

RAPPORT SAE2.1 2025



17 AVRIL

Réalisé par :
Malo Reboux
Dimitri Solar

```
selected=false;e.selected=true;e.startselected=true;c._trigger(
sion:"1.8.16")))(jQuery);
ui.mouse,{widgetEventPrefix:"sort",option
ith:false,containment:false,cursor:"auto"
items:">
ert:false,scroll:true,scrollSensitivity:
e={};this.element.addClass("ui-sortable")
s.length?d.axis==="x"||/left|right/,test
play")):false;this.offset=this.element.off
le").unbind(".sortable");this._mouseDown
n(d,c){if(d===
get()[c?"addClass":"removeClass"]("ui-sort
is,arguments)),_mouseCapture:function(d,c)
ptions.type=="static")return false;this._
=a(this);return false}};if(a.data(d,target
e).find("*").andSelf().each(function(){if(
urrentsFromItems();return true),_mouseStar
.refreshPositions();this.helper.css("top",
is.currentItem.offset();this.offsets[scor
t};this.helper.css("position","absolute"))
```

Sommaire

- **Introduction (Page 3)**
 - **Description du sujet**
 - **Objectifs**
 - **Brainstorming**
- **Difficultés et Répartition du travail (Page 4)**
- **Fonctionnalités 1 (Page 5)**
- **Fonctionnalités 2 (Page 6)**
- **Fonctionnalités 3 (Page 7)**
- **Fonctionnalités 4 (Page 8)**
- **Présentation de la structure et Diagramme de classe (Page 8 et 9)**
- **Algorithme des groupes de blocs (Page 10 et 11)**
- **Conclusion (Page 12)**

Introduction

Description du sujet :

Ce projet nommé SameGame consiste à créer un jeu, en Java, se déroulant sur un plateau rectangulaire rempli de blocs de couleurs, de taille 15 par 10, où le but est de détruire des groupes de blocs plus ou moins grand afin de marquer plus ou moins de points.

Objectifs :

En plus de la description donnée ci-dessus nous nous sommes fixés pour objectifs de réaliser un jeu intuitif, avec une interface utilisateur claire et une expérience claire.

Brainstorming :

Au cours de notre période de brainstorming pour le projet SameGame, nous avons pris soin de structurer notre travail en plusieurs étapes essentielles pour garantir un développement fluide et cohérent de notre projet. Nous avons d'abord élaboré un programme détaillé, comprenant les tâches nécessaires à l'achèvement du projet. Cela nous a permis de mieux organiser notre emploi du temps, en définissant des missions claires à accomplir chaque semaine, tout en préservant à la finalisation des tâches en cours et à la planification des nouvelles activités pour la semaine suivante.

Pour assurer une gestion optimale et un suivi constant de notre projet, nous avons organisé nos réunions à différents moments et dans divers lieux, aussi bien en présentiel dans les salles informatiques que virtuellement sur Discord. Cette flexibilité nous a permis de maintenir une communication fluide et d'ajuster notre travail en temps réel. Ces réunions étaient cruciales pour discuter de l'évolution du jeu, échanger nos idées que nous avons envisagées, et réexaminer les fonctionnalités qui avaient été temporairement mises de côté.

Nos échanges ne se limitaient pas aux seuls jours de la semaine ; le week-end a également joué un rôle fondamental dans notre collaboration. Ces sessions étaient dédiées à des discussions plus approfondies sur la progression du projet. Elles ont permis de traiter des points de vue variés, de confronter nos opinions, et même de résoudre certains désaccords qui auraient pu émerger. Cependant, notre engagement commun envers l'objectif final a toujours été maintenu, ce qui a assuré la satisfaction générale de toute l'équipe.

Cette approche collaborative, combinée à une gestion rigoureuse et à des échanges réguliers, a non seulement renforcé notre cohésion en tant qu'équipe, mais a également enrichi la qualité et la diversité des idées intégrées dans le jeu. Cela a permis de créer une dynamique de travail productive, essentielle à la réussite du projet SameGame.

Difficultés

Après avoir exposé nos objectifs lors du brainstorming initial du projet, nous avons rapidement été confrontés à des difficultés. Bien que notre première session de brainstorming ait permis de définir des objectifs clairs, il était encore difficile de passer de manière concrète à la réalisation d'un véritable projet. Afin de surmonter ces obstacles, nous avons organisé une réunion pour examiner de manière approfondie nos forces et faiblesses spécifiques par rapport au projet. Lors de cette réunion, nous avons eu l'occasion d'analyser en détail les défis rencontrés, qui étaient en grande partie cohérents avec les difficultés identifiées lors du brainstorming initial.

