

# COMMENT ON A CREEER NOTRE JEU ?

RÉALISER PAR :

CLÉMENT FURNON

YOUNESS ZAITOUNI

ANASS SEBAI

MINA MOUALY



# PRÉSENTATION



PRÉSENTATION DU JEU



RÈGLES DU JEU



STRUCTURE DE BASE



IMPLÉMENTANT DU JEU



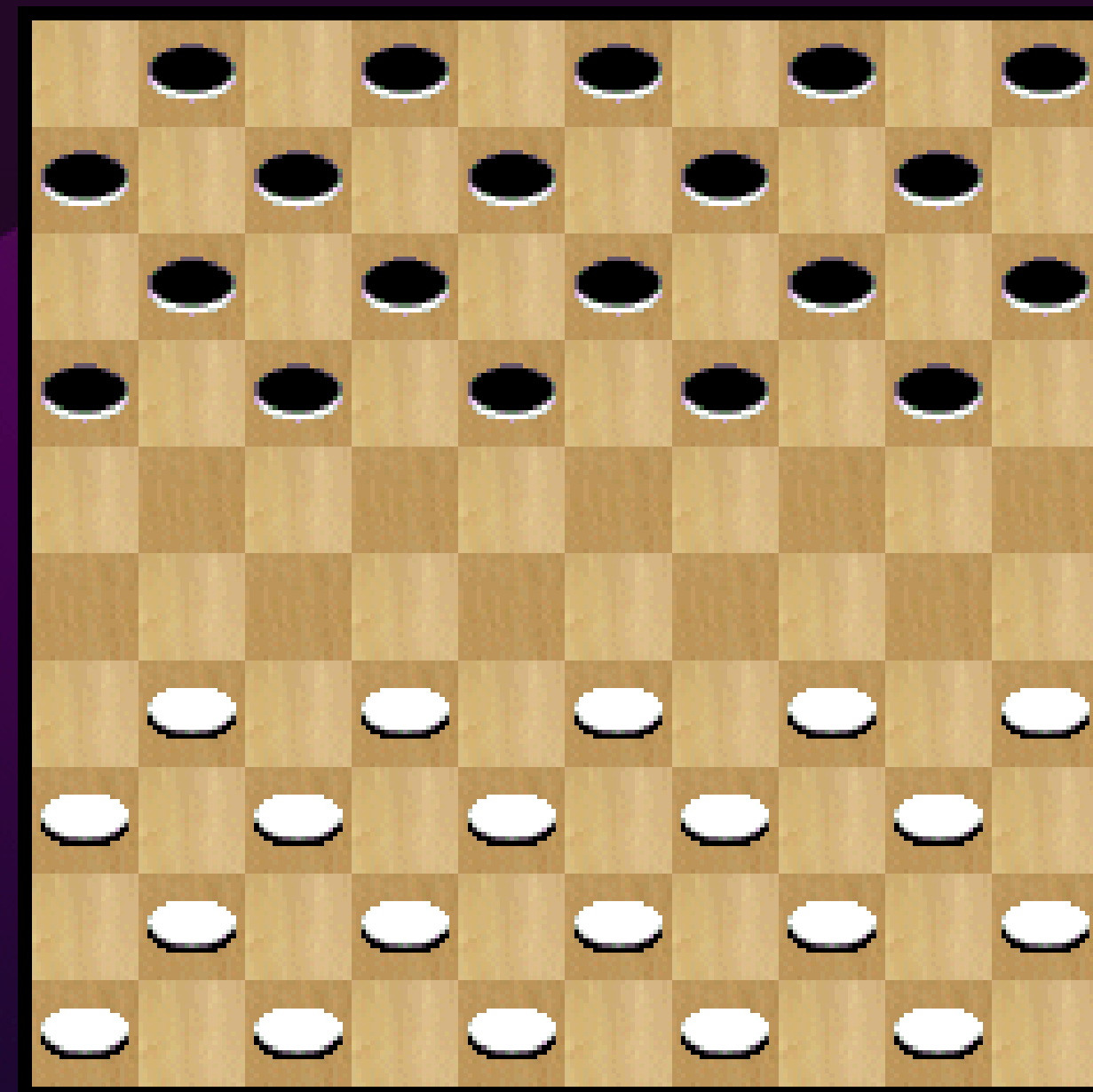
IMPLÉMENTATION DE L'IA



INTERFACE GRAPHIQUE



