



Casse-tête des huit dames

RAPPORT DU TP8

GHAMRI Samy
Groupe TP2

GUYOLLOT Alan
Groupe TP2

Du 7 Avril au 14 Avril 2020

Chargé de TP : Monsieur Giraudo

Matière : Perfectionnement à la programmation en C

Etablissement / Formation : Université Gustave Eiffel – Licence 2 Maths-info Semestre 4

Introduction

L'objectif de ce TP est d'implanter le casse-tête des huit dames. Nous utilisons et approfondissons ici les notions de

- découpage d'un projet en modules ;
- interface NCURSES ;
- opérateurs bit à bit.

L'objectif est de réaliser une interface graphique permettant à un utilisateur de placer à la souris (ou au clavier) des dames sur un échiquier. Le programme signale si le placement est gagnant ou non. Pour mener tout ceci à bien, il sera question — bien que ceci n'est pas évident de prime abord — de manipulation de bits et d'emploi d'opérateurs bit à bit. Nous verrons que représenter une situation de jeu par des suites de bits et opérer dessus à l'aide d'opérateurs bit à bit permet de gagner en efficacité.

Corps du rapport

Exercice 1 :

1. Créer un entier tel que son 15e bit est à 1 et les autres sont à 0.

$(1 \ll 14)$

2. Créer un entier tel que son 14e bit est à 0 et les autres sont à 1.

$\sim(1 \ll 13)$

3. Mettre le 13e bit d'un entier x à 1.

$x = x | (1 \ll 12)$

4. Mettre le 12e bit d'un entier x à 0.

$x = x \& \sim(1 \ll 11)$

5. Tester si le 11e bit d'un entier x est un 1.

$x \& (1 \ll 10)$

6. Tester si les 3 premiers bits d'un entier x sont tous 1.

$(x \& 7) == 7$

7. Tester si les 4 premiers bits d'un entier x sont tous 0.

$(x \& 15) == 0$

8. Tester l'égalité des 10e bits de deux entiers x et y.

$(x \& (1 \ll 9)) \& (y \& (1 \ll 9))$

9. Tester si deux entiers x et y possèdent au moins un bit à 1 situé en un même indice.

$x \& y$

10. Tester si tous les bits des deux entiers x et y qui sont en mêmes indices sont égaux.

$x == y$

11. Tester si tous les bits des deux entiers x et y qui sont en mêmes indices sont différents.

$x == \sim y$

Modularisation du projet

1. Types :

- Graphique (fichier contenant toutes les fonctions d'affichage)
- Dame (fichier contenant toutes les fonctions concernant les dames)
- Structures (fichier contenant toutes les structures concernant la grille du jeu)
- Include (contient tous les includes nécessaires de base)