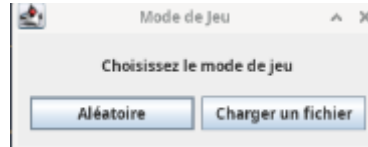
Répartition

Dimitri Solar a pris en charge la modélisation du plateau et de ses blocs, ainsi que l'identification des groupes de blocs mais, il a également assuré la prise en charge de l'affichage du plateau après la destruction d'un groupe de blocs. Malo Reboux quant à lui s'est chargé de la responsivité de la fenêtre du plateau, de l'affichage du score, du choix du mode jeu avant de commencer une partie, plateau aléatoire ou prédéfini, de la fenêtre de fin de partie, de la javadoc et aussi de l'écriture du rapport.

Cette approche collaborative a renforcé notre compréhension mutuelle des domaines de compétences de chacun et a facilité le partage d'idées et de compétences. Les échanges réguliers ont également contribué à améliorer notre efficacité, en nous permettant de résoudre rapidement les problèmes et d'avancer de manière plus fluide dans le développement du projet. Grâce à cette collaboration étroite, nous sommes parvenus à surmonter les défis initiaux et à progresser de manière significative dans la réalisation de notre projet commun.

Fonctionnalités 1

Image du menu :



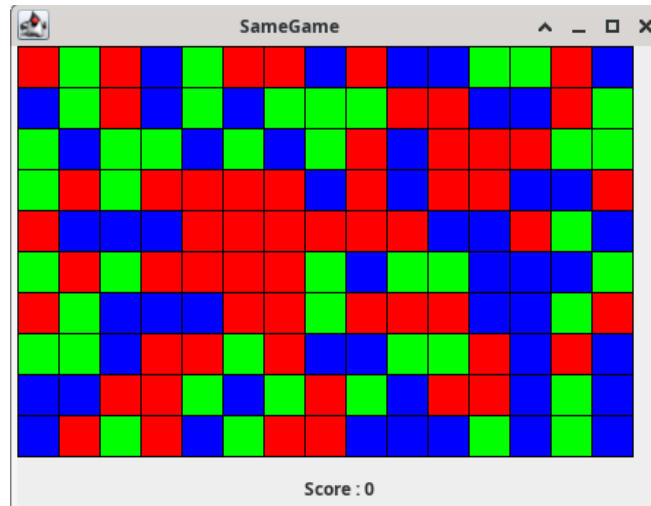
Lorsque le programme est exécuté, la fenêtre du menu s'ouvre nous permettant de choisir le mode de jeu, si « Aléatoire » est cliqué, un plateau aléatoire sera généré. Dans l'autre cas si « Charger un fichier » est choisi, le joueur devra choisir un fichier texte contenant seulement les lettres B, V et R aligné comme ceci :

```
RVVRRVRVBBBBRBV
BVVVVRBVVBRRVRB
VBBVRBVRRBRRRR
BRBVBRBBVBRVRV
RVBVBRRRBRRRBV
RVVBBRBVBVVRB
BRBRBBRBVRVRV
VRRVBBVBBRVVV
BVRRVBRVRRRBVV
BBRBBBBRVRRVRB
```

Cette grille de lettre générera des blocs des couleurs données (Rouge pour R, Vert pour V, Bleu pour B). Dans le cas où le fichier choisi ne contient pas le bon nombre de lettre ou tout simplement autre chose le jeu retournera un message d'erreur et lancera une partie en mode aléatoire.

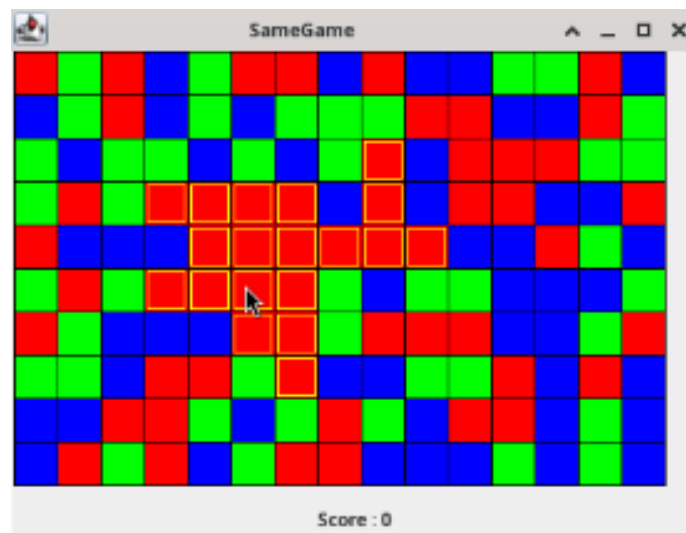
Fonctionnalités 2

Image du début de partie :



Lorsque que la partie est lancée nous arrivons sur un plateau 15x10 où le joueur pourra cliquer sur un groupe de blocs (2 blocs ou plus) afin de marquer des points.

