



SAMEGAME



Table des matières

I} INTRODUCTION	1
II} DESCRIPTIONS DES FONCTIONNALITES	2
III} STRUCTURE DU PROGRAMME	6
IV} IDENTIFICATION DES GROUPES	7
V} CONCLUSION	8

06 AVRIL 2021
LENI BOSCHER
TANGUY DOMERGUE

I} INTRODUCTION

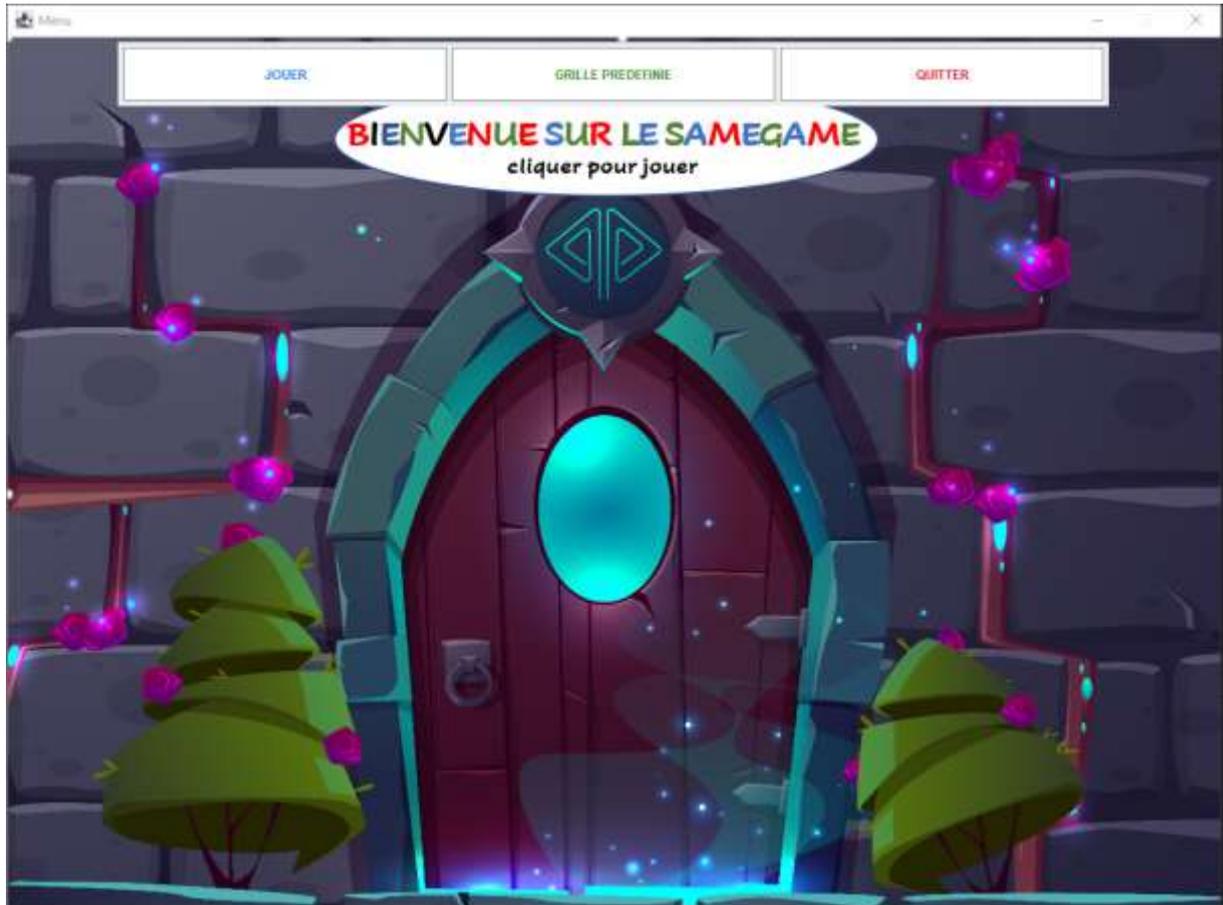
Le projet que nous avons réalisé se nomme le Samegame, il a été réalisé à l'aide de connaissances acquises en Java.

