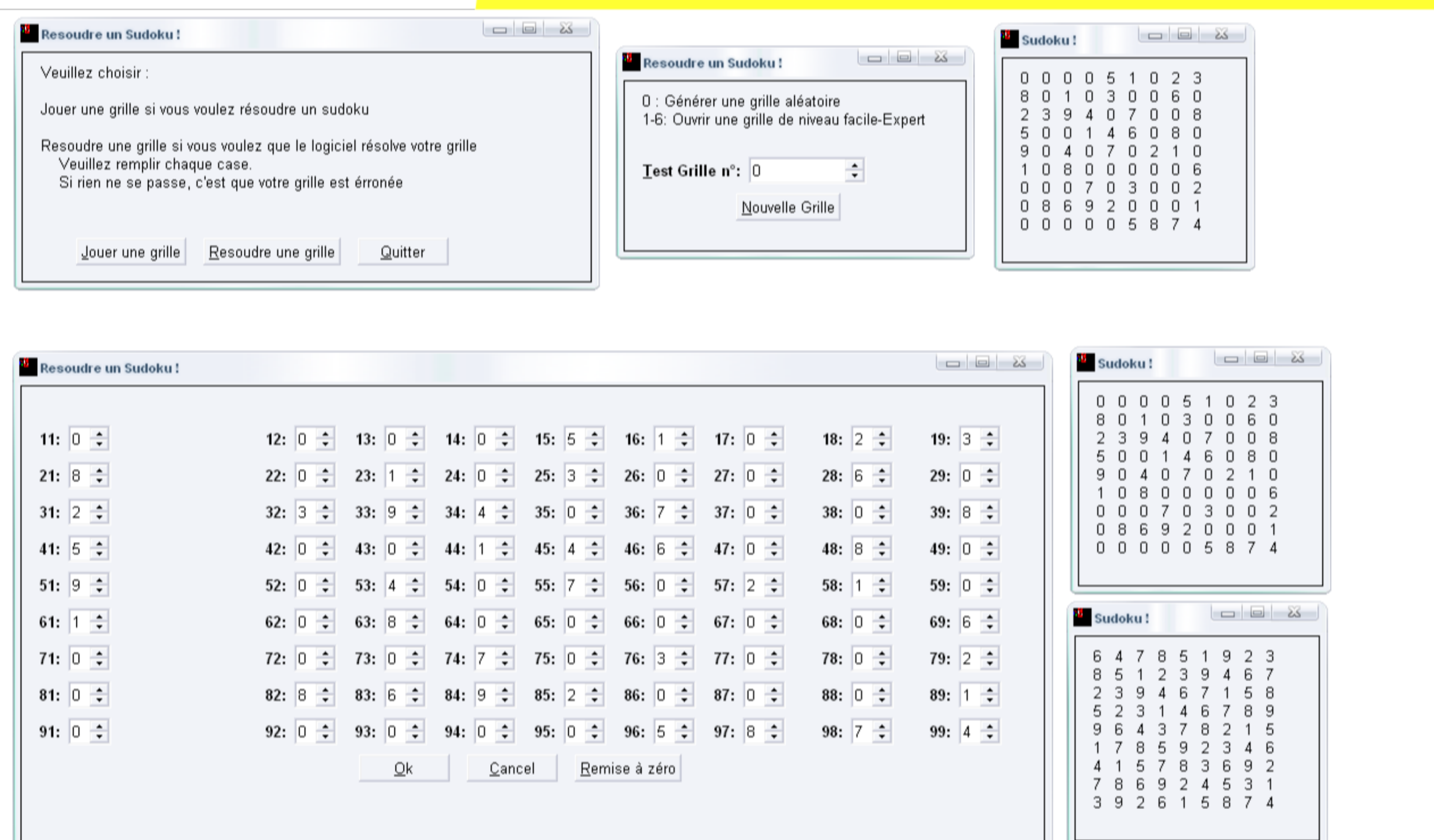


# Résolution et génération de Sudoku

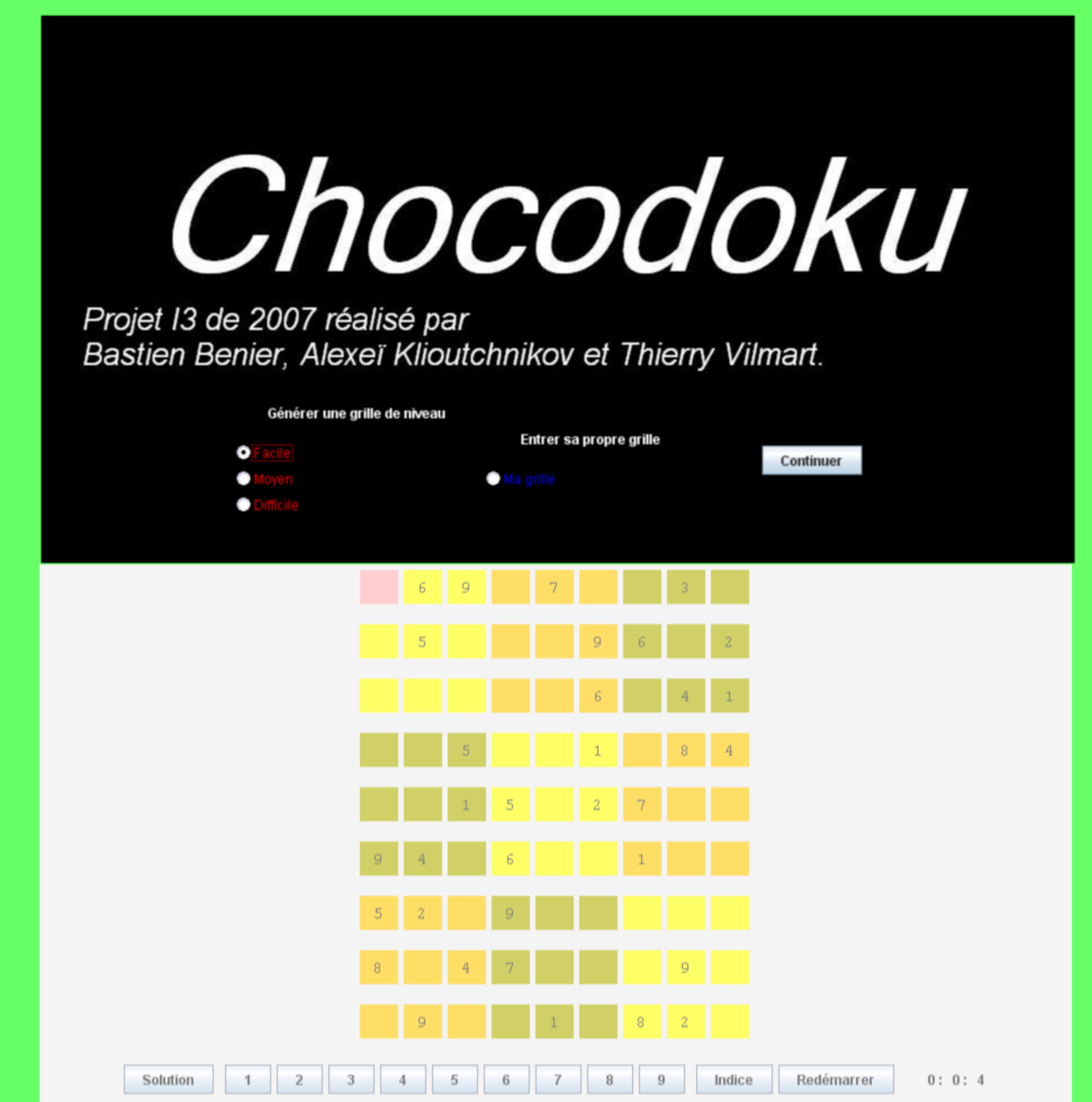
## Programmation logique par contraintes

EN LANGAGE PROLOG



5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

EN LANGAGE JAVA



POURQUOI ?

ASPECT DECLARATIF DU LANGAGE

écrire un programme en prolog consiste à bien déclarer le problème (sous forme de prédicats).

%Fonction vérifiant que la longueur des éléments d'une liste est 9  
NeufElements([]).

NeufElements([X|R]) :- length(X,9),NeufElements(R).

PUISSANCE DU RETOUR ARRIERE "BACKTRACKING"

La machine prolog se charge de mémoriser tous les états de variables (on peut revenir en arrière si la recherche de solutions n'aboutit pas)

PROGRAMMATION LOGIQUE AVEC CONTRAINTES

Il est possible de fixer des contraintes qui sont examinées à chaque pas élémentaire lors de la recherche de solutions, ce qui rend la recherche beaucoup plus efficace.

%Toutes les valeurs d'une ligne doivent être différentes  
all\_different(List).

```
% Résout et choisit une solution
sudokuContrainte(L) :- domaines(L),
    contraintes(L),
    choisirSol(L).
```

```
for (int i = 0; i < n; i++)
{
    pb.post(pb.allDifferent(cols[i]));
    pb.post(pb.allDifferent(rows[i]));
}
```