Fonctions utilisées :

random(min,max,Xaleatoire)

nth0(indice,liste,element)

length(liste,Nelements)

Installation du jeu

case(ligne,colonne,LP,Altitude) :-

adjacenceCase([L1,C1,\_,\_],[L2,C2,\_,\_])

ligneDroite([L1,C1,\_,\_],[L2,C2,\_,\_])

caseNormale() random

caseSpeciale() random

personnage() // ListePerso initiale

*joueur() //penser à creer une listeJoueur*

*tueurAGage()*

policier()

cible()

innocent() :- a été tué par le joueur, n’est pas dans sa liste

Etat du joueur :

enVie()

Arreté()

Mort()

Déroulement

//Liste perso 2 qui est mise à jour à chaque tour

premierJoueur()

compteur is 2

deplacerPerso()

EliminerPerso()

* tirer()
* poignarder()
* sniper()

ControlerIdentitePersonnage()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tueur à gages ni arrêté ni éliminé 2 pts

Élimination d’une cible 1 pt

Arrestation d’un tueur à gages adverse 1 pt

Élimination d’un tueur à gages adverse 3 pts

Élimination d’un innocent - 1 pt

Élimination d’un policier - 1337 pts

LP=["Tigre","Police1","Police2","Police3","Phoque","Panda","Koala","Chat","Chatte","Vautour","Girafe","Pandate","Crocodile","Souris","Poulet","Loup"]

// Installation plateau

repeat,

WHILE (il faut avoir 16 cases)

Nnormal=0.

L=random(1,16,L)

C=random(1,16,C)

Personnage=random(0,15,Personnage)

LengthNormal=length(ListeCaseNormale)

if(not(case(\_,\_,[Personnage])))

if(not(case(L,C,\_)))

if(Nnormal<8)

case(L,C,[Personnage],0).

Nnormal is Nnormal+1.

else

case(L,C,[Personnage],1).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

:- dynamic(faits/2).

% predicat pour obtenir un nombre aléatoire plus grand que 0

nombre\_de\_faits(Base, N) :-

N1 is random(Base),

( ( N1 = 0, nombre\_de\_faits(Base, N));

N = N1).

test :-

% On vide la base des faits

retractall(faits(\_,\_)),

% on selectionne au plus 10 faits

nombre\_de\_faits(10, N),

write('Nombre de faits '), write(N), nl,

% on cree les N faits alétaoires

forall(between(1,N, \_),

( X is random(100), Y is random(100),

assert(faits(X,Y)))),

% edition des résultats

forall(faits(A,B), writeln(faits(A,B))).