

```
/* [SYSTEM] is_in_range : Permet de savoir si le joueur en question est à portée d'une case */  
int is_in_range(int xPos, int yPos, int x, case_board **case_data, int state)
```

# Projet Tutoré 2021

---

```
case_data[xPos][yPos].xPos);  
case_data[xPos][yPos].yPos);
```

```
case_data[xPos][yPos].xPos);  
case_data[xPos][yPos].yPos);
```

```
/* (aucunes cases in range) */
```

```
xPos;
```

5 AVRIL

---

IUT Sénart-Fontainebleau  
Lucas Grandjean, Adil Hammerschmidt



# Projet Tutoré : SameGame

## Table des matières

Projet Tutoré : SameGame .....	3
Table des matières .....	3
Présentation du rapport .....	3
Présentation du jeu .....	3
Le fonctionnement du jeu .....	4
Fonctionnement d'une partie .....	4
Schéma du programme.....	5
Conclusion.....	6
Point de vue de Adil.....	6

## Présentation du rapport

**Dans le cadre de notre cursus scolaire, pour notre première année de DUT Informatique à l'IUT de Sénart-Fontainebleau, il nous a été demandé de réaliser un jeu de SameGame en Java.**

Vous pourrez alors, à travers ce rapport, trouver tous les détails de la création du jeu, les fonctionnalités, des explications et bien d'autres encore.

Pour une navigation plus fluide, je vous invite à vous référer à la table des matières ci-dessus.

## Présentation du jeu

### **Qu'est-ce que le SameGame ?**

Pour y répondre, laissez-moi vous citer les règles du jeu d'après les consignes de notre projet tutoré :

**Dans ce jeu, une grille est remplie de blocs de trois types et votre but est de la vider. Un groupe de blocs adjacents et de 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.**

Le joueur va avoir la possibilité de jouer en solo. Une partie est de courte durée mais elle demande de faire preuve de réflexions pour accumuler le plus de points.

### Le fonctionnement du jeu

Le jeu a été codé en Java.

Grâce à un fichier « Makefile », l'utilisateur peut directement accéder au jeu en tapant la commande « make run » dans un terminal.

Dès lors, le menu se lance. Il vous sera proposé 3 options.

- **Jouer** permet de lancer une partie à partir d'une grille aléatoirement générée.
- **Charger une grille** permet de lancer une partie à partir d'une grille déjà enregistrée/crée sous le format txt.
- **Quitter** permet de fermer la fenêtre.

Une fois que vous lancez une partie, vous vous retrouverez face-à-face avec une grille de 15x10. Les blocs auront une couleur aléatoire, ou prédéfinie selon la méthode. Chaque bloc possède ses propres coordonnées.

Le joueur va alors pouvoir cliquer sur le groupe de blocs qu'il souhaite afin de les supprimer et d'obtenir des points.

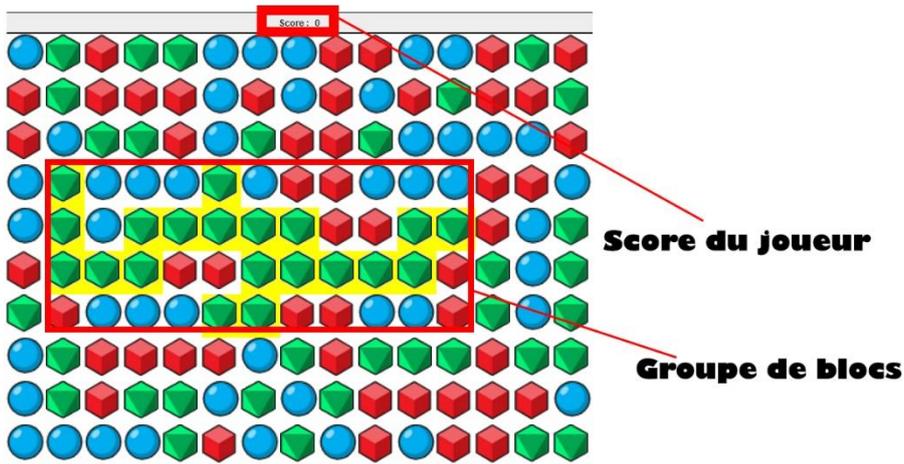
La partie se déroule jusqu'à ce qu'il n'y ait plus de combinaisons de blocs disponibles.