# PRÉSENTATION DU JEU

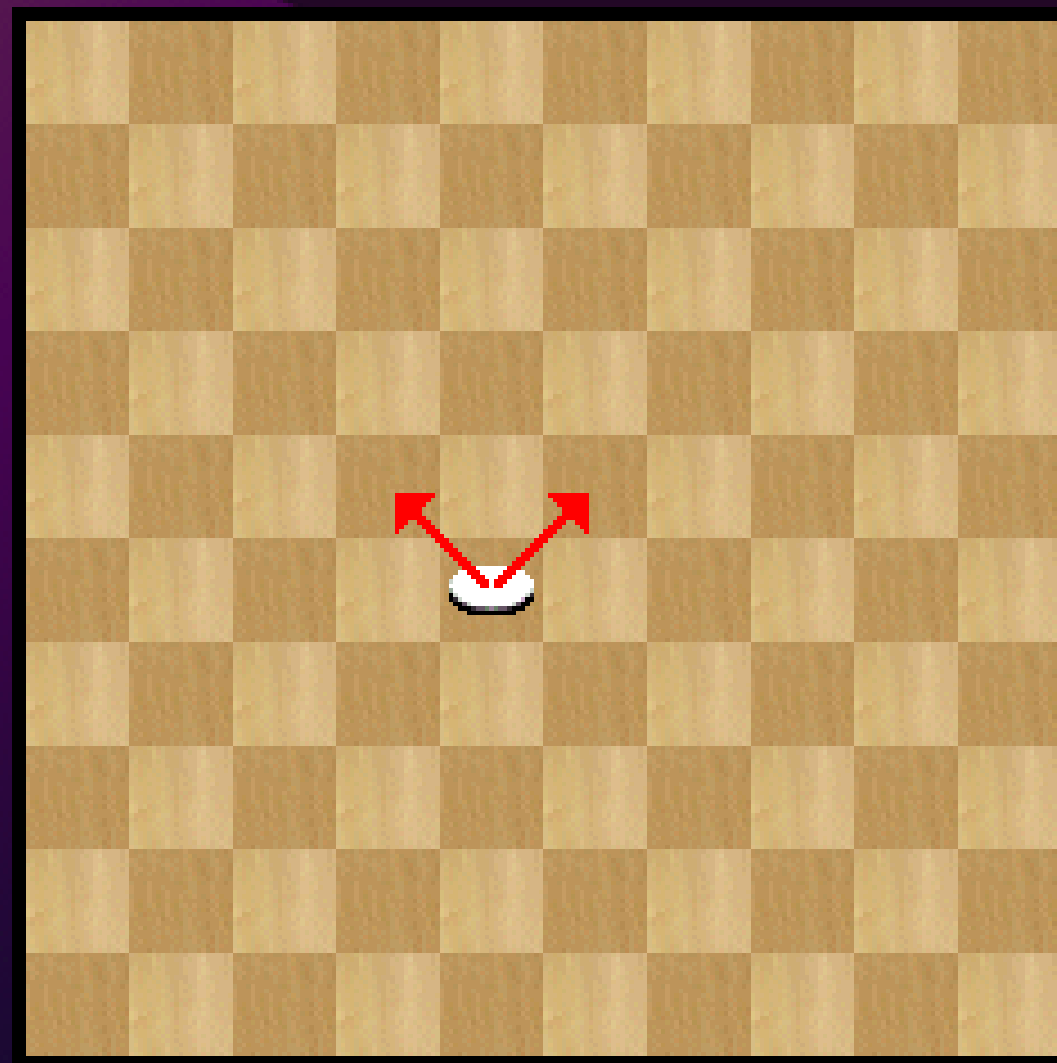


JEU DE DEMES

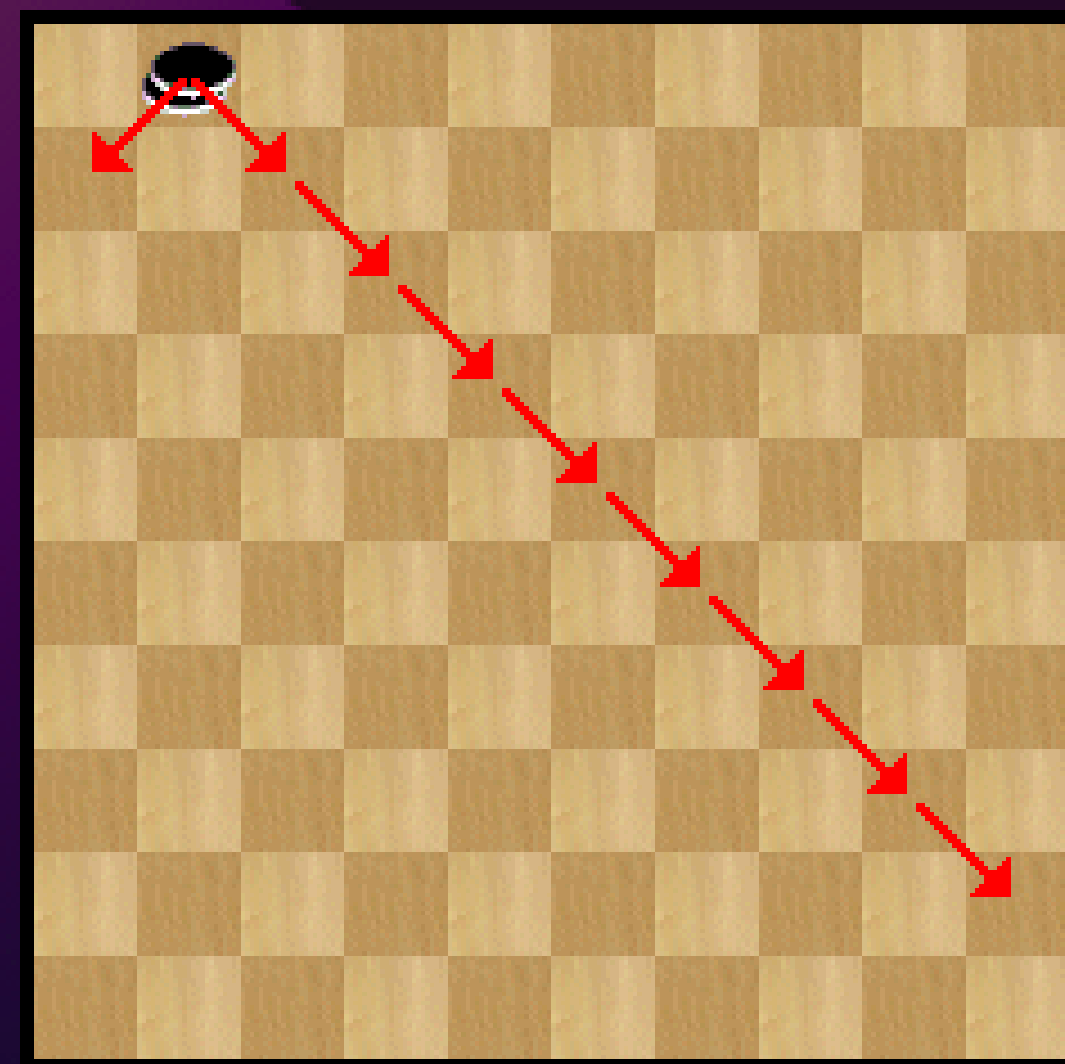


# REGLE DU JEU

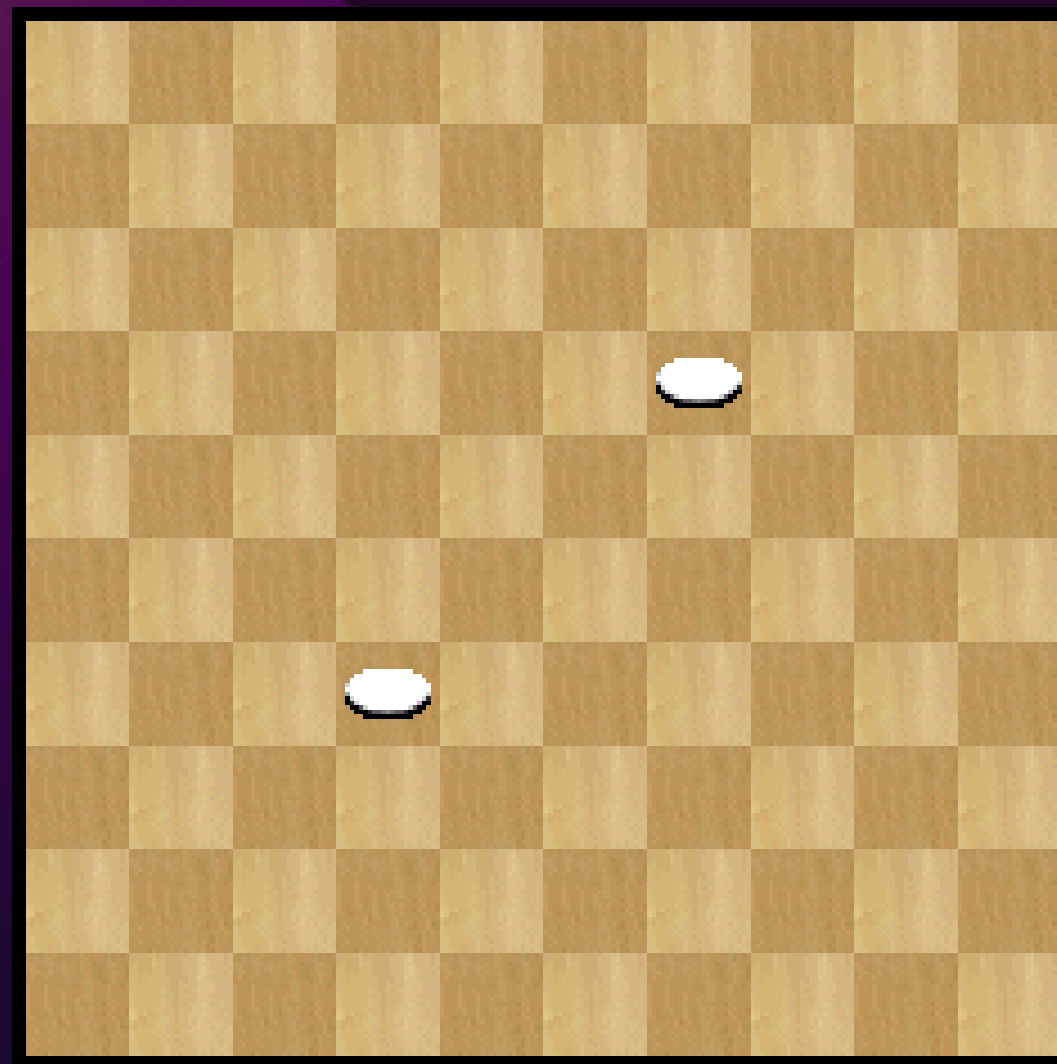
LE DEPLACEMENT  
D'UN PION SE FAIT  
SUR LA DIAGONAL



UN PION QUI  
ATTEINT LE CÔTÉ  
INVERSE DEVIENT  
UNE DEMME



UNE PARTIE EST  
FINIE SI L'UN DES  
