**Algorithmes Utilisés - Dodgem**

## 1) Création du plateau

Pour permettre la création du plateau, le plus pratique est d’utiliser une liste de listes pour un tableau 2D.

Chaque case prend en compte :

* 0 ⬄ pour la valeur ‘neutre’,
* 1 ⬄ pour le pion du joueur (J1) ‘pion1’,
* 2 ⬄ pour le pion du joueur (J2) ‘pion2’.

Aussi, chaque joueur dispose d’un sigle pour différencier son jeu et donc leurs pions sur le plateau se distinguent par :

* ‘x’ ⬄ pour J1,
* ‘o’ ⬄ pour J2.

Par ailleurs, on laisse l’opportunité aux joueurs de choisir la taille du plateau

## 2) Placement de départ

La fonction **init\_depart** place les pions du J1 sur la colonne de gauche (sauf la case du bas), J2 est sur la ligne du bas (sauf la case du coin en bas à gauche). Cela correspond à la position d’un départ classique de Dodgem.

## 3) Vérification de l’existence d’une case

La fonction **in\_bounds** vérifie que les coordonnées (i, j) sont bien dans notre plateau. Cela est important pour éviter de sortir du tableau quand on déplace un pion.

## 4) Choix du déplacement d’un pion

La fonction **selectPawn** permet au joueur de choisir la case de son pion. Elle vérifie que la case appartient bien au joueur et que ce pion peut se rendre dans cette direction.

## 5) Vérification du déplacement d’un pion

La fonction **possiblePawn** teste les 4 directions (haut, droite, bas, gauche). Si au moins une direction est bonne, le pion est « déplaçable ». Sinon, on redemande une case car son accès est impossible.

## 6) Les directions et les règles de déplacement

Les directions sont sous forme de couple (di, dj), préalablement définis. Par exemple, (0, 1) veut dire : même ligne, colonne +1 (vers la droite).

La fonction **possibleMove** vérifie plusieurs choses :

* Que la case contient bien un pion du joueur
* Qu’il n’y ait pas de retour arrière possible ⬄ J1 ne peut pas revenir à gauche et J2 ne peux pas se déplacer vers le bas
* Que la sortie n’est possible que vers la droite pour J1 et que vers le haut pour J2
* Que la case d’arrivée soit dans le plateau et libre.

## 7) Sortir un pion

La fonction **is\_exit\_move** dit si le pion est sur un bord qui permet de sortir : J1 peut sortir en allant à droite depuis la dernière colonne ; J2 peut sortir en allant vers le haut depuis la première ligne. La fonction move enlève le pion s’il sort, sinon elle le déplace sur la case voisine.

## 8) Un tour de jeu complet

La fonction **dodgem** gère la partie :

Affichage du plateau

Regarde si J1 ou J2 a gagné (tous ses pions sont sortis)

Si le joueur courant choisit un pion (**selectPawn**) et une direction (**selectMove**)

Vérifie que le coup est autorisé (**possibleMove**)

Applique le coup (**move**), puis c’est à l’autre joueur. répète jusqu’à la victoire.

## 9) Condition de victoire

La fonction **win** compte les pions d’un joueur. Si ce nombre est égal à 0, cela veut dire que tous ses pions sont sortis : il a gagné.

# Conclusion

En résumé : le programme construit la grille, place les pions de départ, demande au joueur quel pion bouger et dans quelle direction, vérifie les règles simples (pas de retour en arrière, bonne sortie, case libre), effectue le mouvement, et teste la victoire.