Super Mario Bros. Is Harder/Easier than We Thought

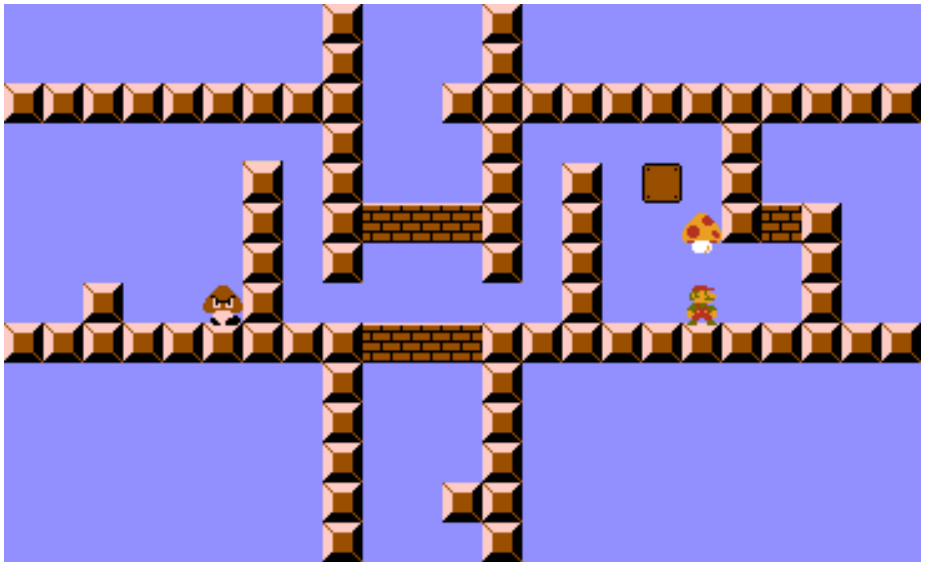
FUN 2016

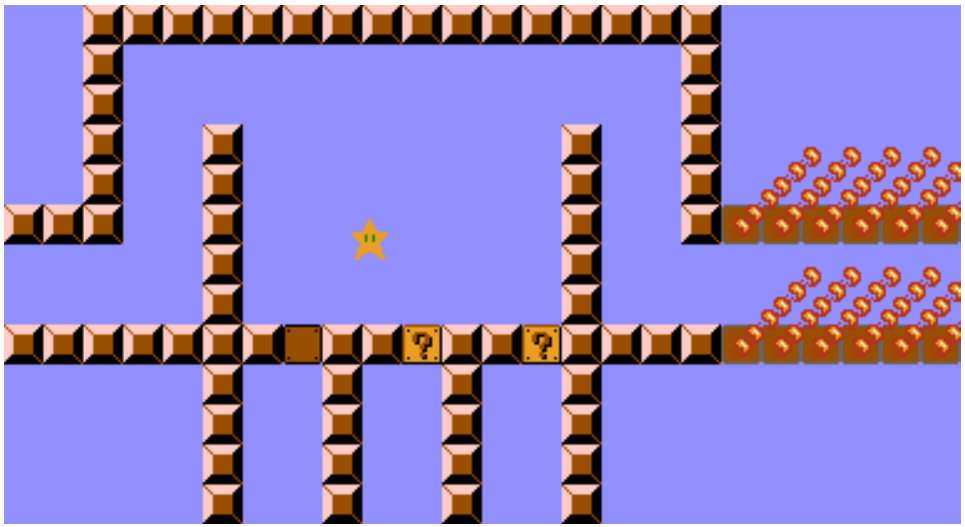
Erik D. Demaine, Giovanni Viglietta, Aaron Williams

