

**SameGame**

## **Sommaire :**

Introduction.....3p

Description des fonctionnalités qui ont été ajoutées au jeu.....3p-5p

User-Story.....6p-7p

Présentation du diagramme de classe.....8p-9p

Conclusion ..... 10p

## **Introduction :**

SameGame est un jeu constitué d'une grille avec des blocs de trois types différents dispersés aléatoirement ou bien placés par le joueur par le biais d'un fichier. Le but est de vider la grille en faisant le plus grand Score. Pour cela, il vous suffit juste de réunir deux blocs du même types au minimum afin de vider la grille.

## **Description des fonctionnalités qui ont été ajoutées au jeu :**

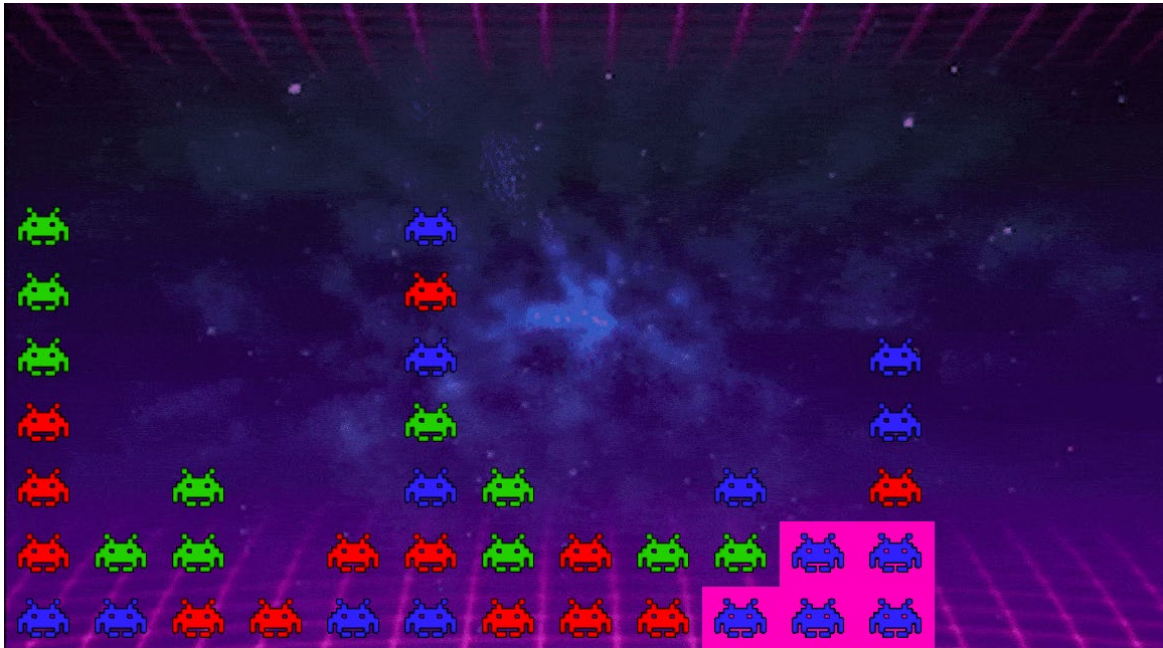
Le jeu actuel, reprends plusieurs fonctionnalités qui ont déjà été implémentées auparavant tels que :

- La possibilité de lancer une partie avec une grille aléatoire
- La possibilité de lancer une partie avec une grille écrite à l'avance par l'utilisateur
- La possibilité de jouer des coups pour l'utilisateur dans une partie
- La récupération des pions voisins à la sélections du pion à détruire
- Les points cumulés durant la partie sont calculés.

Des fonctionnalités qui ont été demander par le client en plus des fonctionnalités ont été implémenté également comme :

- Une **API interaction-moteur**, cet API permettra de faire le lien entre le jeu et les bots pour ces bots puissent influencer la partie.
- Un moyen de **visualiser la partie** avec un **historique des coups** du bot sur la grille
- Une **interface Homme-Machine** permettant à l'utilisateur de voir la partie du bot en direct. Il sera précisé que la partie est jouée par une machine et non pas un humain.