Les règles de ce jeu sont simples, à l'aide de votre réflexion vous allez devoir vider une grille de dimension 10*15, cette grille est composée de 3 types de blocs (rouge, bleu, vert), Deux blocs sont adjacents s'ils ont un côté en commun. 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. Votre score se verra augmenter en fonction de la taille des groupes que vous détruisez.

II} DESCRIPTIONS DES FONCTIONNALITES

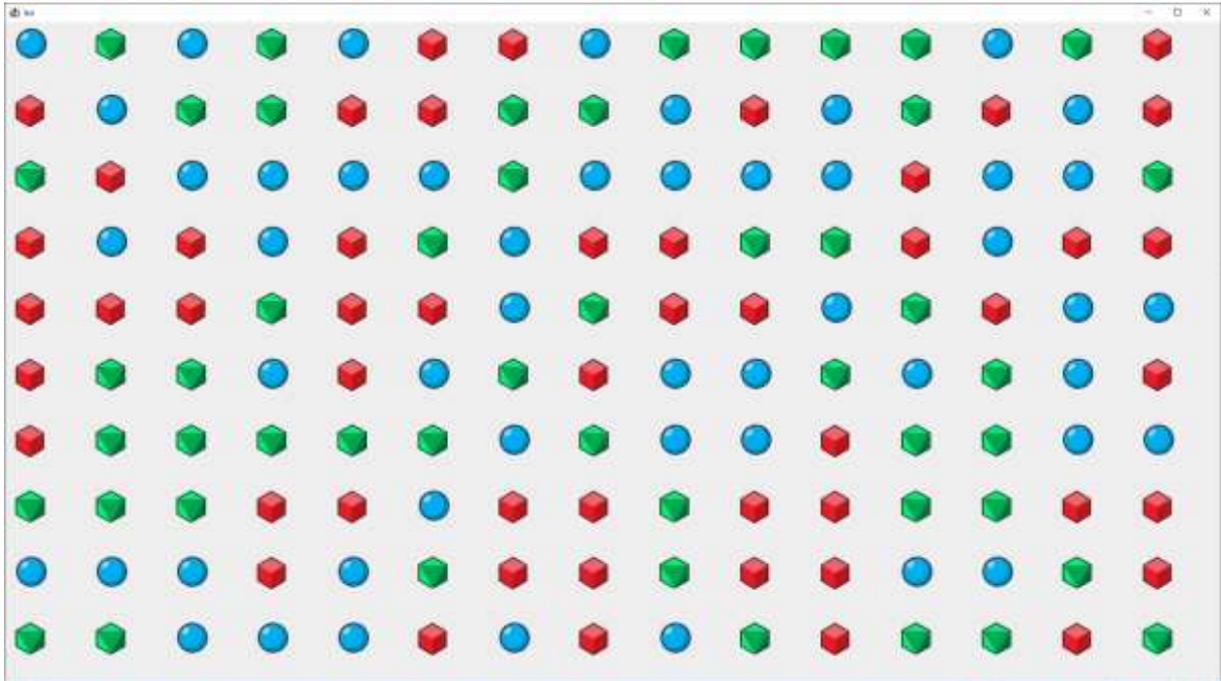
Cette partie est consacrée aux diverses fonctionnalités de notre programme.

Commençons par notre menu :

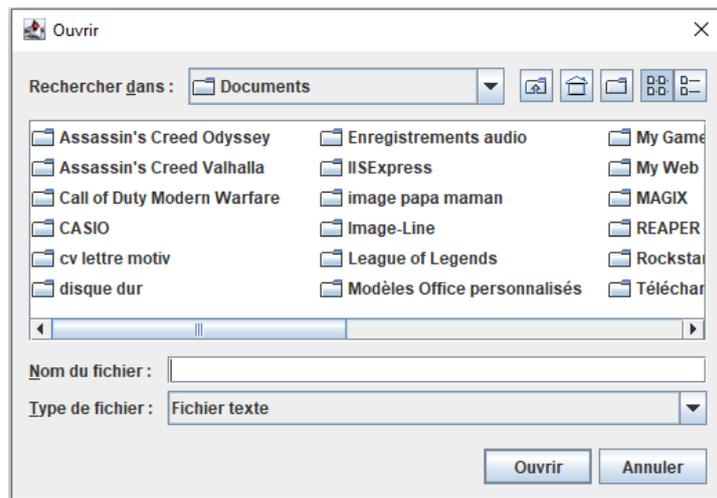


Notre menu est composé de deux panels, un qui contient le fond et l'autre qui contient les trois boutons qui vont nous permettre d'accéder aux autres fonctionnalités.

