lorsque le jeu est lancé, on a le choix de générer une grille aléatoirement, ou alors, on peut sélectionner un fichier sous un format précis rempli de caractères 'R' 'V' et 'B', ainsi la grille se créer en fonction du fichier sélectionné. Chaque type de bloc correspond à un caractère, c'est ce qui permet d'adapter correctement la grille.
Les règles du jeu sont simples, plus on retire de blocs en un seul coup, plus notre score augmente. Pour que le gameplay soit plus agréable, lors du survol d'une case, si celle-ci appartient à un groupe de bloc, celui-ci est mis en surbrillance.

Les fonctionnalités du programme :

Les principales fonctionnalités du programme sont contenues dans notre fichier *Grille.java*, nous allons donc vous les présenter les unes après les autres.

Le menu :

L'affichage du menu :

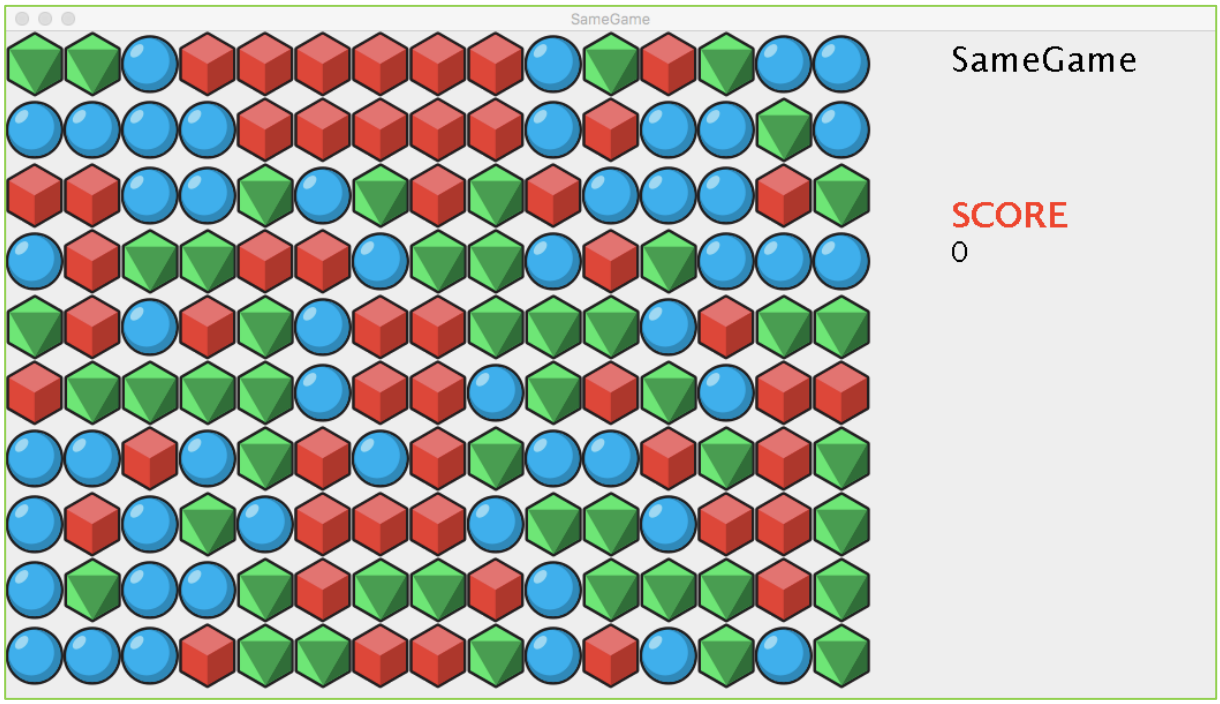
Le menu est obtenu dans le constructeur de notre objet *Grille*. Lorsque nous créons un nouvel objet grille, le menu est automatiquement généré. C'est grâce au constructeur de notre classe, que lorsque nous démarrons le jeu, la première chose qui apparaît est le menu et non la grille de jeu. Ce menu est composé de 3 boutons, ayant une fonctionnalité différente.

La grille aléatoire :

Comme nous l'avons vu précédemment le menu est composé de trois boutons et chacun apporte une fonctionnalité.