Comme vous pouvez le constater ci-dessous :

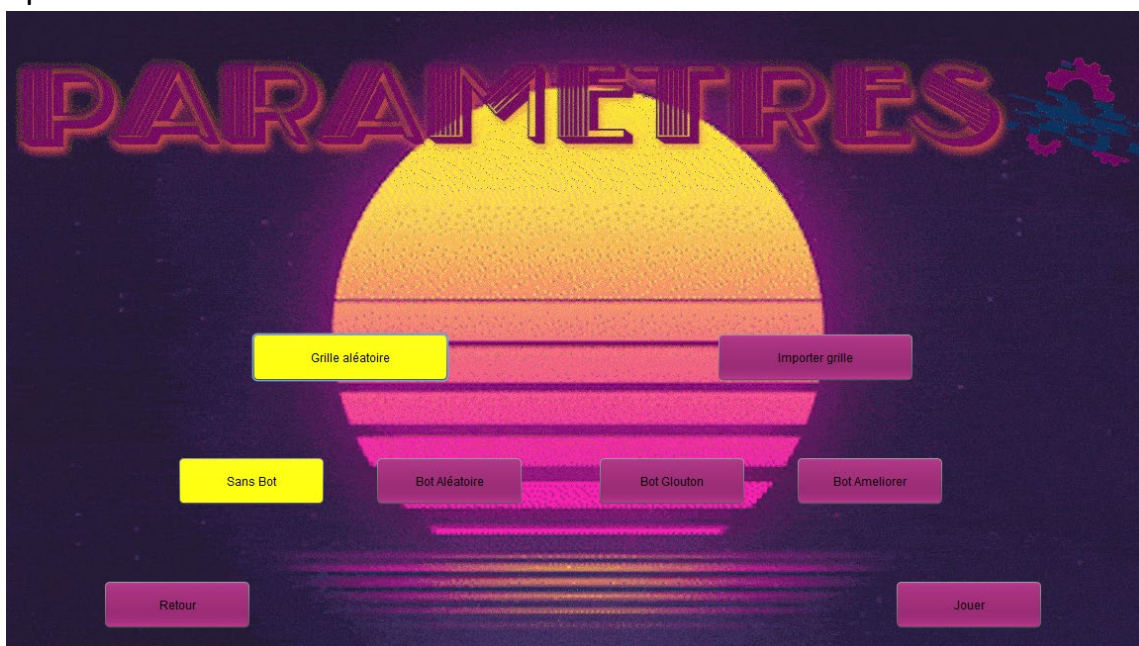


-**3 bots différents** ont été ajoutés au jeu :

-Un **bot aléatoire** : ce bot va jouer sans logique apparente et détruire les pions aléatoirement

-Un **bot Glouton** : ce bot va viser les paquets de pions les plus gros sur le plateau en priorité

-Un **bot amélioré** : ce bot va essayer de faire le meilleur score possible en fonction de la grille en fabriquant des paquets les plus gros possibles. Ces bots peuvent être activés depuis le menu paramètres comme représenté ci-dessous.



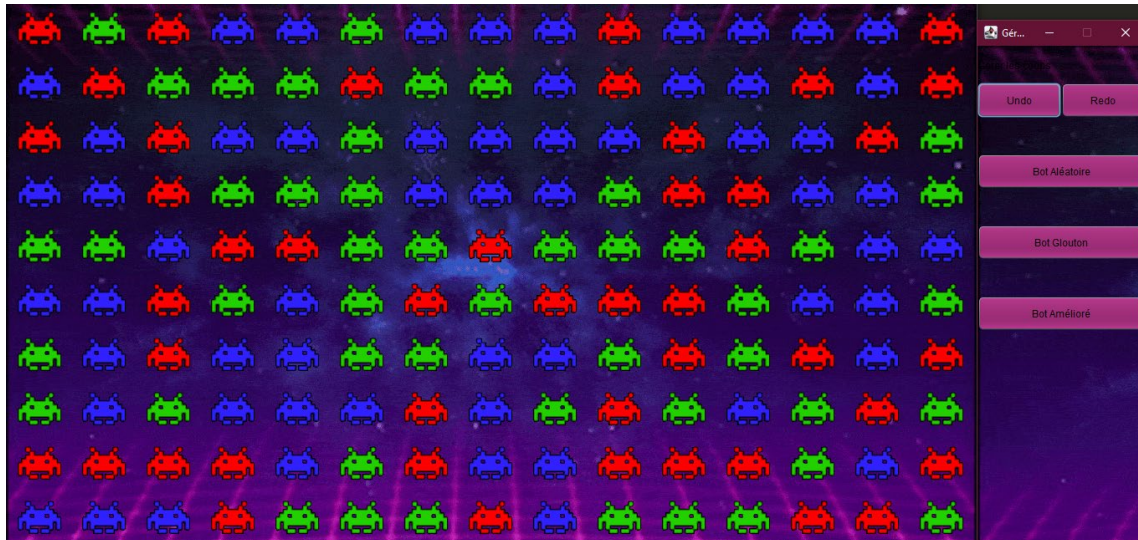
Mais également des fonctionnalités supplémentaires demander par le