Le bouton **JOUER** nous permettra de générer une grille 10*15 configurée aléatoirement



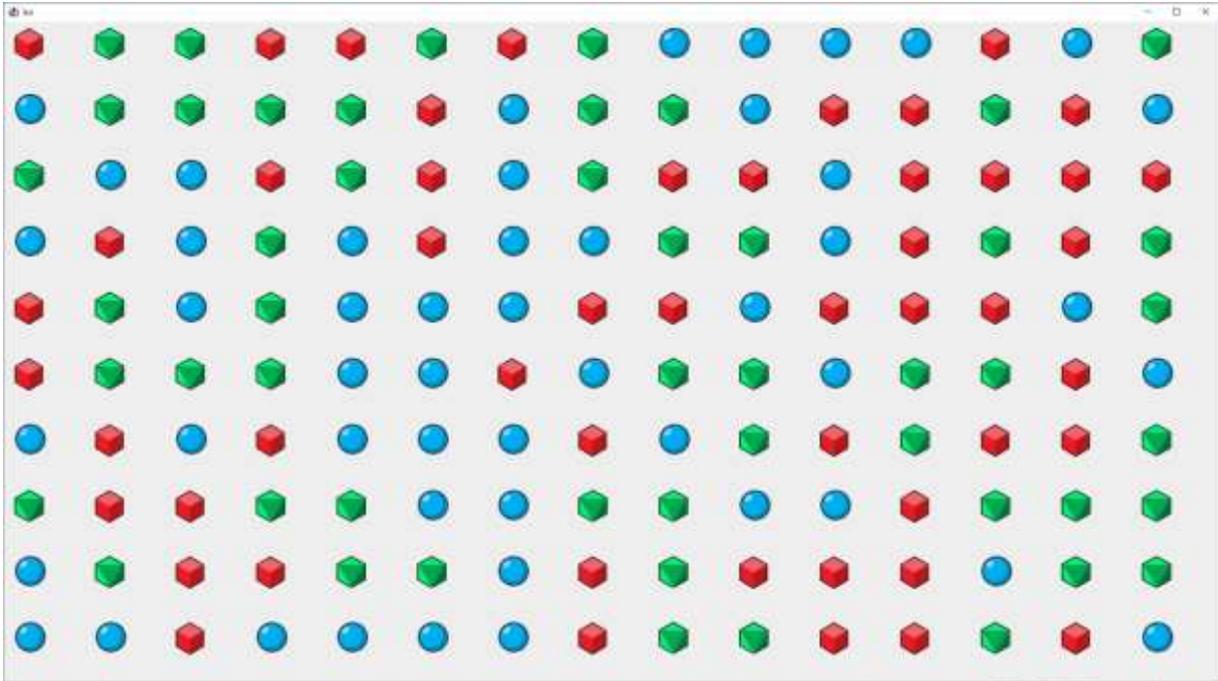
Le bouton **GRILLE PREDEFINIE** permet de générer une grille à partir d'un fichier, il va donc vous rediriger vers une page qui vous demandera de sélectionner un fichier :



Prenons pour exemple le fichier test.txt qui contient :