DEUX JOUEURS N'A  
PLUS DE PION/DEME







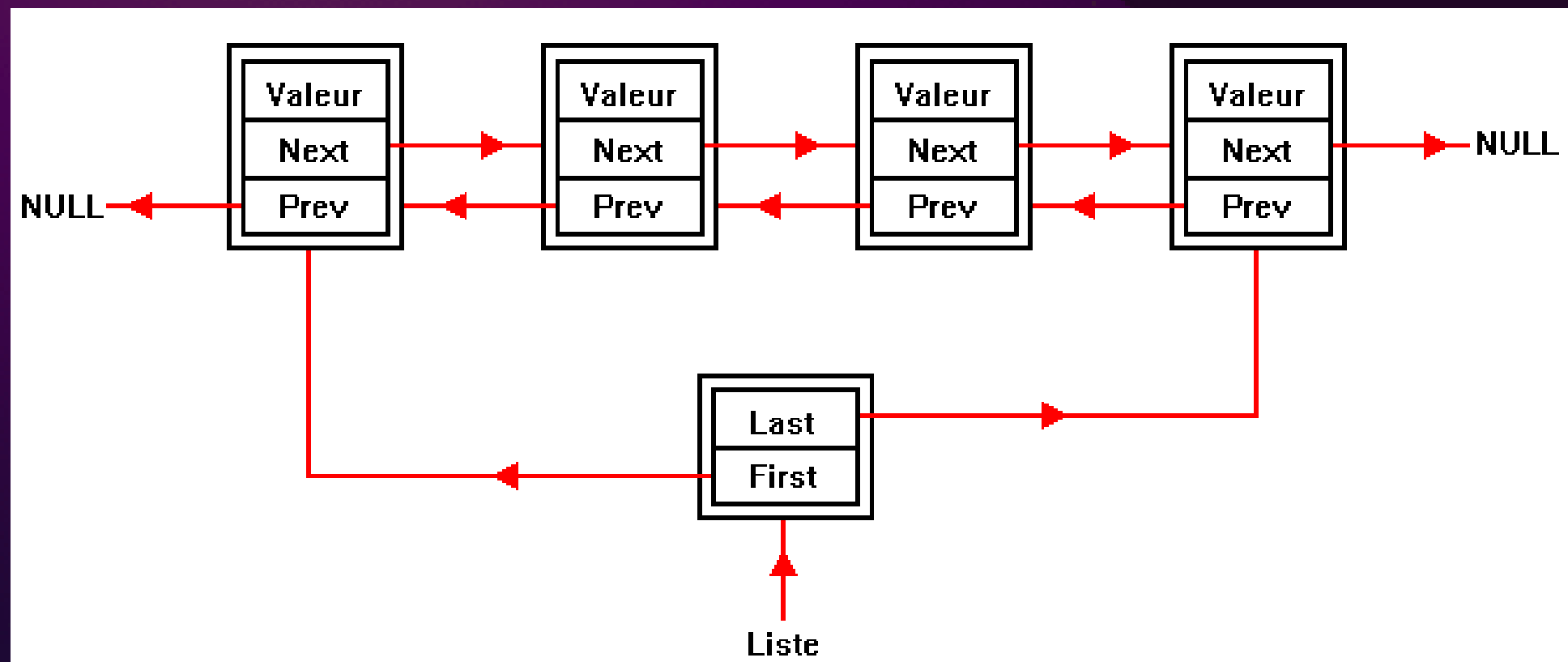
# STRUCTURE DE BASE

# NOEUD

```
typedef struct Item_s {  
    char size; // size of board (< 255)  
    char *board; // board is an array of small int  
    char blank;  
    int f, g, h; // cost, heuristic, ...  
    int depth;  
