

```

/*-----*/
/* DUELLISTE.PRO *           Expert en duels           * Christian JOUSSELIN */
/*-----*/
duelliste :- def_wind(0,79,0,25,'DUELLISTE'),
             def_wind(1,78,1,17,'Surface de jeu'),
             def_wind(15,44,18,20,'Joueur'),def_wind(1,14,18,23,'Menu'),
             def_wind(45,78,18,20,'Expert'),def_wind(15,78,21,23,"),
             repeat,w(["F1: Aide","F2: Jeu","F10: Fin"],'Menu',C),
             m(1,C),!,res_wind,res_wind,res_wind,res_wind,res_wind,res_wind.
m(1,1059) :- aide(A),w(A,'Surface de jeu',_),fail.
m(1,1060) :- w(["Entrer le nom du jeu suivi de .<-+: "],"),
             read(J),nonvar(J),newzone(jeu),consult(J),repeat,
             w(["F1: Aide","F2: Init. ","F3: Lancer","F10: Fin"],'Menu',C),
             m(2,C),!,save(J),newzone(root),delete_gc(jeu),fail.
m(2,1059) :- regles(R),w(R,'Surface de jeu',_),fail.
m(2,1060) :- retract(etat(_)),fail.
m(2,1060) :- w(["Situation initiale? suivi de .<-+: "],"),
             read(S),nonvar(S),init(S,E),assertz(etat(E)),
             window('Surface de jeu'),affiche(E),fail.
m(2,1061) :- repeat,coups,!,fail.
m(_,1068).

coups      :- not jouer('Joueur').
coups      :- not jouer('Expert').

jouer(N) :- etat(E1),possibilites(E1,L),encore(N,E1,L),act(N,E1,L,A),
            modifie(E1,A,E2),window('Surface de jeu'),affiche(E2),
            retract(etat(_)),assertz(etat(E2)).

a(_,L,A) :- read(A),nonvar(A),member(A,L),!.
a(,_,__) :- w(["Action interdite ! Un conseil (o/n)? "],",110),!,fail.
a(E,_A) :- w(["Je joue _ ta place: "],',Joueur'),expert(E,[_A]),write(A).

act('Joueur',E,L,A) :- repeat,w(["Ton jeu puis .<-+: "],',Joueur'),a(E,L,A),!.
act('Expert',E,_A) :- w(["Je joue "],',Expert'),expert(E,[_A]),write(A).

encore(N,E,_) :- termine(E,[1,nil]),w(["Bravo. C'est gagn_ !"],N),!,fail.
encore(N,E,_) :- termine(E,[0,nil]),w(["Dommage. C'est perdu !"],N),!,fail.
encore(N,_,[]) :- w(["Plus de possibil_ s !"],N),!,fail.
encore(N,_,_) :- w(["C'est a "],"),write(N),writes(" de jouer.").

aide(["DUELLISTE:","","Mini syst_me expert adapt_ aux duels. ","","Appuyer sur une touche."]).
/*-----*/
expert(Eta,Sol) :- termine(Eta,Sol),!.
expert(Eta,Sol) :- base(Eta,Sol),!.
expert(Eta,Sol) :- expert1(Eta,Sol),assertz(base(Eta,Sol)).

expert1(Eta,Sol) :- possibilites(Eta,L),t_act(Eta,L,Sol),!.
expert1(Eta,[0,A]) :- possibilites(Eta,[A_]).

t_act(_,[_],_) :- fail.
t_act(E1,[A_],[1,A]) :- modifie(E1,A,E2),not expert(E2,[1,_]),!.
t_act(E,[_L],Sol) :- t_act(E,L,Sol).
/*-----*/
member(A,[A_]) :- !.
member(A,[_T]) :- member(A,T).

w(L,W,K) :- w(L,W),getk(K),clear_scr.
w(L,W) :- window(W),clear_scr,pos(0,0),w_ligne(L).
w_ligne([L]) :- writes(L),!.
w_ligne([L|T]) :- writes(L),nl,w_ligne(T).
/*-----*/
noscreen,newzone(root)?

```

/\*-----\*/  
/\* 3ème Année SIT 95-96 \*                      Expert en duels                      \* Ecole CENTRALE \*/  
/\*-----\*/

Prédicat principal:

Définitions des fenêtres de travail et de jeu

Appel du menu principal de DUELLISTE

Prédicat menu:

Touche F1: Aide. Affiche le texte d'aide sur la surface de jeu.

Touche F2: Appel du jeu. Interroge pour connaître le nom du fichier de jeu, puis charge en zone jeu ce fichier et appelle le menu de lancement du jeu choisi. En sortie de ce menu la zone jeu enrichie de l'apprentissage est sauvee dans le fichier jeu.

Touche F1: Règles. Affiche le texte de règles sur la surface de jeu.

Touche F2: Initialise le jeu choisi: interroge pour connaître la situation initiale et appelle le prédicat de jeu "init" pour avoir l'état initiale et le mémorise dans le fait: "etat". Affiche la situation initiale grace au prédicat de jeu: "affiche".

Touche F3: Fait jouer alternativement le joueur et l'expert jusqu'à la fin de la partie.

Touche F10: Sortie des menus.

Coups : Fait jouer le Joueur.

Coups : Fait jouer l'Expert.

Jouer : Liste les possibilités, non terminales, fait agir celui qui doit jouer en fonction d'elles, affiche la situation modifiée et actualise l'état correspondant à la nouvelle situation.

Lit et teste si la lecture est une action possible.

Propose un conseil dans le cas contraire.

En cas de conseil appelle l'expert et joue l'action de l'expert.

Attend le jeu du joueur et le teste.

Jeu de l'expert.

Situation: terminale gagnante; arrête la partie.

terminale perdante; arrête la partie.

terminale par manque de possibilité; arrête la partie.

non terminale.

Texte d'aide.

/\*-----\*/

expert: Teste si la situation est terminale par axiome.

Teste si la situation a déjà été apprise.

Sinon appelle l'expert1 et mémorise le résultat dans "base".

expert1: Liste les possibilités et les teste une à une.

En cas d'aucune action gagnante joue la première action possible.

Test action: Liste d'action vide: aucune action gagnante possible.

Modifie la situation avec l'action courante et cherche une

situation perdante pour l'adversaire; si pas possible continue.

/\*-----\*/

Prédicat d'appartenance à une liste.

Ecriture d'un texte et lecture d'une touche.

Ecriture d'un texte.

Ecriture d'une ligne.

/\*-----\*/