

Jeu Matrice — Résumé technique

Déplacement diagonal (grille $m \times n$)

Table des matières

1 Présentation du projet	2
2 Représentation de la matrice	2
2.1 Dimensions	2
2.2 Position de la pièce	2
3 Règles de déplacement	2
3.1 Conditions aux bords	2
4 Affichage : canvas et cases carrées	3
5 Entrées : ordinateur et tactile	3
5.1 Ordinateur	3
5.2 Tactile (téléphone)	3
6 Structure de l'application	3

1 Présentation du projet

Le **Jeu Matrice** est une application web (HTML, CSS, JavaScript) dans laquelle une pièce se déplace *uniquement en diagonale* sur une grille. L'utilisateur choisit la taille de la grille (m lignes $\times n$ colonnes, par défaut 6×6 , maximum 10×10), puis déplace la pièce soit avec des boutons (ordinateur), soit par un geste de glissement diagonal sur l'écran (téléphone).

2 Représentation de la matrice

2.1 Dimensions

La grille est une **matrice** de dimensions $m \times n$:

- m : nombre de **lignes** (indice i , de 0 à $m - 1$) ;
- n : nombre de **colonnes** (indice j , de 0 à $n - 1$) .

Contraintes : $1 \leq m \leq 10$ et $1 \leq n \leq 10$. Les valeurs sont saisies sur l'écran d'accueil et validées au clic sur *Jouer*. Si m ou n est hors de cet intervalle (ou non numérique), un message d'erreur s'affiche et la partie ne démarre pas.

2.2 Position de la pièce

La position est donnée par un couple d'indices (i, j) :

- i : numéro de **ligne** (0 = première ligne en haut) ;
- j : numéro de **colonne** (0 = première colonne à gauche).

Position initiale au démarrage d'une partie : la pièce est placée au **centre** de la grille :

$$(i_0, j_0) = (\lfloor m/2 \rfloor, \lfloor n/2 \rfloor).$$

3 Règles de déplacement

Les déplacements sont **uniquement diagonaux** : à chaque action, la pièce change à la fois de ligne et de colonne. Quatre directions sont possibles :

Code	Sens	Δi	Δj
G-H	Gauche-Haut	-1	-1
G-B	Gauche-Bas	+1	-1
D-H	Droite-Haut	-1	+1
D-B	Droite-Bas	+1	+1

Formules de mise à jour :

$$\begin{aligned} G-H &: (i, j) \mapsto (i - 1, j - 1) \\ G-B &: (i, j) \mapsto (i + 1, j - 1) \\ D-H &: (i, j) \mapsto (i - 1, j + 1) \\ D-B &: (i, j) \mapsto (i + 1, j + 1) \end{aligned}$$

3.1 Conditions aux bords

Un déplacement est **refusé** si la nouvelle position sort de la grille :

$$i_{\text{new}} \notin [0, m - 1] \quad \text{ou} \quad j_{\text{new}} \notin [0, n - 1].$$

Dans ce cas, la position (i, j) ne change pas et l'application signale un **débordement** (affichage en rouge, message « Hors limites ! », puis retour à l'affichage normal après un court délai).

4 Affichage : canvas et cases carrées

La grille est dessinée dans un élément `canvas` (HTML5). Pour que les cases soient **carrées**, le canvas est redimensionné en un carré de côté $c = \min(L, H)$, où L et H sont la largeur et la hauteur de la zone réservée à la grille. On pose ensuite :

$$\text{largeurCellule} = \frac{c}{n}, \quad \text{hauteurCellule} = \frac{c}{m}.$$

En JavaScript, `zoneJeu.width` et `zoneJeu.height` sont mis à c , puis les traits de la grille et le cercle représentant la pièce sont tracés à partir de (i, j) et de ces dimensions. La pièce est dessinée au centre de la case (i, j) ; en cas de débordement, elle est affichée en rouge pendant un court instant.

5 Entrées : ordinateur et tactile

5.1 Ordinateur

Quatre boutons (G-H, G-B, D-H, D-B) sont affichés sous la grille. Un clic sur l'un d'eux déclenche l'appel à la fonction `deplacerDiagonal(direction)` avec la direction correspondante. Un clic = un déplacement d'une case en diagonale (si le déplacement est autorisé).

5.2 Tactile (téléphone)

Sur écran tactile, le déplacement est déclenché par un **geste de glissement** (swipe) sur la zone du canvas.

1. **touchstart** : au doigt posé, on enregistre les coordonnées de départ (x_0, y_0) via `touches[0].clientX` et `touches[0].clientY`.
2. **touchend** : au doigt levé, on lit les coordonnées de fin (x_1, y_1) via `changedTouches[0]`.
3. On calcule $\Delta x = x_1 - x_0$ et $\Delta y = y_1 - y_0$.
4. Pour que le geste soit pris en compte :
 - $|\Delta x| \geq 35$ px et $|\Delta y| \geq 35$ px (seuil minimal);
 - le geste doit être suffisamment diagonal : $|\Delta x|$ et $|\Delta y|$ ne doivent pas être trop déséquilibrés (conditions du type $|\Delta x| \geq 0,5 |\Delta y|$ et $|\Delta y| \geq 0,5 |\Delta x|$).
5. La direction du déplacement est déduite des **signes** de Δx et Δy :

$\Delta x < 0, \Delta y < 0$	G-H (gauche-haut)
$\Delta x > 0, \Delta y < 0$	D-H (droite-haut)
$\Delta x < 0, \Delta y > 0$	G-B (gauche-bas)
$\Delta x > 0, \Delta y > 0$	D-B (droite-bas)

Un seul geste valide provoque un seul appel à `deplacerDiagonal`, donc un déplacement d'une case. `preventDefault()` sur les événements touch permet d'éviter le défilement de la page.

6 Structure de l'application

- **Écran d'accueil** : saisie de m et n , boutons Aide, À propos, Jouer. Validation de $m, n \in [1, 10]$ avant démarrage.
- **Overlays** : pages Aide (règles des 4 directions, usage boutons et tactile) et À propos (auteur).
- **Zone de jeu** : canvas (grille + pièce), affichage de la position (i, j) , message « Hors limites » si débordement, boutons G-H, G-B, D-H, D-B.

Le dessin est mis à jour après chaque déplacement (et après redimensionnement de la fenêtre). La grille et la pièce sont entièrement gérées en JavaScript (calcul des coordonnées à partir de (i, j) , `largeurCellule`, `hauteurCellule`).