JOUER AVEC UNE GRILLE ALEATOIRE

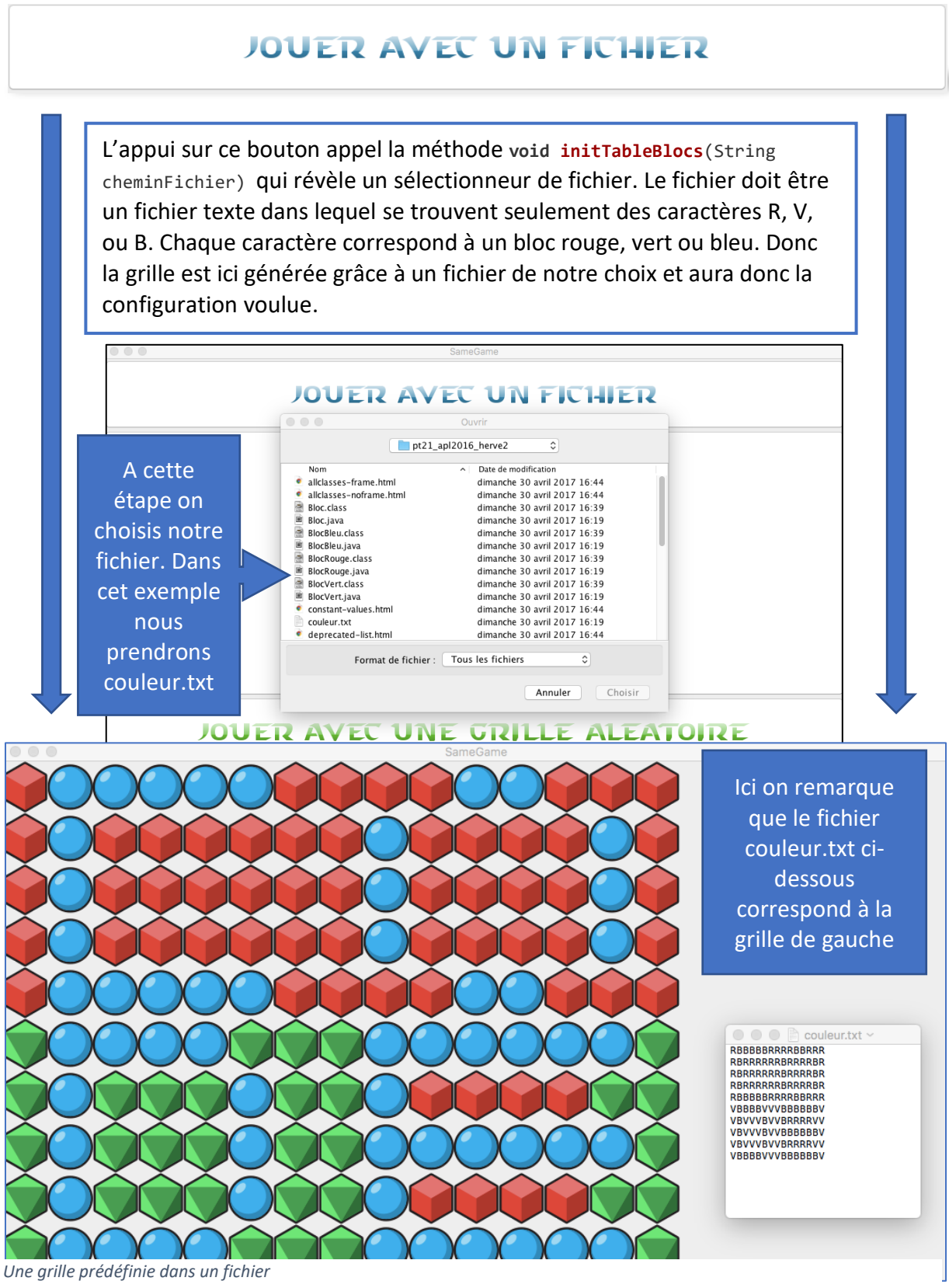
L'appui sur ce bouton, permet de générer une grille de jeu aléatoire. La méthode `void initTableBlocs()` permet de « typer » chaque bloc de la grille de façon à ce que cette grille soit complètement hasardeuse. Ensuite les 3 boutons du menu disparaissent pour laisser place à la grille.



Une grille complètement aléatoire

La grille générée par un fichier :

Le joueur peut aussi jouer avec une grille prédéfinie dans un fichier.

