

Rapport Projet 2026 - Jeu Dékal

Guillaume BEUVOT, François LASOTA

10 Mai 2026

Table des matières

1	Éléments du jeu	1
1.1	Jeu	1
1.2	Joueur	1
1.2.1	Humain	1
1.2.2	Ordinateur	1
1.3	Carte	1
1.4	Case	2
1.5	Grille	2
1.6	Directions	2
1.7	Voisins	2
2	Diagramme UML du projet	3
3	Lancement	3

1 Éléments du jeu

1.1 Jeu

La classe Jeu permet de lancer une partie, de gérer les tours et d'initialiser la pioche de départ (mélanger les cartes).

Le jeu possède un ensemble de joueurs.

1.2 Joueur

Les joueurs ont un nom et une grille, et ont la possibilité de piocher une carte dans leur grille et la mettre au centre ainsi que de tirer une carte au centre et la placer dans la grille.

Un joueur peut être un humain ou un ordinateur, la différence se fait dans le choix de chaque action.

1.2.1 Humain

Un humain peut librement choisir quelle carte tirer dans la grille, quelle carte prendre au centre et la placer dans la grille. Pour tirer une carte dans la grille, l'humain renseignera deux valeurs, une abscisse et une ordonnée. Pour renseigner ou placer la carte dans la grille, il renseignera la direction vers laquelle pousser le reste des cartes.

1.2.2 Ordinateur

Un ordinateur effectue tous les choix à l'aide d'un générateur de nombres aléatoires.

1.3 Carte

Une carte contient sa valeur et peut être visible ou non.

1.4 Case

Une case contient une carte. Il y a 4×4 cases fixes dans la grille, ce sont les cartes qui changeront de case au cours de la partie.

Chaque case contient l'ensemble de ses voisins ainsi que leur direction (HAUT, BAS, GAUCHE, DROITE). Il est possible de retirer une carte d'une case. Une case sans carte est une case vide. Il est aussi possible de placer une carte dans une case, elle devient alors non-vide. Lors du décalage de cartes (lorsqu'une ligne ou une colonne est décalée), chaque case non vide appelle récursivement la méthode de décalage sur la case voisine dans la direction vers laquelle le décalage s'effectue. Lorsqu'une case vide est rencontrée, alors chaque case rencontrée garde comme carte la carte de la case précédente.

Pour déterminer son score, une case renvoie la valeur de a carte si aucun voisin ne possède de carte de la même valeur, 0 sinon.

1.5 Grille

Une grille possède en mémoire sa première case (en HAUT à GAUCHE). Les positions de toutes les cases sont déterminées par voisinage, ainsi que les coordonnées de la case vide, qui servira à décaler les lignes ou les colonnes.

Pour obtenir une case par ses coordonnées, une méthode récursive est utilisée en prenant en paramètre une case et des coordonnées. Chaque appel fait décrémenter l'ordonnée puis l'abscisse jusqu'à atteindre le couple de valeurs (0,0). Pour chaque appel, le voisin de la case est pris en paramètre (Voisin du BAS si l'ordonnée est décrémentée, voisin de DROITE si l'abscisse est décrémentée). Si le couple (0,0) est atteint, la méthode renvoie la case mise en paramètre.

La grille est remplie à l'aide de la méthode de recherche de cases.

Pour calculer le score d'une grille, il suffit de calculer le score de chaque case.

Une grille peut placer ou retirer une carte dans l'une de ses cases.

1.6 Directions

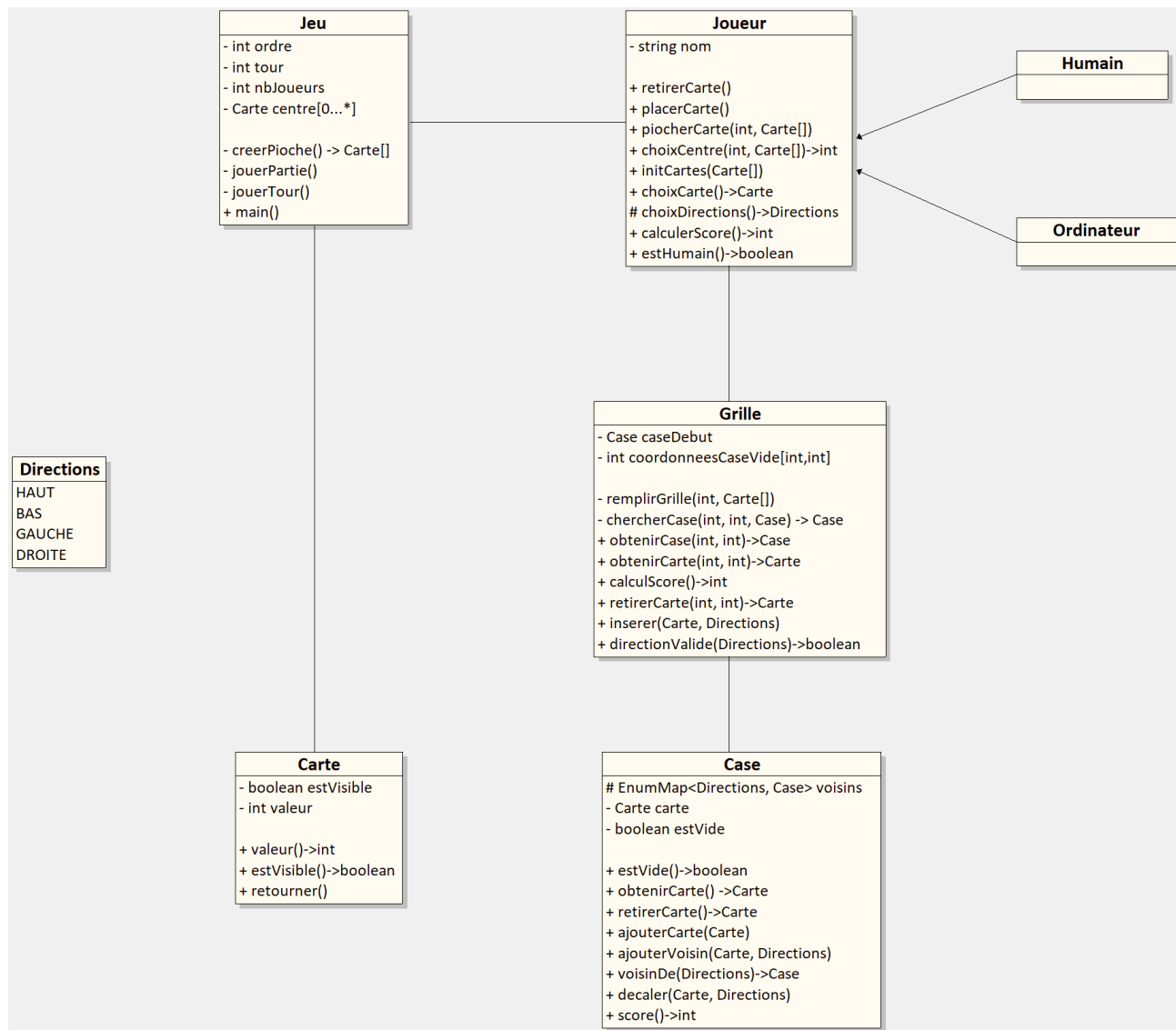
Les directions sont gérées par le type énuméré Directions.

1.7 Voisins

Les voisins sont gérés par une EnumMap<Directions, Case>

2 Diagramme UML du projet

Le diagramme UML du projet est le suivant :



3 Lancement

Le jeu peut être lancé avec la commande `./run.sh`