

Programmation C TP 13

L'objectif de ce TP est de programmer le problème des 8 dames avec interface graphique utilisant la LibMLV.

Un célèbre problème d'échecs consiste à placer 8 dames sur un échiquier sans qu'aucune dame n'en menace une autre. Un échiquier est un quadrillage de taille 8x8 et une dame menace toutes les cases situées sur la même colonne, ligne ou diagonales.

Pour établir la liste des cases menacées, nous allons utiliser des masques binaires. L'échiquier est composé de 64 cases et toute case est soit menacée, soit libre. Au final, 64 booléens sont suffisants pour enregistrer les cases menacées. Dans le but d'utiliser la structure de données la plus fine et la plus légère en ressources, nous allons commencer par numéroté les cases de l'échiquier comme il suit:

63	62	61	60	59	58	57	56
55	54	53	52	51	50	49	48
47	46	45	44	43	42	41	40
39	38	37	36	35	34	33	32
31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0

Ensuite, le type C représentant un vecteur de 64 booléens le plus léger est l'entier long non signé. En effet, un `unsigned long int` est un nombre positif sur 8 octets (`sizeof(unsigned long int) = 8`). Comme chaque octet est composé de 8 bits, on aura bien 64 bits soit 64 booléens pour notre plateau.

Cette structure de données légère mais difficile à manipuler est cruciale dans les intelligences artificielles jouant aux échecs. La mémoire vive est toujours limitée même sur les serveurs de calculs. Ainsi les programmes fins jouant aux échecs utilisent des types similaires, on parle de bitboard (<http://en.wikipedia.org/wiki/Bitboard>). L'efficacité n'est pas cruciale pour ce TP dans lequel on ne calculera pas 5 coups d'avance. Toutefois, ces bitboards de 64 bits sont l'occasion de se familiariser avec les opérations bit à bit en C.

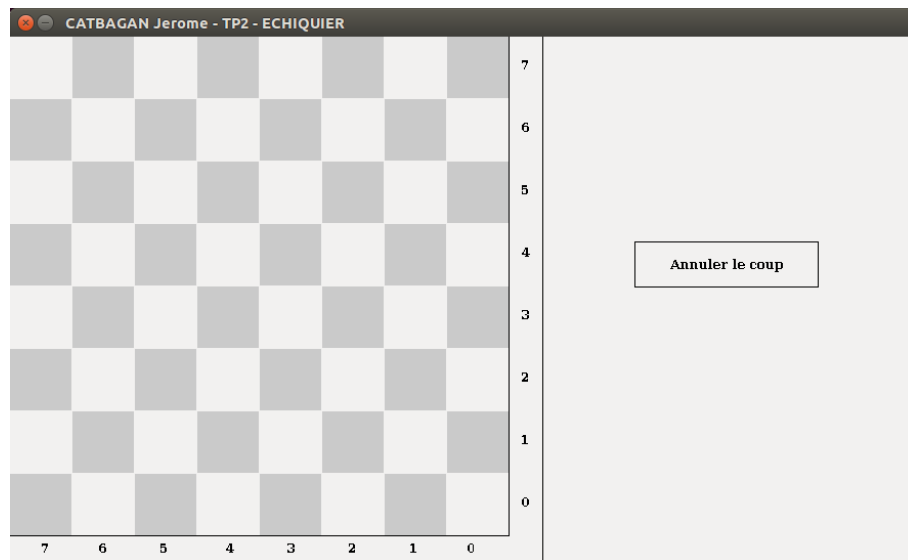
Exercice préliminaire :

- Écrire une fonction `int bit_value_ULI(unsigned long int n, int position)` qui retourne 1 si le bit numéro `position` vaut 1 dans `n` et retourne 0 sinon.
- Écrire une fonction `void print_ULI(unsigned long int n)` qui affiche les 64 bits de l'entier long `n`.
- Écrire une fonction `set_positive_bit_ULI(unsigned long int* n, int position)` qui met à jour la valeur pointée par `n` tel que le bit numéro `position` soit à 1.

