

## Learning the Parity Function :

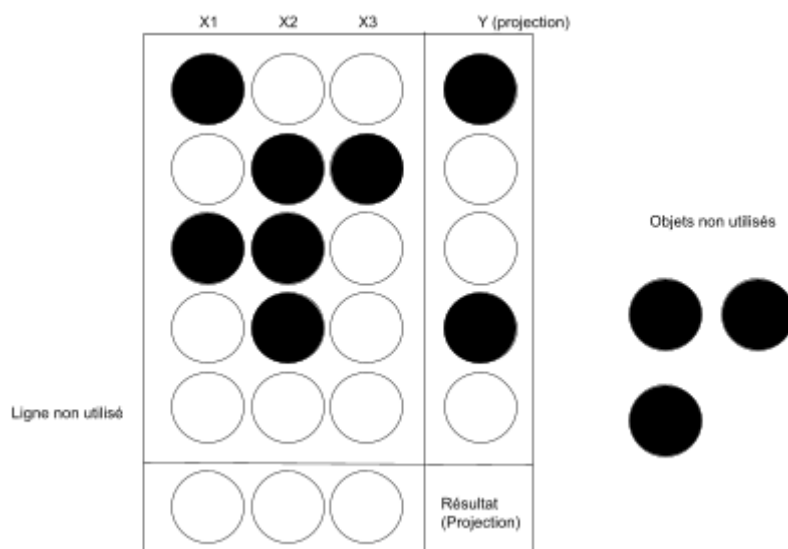
Token+contraint :

des objets en cercle à placer dans des cercles (contraintes)

si objets sur x1 alors  $x1=1$  sinon = 0

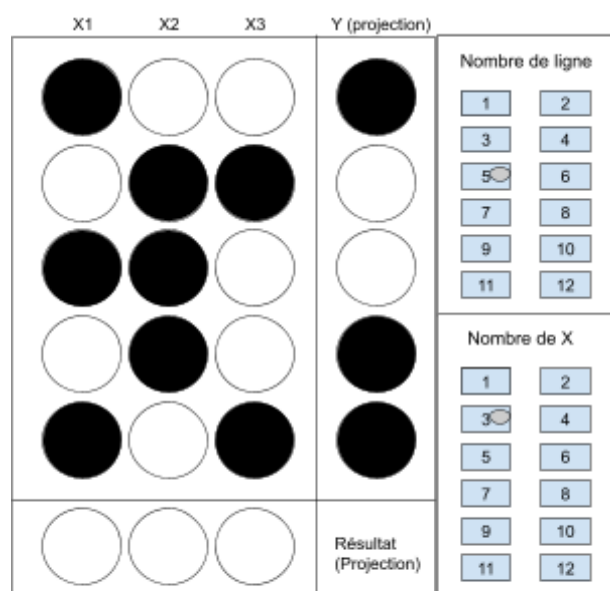
s'il n'y a pas d'objet sur une ligne alors la ligne n'est pas utilisée

On peut faire aussi avec une projection qui permettra de modifier le nombre de vecteurs, il y aura donc plus de liberté, mais plus d'erreurs pour l'utilisateur.



Projection:

permet de choisir le nombre de ligne en plus



Objets non utilisés



Tour de Hanoi:

Le système projette l'état initial et l'état but

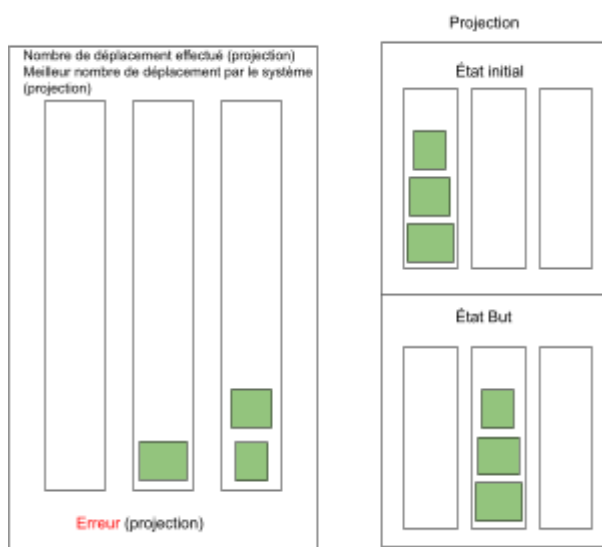
Le nombre de coût effectué est initialisé quand la configuration des objets est la configuration de l'état initial