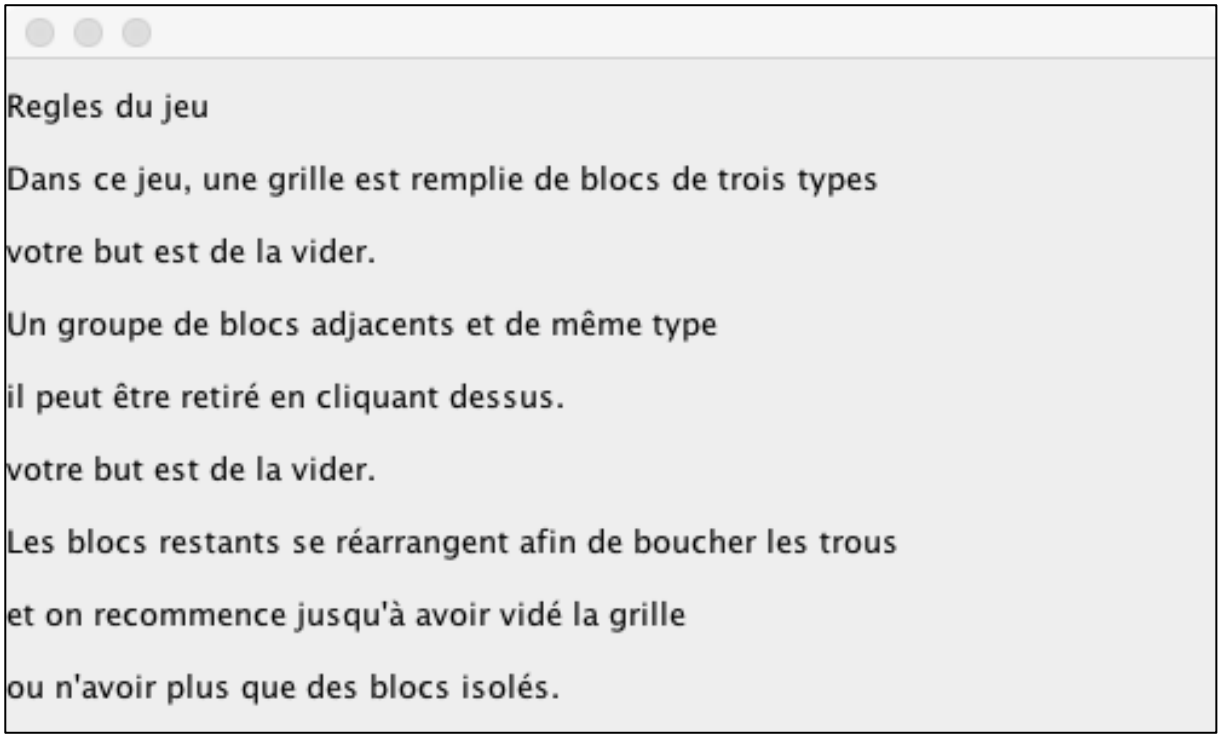


Une grille prédéfinie dans un fichier

Les règles du jeu :

Le dernier bouton de notre menu est attribué à une fonctionnalité que nous dirons « cachée », car rien ne l'indique.

L'appui sur ce bouton, ouvre une nouvelle fenêtre qui contient les règles du jeu.



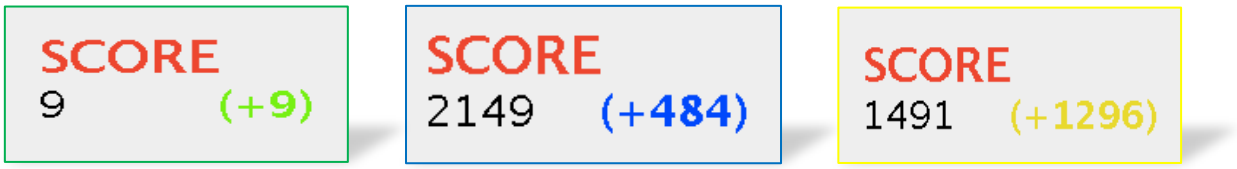
La grille :

Le score :

Comme vous avez pu le voir sur les précédentes images les différentes grilles contiennent un affichage du score. Cet affichage est différent en fonction du moment de la partie mais aussi en fonction du score lui-même. En cours de partie, sa couleur est différente si le score que nous réalisons en un coup est plus ou moins élevé. Si le score réalisé est inférieur à 250 il s'affiche en vert, s'il est supérieur ou égal à 250 il s'affiche en bleu et s'il est supérieur ou égal à 500 il s'affiche en jaune. Le score total, lui, ne change pas de couleur et reste toujours en noir.
A la fin de la partie, le score s'affiche en rouge au milieu gauche de la fenêtre.

