

# Le jeu de Sokoban

---

Vous connaissez certainement ce jeu :

Un terrain rectangulaire. Certaines cases sont occupées par des murs, qui bloquent le passage. Sur le terrain, il y a des caisses. il y a un personnage déplacé (4 directions) par le joueur qui peut aller dans une case vide voisine ou pousser une série de caisses, si il y a une case vide derrière objectif : amener toutes les caisses vers des destinations indiquées.

<https://www.jeux-gratuits.com/jeu-sokoban.html>

## Exemple de situation

	0	1	2	3	4	5	6	7	8	
0	.	.	#	#	#	#	.	.	.	
1	#	#	#	.	.	#	#	#	#	# = mur
2	#	.	.	.	.	.	C	.	#	P = personnage
3	#	.	#	.	.	#	C	.	#	C = caisse
4	#	.	x	.	x	#	P	.	#	x = destination
5	#	#	#	#	#	#	#	#	#	. = case vide

## Objectif généraux du programme

Permettre à un joueur de mener des parties sur des plateaux différents.

Les plateaux seront obtenus à partir de différentes sources

- des fichiers
- une base de données.

## Objectif 1 : Un jeu jouable.

- coder en dur un plateau de jeu puis permettre les interactions avec le personnage.

## Objectif 2 : chargement du plateau de jeu à partir d'un fichier.

## Objectif 3 : création d'une API REST distante

Créer une API distante et la consommer pour charger les map.

## Objectif 4 en option : chargement du plateau à partir d'un BD et administration.

On veut maintenant mettre les plateaux dans une base SQLite (par exemple).

BOARDS

board_id	name	rows	cols
----------	------	------	------

board_id	name	rows	cols
"simple"	"A simple board"	5	10
"difficult"	"A really tough one"	...	...

## ROWS

board_id	row	description
"simple"	0	"#####"
"simple"	1	"#x.x#....#"
"simple"	2	"#...CC.P.#"
"simple"	3	"#.....#"
"simple"	4	"#####"

Dans votre projet, définir une autre classe Administrator :

## ADMINISTRATION INTERFACE - USE WITH CAUTION

1. Create new database
2. List boards
3. Show board
4. Add board from file
5. Remove board from database [DANGEROUS]
6. Quit.

## Condition de remise

Sur git, screencast de votre application, dépôt du projet clean et d'une APK