client ont aussi été ajouté comme :

-**Undo** : qui permet de revenir au mouvement précédent.

-**Redo** : qui permet de revenir au mouvement suivant.

Cette fonctionnalité est disponible lorsque que l'on joue une partie normalement comme vous pouvez le voir.



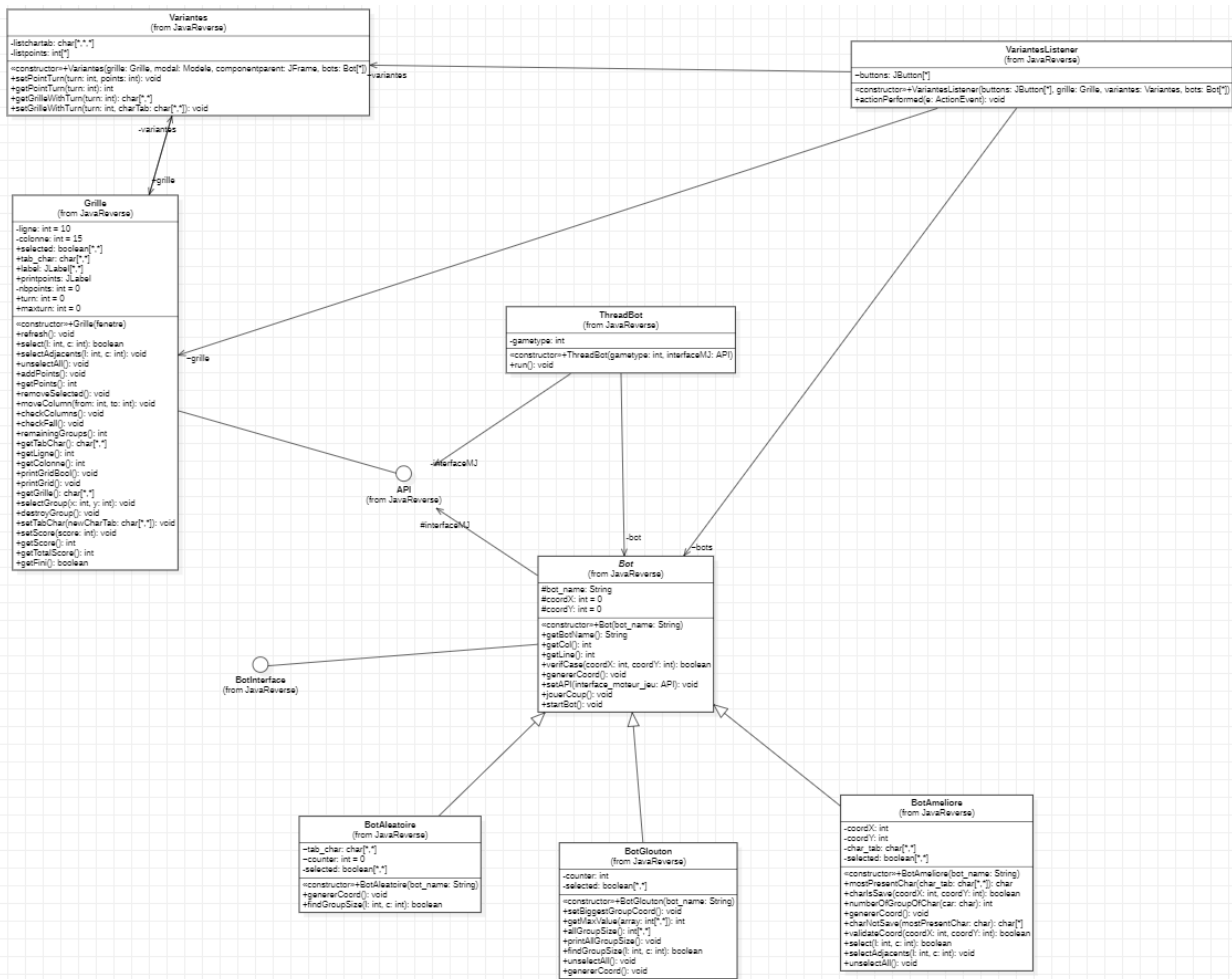
## User-Story

Sophie, sans se poser trop de questions avant de jouer, lance le jeu et apparaît sur le menu principal, il y a trois boutons, elle appuie sur le « Jouer ». Ceci lance une partie et fait apparaître deux fenêtres : une fenêtre où il y a une grille de 15x10 où les blocs d'aliens apparaissent. Et une deuxième où on peut contrôler la partie soit en revenant en arrière ou en avant, ou bien effectuer des actions avec des bots. Elle décide de détruire des blocs du même groupe de blocs en cliquant dessus après que le groupe soit mis en valeur. Ce qui fait descendre, par gravité vers le bas, les autres blocs qui n'étaient pas du même groupe que celui qui avait été sélectionné. Lorsqu'une colonne de blocs est vide entre deux autres colonnes, par gravité vers la gauche, les blocs sont décalés. Cependant Sophie a fait une erreur lors de sa partie et décide revenir en arrière grâce au bouton « Undo » situé sur la fenêtre de gestion de la partie. Après cet imprévu elle continue sa partie jusqu'à douter sur un choix qu'elle a fait ce qui lui fait utiliser les boutons « Undo » puis « Redo » puisqu'elle voulait vérifier si son choix était satisfaisant ou non. A la fin de la partie lorsqu'elle a fini de détruire tous les groupes, il apparaît sur le premier fenêtre le score de la partie et une phrase qui indique que la partie est terminée avec son score affiché. Il y a trois autres boutons, elle décide de cliquer sur « Menu » ce qui lui permet de retourner sur le menu principal, car elle hésite à recommencer une partie avec des paramètres personnalisés. Finalement elle décide que non et quitte en appuyant sur le bouton « Quitter ».