Exemple :

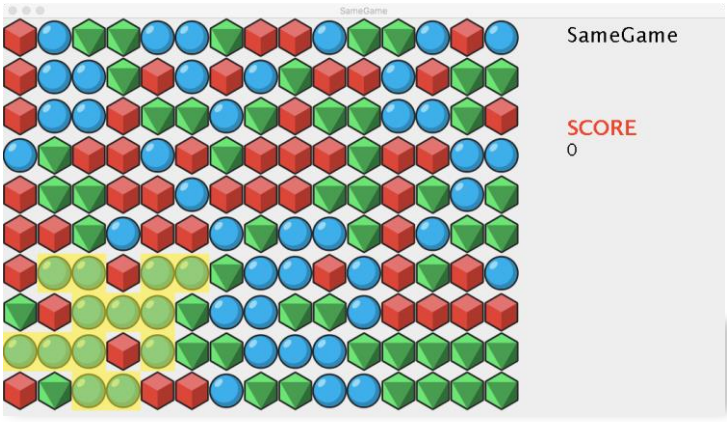
Score en cours de partie :



Score en fin de partie :

Le survol des blocs :

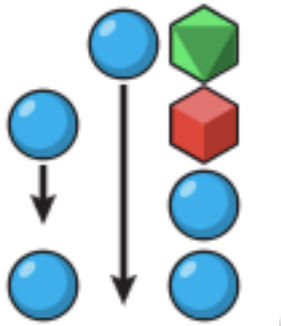
Lors d'une partie en cours, si l'on survole un bloc et que celui-ci appartient à un groupe de blocs, le groupe est mis en surbrillance pour que le joueur puisse voir clairement quels blocs vont disparaître ce qui aide le joueur à réaliser le meilleur score possible.



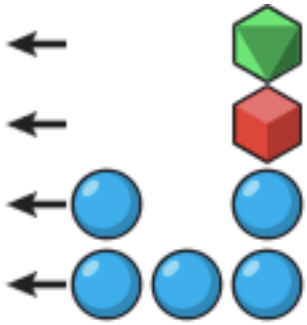
Ici la souris survol un des blocs bleus. Donc tous les blocs qui appartiennent au groupe sont mis en surbrillance.

La réorganisation des blocs :

Pendant une partie, l'utilisateur peut effacer des groupes de blocs, donc la disposition des blocs doit se réorganiser sinon il y aurait des trous dans la grille. Lorsqu'il y a un trou sous un bloc, celui-ci doit descendre, puis, si une colonne sur la gauche est vide, toutes les colonnes de droite doivent se déplacer vers la gauche pour combler la colonne vide. Le programme vérifie donc constamment ces informations pour réorganiser les blocs.



