

WIKIPEDIA

Sokoban

Sokoban (倉庫番 *sōko-ban*, "warehouse keeper") is a type of puzzle video game, in which the player pushes crates or boxes around in a warehouse, trying to get them to storage locations.

Sokoban was created in 1981 by Hiroyuki Imabayashi, and published in December 1982 by Thinking Rabbit, a software house based in Takarazuka, Japan.

Contents

Rules

Selected Sokoban releases 1982–2016

Reception

Implementations of Sokoban

Scientific research on Sokoban

Sokoban variants

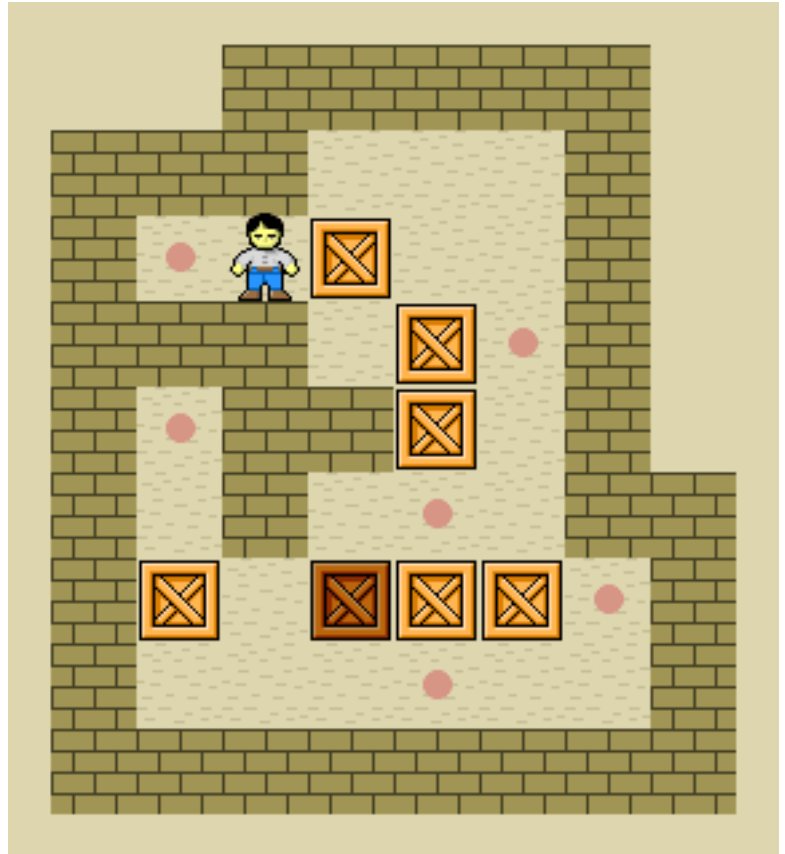
See also

References

External links

Rules

The game is played on a board of squares, where each square is a floor or a wall. Some floor



A Sokoban puzzle being solved.

squares contain boxes, and some floor squares are marked as storage locations.

The player is confined to the board, and may move horizontally or vertically onto empty squares (never through walls or boxes). The player can also move into a box, which pushes it into the square beyond. Boxes may not be pushed into other boxes or walls, and they cannot be pulled. The number of boxes is equal to the number of storage locations. The puzzle is solved when all boxes are at storage locations.

Selected Sokoban releases 1982–2016

Year	Title	Country	Platform
1982	Sokoban (倉庫番)	Japan	<u>NEC PC-8801</u>
1983	Sokoban [Extra Edition] (倉庫番[番外編])	Japan	<u>NEC PC-8801</u>
1984	Sokoban 2 (倉庫番2)	Japan	<u>NEC PC-8801</u>
1988	Soko-Ban	US	<u>IBM-PC</u> and <u>compatible</u>
1989	Soko-ban Perfect (倉庫番Perfect)	Japan	<u>NEC PC-9801</u>
1991	Soko-ban Revenge (倉庫番Revenge)	Japan	<u>NEC PC-9801</u>
2016	Sokoban Touch (倉庫番Touch)	Japan, US	<u>Android</u> and <u>Apple iOS</u>