    struct Item_s *parent; // needed for tree structure in game search  
    struct Item_s *prev, *next; // needed for chained list  
} Item;
```

# LISTE

```
typedef struct {  
    int numElements;  
    char *name;  
    Item *first;  
    Item *last;  
} list_t;
```





# IMPLÉMENTANT DU JEU

# COMMENT POSITIONNER LES PIONS ?

`char *initGame()`

	1		1		1		1		1
1		1		1		1		1	
	1		1		1		1		1
1		1		1		1		1	
	2		2		2		2		2
2		2		2		2		2	
	2		2		2		2		2
2		2		2		2		2	

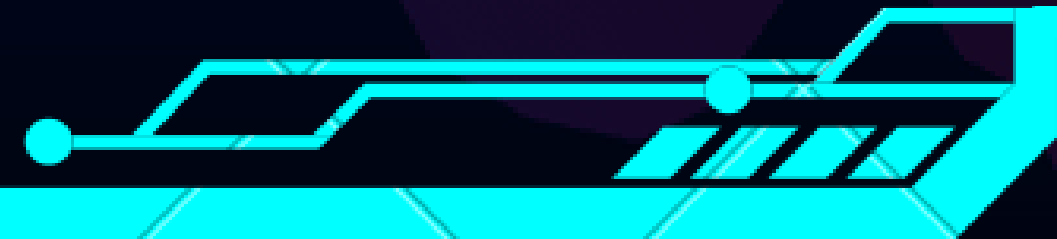
# LE DEPLACEMENT EST-IL VALIDE ?

```
int isValidPositionJeu( char *plateau, int pos_init, int  
pos_final)
```