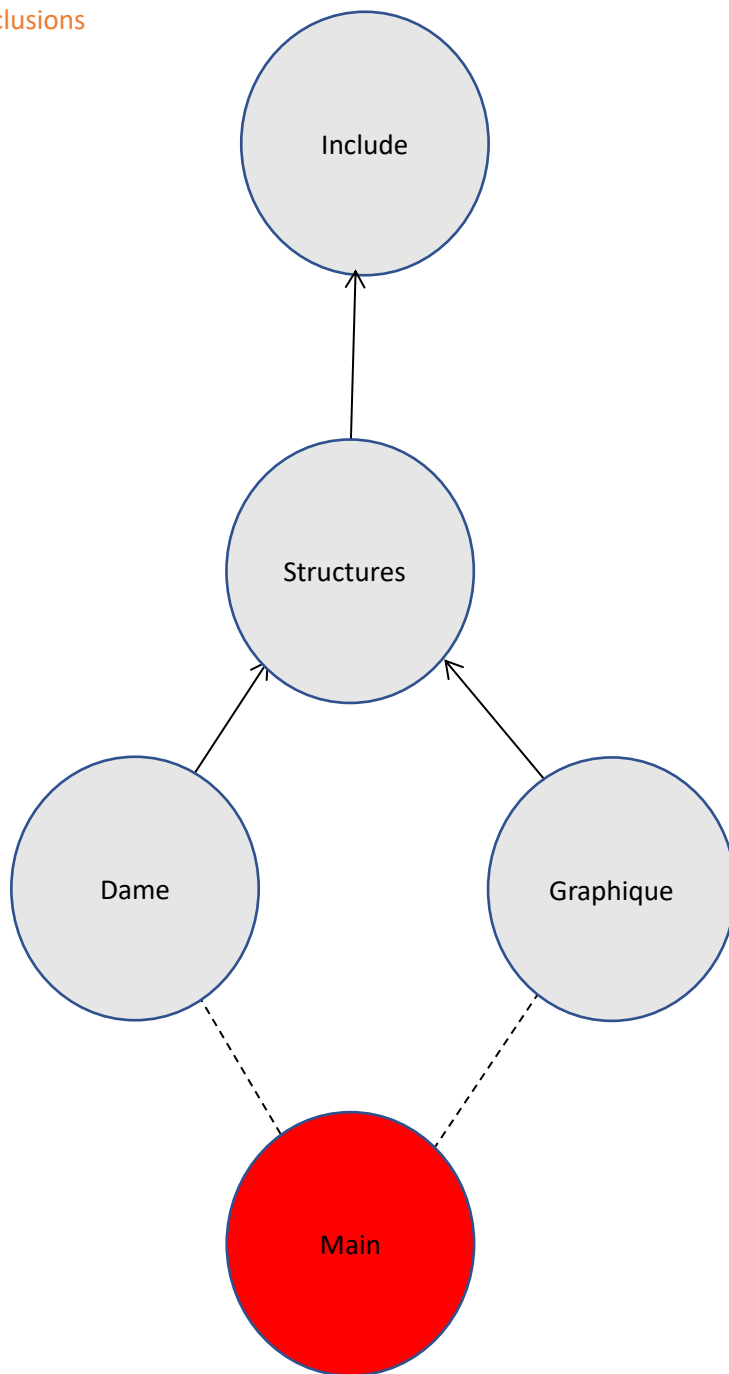
2. Module :

- Création d'un module pour chaque type considéré

3. Liste :

- Lister les fonctions principales à mettre en œuvre et les distribuer selon les modules précédents .
 - Dans le module **Graphique**
 - Déclarer et définir les fonctions
 - afficher_ligne_position_pair
 - afficher_ligne_position_impair
 - affiche_grille_position
 - affiche_demande_verification
 - affiche_resultat_verification
 - affiche_victoire
 - Dans le module **Dame**
 - Déclarer et définir les fonctions
 - est_case_occupee
 - placer_dame_position
 - afficher_ligne
 - afficher_position
 - calculer_cases_attaques
 - init_tab_cases_attaques
 - est_sans_attaques_mutuelle
 - init_couple
 - retirer_dame_position
 - Dans le module **Structures**
 - Déclaration de la structure Case et position
 - Dans le module **Include**
 - Contient les includes de base

Graphe des inclusions



Conclusion

Grâce à ce TP, nous avons pu réutiliser de nouveau des **assert()** qui sont très pratiques pour orienter les choix de l'utilisateur et éviter les erreurs du programme et modulariser notre projet afin de le découper en plusieurs petites parties et ainsi faciliter son entretien. Nous avons pu aussi réutiliser l'interface NCURSES pour faire l'affichage de cette application, découvrir la manipulation de bits et l'utilisation des opérateurs bits à bits.

La seule difficulté rencontrée fut de savoir comment manipuler les bits au début.