Bob veut jouer au SameGame avec des bots personnalisés, plus précisément il veut jouer avec un bot glouton. Pour cela il lance le jeu et apparaît sur le menu principal. Il y a trois boutons, il appuie sur le bouton « Paramètres ». Il veut aussi personnaliser sa grille avec un fichier créée par ses soins. Il clique donc sur le bouton « Importer grille » ce qui permet d'ouvrir une boîte de dialogue permettant la sélection d'un fichier et donc de choisir son fichier qu'il a créée. Pour choisir le bot glouton, il clique ensuite sur le bouton « Bot Glouton ». Et cliquer sur « Jouer » pour pouvoir jouer avec les paramètres personnalisés. Tout comme pour Sophie, le principe du jeu est le même cependant, ce n'est pas Bob qui joue vraiment mais le bot glouton sélectionné précédemment. Cependant, Bob peut voir la partie se dérouler en direct devant ses yeux, il peut être témoin des décisions du bot. La partie se fini rapidement vu que c'est un bot qui joue. Il y a toujours les systèmes de gravité qui font tomber les blocs. Le bot glouton est un bot amélioré du bot aléatoire. Un bot aléatoire est un bot qui comme le nom l'indique qui choisit les groupes de blocs aléatoirement sans aucune réflexion. Le

glouton effectue un minimum de réflexion avant de choisir les blocs. Après la partie terminée, Bob est redirigé sur le menu de fin de partie avec son score. Il décide de rejouer avec un bot amélioré, un bot glouton de meilleure intelligence. Il clique donc sur « Menu », puis depuis le menu principal le bouton « Paramètres ». Il reprend sa grille depuis un fichier en cliquant sur « Importer grille » et choisi donc un « Bot Amélioré » et sur « Jouer ». La partie se joue encore une fois et termine aussi avec le menu de fin. Cette fois ci Bob quitte le jeu en cliquant sur le bouton « Quitter ».