Action descente des blocs

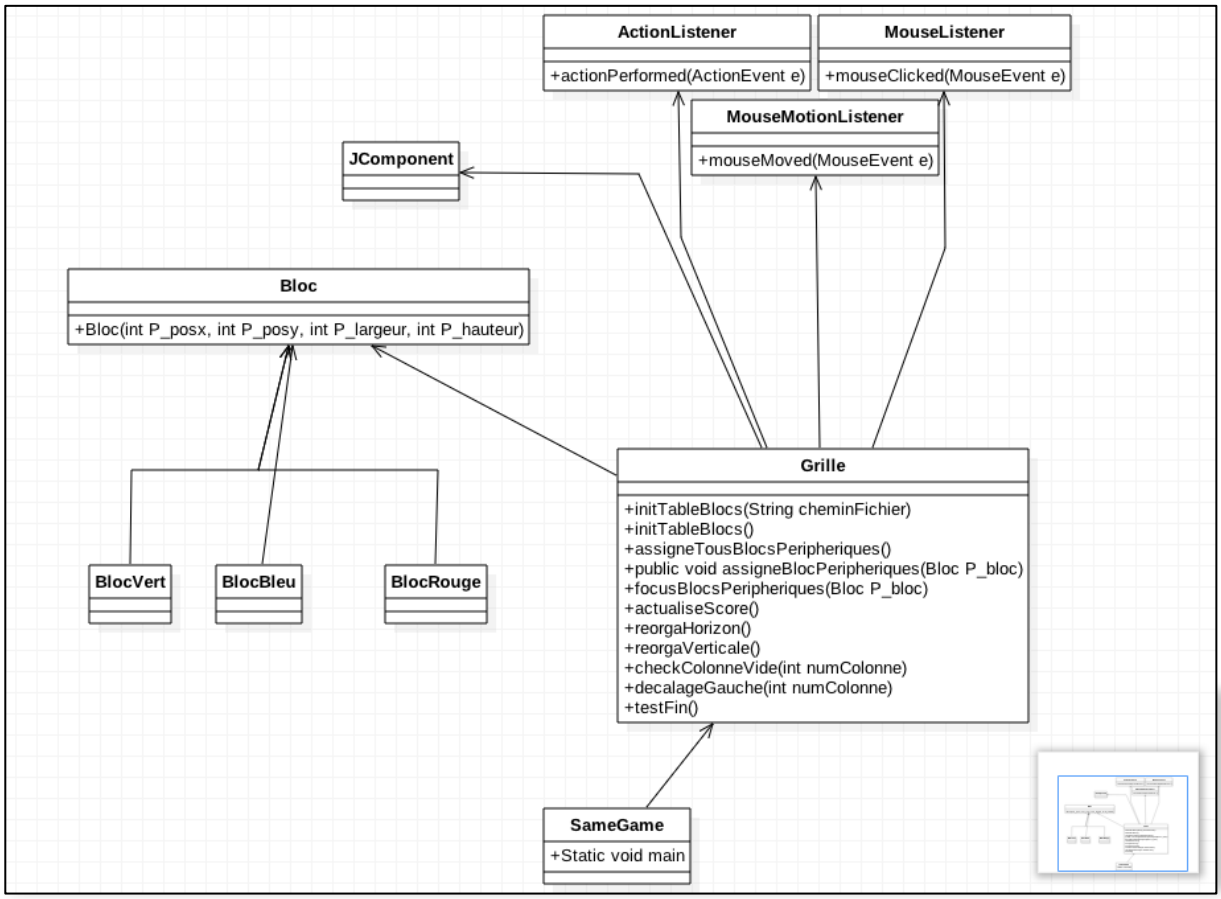


Action décalage colonne

La découpe du programme :

Nous allons maintenant vous expliquer comment nous avons décidé de découper notre programme. Voici ce découpage illustré à l'aide de ce diagramme de classes

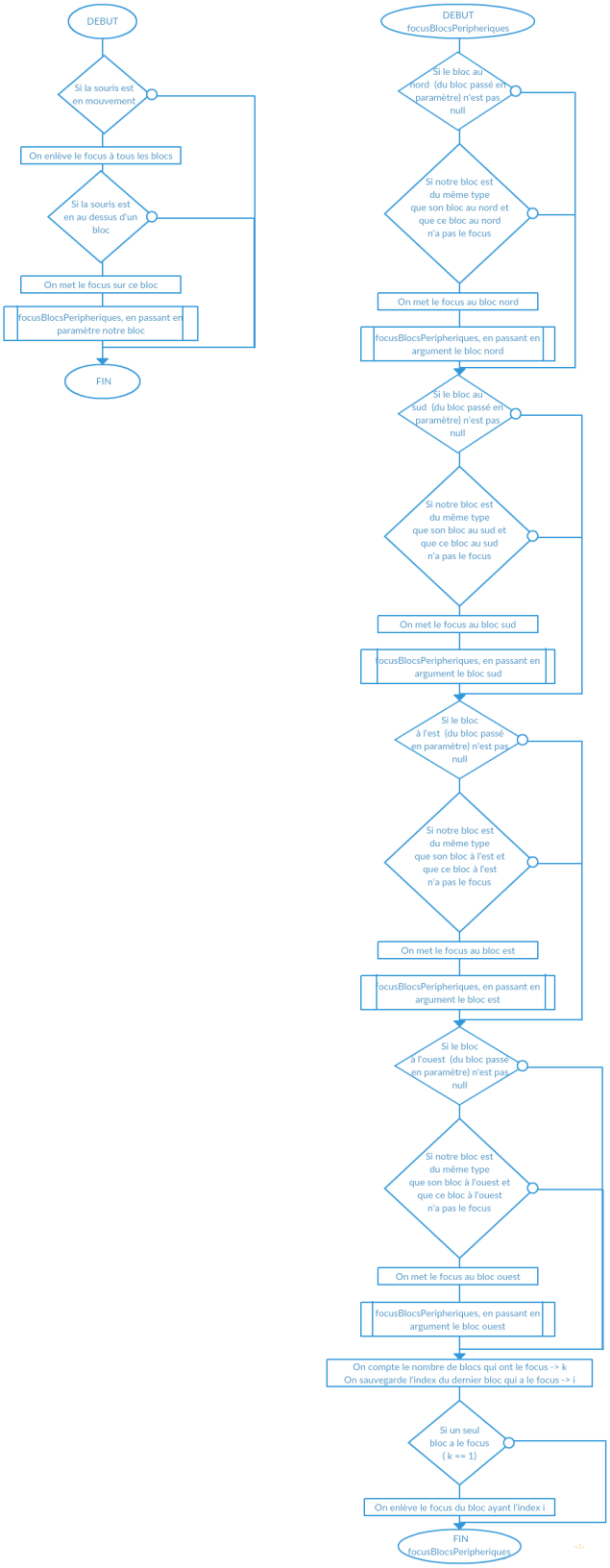
Diagramme de classes :