Aucune amélioration effectuée

Annexe 1

Instructions pour compiler le projet

Pour compiler le projet modularisé

*Il suffit d'utiliser la commande **make***

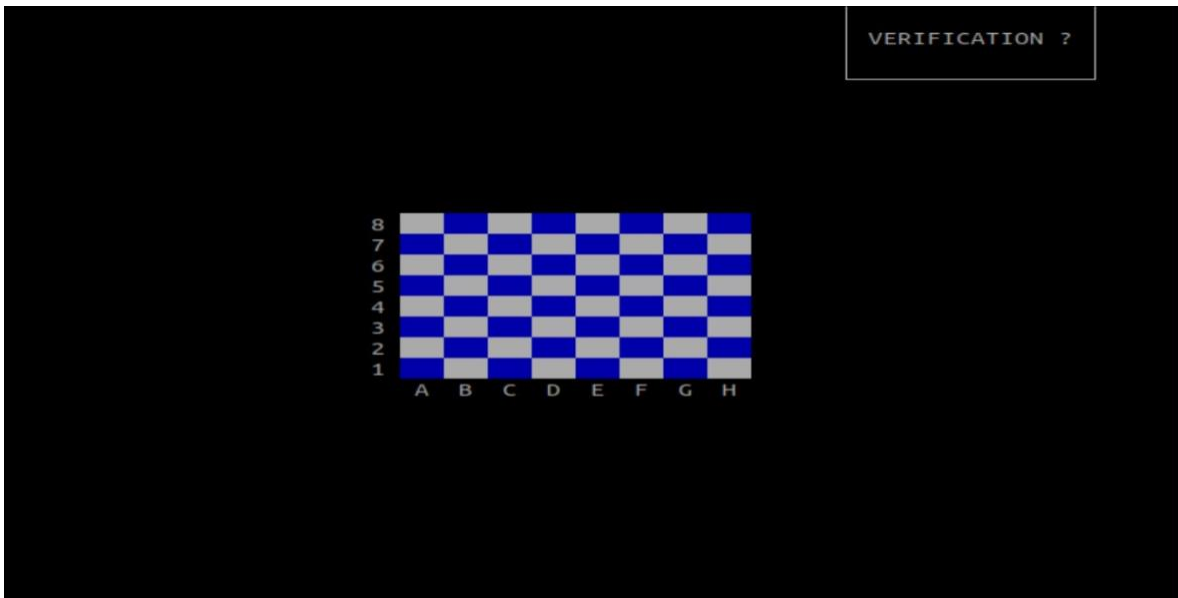
Annexe 2

Documentation du projet

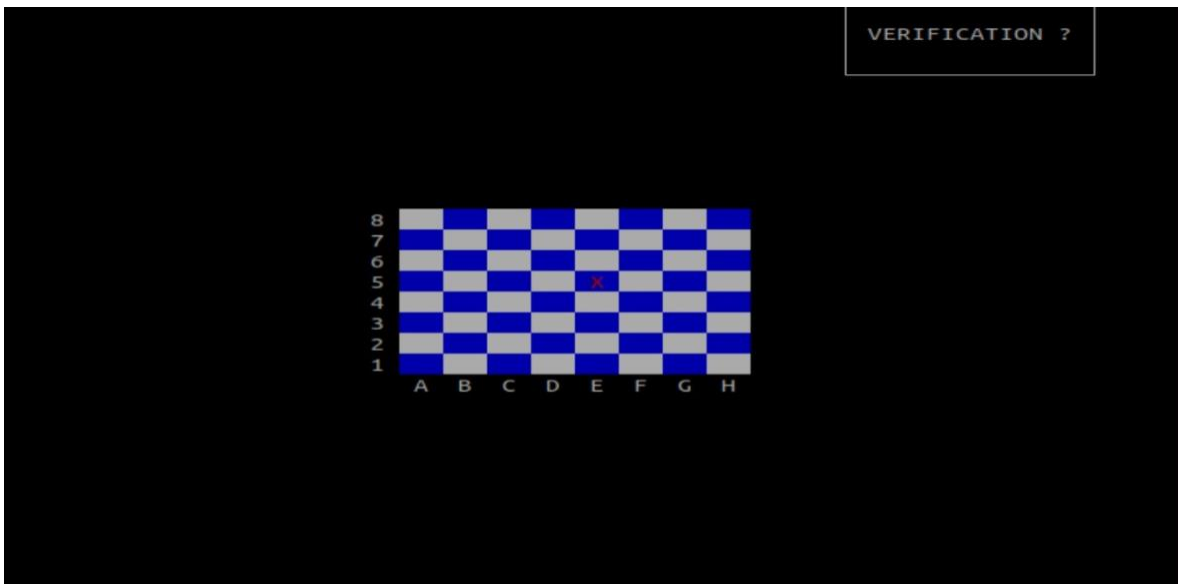
Vous devez être sous Linux

Tout d'abord, il faut télécharger les fichiers du projet et se déplacer où sont situés ces fichiers. Ensuite, vous ouvrez le terminal à partir de ce fichier (ou vous utilisez la commande **cd** dans le terminal pour vous déplacer dans le dossier où vous avez tout téléchargé). Puis vous allez utiliser la commande **make** dans le terminal et pour cela vous avez juste à écrire **make** puis appuyer sur la touche **Entrée**, cela va créer le fichier exécutable. Pour lancer le jeu vous devez écrire **./Dame** dans le terminal et appuyer sur la touche **Entrée**.

