

Nom : NANA KAMDOUN
Prenom : JIRES JOSPIN
Matricule : 23V2009

UNIVERSITE DEYAOUNDE 1
sous la supervision de
DR MESSI

TP : Creation du Ludo en console et interface graphique

I) - LE CONTEXTE

Objectif du Jeu

L'objectif est d'être le premier joueur à faire parcourir à ses quatre pions le circuit complet du plateau et à les amener dans sa "maison" (zone d'arrivée). Les pions se déplacent selon le résultat obtenu au dé, et les joueurs peuvent capturer les pions adverses en atterrissant sur la même case.

Éléments du Jeu

- **Un plateau de jeu** : En forme de croix avec un parcours coloré
- **16 pions** : 4 pions de chaque couleur (généralement rouge, vert, jaune et bleu)
- **Un dé** : Pour déterminer les déplacements
- **Zones de départ** : Une pour chaque joueur
- **Zones d'arrivée** : Les "maisons" où les pions doivent terminer leur parcours

Points Clés

- Un 6 est nécessaire pour sortir un pion de sa zone de départ
- Les pions se déplacent dans le sens des aiguilles d'une montre
- Les pions peuvent être capturés et renvoyés à leur zone de départ
- Stratégie et chance sont équilibrées
- Le jeu favorise l'interaction sociale et la compétition amicale

II) PROBLEME A RESOURDRE AVEC SOLUTION

1. Problème d'affichage du plateau

Problème actuel : Le plateau n'est pas entièrement visible dans le terminal.

Solution proposée :

- Ajuster le décalage vertical (DECALAGE_Y_na) dans la fonction dessiner_forme()
- Modifier les coordonnées pour centrer le plateau
- Implémenter une gestion dynamique de la taille du terminal

```
const int DECALAGE_Y_na = -7;  
  
// Ajuster cette valeur pour remonter le plateau
```

2. Amélioration du son

Problème actuel : La fonction son n'est pas optimale sous Ubuntu.

Solution proposée :

- Utiliser la bibliothèque SDL pour le son
- Implémenter une alternative avec beep système

```
// Alternative pour le son sous Ubuntu  
  
void emettre_son_na(int frequence_na, int duree_na) {  
    printf("\a"); // Beep système  
    fflush(stdout);  
    usleep(duree * 1000);  
}
```

3. Gestion du terminal

Problème actuel : Problèmes de compatibilité avec différentes tailles de terminal.

Solution proposée :

```
void adapter_affichage_na() {  
  
    struct winsize w;  
  
    ioctl(STDOUT_FILENO, TIOCGWINSZ, &w);  
  
    // Calculer l'échelle optimale  
  
    float scale_x_na = (float)w.ws_col_na / LARGEUR_MIN_na;  
    float scale_y_na = (float)w.ws_row_na / HAUTEUR_MIN_na;  
    float scale_na = (scale_x_na < scale_y_na) ? scale_x_na : scale_y_na;  
  
    if (scale_na < 1.0) {  
        // Redimensionner automatiquement  
    }  
}
```

5. Interface utilisateur

Problème actuel : Navigation dans les menus peu intuitive.

Solution proposée :

- Simplifier la navigation avec des touches plus intuitives

III) LA FONCTION PRINCIPALE DU JEU LUDO

1. Fonction Principale : jouer()

C'est la fonction centrale qui gère le déroulement du jeu :

```
void jouer_na() {  
    // Boucle principale du jeu  
    do {  
        system("clear");  
        dessiner_forme_na();  
        touche_na = getchar();  
  
        if(touche_na == 'o' || touche_na == 'O') {  
            // Lancer le dé  
            valeur_de_na = rand() % 6 + 1;  
  
            // Gérer le déplacement des pions  
            // Vérifier les collisions  
            // Mettre à jour le tour des joueurs  
        }  
    } while(touche != 'q' || touche != 'Q');  
}
```

Responsabilités principales :

- Gestion des tours de jeu
- Lancement du dé
- Sélection et déplacement des pions

- Contrôle des règles du jeu

2. Structure du Jeu

Hiérarchie des fonctions principales :

```
main_na()
├─ menu_na()
│   └─ sous_menu_na()
│       └─ jouer_na()
│           ├── dessiner_forme_na()
│           ├── deplacer_jetons_na()
│           ├── tuer_jetons_na()
│           └─ lancer_de_6_na()
```

3. Composants Essentiels

- **Structure de données principale :**

```
typedef struct Jetons_na{
    int x;
    int y;
} Jetons_na;
```

- **Variables globales importantes :**

```
Jetons jetons[4][4];    // Positions des pions
char noms[4][25];       // Noms des joueurs
int nb_joueurs;         // Nombre de joueurs
int tour_joueur;        // Tour actuel
```

4. Flux de Contrôle

1. Initialisation du jeu (chargement_na())
2. Affichage du menu principal (menu_na())
3. Configuration de la partie (sous_menu_na())

- Sélection du nombre de joueurs
 - Saisie des noms
 - Choix des caractères
4. Boucle de jeu principale (jouer_na())
- Lancer le dé
 - Déplacer les pions
 - Gérer les interactions
 - Vérifier les conditions de victoire

IV)- DESCRIPTION DES STRUCTURES DE DONNÉES ET VARIABLES DU JEU LUDO

1. Structure Principale

```
typedef struct Jetons_na {
    int x; // Coordonnée horizontale
    int y; // Coordonnée verticale
} Jetons_na;
```

Cette structure fondamentale gère le positionnement des éléments sur le plateau de jeu.

2. Tableaux et Matrices

- `Jetons_na jetons_na[4][4]` : Stocke les positions des jetons pour 4 joueurs (4 jetons chacun)
- `Jetons_na des[4]` : Positions des dés pour chaque joueur
- `Jetons_na jetons_na_temp[4][4]` : Sauvegarde temporaire des positions des jetons

3. Variables de Contrôle du Jeu

| Variable | Type | Description |
|------------|------|--------------------------|
| drapeau_na | bool | Indicateur d'état du jeu |

| | | |
|---------------|------|----------------------------------|
| son_na | bool | Activation/désactivation du son |
| nb_joueurs_na | int | Nombre de joueurs dans la partie |

4. Constantes de Configuration

- HAUTEUR_MIN_na : 42 (hauteur minimale du plateau)
- LARGEUR_MIN_na : 60 (largeur minimale du plateau)

5. Variables de Position et d'État

- Variables de fin de jeu (coordonnées finales):
 - fin_j1_y_na, fin_j2_x_na, fin_j3_x_na, fin_j4_y_na
- Tableau des gagnants: gagnants_na[4]
- Variables de tour de jeu:
 - tour_joueur_na
 - valeur_de_na
 - temp_de_na

6. Caractères et Noms

```
char noms_na[4][25] = {"Joueur1", "Joueur2", "Joueur3", "Joueur4"};
char caracteres[4] = {254, 232, 234, 224};
```

Ces tableaux gèrent l'affichage et l'identification des joueurs dans le jeu.