# COMMENT DEPLACER LES PIECES ?

```
char *moveJeu( char *plateau, int pos_init, int pos_final)
```



# COMMENT SAVOIR SI UNE PARTIE EST FINIE ?

```
int partieFini(char *plateau)
```

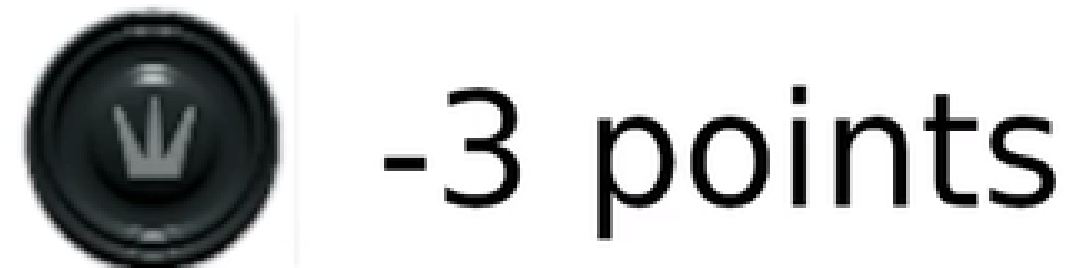
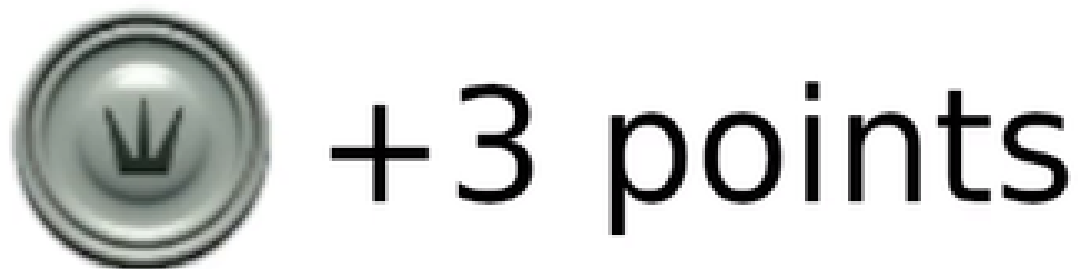
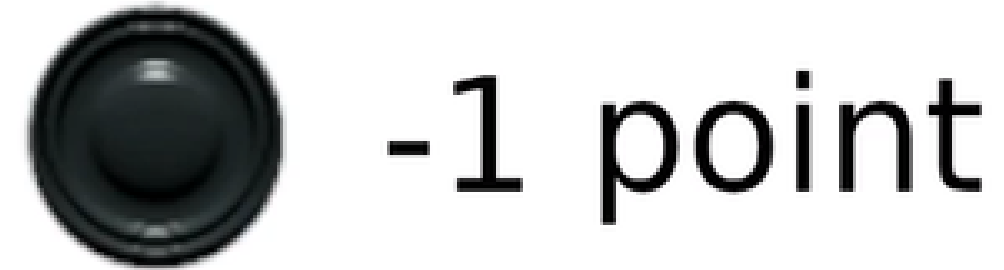
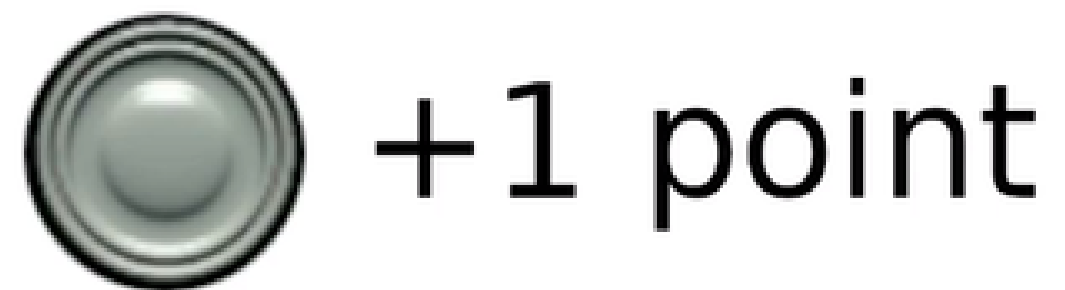