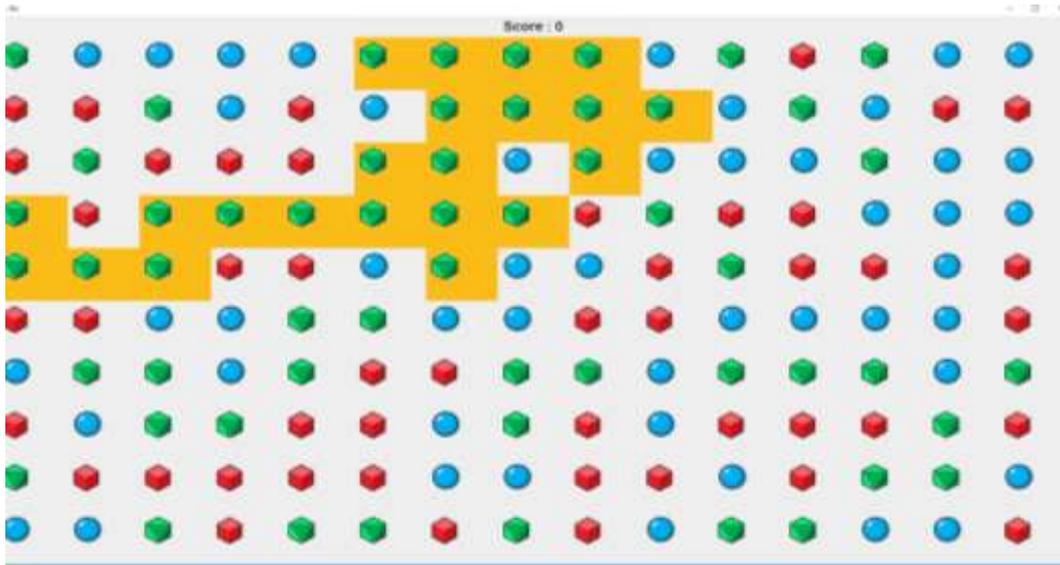
```
RVRRRVB BBBRBV
B V V V R B V B R R V R B
V B B R V R B V R R B R R R R
B R B V B R B B V V B R V R V
R V B V B B R R R B R R R B V
R V V B B R B V V B V R B
B R B R B B B R B V R R R V
V R R V B B V V B B R V V V
B V R R V V B R V R R R B V V
B B R B B B R V R R V R B
```

A l'aide d'un programme pouvant lire ce fichier, on va créer une grille en fonction des caractères qu'il va lire (R pour Rouge, V pour Vert, B pour Bleu), ce qui va donner la grille suivante :

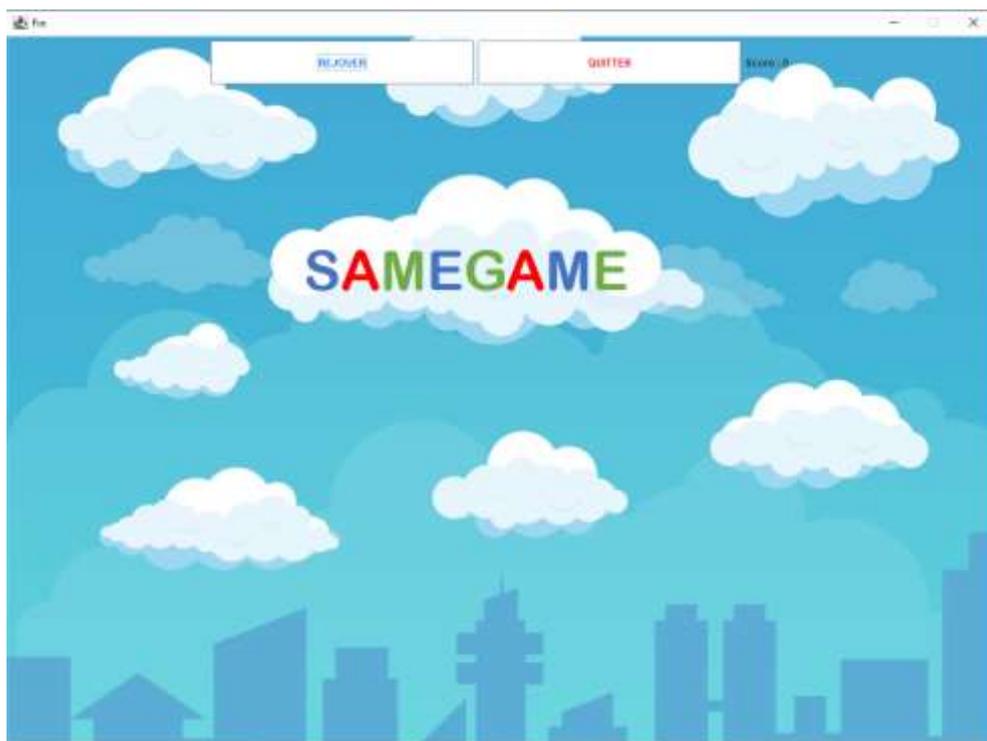


Le bouton **Quitter** ferme entièrement le programme.