En cas d'erreur, un message en rouge écrit erreur (si un objet plus grand et au dessus d'un plus petit ou si deux objets sont enlevés en même temps)

on peut soustraire un coût effectué si l'utilisateur retourne sur l'état d'avant

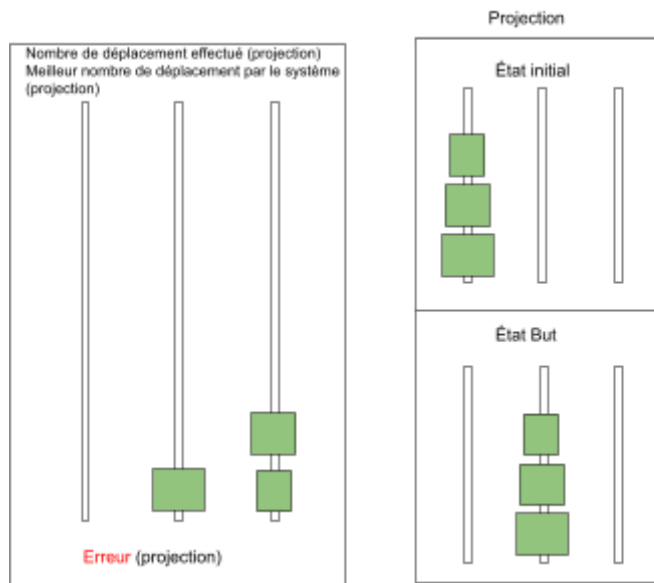
ex: token +contraint

les objets de différentes tailles (token) peuvent être placés dans un rectangle (contraint) et peuvent être déplacés dedans pour connaître la position de l'objet dans la tour



