**Sokoban**

**1. Introduction**



Menu

This is a simple Sokoban version, made with Rapid Q, with ten built in levels; with the attached design tool it is easy to make your own levels. To deploy those levels needs a minimum amount of effort.

To play the game merely the **Sokoban\_GB.EXE** file is needed (**Sokoban.EXE** is the Dutch version).

To design levels merely the **SokobanBuilder.EXE** file is needed.

To make your own Sokoban edition you need Rapid Q and the resources in the ZIP; feel free to make your own versions of the graphic elements of course. The rest of this text is a step by step guide for making your own edition of the game.

**2. Preparation**

Make a *directory* (preferably **C:\Sokoban**), copy the ZIP there and then unpack it there. Compile **SokobanBuilder.EXE** from the BAS file (you might want to make a desktop link to it). Compile **Sokoban\_GB.EXE** from the BAS file, then save the BAS file under a new name (replace the **\_GB** bit); use that for making your own edition.

**3. Building a level**

The level tool is a simple drag & drop application, it contains a few checks (one warehouse worker placed, number of crates versus number of targets) but it is all too easy to make unplayable levels; sadly there is no good way to check for that.

Levels are saved in the form of three files; use a ONE WORD level title when saving (use no spaces or odd characters in the title save name). The in-game level title can be edited later, in the TXT file or game source code.

The three files used are;

* SBL matrix format used for the design tool
* BMP level map
* TXT text format data for use in the game source code

The text file (based on the level shown above) looks like this;

 data 1,1,5,5,5,5,5,5,5,5,5,5,1,1, 1,6,1,1,1,1,1,1,1,1,1,1,6,1, 5,1,6,6,1,1,1,1,1,1,6,6,1,5, 5,1,6,5,1,1,1,1,1,1,5,6,1,5

 data 5,1,1,1,1,1,2,1,2,1,1,1,1,5, 5,1,1,1,1,2,1,2,1,2,1,1,1,5, 5,1,1,1,2,1,2,1,2,1,1,1,1,5, 5,1,1,1,1,2,1,2,1,2,1,1,1,5

 data 5,1,1,1,2,1,2,1,2,1,1,1,1,5, 5,1,1,1,1,2,1,2,1,1,1,1,1,5, 5,1,6,5,1,1,1,1,1,1,5,6,1,5, 5,1,6,6,1,1,3,1,1,1,6,6,1,5

 data 1,6,1,1,1,1,1,1,1,1,1,1,6,1, 1,1,5,5,5,5,5,5,5,5,5,5,1,1, Vrijbaan

The level is saved in the form of the element numbers with four columns of data per line. Change the level name here to what it should be / look like in-game.

**4. Making your own edition**

Several icons have been included in the ZIP (of course you could make your own), so load your renamed game source code and pick an icon for it. Scroll down to the end of the code and select the level to be replaced;



Now open the TXT file containing the level you want (using Notepad or some such) and select the data in it, then copy it to the buffer (press **Ctrl** + **C**). Switch to Rapid Q and overwrite the old level data (press **Ctrl** + **C**).

To personalize your version more the main colours used are placed at the start of the source code;

 air=&ha04030 : ground=&hd4a6b4 : tekst=&h247efc

The advantage of using hexadecimal values is that for colour they are built up thus;

&h blue component green component red component

With each component having a value from 00 (zero, darkest) to FF (255, brightest) it is easy to quickly make the colour you want. As an example change the value of **air** to &h30ffff and run the program;

  

All backgrounds (except the menu graphic) become a bright yellow. **Ground** is used for the base floor tiles and **tekst** (text in Dutch) is used for the menu ‘level select’ line.

Changing the **ground** colour only changes the base floor tile colour, so (see the picture) you’ll need to change the colour of the other floor tiles;





Of course there is a lot more that can be changed[[1]](#footnote-1). Graphics, design & programming;

Ad C. Lockhorst

Lupinepad 7

3202 BT Spijkenisse

alockhorst@chello.nl

P.S. yes the text in the pictures here is in Dutch, I couldn’t be arsed to translate it. The source code is mostly English, it contains comments in English too. One obvious and handy exception is the use of **tekst** – Rapid Q would flip if you used text for that.

1. Actually… I only made **Sokoban** as a basis for a game I’m working on which I’m calling **LingoBan**; in it you move SYMBOL crates about. It’s almost done!

   [↑](#footnote-ref-1)