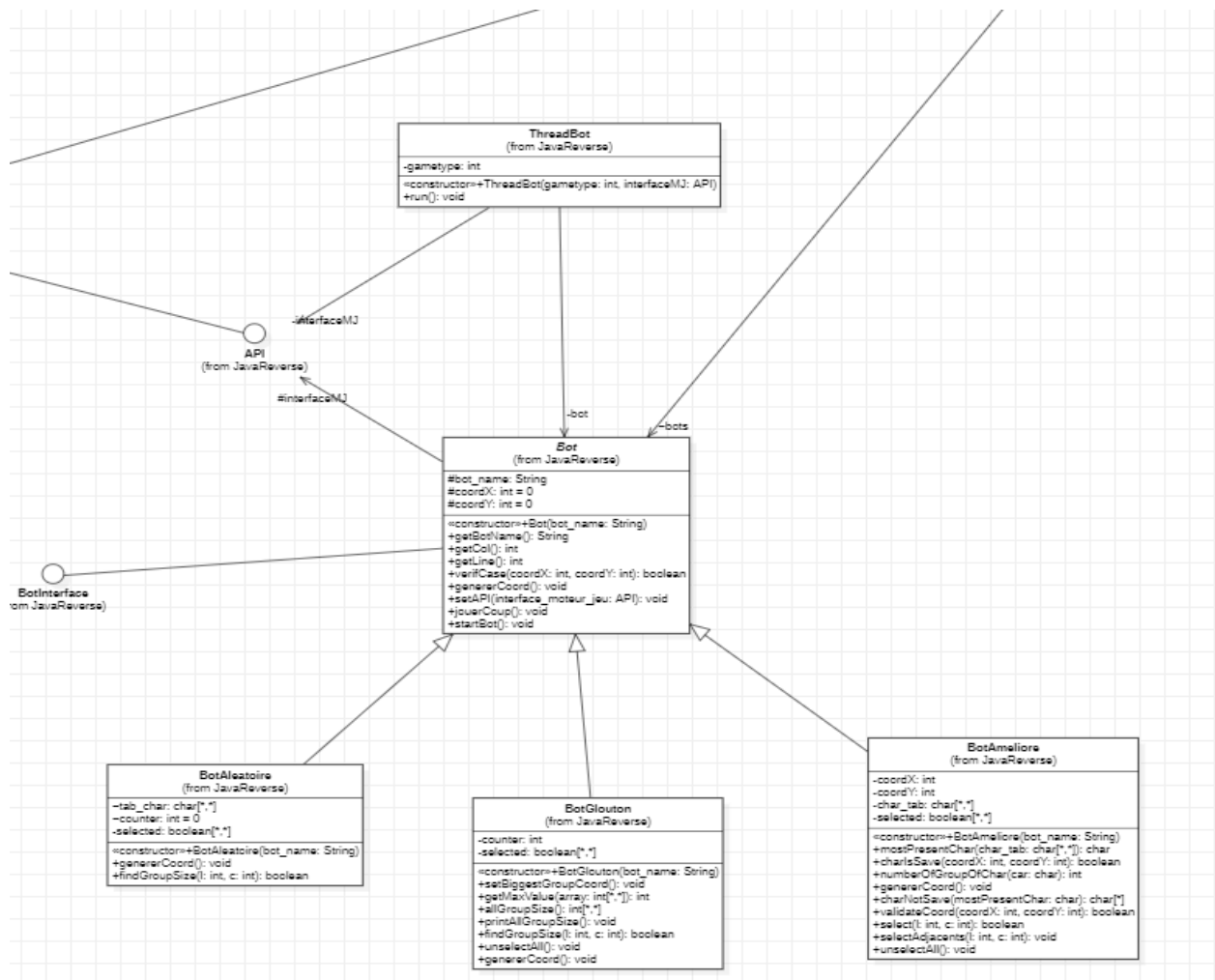
## Présentation du diagramme de classe :