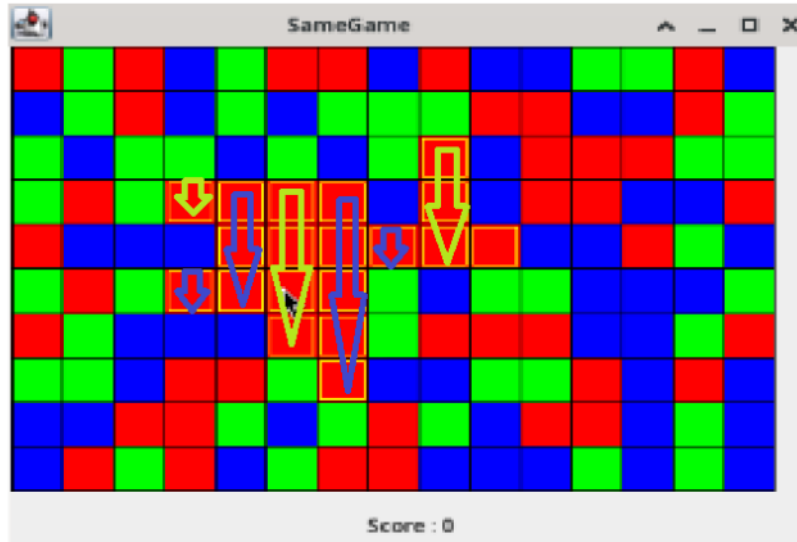
Image de la surbrillance d'un groupe de blocs :



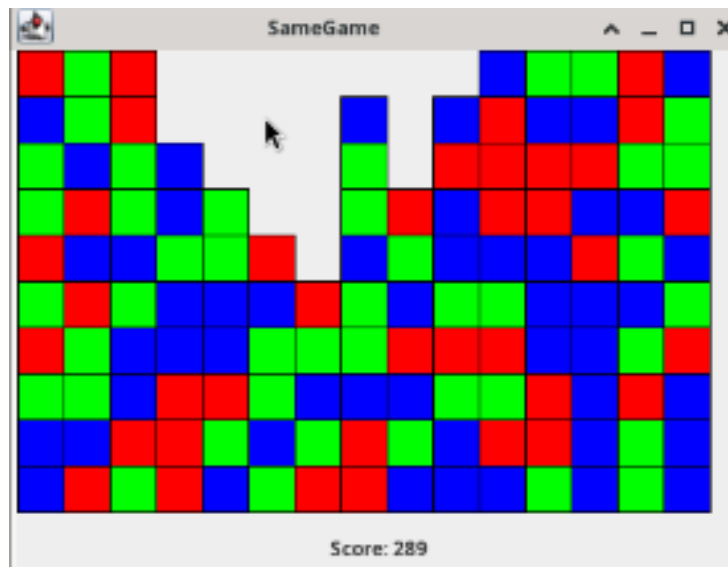
Lorsque le joueur passe sa souris sur un groupe de blocs, il est mis en surbrillance afin de faciliter la vision celui-ci.

Fonctionnalité 3

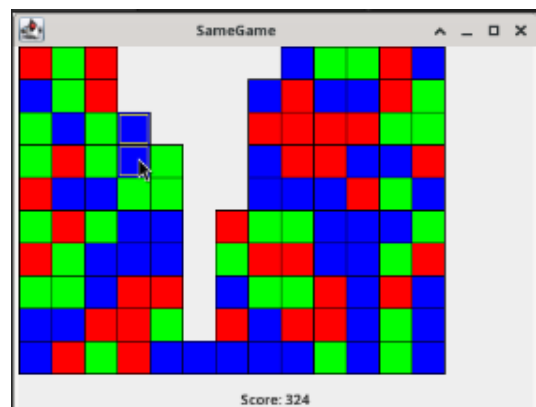
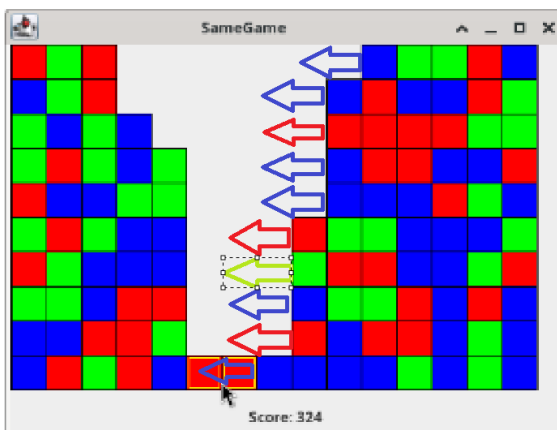
Image de la descente des blocs :



