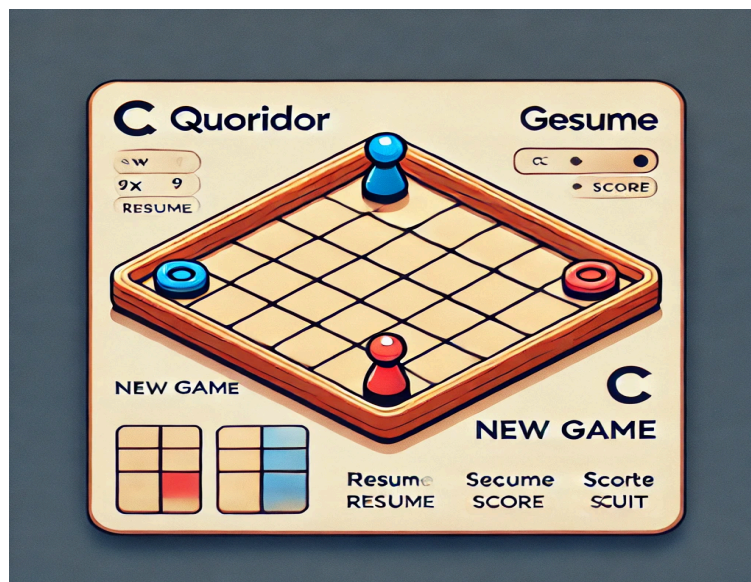


Le Quoridor

Règles: Le Quoridor est un jeu de stratégie où chaque joueur doit atteindre le côté opposé du plateau en déplaçant son pion ou en plaçant des barrières pour ralentir l'adversaire. Le premier à y parvenir gagne la partie ! 🎯

Actions possibles: Déplacer son pion ou poser des barrières pour gêner les adversaires (sans les bloquer complètement).

Exigences fonctionnelles prioritaires:



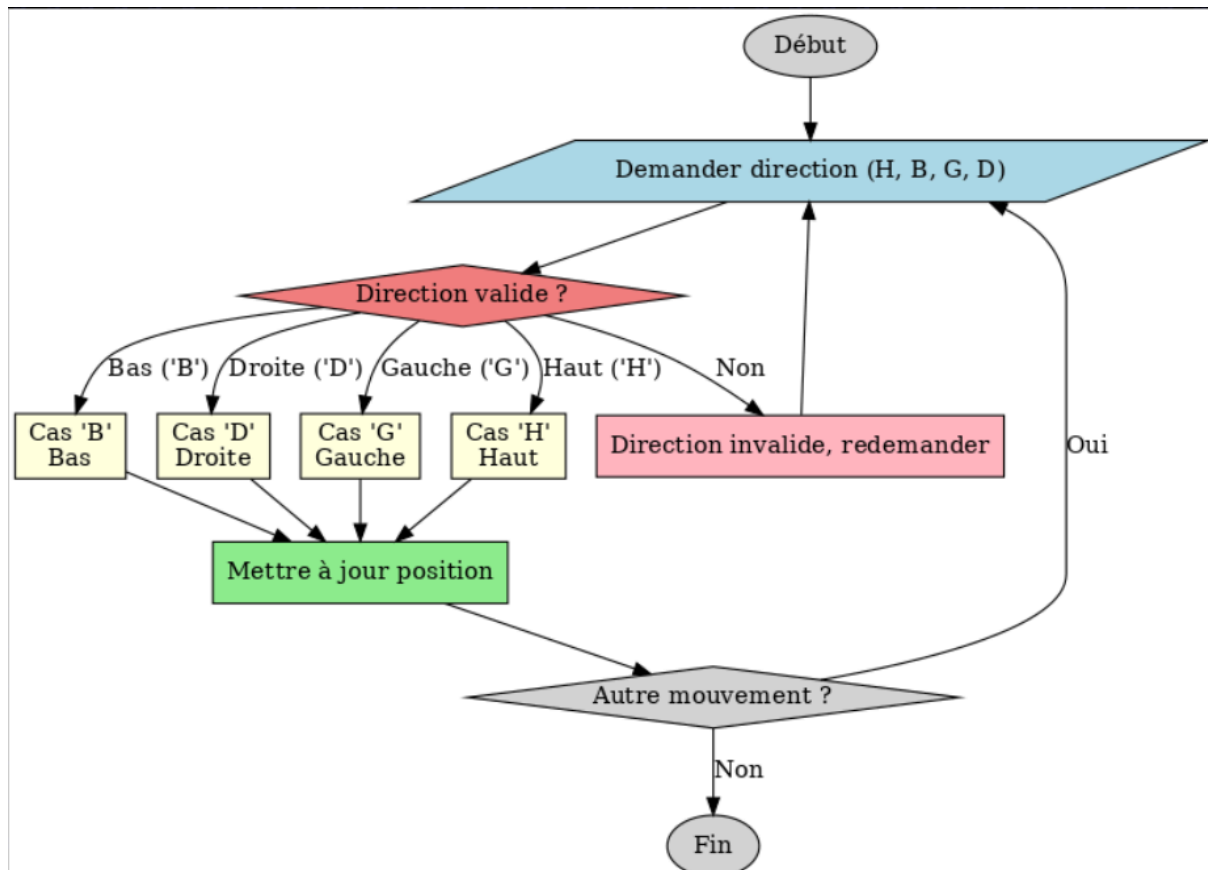
- Plateau : Générer une grille 9x9.
- Déplacement : Déplacer les pions selon les règles du jeu.
- Barrières : Placer des barrières en respectant les contraintes.
- Tours de jeu : Gérer l'ordre et les actions des joueurs.
- Victoire : Détecter le premier à atteindre le bord opposé.
- Scores : Sauvegarder et afficher les scores.
- Menu : Options : Nouvelle partie, reprise, scores, quitter.
- Sauvegarde : Permettre d'interrompre et reprendre une partie.

