



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ×

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

**DOCUMENTATION POUR LE
DEVELOPPEUR**

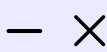


MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE



DESTINATAIRE



KHITEM BLIDAQUI

UNIVERSITÉ DAUPHINE PSL

ARCHITECTURE



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

ARCHITECTURE

Notre projet est structuré en deux packages :

- Package “model” : il regroupe les classes utilitaires.
- Package “controller” : il regroupe les classes relatives à JavaFX et aux méthodes nécessaires au bon déroulement du jeu.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

PACKAGE "MODEL" - BOARD

La classe Board représente le tableau de jeu. Elle contient tous les points et les lignes placés dessus.

Son initialisation place les points de base selon la forme choisie (Croix ou Tsunami). La classe offre plusieurs méthodes utiles, telles que l'ajout d'un point, l'ajout d'une ligne, la vérification de la possibilité d'un alignement dans une même direction (mode 5T), et la détermination du point appartenant à quelle ligne.

La classe Board est abstraite et étendue par les classes Board5D et Board5T, en fonction du mode de jeu choisi. Ces classes sont instanciées dans la classe GameManager.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

PACKAGE "MODEL" - POINT

La classe Point symbolise un point placé.

Les points sont instanciés dans la classe Board.

Cette classe possède des fonctions utilitaires, notamment pour vérifier si un point est égal à un autre.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

PACKAGE "MODEL" - LIGNE

La classe Ligne symbolise une ligne jouée.

Également instanciées dans la classe Board, les lignes sont composées de 5 points.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

DESTINATAIRE

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

PACKAGE "MODEL" - PLAYER

La classe Player symbolise un joueur.

Cette classe est utilisée pour sauvegarder le nom d'un joueur lorsqu'une partie est lancée.

RA



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

PACKAGE "CONTROLLER" - GAMEMANAGER

La classe GameManager permet de créer le tableau de jeu contenu dans la classe Board.

Elle contient également les dimensions du tableau de jeu, utiles pour l'affichage et la vérification de la validité des points.



MORPION SOLITAIRE

DARYL MARTIN-DIPP

GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI

UNIVERSITÉ DAUPHINE PSL

PACKAGE "CONTROLLER" - HOMECONTROLLER

La classe HomeController est le contrôleur associé à la vue d'accueil de l'application.

Elle gère les interactions utilisateur liées au lancement du jeu Morpion Solitaire, à la configuration des options (nom d'utilisateur et mode de jeu), et à l'affichage du tableau des scores.

Cette classe utilise JavaFX pour la gestion de l'interface utilisateur.

Elle offre des méthodes pour lancer le jeu, charger le tableau des scores, et configurer les options de départ.

De plus, elle présente des styles visuels dynamiques pour les boutons de la page d'accueil.

La sélection du mode de jeu influence le comportement du jeu, avec des ajustements spécifiques pour les différents modes ("5T," "5D," "5DTsunami," et "5TTsunami").



MORPION SOLITAIRE

DARYL MARTIN-DIPP

GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI

UNIVERSITÉ DAUPHINE PSL

PACKAGE "CONTROLLER" - MORPIONSOLITAIRECONTROLLER

La classe `MorpionSolitaireController` est le contrôleur associé à la vue du jeu Morpion Solitaire, gérant les interactions utilisateur, l'affichage du score, et la logique du jeu.

Elle utilise JavaFX pour l'IHM, communique avec le modèle (`GameManager` et `Player`), et inclut des fonctionnalités spécifiques au mode "5T".

La classe respecte le modèle MVC en séparant la logique du jeu de l'interface utilisateur.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

PACKAGE "CONTROLLER" - VIEWMORPIONSOLITAIRE

La classe ViewMorpionSolitaire est responsable du rendu graphique du jeu Morpion Solitaire.

Elle utilise JavaFX pour dessiner le plateau de jeu, les points, les lignes, et gère les interactions utilisateur, notamment pour le mode "5T".

