

:JoueurHumain

:jeu

:Plateau

:Trou

estJoueurAffame(joueurAdverse)&& !plateau.peutNourrirAdversaire(joueurCourant)

ALT true

True

ALT

finDePartie == true

ALT False

false

ALT : true

(plateau.estJoueurAffame(joueurAdverse))

True

message  
joueur adverse affamé

choisirTrou(joueurCourant, ligneCourante)

joueurCourant.choisirUnTrou(plateau.getNbrTrou())

Random()

int trou de 0 a 5

indiceLigne

indiceLigne

plateau.estTrouAffame(trou passé)

isAffamed()

while :plateau.estTrouAffame(trou passé)

ce trou est affamé

return true

return true

ALT

joueurCourant.choisirUnTrou(plateau.getNbrTrou())

random()

int trou de 0 a 5

return false

return false

return indiceLigne

plateau.semer(indiceLigne)

return dernier indiceaprès semaille

while :(plateau.champsAdverse(indiceLigne,ligneCourante)) &&  
(plateau.recoltable(dernier indice))

return true

plateau.recolter(indiceLigne)

return Score de la récolte

Trou.isRecoltable()

return true

ALT

return false

ALT:plateau.estJoueurAffame(joueurAdverse)

(plateau.estJoueurAffame(joueurAdverse))

True

recolte = 0

plateau.copyTabjToTabLigne()

reinitialize tabline invalide le coup

alt:plateau.estJoueurAffame(joueurAdverse)

(plateau.estJoueurAffame(joueurAdverse))

False

incrementeScore(joueurCourant, recolte)

plateau.copyTabLigneToTabj()

actualise les tableau reel(valide le coup)