Lorsque vous avez lancer une partie un programme vous permettra de voir plus facilement les groupes adjacents pour vous assurer un bon score



Une fois la partie terminée une nouvelle fenêtre va apparaître :



Votre score sera affiché en haut a droite et vous aurait à disposition deux bouton :

Le bouton **REJOUER** vous redirigera vers la fenêtre d'accueil.

Le bouton **QUITTER** stoppera le programme.

III} STRUCTURE DU PROGRAMME

Notre programme comporte :

3 fenêtres principales (FrameAccueil | FrameEnd | FrameJeu) vues précédemment.

Une fenêtre ChooseFramePredefini qui sera une fenêtre intermédiaire qui propose à l'utilisateur de rentrer un fichier qui sera ensuite lu dans la classe FrameJeu.

2 class (BackgroundPanelAccueil | BackgroundPanelEnd) qui héritent de JPanel permettant de décorer le menu d'accueil et le menu de fin de partie.

Une class Couleur qui hérite de JPanel qui a pour but de dessiner nos icones.

3 class Red, Blue, Green qui héritent de Couleur permettant de générer plusieurs icones différentes.

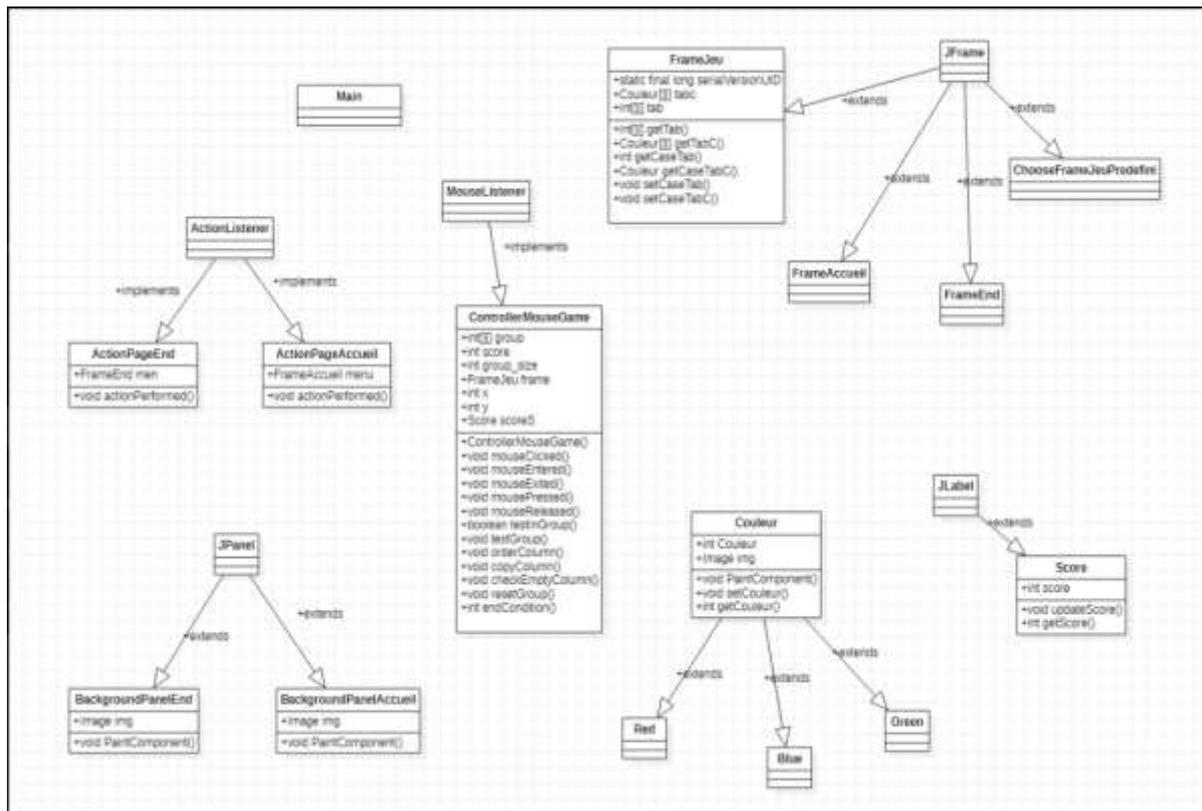
Une class score qui hérite d'un JLabel, qui affichera le score à la fin de la partie et permet de le mettre à jour.