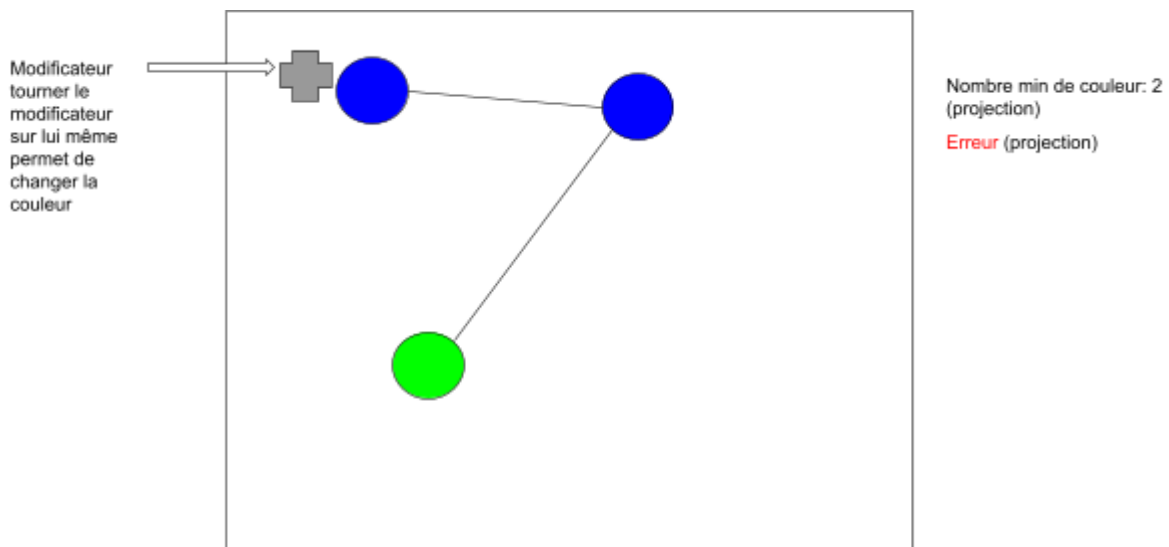
ex : Projection

Les rectangles sont remplacés par des tiges affichées via un vidéoprojecteur quand les objets ne sont pas sur la tige ils ne sont pas pris en compte par le système. Avec ce système on pourra augmenter le nombre de pique aisément comme avec tinkerLamp. Mais plus de liberté et plus d'erreurs pour l'utilisateur.



Coloration de graphe:

Utilisation de l'interface Sensetable, à la place d'utiliser des grandes valeurs l'objet modificateur va permettre de changer de couleur selon à choix d'une dizaine de couleur. Un message peut être affiché par le système pour dire cb de couleur propose le meilleur résultat et s'il y a une contrainte par respecter (en rouge)  
Pour éviter de manipuler des couleurs on peut mettre des chiffres à la place



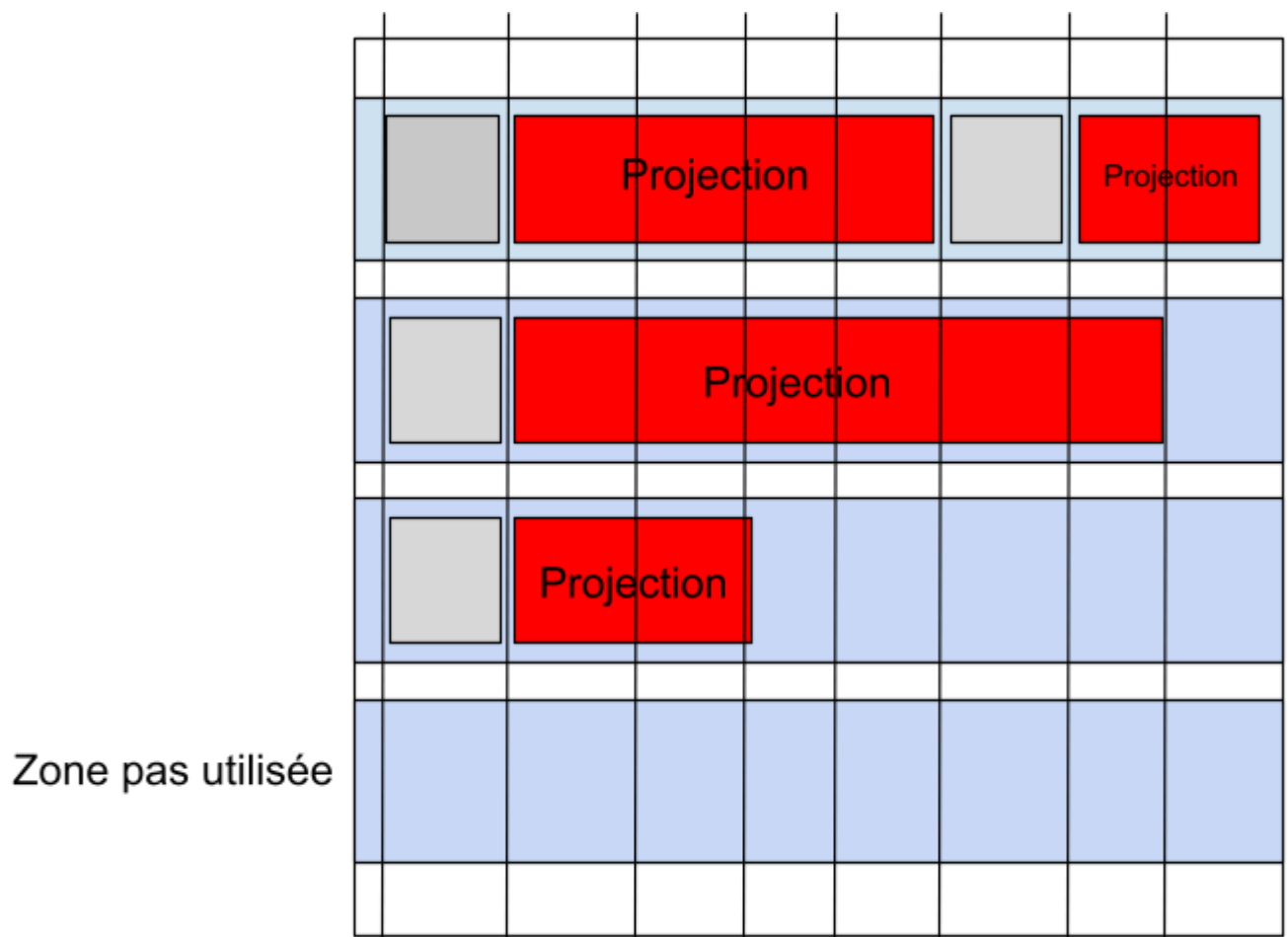
Job shop scheduling (ou problème d'ordonnancement )