🌐 Objectif : Offrir une expérience de jeu fluide et conforme aux règles du Quoridor ! 🎯

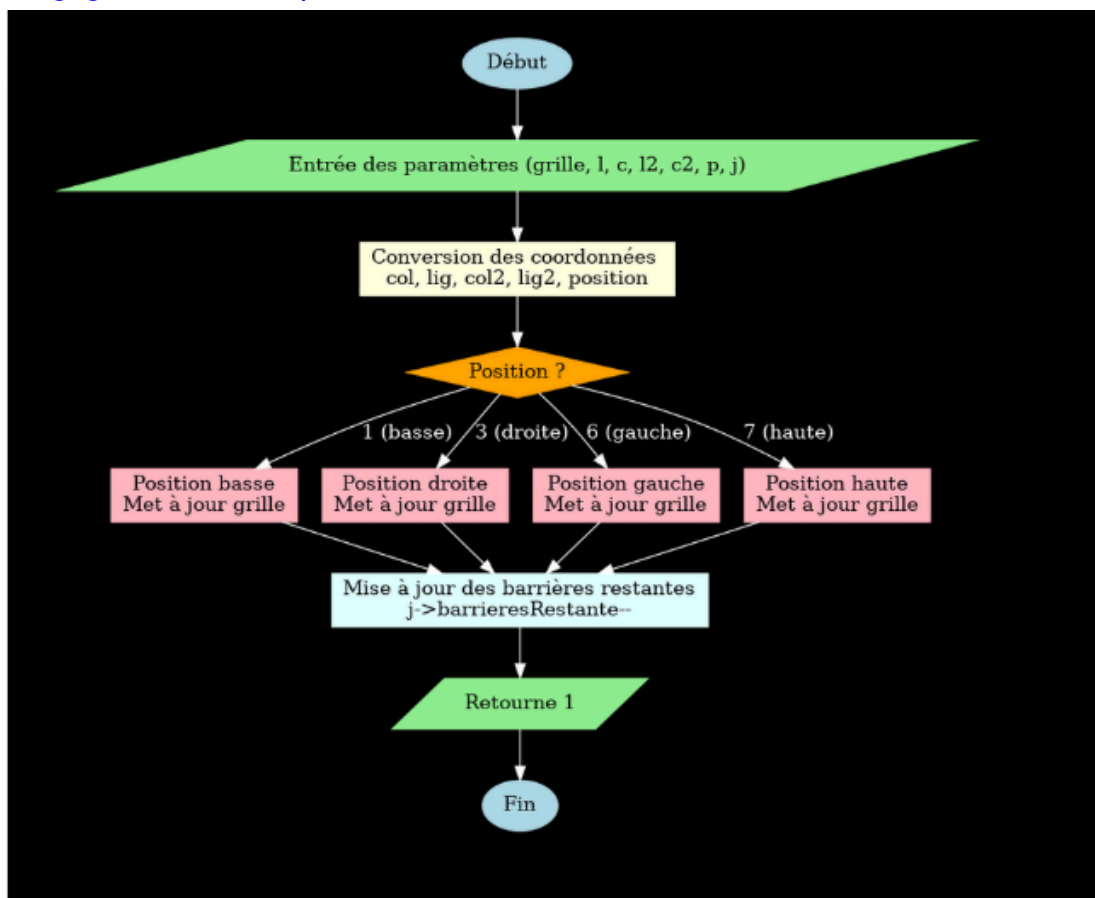
Déroulement de la partie :