Notre diagramme de classes explique clairement la découpe de notre programme avec également, les méthodes que chaque classe contient. On remarque que nous avons décidé de faire une classe *Bloc* pour que tous nos blocs de différentes couleurs héritent de cette classe.

L'Algorithme de recherche des groupes :

Pour trouver les groupes, nous avons utilisé une fonction récursive. Voici l'algorithme de cette fonction :



Conclusion Personnelle :

Conclusion de Flavien :

Ce projet fut complètement différent du premier que nous avions eu à faire. Un nouveau projet, un nouveau langage de programmation. J'ai trouvé que ce projet fut plus simple à réaliser car nous avions déjà eu une expérience et nous avons donc mieux géré notre travail, notre temps, notre répartition du travail, ainsi que la façon de s'échanger les fichiers grâce a git ce qui nous a permis de mieux comprendre ce que l'autre avait amélioré sur le code source. Je trouve que ce projet ma ouvert les yeux sur ce qu'est réellement un projet contrairement au premier. Je pense que le fait de réaliser ce projet m'a permis de comprendre énormément de chose que je n'avais pas assimilée durant la deuxième partie de l'année ce qui aura été bénéfique pour moi. Ce projet est également une source de motivation pour moi car je trouve cela enrichissant et ludique de pouvoir réaliser des jeux que nous pouvons montrer à nos proches en leur disant que nous sommes les auteurs de ce jeu. Cette expérience m'a donc montré deux aspects totalement différents du monde informatique, l'expérience acquise en est donc d'autant plus grande.

Je peux donc affirmer que ce projet m'a beaucoup apporté dans le domaine et a été en fin de compte un travail agréable car le résultat est concret et très satisfaisant.

Conclusion de Thibault :

Lors de la réalisation de ce projet, j'ai d'abord remarqué que les méthodes de programmation étaient assez différentes du premier projet. J'ai d'abord compris que pour programmer en Java, il fallait penser différemment que lorsqu'on programme en C. J'ai par ailleurs été surpris par la puissance de Java, la possibilité de faire beaucoup de choses avec peu de lignes de code, avec l'utilisation des classes et des méthodes disponibles dans la bibliothèque. Ce projet a pour moi était plus rapide à réaliser que le premier, mais je ne l'ai pas trouvé plus simple pour autant. J'ai eu du mal à bien « diviser » le code sous forme de classes. La répartition des tâches n'a parfois pas été évidente avec mon camarade car nous devions toujours nous tenir informé de nos « avancées » et chacun était dépendant de l'autre. Malgré cela, l'utilisation de GIT nous a permis un meilleur échange du code, et donc de communication. Lors de mon premier projet, je n'étais pas en binôme. Le fait d'être deux m'a permis de voir qu'on n'avait pas tous la même façon de penser, mais surtout que chacun pouvait apporter des solutions aux problèmes de l'autre, et vice-versa.

Ce projet m'a permis de mieux comprendre la programmation objets et également d'améliorer ma capacité à me faire comprendre et à expliquer des choses aux autres.