# IMPLÉMENTANT DE L'IA

# COMMENT SAVOIR UN BON COUP D'UN MAUVAIS ?

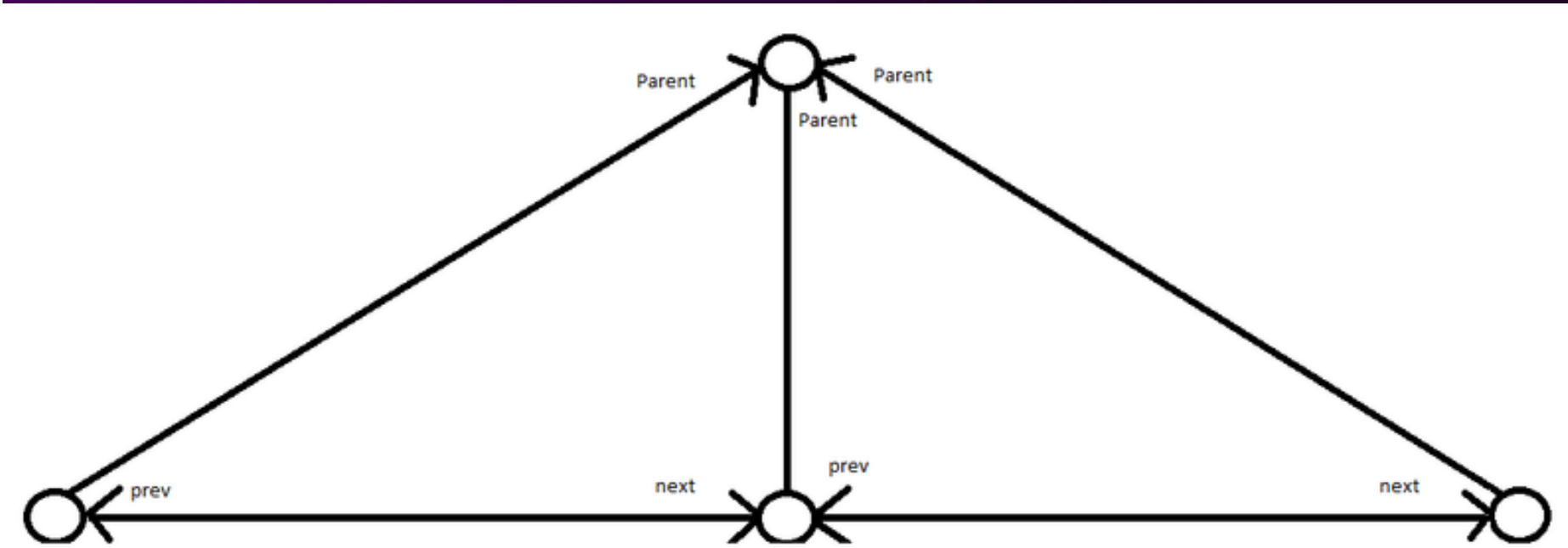
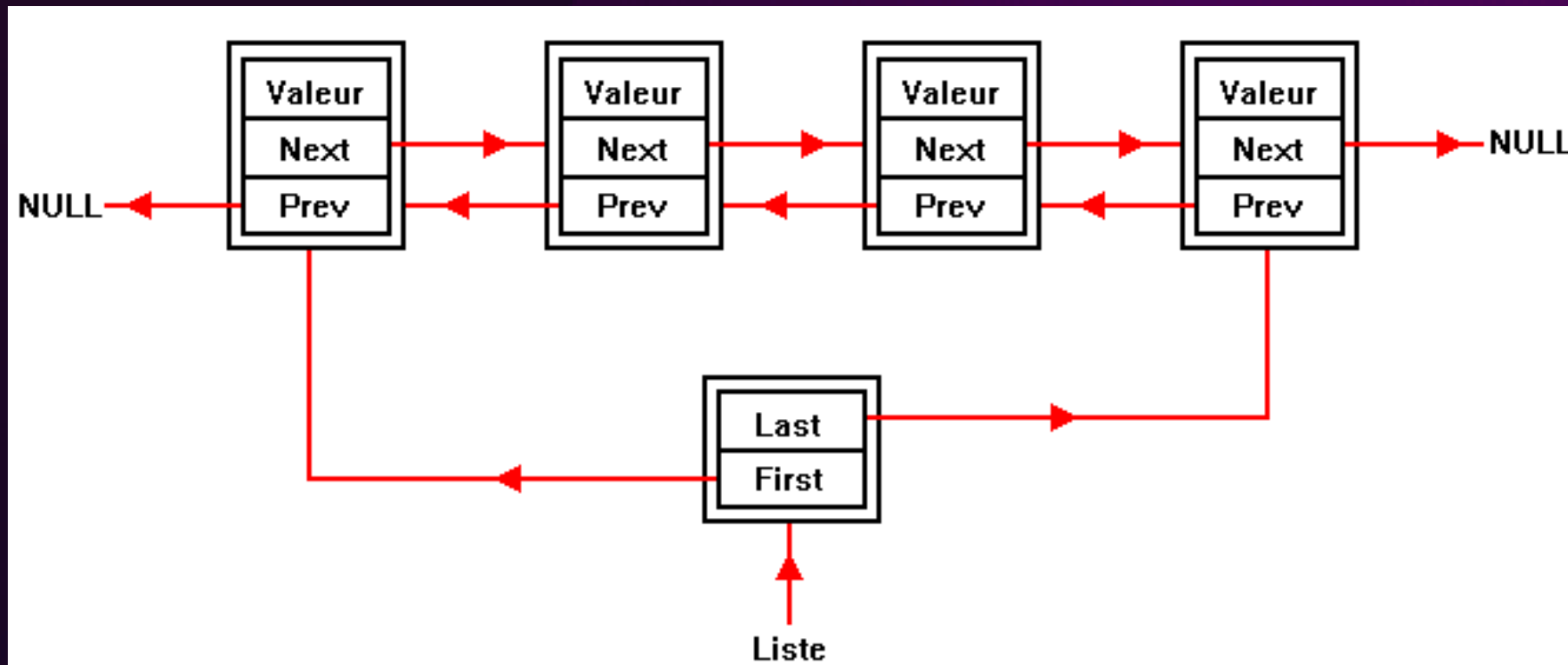
```
int evaluation(Item *node)
```