Déplacement des joueurs :



Logigramme du placement des barrières



Quelques résultats visuels

Le menu :



Le plateau :

```
Le plateau a ete sauvegarde.
  A  B  C  D  E  F  G  H  I
1  .  .  .  .  .  .  .  .  .
2  .  .  .  .  .  .  .  .  .
3  .  .  .  2  .  .  .  .  .
4  .  .  .  .  .  .  .  .  .
5  .  .  .  .  .  1  .  .  .
6  .  .  .  .  .  .  .  .  .
7  .  .  .  .  .  .  .  .  .
8  .  .  .  .  .  .  .  .  .
9  .  .  .  .  .  .  .  .  .

tour du joueur 2 : WIAM
nom:WIAM
numero de joueur:2
jeton:2
nombre de barrieres restante:9
score:0

Que voulez vous faire?
tapez 1 pour se deplacer
tapez 2 pour mettre une barriere
tapez 3 passer son tour
tapez 4 pour se quitter
|
```

Les chiffres 1 et 2 représentent les joueurs, et les B les barrières déjà mises en place

Historique du score :

```

                                     SCORE
--- Scores des joueurs enregistrés ---
Joueur : AMINE:, Score : 0
Joueur : YACINE:, Score : 10
Joueur : A:, Score : 35
Joueur : AARON:, Score : 0
Joueur : WIAM:, Score : 0
appuyez sur un chiffre pour quitter:
```

Quelques bouts de code:

Menu :

```
do {
    afficherMenu();
    printf("Quel est votre choix ? : ");
    scanf("%d", &choix);

    switch (choix) {
        case 1:
            j = remplirJoueur(grille, 1, c, l2, c2, p, &n timer);
            initialisePlateau(grille);
            partie(j, grille, nbj);
            break;
        case 2:
            reprendrePartie();
            break;
        case 3:
            afficherAide();
            break;
        case 4:
            AfficherScores();
            break;
        case 5:
            quitterJeu();
            break;
        default:
            printf("Choix invalide. Veuillez reessayer.\n");
    }
} while (choix != 5);
```

Affichage du plateau de jeu:

```
void plateau(char grille[TAILLE][TAILLE]) {
    int c = 1; // Compteur pour les lignes
    printf("    "); // Alignement des colonnes

    // Boucle pour les colonnes
    for (char c = 'A'; c < 'A' + 9; c++) {
        printf("%c ", c);
    }
    printf("\n");

    // Boucle pour les lignes
    for (int i = 0; i < TAILLE; i++) {
        if (i % 2 == 0) {
            printf("%d ", c);
            c++;
        } else {
            printf(" "); // Espacement pour aligner
        }
        for (int j = 0; j < TAILLE; j++) {
            printf("%c ", grille[i][j]);
        }
        printf("\n");
    }
}
```

Gestion du curseur:

```
// Déplace le curseur à une position précise dans la console
void gotoligcol(int lig, int col) {
    COORD mycoord;
    mycoord.X = col;
    mycoord.Y = lig;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), mycoord);
}
```

Déplacer un joueur:

```
void deplacerJoueur(joueur *j, char direction, char grille[TAILLE][TAILLE]) {
    switch (direction) {
        case 'z': // Haut
            if (j->positionX > 0) j->positionX--;
            break;
        case 's': // Bas
            if (j->positionX < TAILLE - 1) j->positionX++;
            break;
        case 'q': // Gauche
            if (j->positionY > 0) j->positionY--;
            break;
        case 'd': // Droite
            if (j->positionY < TAILLE - 1) j->positionY++;
            break;
        default:
            printf("Direction invalide !\n");
    }
}
```

Placement barrières:

```
void creerBarriere(char *l, int *c, char *l2, int *c2, char *p, char grille[TAILLE][TAILLE], joueur *j) {
    if (j->barrieresRestante > 0) { // Vérifie si le joueur a encore des barrières
        printf("Vous n'avez plus de barrière.");
    } else {
        // Code pour placer la barrière en fonction des coordonnées et direction
    }
}
```