Après la destruction d'un groupe de blocs, les blocs au-dessus vont descendre pour prendre la place de ceux qui ont été détruit afin de créer de nouveaux groupes, comme ceci :

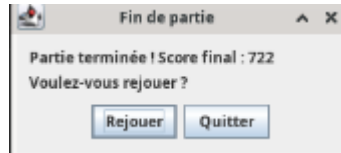


Il en va de même pour les colonnes quand elles deviennent vides :



Fonctionnalités 4

Image de fin de partie :



Lorsque plus aucun groupe de blocs n'existent, une fenêtre s'ouvre affichant le score final et invite le joueur à rejouer ou quitter s'il le souhaite.

Présentation de la structure :

SameGame :

- Description : Point d'entrée du programme

Liens : Appelle ChoixModeJeu

ChoixModeJeu :

- Description : Génère le plateau initial

Liens : Crée des Bloc[][] pour LogiqueJeu

Bloc :

- Description : Représente une case du plateau

Liens : Utilisé par LogiqueJeu

LogiqueJeu :

- Description : Gère les règles et l'état du jeu

Liens :

- Contient Bloc[][]
- avec PanneauJeu

PanneauJeu :

Description : Affiche le plateau et gère l'interface

Liens :

Contient LogiqueJeu

Possède EcouteurDeSouris et SurbrillanceSouris

EcouteurDeSouris :