La classe offre des méthodes pour dessiner différents éléments du jeu, comme les points cliquables, les points placés, les lignes, et les points surlignés.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

CLASSES HORS PACKAGE

La classe `GameUI` permet de lancer la première scène, la page d'accueil.

La classe `Main` permet de lancer la classe `GameUI`.



MORPION SOLITAIRE

DARYL MARTIN-DIPP

GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI

UNIVERSITÉ DAUPHINE PSL

ARCHITECTURE

L'architecture adoptée pour notre projet de Morpion Solitaire présente plusieurs avantages significatifs. La séparation en deux packages distincts, "model" et "controller", suit le principe du MVC (modèle-vue-contrôleur), offrant une structure claire et modulaire.

Dans le package "model," les classes utilitaires telles que Board, Point, Ligne, et Player sont conçues de manière à encapsuler logiquement les éléments du jeu. La classe abstraite Board permet une extension facile pour les modes de jeu spécifiques (Board5D et Board5T). Cette approche favorise une gestion claire des entités du jeu et permet d'ajouter de nouveaux modes avec une relative facilité.

Le package "controller" regroupe les classes responsables de l'interface utilisateur et de la logique du jeu. Les classes telles que GameManager, HomeController, MorpionSolitaireController, et ViewMorpionSolitaire respectent les principes du MVC, garantissant une séparation nette des responsabilités. L'utilisation de JavaFX dans les contrôleurs offre une interface utilisateur dynamique et réactive.

Le choix de séparer le contrôleur du rendu graphique, avec MorpionSolitaireController et ViewMorpionSolitaire, permet une gestion modulaire des interactions utilisateur et du rendu visuel. Cela facilite la maintenance et l'extension du code.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

LISTE DES IMPLÉMENTATIONS



MORPION SOLITAIRE

DARYL MARTIN-DIPP

GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI

UNIVERSITÉ DAUPHINE PSL

DIFFÉRENTS MODES DE JEU

Notre projet intègre un moteur de jeu permettant de jouer au Morpion Solitaire, avec les modes 5T, 5D, 5D Tsunami et 5T Tsunami entièrement opérationnels.

L'implémentation de ces fonctionnalités s'est avérée aisée grâce à l'utilisation de classes préalablement définies pour le positionnement des points selon les schémas prédéfinis.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

INTERFACE GRAPHIQUE

L'élaboration de cette implémentation a été, à mon sens, la phase la plus complexe de notre projet.

Initialement, il a fallu déterminer la bibliothèque JavaFX la plus appropriée à nos besoins.

Etant donné notre expérience antérieure avec les Canvas, nous avons conclu que c'était l'option la plus adaptée pour notre application.

De plus, nous considérons que l'aspect le plus dur pour nous a été de résoudre le problème de la transition d'un clic sur une image, qui renvoie des coordonnées en pixels, à la sélection d'un point caractérisé par des coordonnées x et y.

Heureusement, nous avons découvert quelques astuces, notamment en sauvegardant la hauteur et la largeur du canvas, ainsi que les dimensions du quadrillage.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

UTILISATION DU MODE ST DANS L'INTERFACE GRAPHIQUE

Cette implémentation s'est également révélée très complexe pour nous.

Le défi résidait dans le fait que, jusqu'à présent, dans l'invite de commande, lors de la détection de deux lignes pouvant se toucher, l'utilisateur était invité à choisir entre les deux points possibles.

Cependant, transposer cette logique sur l'interface graphique s'est avérée très difficile. Nous ne souhaitions pas utiliser de menu déroulant, mais plutôt, comme sur d'autres jeux en ligne, des points surlignés en attente d'un clic.

Cependant, nous avons rencontré des difficultés à permettre à l'utilisateur, tout en étant dans la fonction de choix, de sélectionner l'un des deux points.

Après de nombreuses réflexions et plusieurs heures d'essais, nous avons réussi à diviser la méthode de demande en deux, en enregistrant les informations nécessaires dans des variables globales.