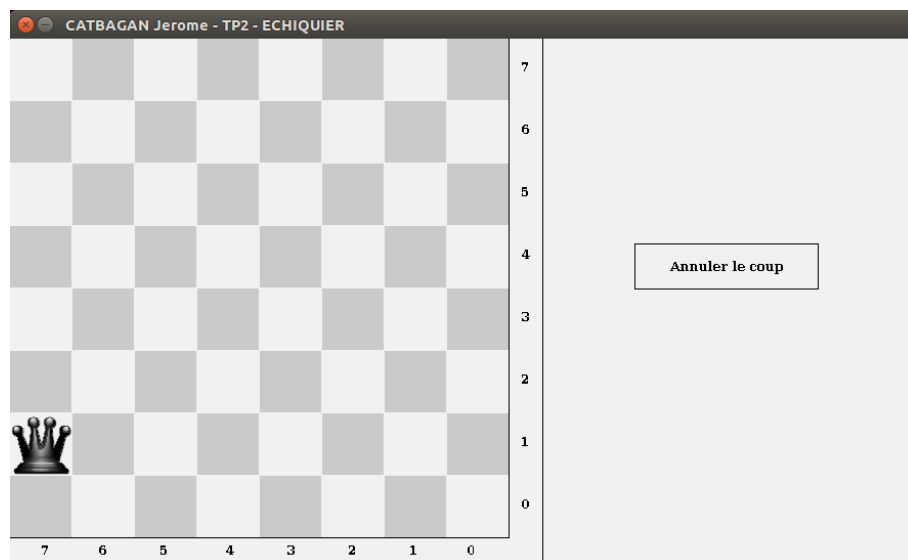
- Écrire une fonction `set_negative_bit_ULI(unsigned long int* n, int position)` qui met à jour la valeur pointée par `n` tel que le bit numéro `position` soit à 0.

Maintenant que vous avez réussi à jouer avec les bits d'un entier long, essayez de proposer un jeu graphique pour le problème des huit dames. Pour fixer les idées, voici une manière de proposer le jeu graphiquement.

Tout d'abord le plateau de jeu vide :

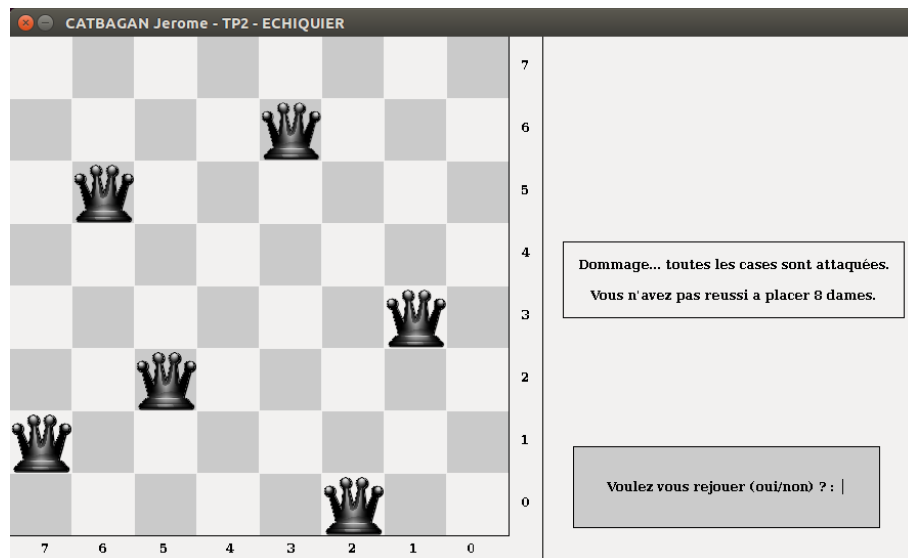


En cliquant sur une case non menacée, l'utilisateur peut placer une dame. Que ce soit en changeant la couleur de la case ou bien ici en y plaçant une image, on obtient alors:



Deux fins sont possibles pour ce jeu :

Soit l'utilisateur arrive à placer 8 dames et alors il a gagné lorsqu'il place la huitième. Soit il arrive dans une position où il n'a pas encore tout placé mais pourtant toutes les cases sont déjà menacées, il a alors perdu la partie. C'est ce qui est proposé dans l'image qui suit.



Suggestions d'amélioration :

- Rajouter la possibilité d'annuler le dernier coup (ou plus si vous le souhaitez...).
- Rajouter un compteur de temps.
- Rajouter du son.
- Glisser des animations et des couleurs.