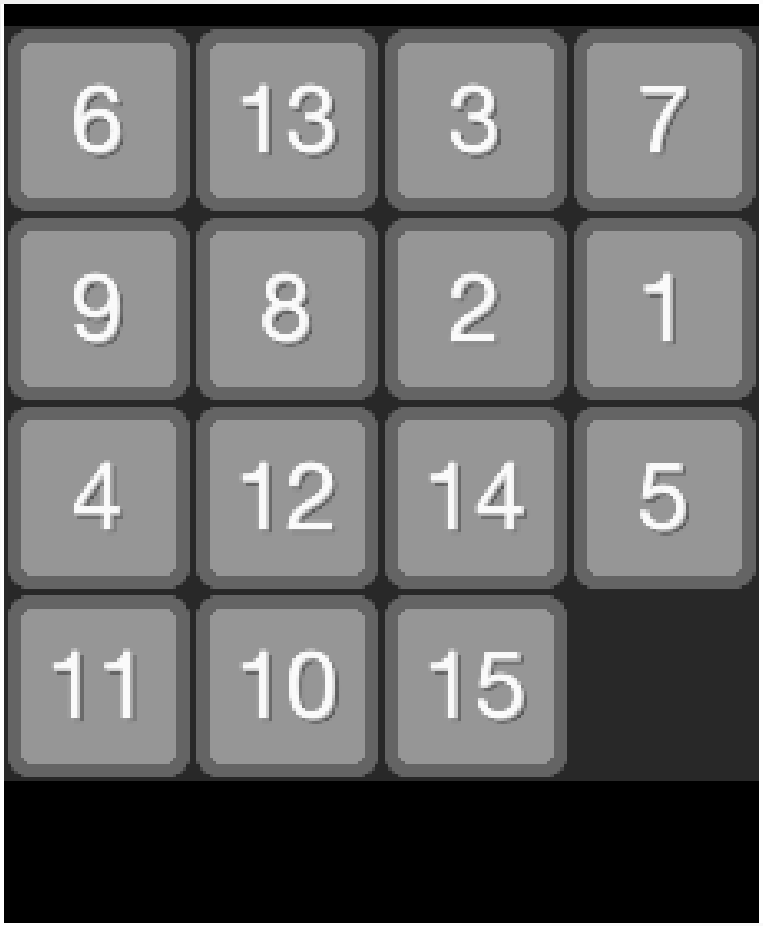
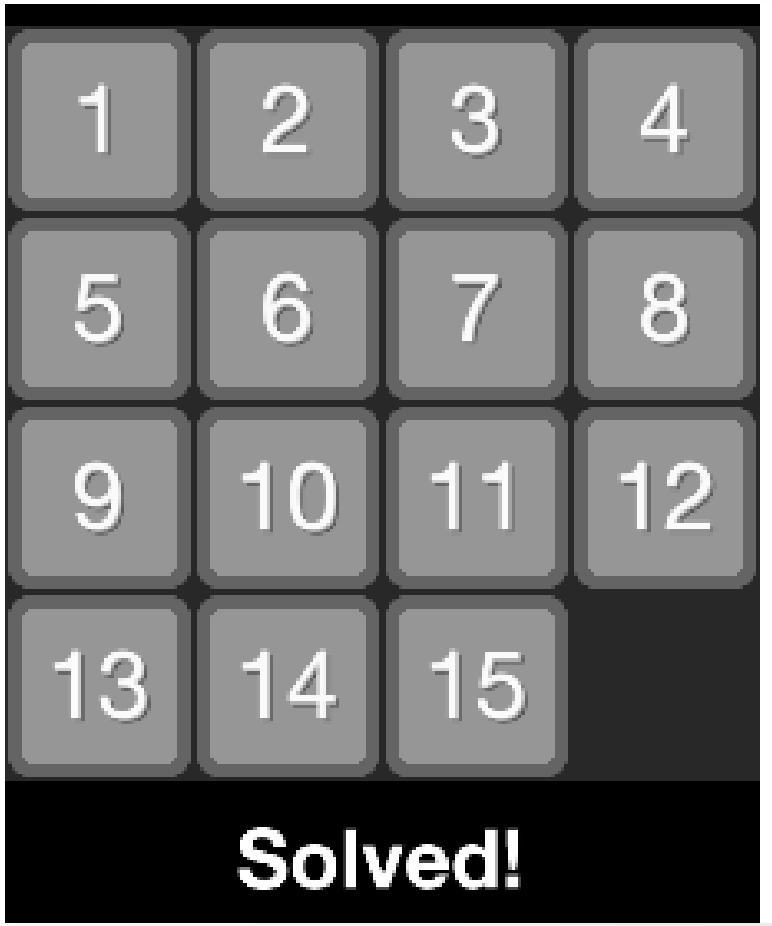
Game 15

- Fifteen puzzle: 4x4 tiles





```
Subalg. searchSpace( initialConfig )
    reachedConfig ← { initialConfig }
    unExpandedConfig ← { initialConfig }

    while unExpandedConfig ≠ ϕ do
        config := extractOne (unExpandedConfig)
        @for any valid successor succ of config do
            if succ ∉ reachedConfig then
                if isFinal(succ) then
                    // … !! process solution
                endif
                reachedConfig ← {succ}∪ reachedConfig
                unExpandedConfig ← {succ} ∪ unExpandedConfig
            endif
        endfor
    endWhile
End_searchSpace
```

Robot in a maze (1)

Consider a maze (rectangular shape) with occupied cells (X) and free cells (*). Consider a robot (R) in this maze, and a goal position in this maze.

(a) Verify if the robot can reach the goal position.

(b) Determine a path (if exists).

(c) Determine the shortest path (if exists).

X	*	X	X	*	*	*
*	X	*	*	X	*	*
G	*	*	*	*	*	*
*	X	*	S	*	*	X
*	X	*	*	*	*	X
*	X	*	*	X	*	*
*	X	*	X	*	*	*

Robot in a maze (2)

Consider a maze (rectangular shape) with occupied cells (X) and free cells (*). Consider a robot (R) in this maze

(a) Verify if the robot can get out of the maze (can reach any of the margins).

(b) Determine a path (if exists) to get out of the maze.

(c) Determine the shortest path (if exists) to get out of the maze.

X	X	X	X	X	X	X
X	S	*	*	*	*	*
X	X	X	*	X	X	X
*	*	*	*	X	X	X
X	X	X	X	X	X	X
X	X	X	X	X	X	X
X	X	X	X	X	X	X