Description : Gère les clics souris

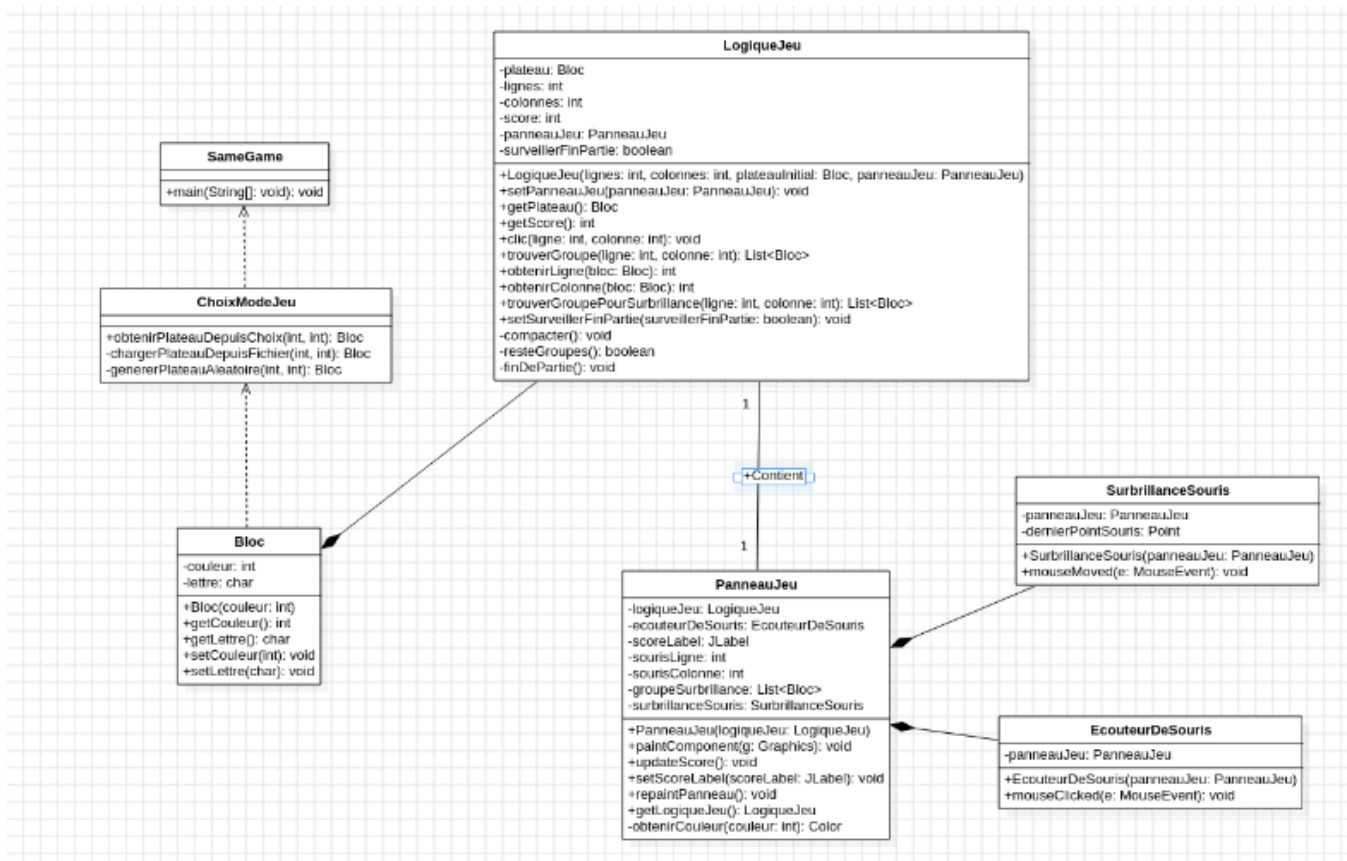
Liens : Appartient à PanneauJeu

SurbrillanceSouris :

Description : Gère le survol souris

Liens : Appartient à PanneauJeu

Diagramme de classe :



(Note : sur StarUML, nous n'avons pas trouver comment mettre des [] pour les types des variables donc nous allons préciser les manques ici :

Dans ChoixModeJeu :

Les types des 3 méthodes de type « Bloc » sont « Bloc[][] »

Dans LogiqueJeu :

Le type de la variable plateau est « Bloc[][] » et dans la méthode LogiqueJeu le type de la variable en argument « plateauInitial » et de type « Bloc[][] ».

)

Algorithme des groupes de blocs

Étapes de l'Algorithme :

1. Initialisation :

- Vérifie si la case de départ est vide → retourne une liste vide.
- Crée :
 - Une liste groupe pour stocker les blocs connectés.
 - Une matrice visite (booléens) pour marquer les cases visitées.
 - Une file pour l'algorithme de parcours en largeur.

2. Algorithme de parcours en largeur :

- Ajoute la position initiale à la file et la marque comme visitée.
- Mémoire la couleur de référence du bloc de départ.

3. Exploration des Voisins :

- Tant que la file n'est pas vide :
 1. Défile la position courante.

2. Ajoute le bloc au groupe.

3. Vérifie les 4 directions (haut, bas, gauche, droite) :

- **Si le voisin est :**
 - **Dans les limites du plateau**
 - **Non visité**
 - **De même couleur**
 - **Non nul**
→Ajoute à la file et marque comme visité.

4. Retour du Résultat :

Renvoie la liste groupe contenant tous les blocs connectés.

Conclusion

Ce projet a été pour nous une expérience riche en enseignements et en défis, marquant une étape importante dans nos cursus en programmation. Face à ce projet nous avons ressenti de l'enthousiasme comme de l'agacement face à certains problèmes, mais cela nous a permis de surmonter certaines difficultés dont nous n'aurions peut-être jamais eu connaissance.

Dimitri Solar :

Au cours de ce projet nous avons rencontré plusieurs difficultés, comme, au départ le fait de faire une grille cliquable, cela nous a posé bien des problèmes et des erreurs lors de l'exécution de notre programme. Également la gestion des groupes de blocs a aussi été source de problèmes, en effet lors de nos premiers essais le résultat attendu n'était pas ce que nous espérions, ce n'est que par la suite que nous sommes arrivés à corriger ce problème.

Malo Reboux :

La gestion du fichier de la grille prédéfini a aussi été difficile à obtenir, mais après persévérance nous sommes arrivés à un résultat convenable.

La création d'une javadoc a été également très enrichissante, cet exercice que nous n'avions pas souvent fait avant nous permet de relire notre code tout en le décrivant pour faciliter sa compréhension.

Pour conclure sur ce projet, tous les défauts que nous avons trouvés ont été corrigés, il nous a également permis d'améliorer notre travail en équipe, d'améliorer nos compétences en communication et à approfondir nos connaissances en développement.