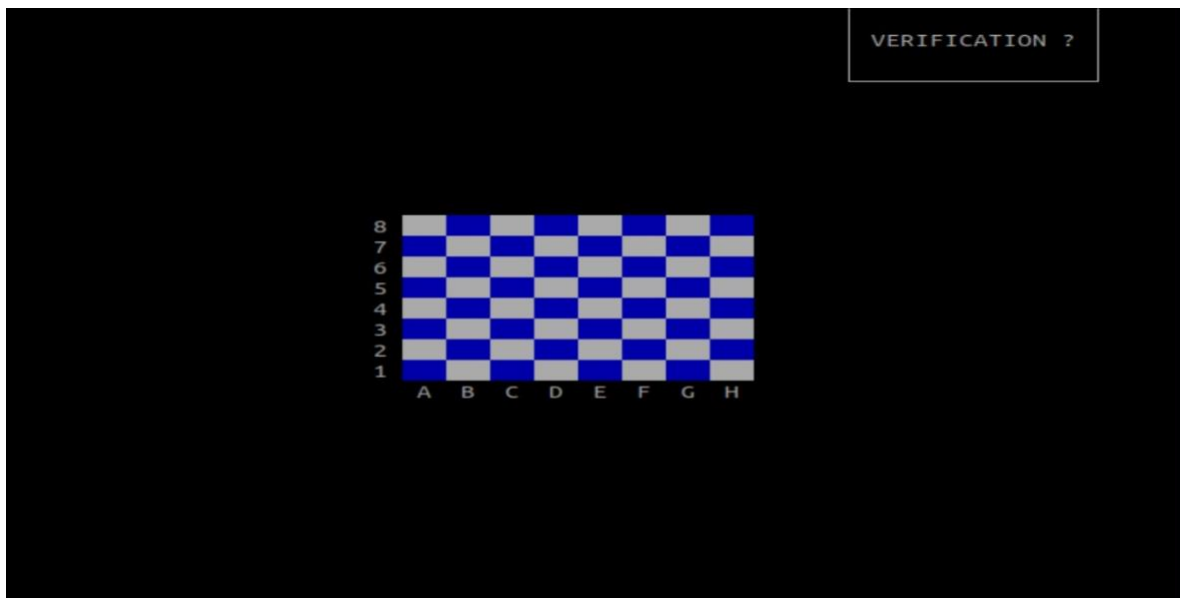
Cette image s'affiche :



Si on veut placer une **dame** sur le quadrillage, il suffit de cliquer sur une case du quadrillage. Exemple : on veut ajouter une dame sur la case E5 alors on clique sur la case E5. (Une dame est représenté par **X**).