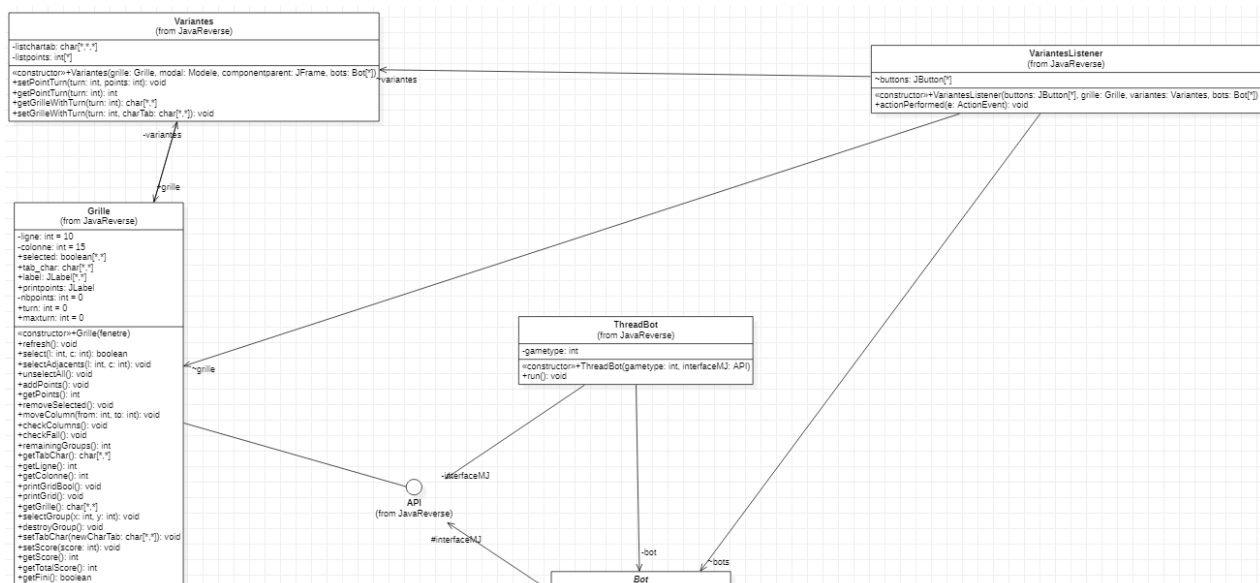
Voici, ci-dessus le diagramme de classe des fonctionnalités qui ont été ajoutées au jeu.

On peut voir au centre du diagramme une classe bot. Cette classe est liée à l'interface de l'API. Elle permet comme vous pouvez voir ci-dessous de récupérer les informations sur la grille comme avoir les coordonnées en x et y de la grille ainsi que de vérifier les cases pour finalement jouer un coup. Cette classe est liée à 3 autres classes qui sont respectivement les trois bots décrits juste au-dessus. Le bot aléatoire va détruire les pions aléatoirement, le bot glouton va détruire en premier le plus gros paquet et enfin le bot amélioré va détruire les paquets de pions les moins présents dans la grille pour que la fin cela-terminer en un gros paquet.





Repris u jeu de base la classe Grille ne change pas en revanche, les classes Variant et VarianteListener ont été ajoutées. Ces classes permettent la mise en place et le fonctionnement des fonctionnalités undo et redo mais aussi d'avoir accès durant la partie en mode normal à un ou plusieurs coups des bots présentés ci-dessus.



## **Conclusion :**

Les fonctions majoritaires ainsi que certaines fonctionnalités demander par le client au fur et à mesure de l'avancement du projet ont bien été réalisées comme expliqué ci-dessus. Le projet est globalement bien abouti malgré le quelque manque de fonctionnalité (en should) qui nous reste à ajouté au projet. Dans l'ensemble, il y a eut un bon travail d'équipe sur le projet.