

```
// Définition des formes des Tetriminos
const tetriminos = {
  I: [
    [
      [0, 0, 0, 0],
      [1, 1, 1, 1],
      [0, 0, 0, 0],
      [0, 0, 0, 0],
    ],
  ],
  O: [
    [
      [0, 1, 0, 0],
      [0, 1, 0, 0],
      [0, 1, 0, 0],
      [0, 1, 0, 0],
    ],
  ],
  T: [
    [
      [1, 1],
      [1, 1],
    ],
  ],
  L: [
    [
      [0, 1, 0],
      [1, 1, 1],
      [0, 0, 0],
    ],
    [
      [0, 1, 0],
      [1, 1, 0],
      [0, 1, 0],
    ],
    [
      [1, 1, 1],
      [0, 1, 0],
      [0, 0, 0],
    ],
    [
      [0, 1, 0],
      [0, 1, 1],
      [0, 1, 0],
    ],
  ],
  S: [
    [
      [0, 0, 1],
      [1, 1, 1],
      [0, 0, 0],
    ],
    [
      [0, 1, 0],
      [0, 1, 0],
    ],
  ],
  Z: [
    [
      [0, 1, 0],
      [0, 1, 0],
    ],
  ],
}
```

```
[0, 1, 1],
],
[
[1, 1, 1],
[1, 0, 0],
[0, 0, 0],
],
[
[1, 1, 0],
[0, 1, 0],
[0, 1, 0],
],
],
J: [
[
[1, 0, 0],
[1, 1, 1],
[0, 0, 0],
[0, 0, 0],
],
[
[0, 1, 1],
[0, 1, 0],
[0, 1, 0],
[0, 0, 0],
],
[
[1, 1, 1],
[0, 0, 1],
[0, 0, 0],
[0, 0, 0],
],
[
[0, 1, 0],
[0, 1, 0],
[1, 1, 0],
[0, 0, 0],
],
],
S: [
[
[0, 1, 1],
[1, 1, 0],
[0, 0, 0],
[0, 0, 0],
],
[
[1, 0, 0],
[1, 1, 0],
[0, 1, 0],
[0, 0, 0],
],
],
```

```

],
Z: [
  [
    [1, 1, 0],
    [0, 1, 1],
    [0, 0, 0],
    [0, 0, 0],
  ],
  [
    [0, 1, 0],
    [1, 1, 0],
    [1, 0, 0],
    [0, 0, 0],
  ],
],
];

// Initialisation de la grille et de la position
const nbLignes = 20;
const nbColonnes = 12;
let grille = Array.from({ length: nbLignes }, () =>
  Array(nbColonnes).fill(" ")
);
let posX = 1;
let posY = nbLignes - 2;
let rotationIndex = 0;

// Sélectionner un Tetrimino aléatoire
function choisirTetriminoAleatoire() {
  const formesDisponibles = Object.keys(tetriminos);
  const formeChoisie =
    formesDisponibles[Math.floor(Math.random() * formesDisponibles.length)];
  return tetriminos[formeChoisie];
}

// Initialisation du Tetrimino actuel avec un Tetrimino aléatoire
let currentTetrimino = choisirTetriminoAleatoire();

// Fonction pour afficher la grille
function afficherGrille() {
  const gridElement = document.getElementById("grid");
  gridElement.textContent = grille.map((row) => row.join(" ")).join("\n");
}

// Fonction pour afficher le Tetrimino dans la grille
function afficherTetrimino(x, y) {
  const forme = currentTetrimino[rotationIndex];
  for (let i = 0; i < forme.length; i++) {
    for (let j = 0; j < forme[i].length; j++) {
      if (forme[i][j] === 1) {
        grille[y + i][x + j] = "X";
      }
    }
  }
}

afficherGrille();

// Fonction pour tourner le Tetrimino
function tournerTetrimino() {

```

```

rotationIndex = (rotationIndex + 1) % currentTetrimino.length;
afficherGrille();
}

```

```

// Fonction de réinitialisation pour supprimer l'ancien Tetrimino
function effacerTetrimino(x, y) {
    const forme = currentTetrimino[rotationIndex];
    for (let i = 0; i < forme.length; i++) {
        for (let j = 0; j < forme[i].length; j++) {
            if (forme[i][j] === 1) {
                grille[y + i][x + j] = " ";
            }
        }
    }
}

```

```

// Gestion des touches pour déplacer et tourner
document.addEventListener("keydown", function (event) {
    effacerTetrimino(posX, posY); // Efface l'ancien Tetrimino de la grille

```

```

    if (event.key === "ArrowDown") {
        // Touche pour tourner
        tournerTetrimino();
    } else if (event.key === "ArrowLeft" && posX > 0) {
        // Déplacement à gauche
        posX--;
    } else if (event.key === "ArrowRight" && posX < nbColonnes - 4) {
        // Déplacement à droite
        posX++;
    } else if (event.key === "ArrowUp" && posY < nbLignes - 3) {
        // Descente
        posY--;
    }
}

```

```

    afficherTetrimino(posX, posY); // Affiche le Tetrimino à sa nouvelle position
});

```

```

// Fonction de jeu
function gameLoop() {
    effacerTetrimino(posX, posY);
    posY--; // Déplacement automatique vers le haut
    if (posY <= 0) {
        // Réinitialisation si le bloc atteint le haut
        posY = nbLignes - 2;
        currentTetrimino = choisirTetriminoAleatoire(); // Nouveau Tetrimino aléatoire
    }
    afficherTetrimino(posX, posY);

```

```

    setTimeout(gameLoop, 1000); // Intervalle de 1 seconde
}

```

```

// Démarre la boucle de jeu
gameLoop();

```