# IA: GENERATION DES COUPS

ETAPE 01


`Item *generation_coups(Item* node, list_t *listeCoups, int i, int j)`



# ETAPE 02

# ETAPE 02





# IA: GENERATION DES COUPS

ETAPE 03

`Item * minmax(Item *node)`

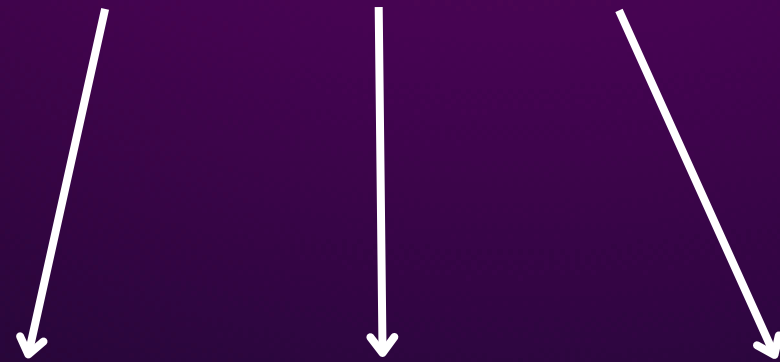




# INTERFACE GRAPHIQUE

# BIBLIOTHEQUE UTILISÉ

**SDL**



Simple DirectMedia Layer

# FONCTIONS UTILISÉES



**DRAWCIRCLE()**



**SDL\_RENDERFILLCIRCLE()**



**AFFICHAGE()**



**INPUTSOURIS()**



**DRAWELLIPSE()**



**SDL\_RENDERFILLELLIPSE()**