#### **Fonctionnement d'une partie**

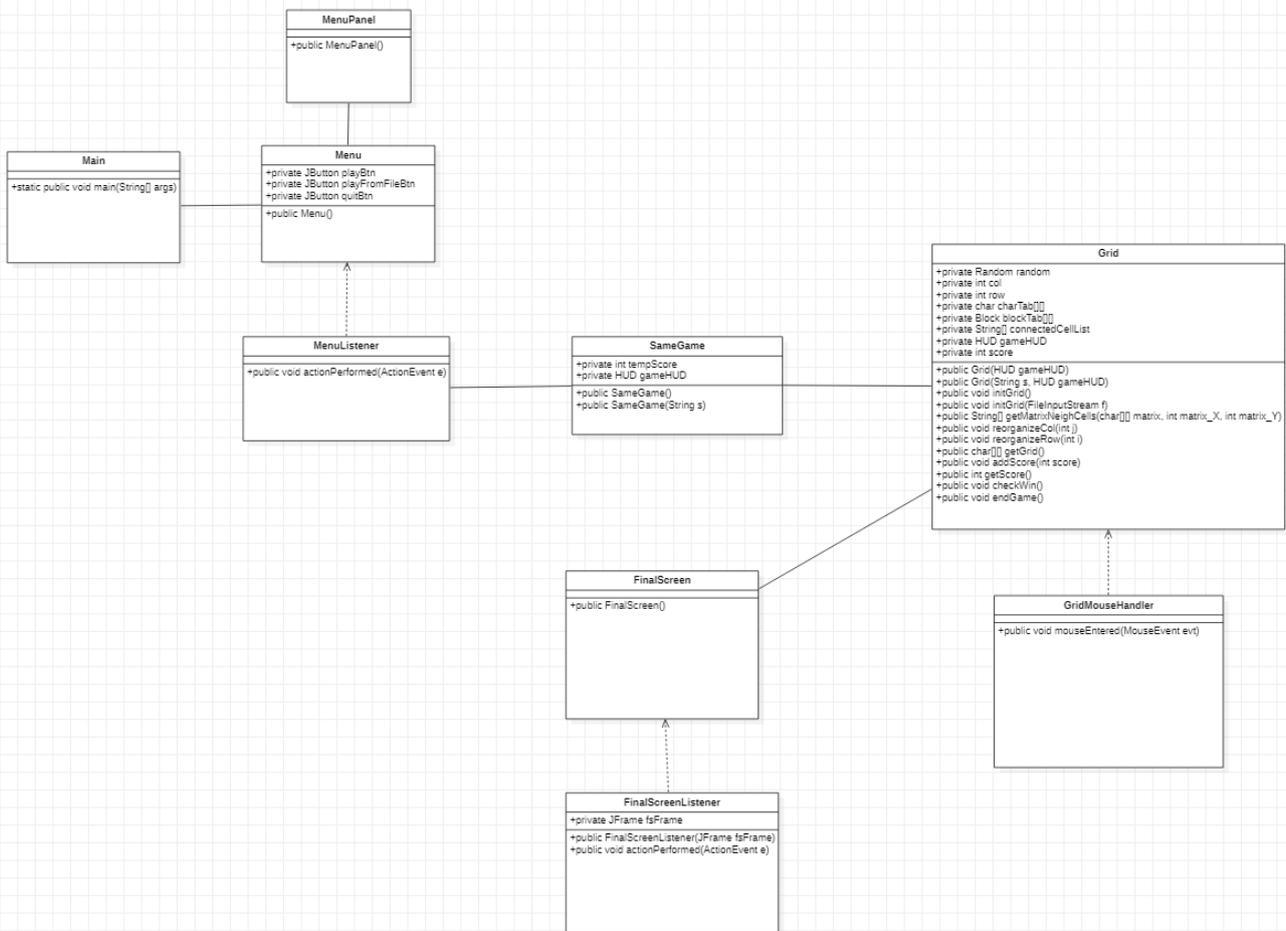
Une grille de 15x10 est générée, et l'algorithme va calculer grâce à de la récursivité les blocs adjacents ayant une valeur (couleur) similaire. Lorsque l'utilisateur clique, les points sont calculés en fonction du nombre de blocs et ils seront alors supprimés.

Les lignes et colonnes se verront alors réorganisés afin de faire tomber les blocs au-

dessus et de supprimer les colonnes vides.



### Schéma du programme



## Conclusion

### **Point de vue de Lucas**

Contrairement à l'ancien projet tutoré, j'ai trouvé cet exercice challengeant et proposant davantage de possibilités.

En effet, au contraire de la bibliothèque graphique limitée requise en C, avoir la possibilité d'utiliser l'intégralité des choix en Java offre une liberté intégrale, permettant de diversifier les solutions à de nombreux problèmes.

Nous avons manipulés les objets, les méthodes et mis en pratique la surcharge, le polymorphisme et bien d'autres choses utiles dans nos projets futurs et je pense que c'est une opportunité en or pour épaissir notre portfolio.

### **Point de vue de Adil**

Ce projet m'a relativement aidé à me familiariser avec la programmation orientée objet, notamment la notion de classe qui prend tout son sens ici. Les outils graphiques proposés par le java nous ont permis de gérer une interface graphique très facilement, là où en C c'était plus fastidieux.