Utilisation de Token+contraint : utilisation d'objets (token) pour représenter les tâches à placer dans des rectangles (contraint) un déplacement libre peut être effectué pour connaître le temps de début de la tâche, des tiges sont présent pour délimiter une valeur de temps

Le temps des tâches est connu grâce au RFID associé à chaque objet.  
 Le temps de la tâche est représenté par une projection d'un rectangle en rouge  
 Certaines zones de contraintes ne sont pas obligées d'être remplies.

Deux possibilités:

- Le système connaît les tâches qui sont dépendantes et affiche les zones impossibles.
- les tâches dépendantes sont dans le même rectangle et le système projette la durée de la tâche posée
- juste mettre un message en cas de conflit (comme SenseBoard)



Utilisation : SenseBoard

Dans le même principe de leur application, on utilise que les objets représentant les tâches et on enlève le système de contrainte, l'objet peut être placé librement, mais ne sera pas pris en compte s'il est mal placé.









Comme avec T+C, on peut utiliser une projection de la durée de la tâche ou alors juste un message d'erreur. Avec ce système on pourra élargir le temps ainsi que le nombre de ranger ou mettre les tâches

## Sudoku

### Utilisation token+contraint

Pour chaque case 3\*3 nous avons 9 objets représentant les 9 nombres, et pour chaque grille une forme spécifique (triangle, rectangle ...) pour éviter de mettre d'autres objets dans d'autres cases, le seul inconvénient dans cette stratégie est le nombre d'objets, en effet on aura besoin de 81 objets qui peuvent être beaucoup

En cas d'erreur, une LED rouge s'allume.