In 1988 *Sokoban* was published in US by Spectrum HoloByte as *Soko-Ban* for the IBM-PC and compatible computers, Commodore 64, and Apple II series. (The game title screen text "*Copyright 1984 ASCII Corp.*" is a reference to copyrights, not the

publishing date.) A review in *Computer Gaming World* praised the game for being "pure and simple, very playable and mentally challenging", citing its addictive qualities.^[2] It was also reviewed in *Dragon* giving it 4½ out of 5 stars.^[3]

Reception

Sokoban was a hit in Japan, and had sold over 400,000 units in that country by the time *Spectrum HoloByte* imported it to the United States.^[4]

Implementations of Sokoban

Implementations of Sokoban have been written for numerous computer platforms, including almost all home computer and personal computer systems. Versions also exist for video game consoles, mobile phones, graphic calculators, digital cameras and electronic organizers.

Scientific research on Sokoban

Sokoban can be studied using the theory of computational complexity. The problem of solving Sokoban puzzles has been proven to be NP-hard.^{[5][6]} Further work showed that it was significantly more difficult than NP problems; it is PSPACE-complete.^[7] This is also interesting for artificial intelligence researchers, because solving Sokoban can be compared to the automated planning that needs to be done by a robot that moves boxes in a warehouse.

Sokoban is difficult not only due to its branching factor (which is comparable to chess), but also its enormous search tree depth; some levels can be extended indefinitely, with each iteration requiring an exponentially growing number of moves and pushes.^[8] Skilled human players rely mostly on heuristics; they are usually able to quickly discard futile or redundant lines of play, and recognize patterns and subgoals, drastically cutting down on the amount of search.

Some Sokoban puzzles can be solved automatically by using a single-agent search algorithm, such as IDA*, enhanced by several techniques which make use of domain-specific knowledge.^[9] This is the method used by *Rolling Stone*,^[10] a

Sokoban solver developed by the University of Alberta GAMES Group. The more complex Sokoban levels are, however, out of reach even for the best automated solvers.^[11]

Sokoban variants

Several puzzles can be considered variants of the original Sokoban game, in the sense that they all make use of a controllable character who pushes boxes around a maze.

Alternative tilings: In the standard game, the mazes are laid out on a square grid. Several variants apply the rules of Sokoban to mazes laid out on other tilings. *Hexoban* uses regular hexagons and *Trioban* uses equilateral triangles.

Multiple pushers: In the variants *Multiban* and *Interlock* the player can control multiple characters.

Alternative goals: Several variants adjust the requirements for completing a level. For example, in *Block-o-Mania* the boxes have different colours, and the goal is to push them onto squares with matching colours. *Sokomind Plus* implements a similar idea, with boxes and target squares uniquely numbered. In *Interlock* and *Sokolor*, the boxes also have different colours, but the goal is to move them so that similarly coloured boxes are adjacent. In *CyberBox*, each level has a designated exit square, and the goal is to reach that exit. In a variant called *Beanstalk*, the elements of the level must be pushed onto the goal in a fixed sequence.

Additional game elements: *Push Crate*, *Sokonex*, *Xsok*, *Cyberbox* and *Block-o-Mania* all add new elements to the basic puzzle. Examples include holes, teleports, moving blocks and one-way passages.

Character actions: In *Pukoban*, the character can pull boxes in addition to pushing them.

Destructible walls: The early *Sokoban* (1982) (NEC PC-8801) game featured levels with destructible walls. In order to solve these levels, the player had to make some of the destructible wall elements disappear by walking up to the wall and push it from a certain side.

Unwalkable targets: Some variants make empty target tiles unwalkable, only becoming walkable when occupied by a crate. This is common for Sokoban-type levels in LaserTank and Chip's Challenge.

Reverse mode: The player solves the puzzle backwards, from the end to the initial position by pulling instead of pushing boxes. Standard Sokoban puzzles can be played in reverse mode, and the reverse mode solutions can be converted to solutions for the standard-mode puzzles. Therefore, reverse mode can also be instrumental in solving standard Sokoban puzzles.

