

## Type de contrainte que chaque interface peut résoudre

Type de T+C

MuseoTUI -> utilisé pour la collaboration-> réflexion sur le parcours souhaité

Hercule même principe

type de contrainte :

- résous les contraintes inégalité (ou arithmétique) (ex: si  $x_1=1$  alors  $x_2 \neq 2$  on peut voir ça avec MuseoTUI quand le nombre de personne =1 le type de personne != enfant de 1 à 3 ans si on associe un 1 à "enfant de 1 à 3 ans" 1 alors on a  $x_1+x_2 \neq 2$ )
- alldiff c'est possible (ex: dans un sudoku on ne peut pas avoir sur une même ligne, colonne et case le même numéro, ce "alldiff" sera caractérisé par le nombre de jeton disponible)

Domaine :

MuseoTUI

$x_1$ = nombre de personne= {1 personnes, 2 personnes , 3-6 personnes}

$x_2$ = âges des personnes = {3-6 ans, 7-12 ans, 12-18 ans, +18 ans}

$x_3$  = expérience = {expert,débutant}

$x_4$ = objectif = {apprendre,jouer,expérimenter}

$x_5$ = handicap= {cognitif,visuel,moteur,auditif}

$x_6$ = durée= {30 min, 1h,2h}

le système pourra interpréter ces informations sous forme de nombre (ex pour  $x_1$ : 1= 1 personne, 2= 2 personnes , 3=3-6 personnes)

autre exemple avec un sudoku:

$x_{ij}$  = {0...9} avec i = la ligne et j la colonne

utilisation : requête, planification

URP-> discussion autour d'une table pour entre différent architecte pour savoir où construire les bâtiments

Contraint :

- alldiff: un bâtiment ne peut pas être dans la même position qu'un autre bâtiment
- peut être  $p_x \neq p$  si sur la position p il y a une route par exemple

Domaine :

- $p_x$  = les différentes positions possibles sur la table de l'objet n (vecteur(x,y))  $\{(0,0)....(N,M)\}$  N= la longueur max de la table M= la largeur max de la table

Utilisation : architecture, trafic routier, tous les domaines qui ont besoin d'agencement spatial

PICO -> bien pour la collaboration car (si on prend l'exemple des antennes) les ingénieurs pourront facilement discuter pour avoir la meilleure optimisation possible

Contraint :

- Alldiff: aucun objet au même endroit
- d'inégalité ou arithmétique :  $px+d \geq py$  ou  $px-py \geq d$  cela veut dire py ne peut pas être à une distance de  $px \leq d$

Domaine :

- $p_n$  = les différentes positions possibles sur la table de l'objet n (vecteur(x,y))  $\{(0,0), \dots, (N,M)\}$  N= la longueur max de la table M= la largeur max de la table
- $n = \{0 \dots N\}$  N= le nombre d'objet

Utilisation : telecom, eau, paratonnerre, tous les domaines qui ont besoin de poser des objets en fonction d'autres objets agencer .

Sensetable

Collaboration -> différentes personnes peuvent discuter autour de la simulation

Contraint

- Contraintes de bornes (possible)-> par exemple un nombre maximal de lapin par rapport au nombre de renard  $2r_x < l < 3r_x$  et il peut aussi avoir un nombre de renard maximal
- ou alors c'est juste une contrainte d'égalité ou arithmétique  $l = 2r_x$

Domaine:

$r_x$  = nombre de renard dans le lieu  $x = \{0 \dots R\}$  R = le nombre max de renard

$x = \{0 \dots X\}$  X= le nombre de lieu

$l_x$  = nombre de lapin dans le lieu  $x = \{0 \dots L\}$  L= le nombre max de lapin

Utilisation : bourse, chimie, ou tous les domaines qui ont une donnée qui varie selon une autre donnée

Supply Chain Visualization

Collaboration -> différentes personnes peuvent discuter autour de la simulation

Contraint:

- dépendance ou arithmétique ex:  $dx + tx < dy$

Domaine:

$dx$  = début de la tâche  $x = \{0 \dots T\}$  T= le nombre max de temps

$x = \{0 \dots X\}$  X= le nombre de tâche

$tx$  = temps de la tâche  $x = \{0 \dots T\}$

utilisation : Toutes entreprises qui ont besoin d'ordonnancement

## SenseBoard

Collaboration -> discuter de la gestion des tâches ensemble

Contraint:

- Dépendance ou arithmétique ex:  $dxia + txia < dyjb$
- Diff ex :  $dxia \neq dyib$  par rapport au x ou  $dxia \neq dyja$

Domaine:

- $dxia$  = début de la tâche x de l'auteur i dans la salle a =  $\{0 \dots n\}$  n= le nombre de tâche
- $x = \{0 \dots X\}$  X= le nombre de tâche
- $i = \{0 \dots I\}$  I= le nombre d'auteur
- $a = \{0 \dots A\}$  = le nombre de salle
- $txia$  = temps de la tâche x =  $\{0 \dots T\}$  T= le nombre max de temps

utilisation : école, Toutes entreprises qui ont besoin d'ordonnancement, ou tous types d'événements (conférence ...)

## ColorTable

discussion autour d'une table pour entre différentes architectures pour savoir où construire les bâtiments

Contraint :

- alldiff: un bâtiment ne peut pas être dans la même position qu'un autre bâtiment
- peut être  $px \neq p$  si sur la position p il y a une route par exemple

Domaine :

- $px$  = les différentes positions possibles sur la table de l'objet n (vecteur(x,y))  $\{(0,0) \dots (N,M)\}$  N= la longueur max de la table M= la largeur max de la table

Utilisation: architecture, réseau routier, tous les domaines qui ont besoin d'agencement spatiale

## TangiSense

Le principe de cette table est de faire collaborer les différents individu

Contraint :

- alldiff: un bâtiment ne peut pas être dans la même position qu'un autre bâtiment
- peut être  $px \neq p$  si sur la position p il y a une route par exemple

Domaine :

- $px$  = les différentes positions possibles sur la table de l'objet n (vecteur(x,y))  $\{(0,0) \dots (N,M)\}$  N= la longueur max de la table M= la largeur max de la table

Utilisation: éducation, architecture, réseau routier, tous les domaines qui ont besoin d'agencement spatiale

## TinkerLamp

Permet la collaboration car les enfants vont discuter pour savoir où placer les étagères

Contraintes:

- diff: les étagères ne peuvent pas être au même endroit  $p_x \neq p_y$
- inégalités ou arithmétique les étagères on besoin d'un espace  $p_x + d \leq p_y$  ou  $p_x - d \geq p_y$

Domaine :

- $p_x$  = position de l'étagère  $x$  = vecteur  $(x, y)$  de 2 dimension de taille  $n \times m$  en fonction de la largeur et la longueur de l'étagère
- $x = \{0 \dots X\}$   $X$  = le nombre d'étagère

Utilisation : éducation , ou domaines qui nécessite de l'agencement

LegoWall

possible discussion et collaboration pour l'ordonnancement

Contraint:

- alldiff :  $x_i \neq x_j$
- temps ou arithmétique ou inégalité  $t_i < t$  l'heure d'arrivé de  $i$  doit être inférieur au temps dans la matrice (temps actuelle)

Domaine

$x_i$  ports maritime où est le bateau  $i = \{0 \dots M\}$   $M$  = le nombre de port maritime

$t_i$  heure d'arrivée de  $i = \{0 \dots T\}$   $T$  = le temps maximal

$t$  le temps actuelle =  $\{0 \dots T\}$

Utilisation : port maritime, ou tous domaines qui ont besoin d'ordonnancement