Cela nous a permis de se souvenir des deux points et de suspendre l'exécution jusqu'à ce que l'un des deux points soit effectivement cliqué.

Proches de plusieurs moments d'abandon, nous avons finalement réussi à le faire fonctionner.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

AFFICHAGE DE L'INDICE

L'implémentation de cette fonction a nécessité la création d'une méthode parcourant l'ensemble de la grille à la recherche des points jouables.

Ce processus a été relativement simple, car il s'agissait essentiellement d'une boucle qui, pour chaque point, vérifiait s'il était jouable.

Cependant, nous avons rencontré une difficulté à ce stade. La méthode vérifiant la possibilité de jouer un point le plaçait automatiquement sur la grille si c'était le cas.

Pour remédier à cela, nous avons introduit l'utilisation de booléens pour déterminer si le coup doit effectivement être joué ou s'il s'agit simplement d'une vérification.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

SUPPRESSION D'UN COUP

L'implémentation de cette fonctionnalité s'est révélée relativement simple pour nous.

La tâche a été grandement facilitée par la sauvegarde de la liste des points et des lignes jouées dans l'ordre.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

AFFICHAGE DU SCORE EN TEMPS RÉEL

La récupération du score s'effectue assez facilement, car nous avons prévu dès le début de l'implémentation de l'inclure dans les attributs de la classe Board.

Initialement, nous ne disposions pas des setters pour gérer les coups supprimés, mais cela a été facilement résolu par la suite.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

AFFICHAGE DES NUMÉROS SUR LES POINTS JOUÉS

Pour résoudre cela, nous avons d'abord envisagé d'ajouter le numéro de ligne comme attribut à la classe Ligne.

Cependant, nous avons rencontré des difficultés pour le récupérer sur le point après l'avoir joué.

De plus, la gestion des suppressions de points nécessitait l'ajout de setters, ce qui rendait notre code plus complexe.

Pour simplifier, nous avons opté pour la création d'une ArrayList stockant tous les points joués par l'utilisateur, les distinguant ainsi des points utilisés lors de l'initialisation de la croix ou du tsunami.

De cette manière, il suffisait d'ajouter le nombre de points à cette liste après avoir placé un point.

Si le point était supprimé du tableau de jeu, il était également retiré de cette liste, simplifiant ainsi le processus.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

MÉTHODE DE RECHERCHE DE SOLUTION ALÉATOIRE

Cette tâche a également été relativement simple, car nous disposions déjà de fonctions permettant de retrouver tous les points non-joués, utilisées notamment pour l'affichage de l'indice.

Notre démarche était donc assez simple : parcourir les points restants à jouer, en choisir un au hasard, le jouer, puis répéter le processus.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

DÉTECTION DU GAME OVER

Nous avons réussi assez facilement à mettre en place la détection de la fin de partie grâce à la même fonction qui trouve tous les points non-joués.

Si cette liste est vide, alors la partie est considérée comme terminée.

De plus, nous avons amélioré l'animation en ajoutant l'affichage du nom et du score du joueur.

Nous avons également inclus la possibilité pour le joueur de quitter le jeu ou de retourner à l'écran d'accueil.



MORPION SOLITAIRE

DARYL MARTIN-DIPP

GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI

UNIVERSITÉ DAUPHINE PSL

BASE DE DONNÉES

Une base de données FreeSqlDatabase a été spécialement mise en place pour le projet.

Cette base de données permet de sauvegarder le classement en utilisant un ID, un PlayerName, un score et un mode.

La mise en place de cette fonctionnalité a été assez simple, car il suffisait de communiquer avec la base de données en envoyant toutes les informations requises lors d'une fin de partie.