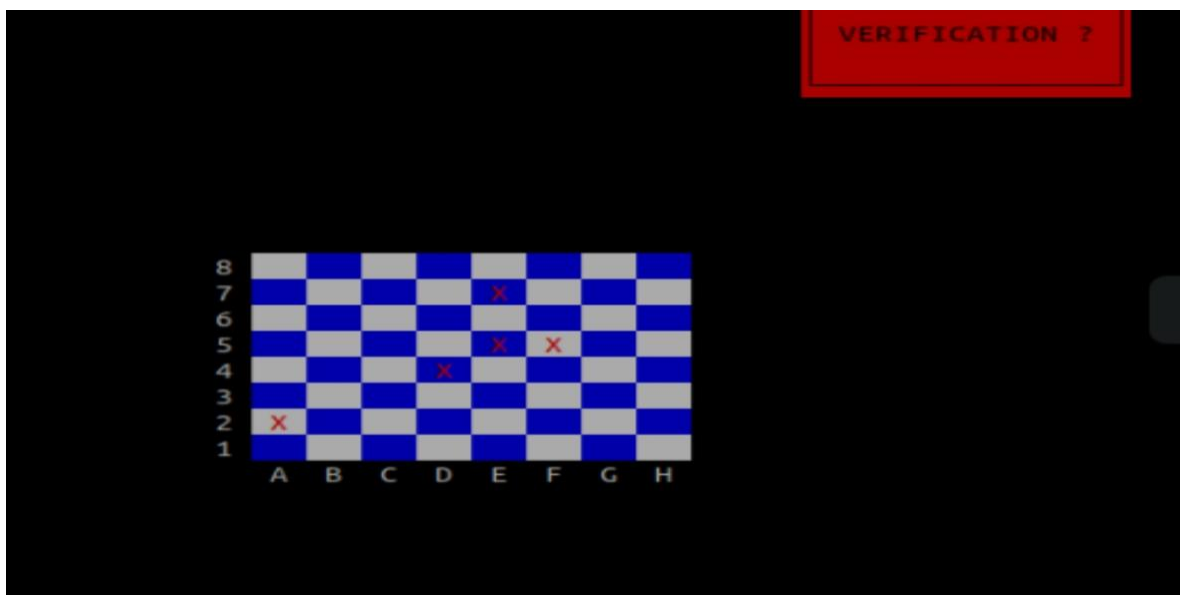


Si on veut enlever la dame, il suffit de cliquer sur la dame qu'on veut enlever. Enlevons la dame située sur la case E5.

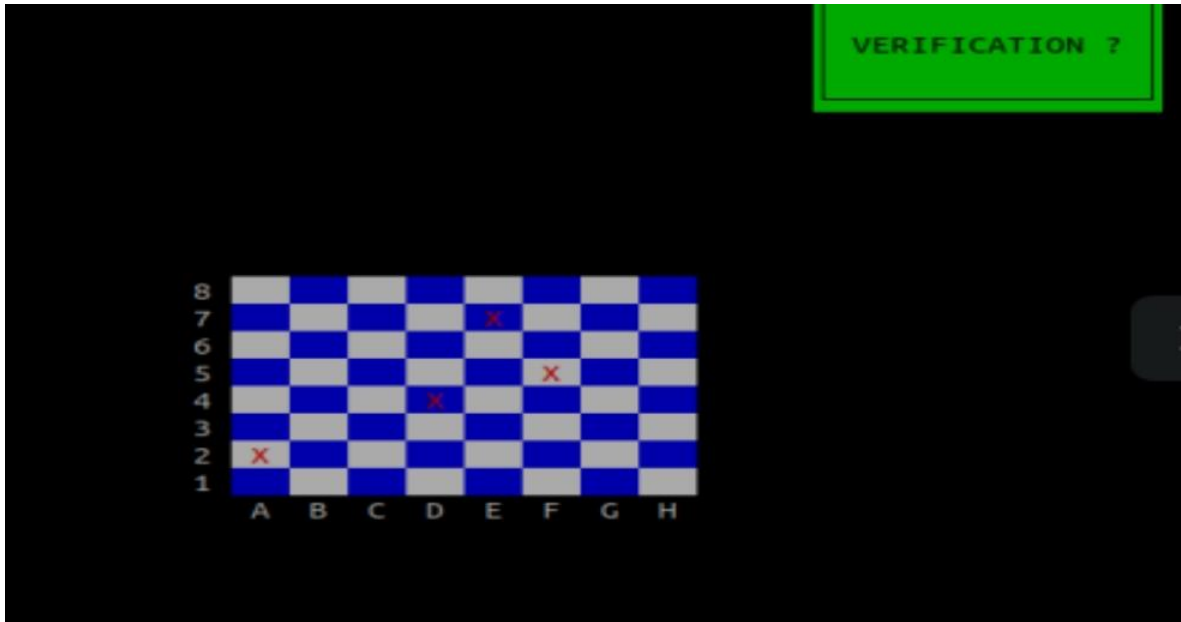


Si on veut savoir si 2 dames ou plus s'attaquent entre elles, il faut appuyer sur le bouton « VERIFICATION ? »

S'il est surligné en rouge, cela signifie qu'au moins deux dames sont en conflit.



S'il se surligne en **vert**, cela signifie qu'aucune dame n'est en confl



Le **but du jeu** est de placer 8 dames qui ne s'attaquent pas entre elles. Si vous arrivez à les placer correctement, et ensuite appuyez sur le bouton « VERIFICATION ? » alors cet écran s'affichera :