## Problème de n reines

### T+C:

On peut utiliser des cases et des objets

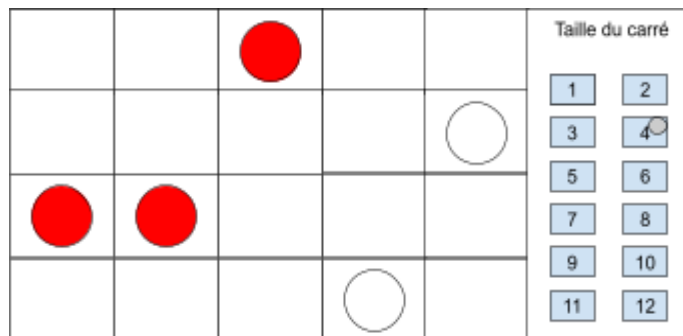
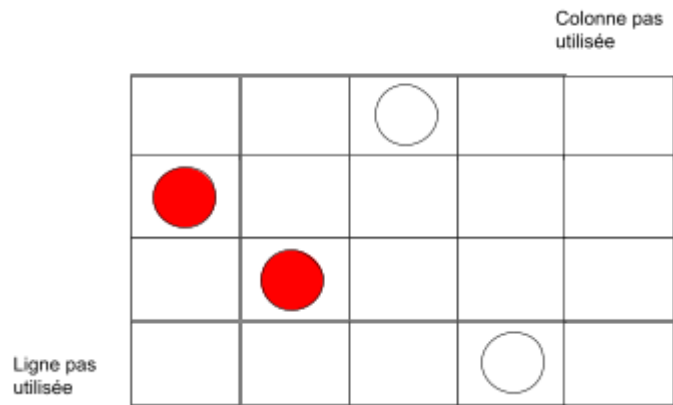
Si deux objets ne respectent pas la contrainte, une LED rouge s'allume, pour enlever la LED il faut remplacer un des deux objets.

Une colonne et une ligne peuvent ne pas être utilisées alors ils ne vont pas être pris en compte par le système.

### Projection:

La projection va permettre de configurer le problème selon notre souhait

A la place des LED on peut utiliser le vidéoprojecteur pour mettre les objets en rouge



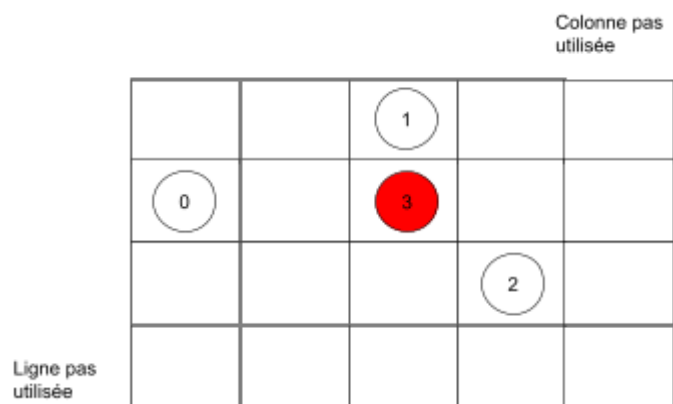
Problème du cavalier d'euler:

T+C:

On peut utiliser des cases et des objets, chaque objet à un numéro pour connaître l'ordre des mouvements

Une ligne et une colonne peuvent ne pas être utilisées alors le système ne prendra pas en compte ces dernières.

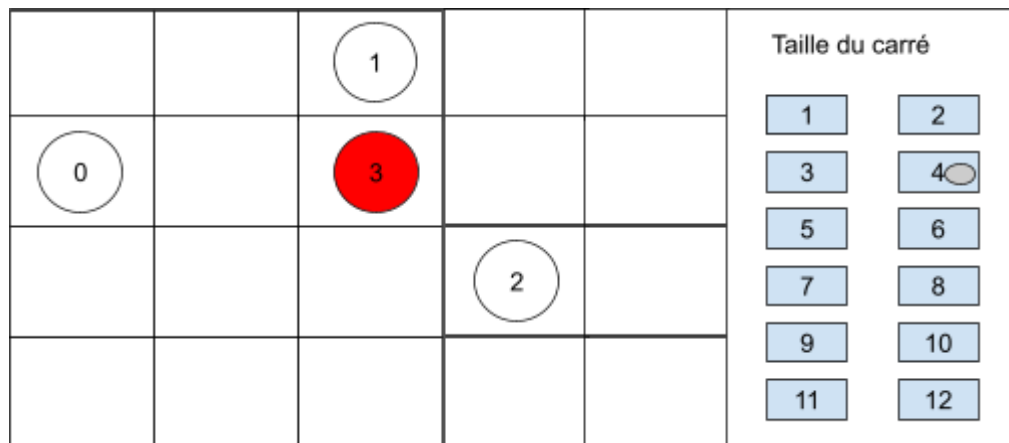
Si un objet ne respecte pas la contrainte, une LED rouge s'allume, pour enlever la LED il faut replacer l'objet.