Étant déjà familiers avec le langage SQL et ayant précédemment utilisé des bases de données avec Java, cette tâche nous était relativement aisée.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ×

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

IMPLÉMENTATIONS
MANQUANTES



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

IMPLÉMENTATIONS MANQUANTES

Malheureusement, plusieurs implémentations que nous souhaitions ajouter au code n'ont pas été réalisables en raison de diverses contraintes.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

AFFICHAGE DE LA MEILLEURE GRILLE

Nous avons réussi à mettre en place des fonctionnalités permettant la génération de grilles.

Malheureusement, nous n'avons pas réussi à implémenter la sauvegarde de la meilleure grille, car le processus était inconnu et malheureusement trop complexe pour nous, ainsi que pour notre code.

Étant donné notre contrainte de temps limité, nous avons préféré nous concentrer sur la consolidation de notre code et des fonctionnalités déjà présentes.

L'objectif était de livrer un travail plus léger mais de qualité, plutôt que d'ajouter des options qui n'étaient pas totalement optimisées.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

AJOUTS D'AUTRES ALGORITHMES

Nous avons tenté à maintes reprises d'ajouter la possibilité d'intégrer d'autres algorithmes dans notre jeu, tels que NMCS, par exemple.

Malheureusement, notre code n'a pas été initialement conçu dans cette optique, et il était trop complexe de le modifier.

Le principal problème résidait dans la copie des tableaux de jeu pour effectuer plusieurs tests sans altérer l'original.

Notre tableau de jeu, étant un attribut de notre classe `MorpionSolitaireController`, ne pouvait pas être copié.

Les méthodes détectant une fin de partie ou recherchant des points manquants étaient donc conçues dans ce sens : elles étaient présentes dans la classe `MorpionSolitaireController` et non pas dans la classe `Board`.

Le jeu n'était donc pas prévu pour avoir plusieurs instances de classes.

Cela a été notre principal regret. Si ce projet était à refaire, nous aurions sans aucun doute pensé à construire notre code dans ce sens là, pour pouvoir sans problème ajouter de nouveaux algorithmes.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

**AFFICHAGE SOUS FORME DE GRAPHIQUE DE LA DISTRIBUTION
DES SCORES OBTENUES, ET MULTI-THREADING**

Malheureusement, n'ayant qu'une seul algorithme de recherche, comme mentionné précédemment, nous n'avons pas l'utilité d'implémenter cela.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ×

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

PROPRETÉ DU CODE



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

PROPRETÉ DU CODE

Nous avons accordé une grande importance à la propreté et à la documentation de notre code.

Toutes les fonctions essentielles sont commentées, de même que les parties plus complexes du code. Nous avons veillé à ce qu'un développeur n'ayant jamais travaillé sur notre programme puisse le comprendre rapidement et le compléter en utilisant des méthodes génériques et utilitaires que nous avons écrites.

Par ailleurs, nous avons adopté une approche modulaire en segmentant notre code et en créant de nombreuses fonctions réutilisables. Cette démarche vise à améliorer la clarté, la propreté et la lisibilité du code.

Selon l'analyse d'IntelliJ, notre code est dépourvu de fautes grammaticales, de fonctions inutiles, de code redondant, et ne contient aucune ligne pouvant être simplifiée.



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

RÉPARTITION DU TRAVAIL



MORPION SOLITAIRE

DARYL MARTIN-DIPP
GUILLAUME KARAOUANE

● DESTINATAIRE — ✕

KHITEM BLIDAQUI
UNIVERSITÉ DAUPHINE PSL

RÉPARTITION DU TRAVAIL

Nous avons réparti équitablement les tâches en fonction de nos compétences respectives.

Daryl MARTIN-DIPP, ayant une bonne expérience en Java, a pris en charge le développement du code, la réalisation des tests, et la conception globale de l'architecture du programme. Il a veillé à maintenir la propreté de notre code et a rédigé les documentations relatives au développement.

Guillaume KARAOUANE a pris en charge l'intégralité de la partie interface graphique, assurant la liaison entre le code et l'affichage. Il a également joué un rôle crucial dans le choix des fonctionnalités à implémenter et des améliorations à apporter.