

TP 4

Au cours de ce TP, vous écrirez en Prolog un jeu vidéo : le jeu des Vaches Zombies. Le principe du jeu est très simple : « Dimitri s'est perdu dans un alpage. La nuit tombe et pour se réchauffer, Dimitri va au milieu d'un troupeau de vaches. Tout à coup, l'une d'elles meurt étrangement et se relève : c'est une vache zombie! Le joueur incarne Dimitri qui doit s'enfuir du troupeau sans se faire mordre par une vache zombie. »

Exercice 1 : Données du jeu

Le jeu se déroule dans un alpage, représenté par un plateau rectangulaire composé de cases.

1. Le terrain est encombré par des rochers et des arbres. Déclarez les relations `rocher(X,Y)` et `arbre(X,Y)` qui retourneront vrai si la case (X,Y) est encombrée par un rocher (respectivement par un arbre).
Indications : La déclaration d'une relation se fait avec `:-dynamic`. Ainsi `:-dynamic r1/1.` déclare une relation unaire `r1`, et `:-dynamic r2/3.` déclare une relation ternaire `r2`.
2. Des vaches paissent tranquillement sur le pâturage.
Déclarez la relation `vache(X, Y, Race, Etat)` qui retourne vrai lorsqu'une vache de race `Race` est sur la case (X,Y) . L'Etat de la vache est vivante ou zombie. Les races possibles sont brune, simmental, alpine_herens.
3. Il reste à placer le héros. Déclarez la relation `dimitri(X, Y)` qui donne la position de Dimitri.
4. Définissez les faits `largeur(X)` et `hauteur(Y)` qui donnent la largeur et la longueur du plateau de jeu (choisissez des valeurs).
5. Définissez les faits `nombre_rochers(N)`, `nombre_arbres(N)`, `nombre_vaches(Race, N)` qui donnent le nombre de rochers, d'arbres et de vaches de chaque race sur le plateau de jeu (choisissez des valeurs).

Exercice 2 : Initialisation 1

1. Écrivez la règle `occupe(X,Y)` qui est vrai si et seulement si la case (X,Y) est occupée par un arbre, un rocher, une vache ou Dimitri.
2. Écrivez la règle `libre(X,Y)` qui retourne dans `X` et `Y` les coordonnées d'une case libre, c'est-à-dire n'ayant ni rocher, ni arbre, ni vache, ni Dimitri.
Indications : Pour obtenir un nombre aléatoire, vous pouvez utiliser `N is random(Max)`. Vous pouvez utiliser `repeat` pour forcer le backtrack sur `random`.

3. Écrivez les règles `placer_rochers(N)`, `placer_arbres(N)`, `placer_vaches(Race, N)` qui placent N rochers, arbres ou vaches sur le plateau de jeu.
Écrivez également la règle `placer_dimitri` qui place Dimitri sur le plateau de jeu.
Attention, vérifiez bien qu'il y a un seul objet par case.
Indication : Pour ajouter un fait $r(x)$, utilisez `assert(r(x))`.
4. Écrivez la règle `vaches(L)` qui retourne dans L la liste des positions occupées par des vaches. Indications : Pensez à utiliser `bagof` ou `findall`.
5. Écrivez la règle `creer_zombie` qui sélectionne aléatoirement une vache et la transforme en zombie.
Indications : Pensez à utiliser `length`. Pour supprimer un fait $r(x)$, utilisez `retract(r(x))`.

Vous pouvez afficher votre plateau de jeu avec les règles suivantes :
`initialisation. affichage.`