

# Rendu de projet pour transfert de technologie

Le projet présente les jeux avec une interface développée en Qt. Pour le moment, seul le jeu des seaux fonctionne.

Le modèle du jeu des seaux est donné ci-dessous.

```
include "globals.mzn";

% Jeu des seaux

enum seaux;
int: max_transferts;
array [seaux] of int: capacite;
array [seaux] of int: init;
array [seaux] of int: fin;

var 1..max_transferts - 1: nb_transferts;
array [1..max_transferts, seaux] of var 0..max(capacite): remplissage;

% État initial et final
constraint forall (s in seaux) (remplissage[1,s] = init[s] /\
remplissage[nb_transferts,s] = fin[s]);
% Conservation de l'eau
constraint sum(init) = sum(fin) /\ forall (t in 1..max_transferts)
(sum(init[..]) = sum(remplissage[t,..]));
% Contrainte sur la solution
constraint forall (t in 1..nb_transferts) (
    exists (s1, s2 in seaux) (
        remplissage[t,s1] > 0 /\ remplissage[t,s2] < capacite[s2]
        /\ remplissage[t,s1] - remplissage[t+1,s1] = remplissage[t+1,s2]
- remplissage[t,s2]
        /\ (remplissage[t+1,s1] = 0 \/ remplissage[t+1,s2] =
capacite[s2])
    )
);
constraint forall (t in 1..max_transferts) (
    forall (s in seaux) (0 <= remplissage[t,s] /\ remplissage[t,s] <=
capacite[s])
);

solve minimize nb_transferts;
```

L'exemple du jeu est donc donné avec un datafile comme suit.

```
% Exemple du jeu des seaux

seaux = {A,B,C};
max_transferts = 10;
capacite = [8,5,3];
init      = [8,0,0];
fin       = [4,4,0];
```