La fonction main qui va être à l'origine du début du programme et qui va initialiser la première page d'accueil.

Une class ControllerMouseGame qui contient toutes les règles du jeu, gère les clics de l'utilisateur, l'aide à mieux visionner les différents groupes.

ActionPageAccueil et ActionPageEnd qui vont attribuer des évènements aux boutons du menu principal et au menu de fin de partie.

Diagramme de class :



IV} IDENTIFICATION DES GROUPES

```

public void testGroup(int x, int y)
{
    if(x>0)
    {
        if((this.frame.getCaseTab(x,y) != 0) && (this.frame.getCaseTab(x,y) == this.frame.getCaseTab(x-1,y)))
        {
            if((testInGroup(x-1,y)))
            {
                this.group_size++;
                this.group[x-1][y] = this.frame.getCaseTab(x-1,y);
                testGroup(x-1,y);
            }
        }
    }
}

```

Algorithme de sélection de groupe :

L'algorithme qui identifie les groupes est la méthode testGroup(), qui ne renvoie rien et prend en argument un indice x et un indice y, correspondant aux indices du bloc survolé.

Cette méthode est récursive car elle vérifie, bloc après bloc si ses adjacents font partie du même groupe ou non. Elle doit vérifier plusieurs conditions afin d'entrer dans la récursivité.

L'algorithme vérifie si le bloc adjacent est de la même couleur, donc est représenté par le même chiffre dans la structure de donnée tab et s'il n'est pas vide. Puis il vérifie si le bloc en question n'a pas déjà été ajouté par la méthode « testInGroup » qui consiste à tester si un bloc est déjà stocké dans le tableau group[][] lorsqu'on survole un groupe. S'il est libre et qu'il ne vérifie pas cette méthode, le tableau group prend la valeur du tableau d'entiers du bloc. La méthode est ensuite rappelée dans la même ligne mais avec la valeur de colonne diminuée.

L'algorithme vérifie de cette manière, au fur et à mesure, le groupe survolé en modifiant ses valeurs de colonnes et lignes différentes.

V} CONCLUSION

Tanguy :

Pour conclure je voudrais vous faire part de mes ressentis, contrairement au premier projet que nous avons eu à rendre, où nous avons eu quelques contraintes au niveau du temps, ce projet a été beaucoup mieux organisé, et nous a permis de gagner du temps, on a pu aussi avoir une meilleure coordination à l'aide de notre répertoire commun sur gogs. Ce projet m'a permis d'enrichir mes connaissances et de rectifié certaines lacunes dans tous les domaines appris lors de ce semestre en Java.

Leni :

De mon côté, j'ai ressenti la facilité que l'on pouvait avoir à coder en Java. Je trouve le langage bien plus intuitif et simple à utiliser que le C, que je trouvais complexe à cause de certaines notions, qui sont bien plus simple en Java. Ceci m'a beaucoup aidé, mais nous trouvons également que notre gestion du projet s'est améliorée, que ça soit la gestion de notre temps et également l'organisation avec git qui nous a aidé, c'est un outil bien pratique.

J'ai malgré cela eu des difficultés, notamment beaucoup de bugs à corriger et je ne maîtrise pas encore totalement mon sujet. Il s'agissait, à mon avis, d'un projet assez difficile à réaliser car les règles du jeu étaient, à mon sens, difficile à aborder dans le code.