Projection:

La projection va permettre de configurer le problème selon notre souhait

A la place des LED on peut utiliser le vidéoprojecteur pour mettre les objets en rouge

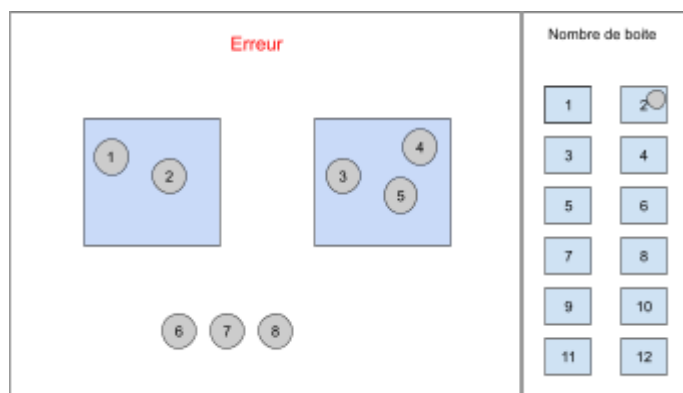


### Problème de Shurr

Comme tangiSense, on fait des boites projeter par le système, avec des objets représentant les différentes billes numérotées. Les objets ne sont pas obligés d'être toutes dans des boîtes, alors le système ne les prendra pas en compte. Mais attention si l'utilisateur utilise l'objet 8 il doit utiliser l'objet 3

Si une contrainte n'est pas respectée (par exemple:  $x=2*y$ ) alors un message d'erreur s'affiche

le nombre de boites n'est pas fixe il peut varier si on le modifie dans le système



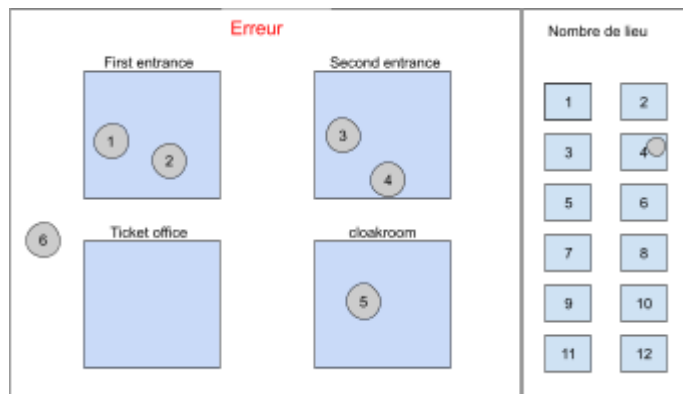
### Problème du Cinéma (ou tout autre organisation des tâches )

Comme tangiSense, on fait des boites projeter par le système, avec des objets représentant les différentes personnes, les personnes sont représenté par des numéros (identifiant). Les objets ne sont pas obligés d'être toutes dans des boîtes, alors le système ne les prendra pas en compte.

Si une contrainte n'est pas respectée (par exemple: il y a personne à ticket office) alors un message d'erreur s'affiche

le nombre d'emplacements peut être changé en fonction de la demande

des petits objets peuvent être placés pour indiquer le nombre de boites ou être fait par le système



### Problème du déménageur

Comme tangiSense, on fait des boîtes projeter par le système, avec des objets représentant les différentes personnes. Les objets ne sont pas obligés d'être toutes dans des boîtes, alors le système ne les prendra pas en compte.

Si une contrainte n'est pas respectée (la somme de la taille des objets dans une boîte dépasse la capacité de la boîte) alors un message d'erreur s'affiche

le nombre de boîtes peut être changé en fonction de la demande

des petits objets peuvent être placé pour indiquer le nombre de boîtes ou être fait par le système