See also

- Logic puzzle
- Sliding puzzle
- Transport puzzle
- Motion planning

References

1. "倉庫番[番外編]". PCマガジン (source code). August 1983. pp. 52–56.
2. Wagner, Roy (May 1988). "Puzzling Encounters". Computer Gaming World (47): 42–43.
3. Lesser, Hartley; Lesser, Patricia; Lesser, Kirk (April 1988). "The Role of Computers". Dragon (132): 80–85.
4. Low, Lafe (November 1988). "News Line; Made in Japan". inCider (43). 14, 15.
5. M. Fryers; M.T. Greene (1995). "Sokoban". *Eureka* (54).

6. Dor, Dorit; Zwick, Uri (1999). "SOKOBAN and other motion planning problems". *Computational Geometry*. **13** (4): 215–228. doi:10.1016/S0925-7721(99)00017-6 (<https://doi.org/10.1016%2FS0925-7721%2899%2900017-6>). ISSN 0925-7721 (<https://www.worldcat.org/issn/0925-7721>).
7. Joseph C. Culberson, [Sokoban is PSPACE-complete](http://cl-informatik.uibk.ac.at/teaching/ss07/alth/material/culberson97sokoban.pdf) (<http://cl-informatik.uibk.ac.at/teaching/ss07/alth/material/culberson97sokoban.pdf>). Technical Report TR 97-02, Dept. of Computing Science, University of Alberta, 1997. Also: <https://webdocs.cs.ualberta.ca/~joe/TR/TR97-02.ps> (<https://webdocs.cs.ualberta.ca/~joe/TR/TR97-02.ps>)
8. David Holland and Yaron Shoham, "Theoretical analysis on Picokosmos 17" (http://web.archive.org/web/20160607071224/http://www.abelmartin.com/rj/sokobanJS/sokoban-jd.blogspot/sokoban_lessons/picokosmos17/analysis.htm).
9. Andreas Junghanns, Jonathan Schaeffer (2001) [Sokoban: Enhancing general single-agent search methods using domain knowledge](http://www.sciencedirect.com/science?_ob=PublicationURL&_tockey=%23TOC%235617%232001%23998709998%23278096%23FLP%23&_cdi=5617&_pubType=J&_auth=y&_acct=C000050221&_version=1&_urlVersion=0&_userid=10&md5=1310880ee355fe1b42ef1722e0a96b15) (http://www.sciencedirect.com/science?_ob=PublicationURL&_tockey=%23TOC%235617%232001%23998709998%23278096%23FLP%23&_cdi=5617&_pubType=J&_auth=y&_acct=C000050221&_version=1&_urlVersion=0&_userid=10&md5=1310880ee355fe1b42ef1722e0a96b15), *Artificial Intelligence* **129**(1-2):219-251 (Special issue on heuristic search in artificial intelligence)

10. Junghanns, Andreas; Schaeffer, Jonathan (1997). "Sokoban: A Challenging Single-Agent Search Problem" (https://webdocs.cs.ualberta.ca/~jonathan/publications/ai_publications/soko.pdf) (PDF). In *IJCAI Workshop on Using Games as an Experimental Testbed for AI Research*. University of Alberta. pp. 27–36.
11. "Solver Statistics - Sokoban Wiki" (http://www.sokobano.de/wiki/index.php?title=Solver_Statistics). Retrieved 8 February 2013.

External links

- Official Sokoban site (<http://www.sokoban.jp/>) (in Japanese)
- The University of Alberta Sokoban page (<http://www.cs.ualberta.ca/~games/Sokoban/>)
- Virkkala, Timo (2011). *Solving Sokoban* (<http://weetu.net/Timo-Virkkala-Solving-Sokoban-Masters-Thesis.pdf>) (PDF) (Master's Thesis). University of Helsinki. Retrieved 24 September 2014.

Retrieved from "<https://en.wikipedia.org/w/index.php?title=Sokoban&oldid=911334087>"

This page was last edited on 18 August 2019, at 07:00 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.