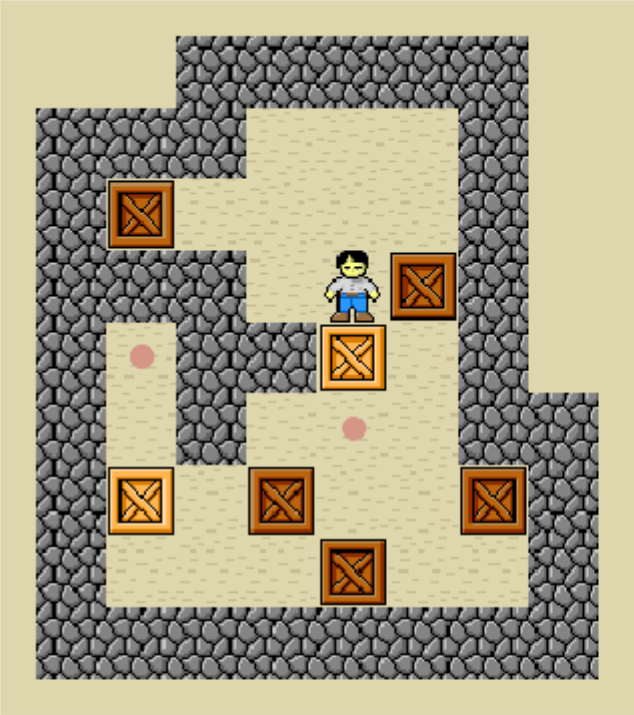
Developers Document



SOKOBAN

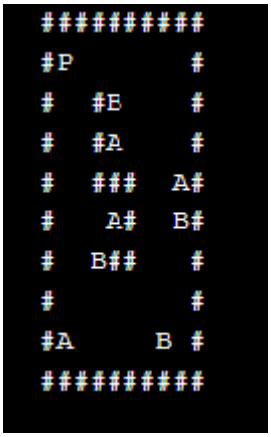
倉庫番

Sokoban (倉庫番, "warehousekeeper") is a puzzle video game in which the player pushes boxes around in a warehouse.
The goal is trying to get the boxes to the targer squares which are called "storage locations"

- CONTENTS :

1.Goal of the program :

Our goal in this program is to draw a map surrounded by a wall and create a character moving around this map in order to place the boxes to the storage squares.
If we put all boxes using the character or the player in the right place then we win the game.



2.Inputs and Outputs :

2.1.Inputs :

- Cordinates of the map → 2D dynamic array : width and height (column and a row).

- Number of boxes "B" and number of storage squares "A" (should be equal A = B).

- Map name.

- Cordinates of the player position : position Xp and Yp.

- Cordinates of the squares position : position Xp and Yp.

- Initialized counter to count how many boxe is already in the storage square.

2.2. Outputs :

- Drawn map with all caracters included

walls = #
Player = P
Boxes = B
Storage Squares = A

3. Description of the program :

In preparation for this game we have to follow those steps :
First of all :

- **Define number of boxes :** input an integer.
- **Define number of storage squares :** input an integer.
- **Create the map :** we input both width and height constants by using numbers of rows and columns. We have called it ‘’map’’
- **Determine player position :** void function includes IF statement to assure that Xp and Yp are inferior to width and height respectively. Note that Xp and Yp should be pointers because the player position is changeable in every single move
- **Determine "Storage square" position :** void function includes strrchr() function in to locate the occurrence of the characters ‘’A’’ in the map. It returns pointers.
- **Determine "Boxes" position :** void function includes strrchr() function in to locate the occurrence of the boxes ‘’B’’ in the map. It returns pointers.
- **Move the player character :** using the enumerated type function.
  So each case do a move : case 'Q': → move character to the left
  case 'D': → move character to the right
  case 'Z': → move character up
  case 'S': → move character down

- **Memory allocation and deallocation :** a function to store variables of 2D array string (the map).
  And when we don't need this storage again we use the deallocation so The variables of the map get released.

- **Counter :** counter function to count how many boxe is already in the storage square + for loop to repeat this step after every boxe take the new right position.

    - **Main function :** Collect all the previous function to work under this main function

  + implement a file handling includes a text which concern the map coordinates.
  + If all boxes are in their right squares → **print "**we won the game''