La Maddalena – June 9, 2016

State of the art: FUN 2014

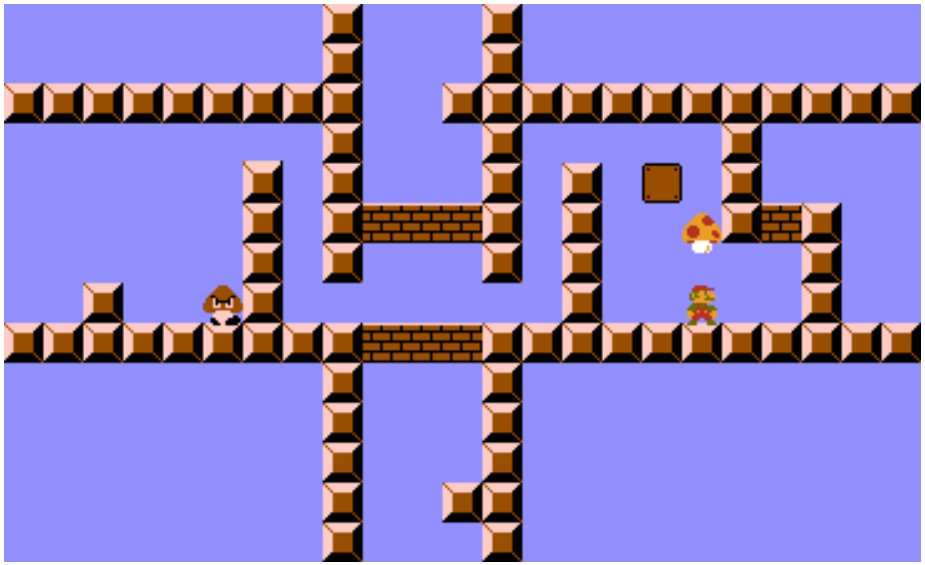
"Classic Nintendo games are (computationally) hard" [Aloupis-Demaine-Guo-Viglietta]

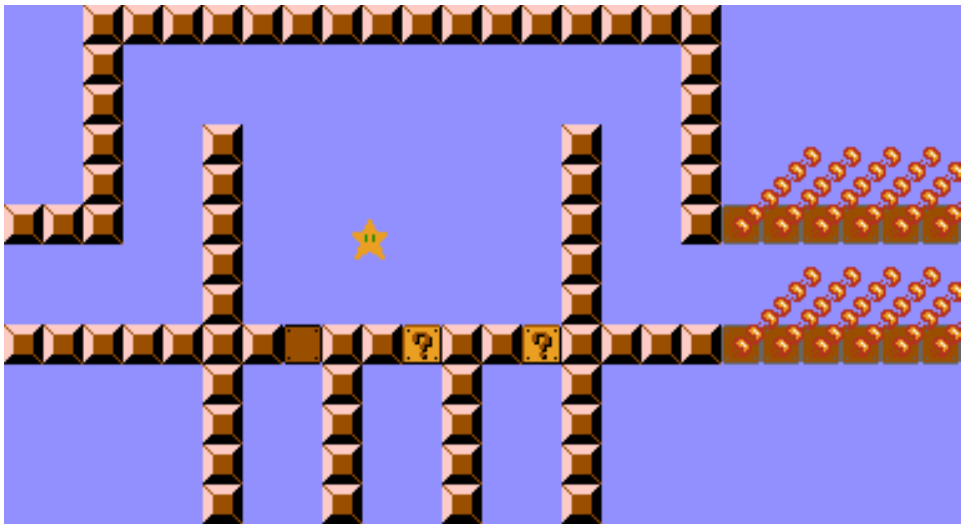




State of the art: FUN 2014

"Classic Nintendo games are (computationally) hard" [Aloupis-Demaine-Guo-Viglietta]



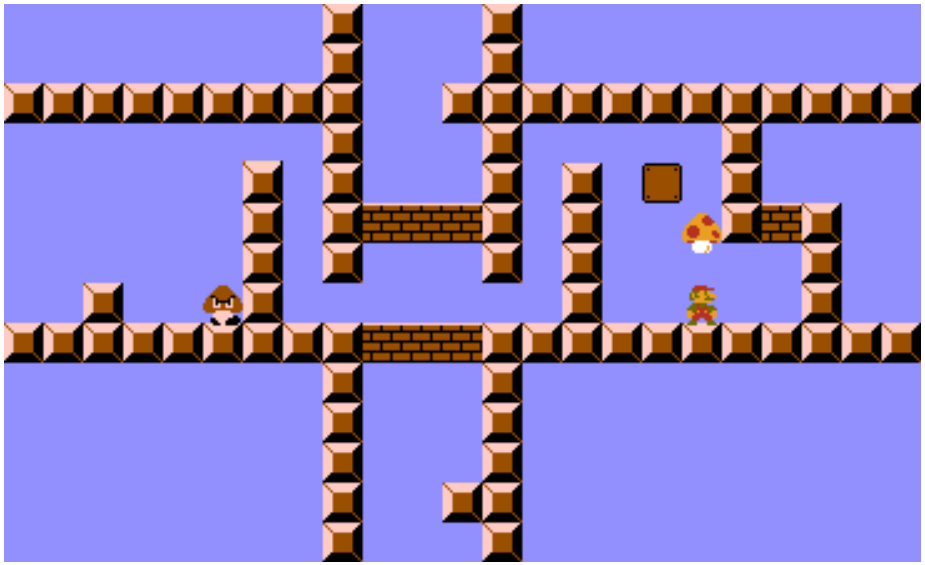


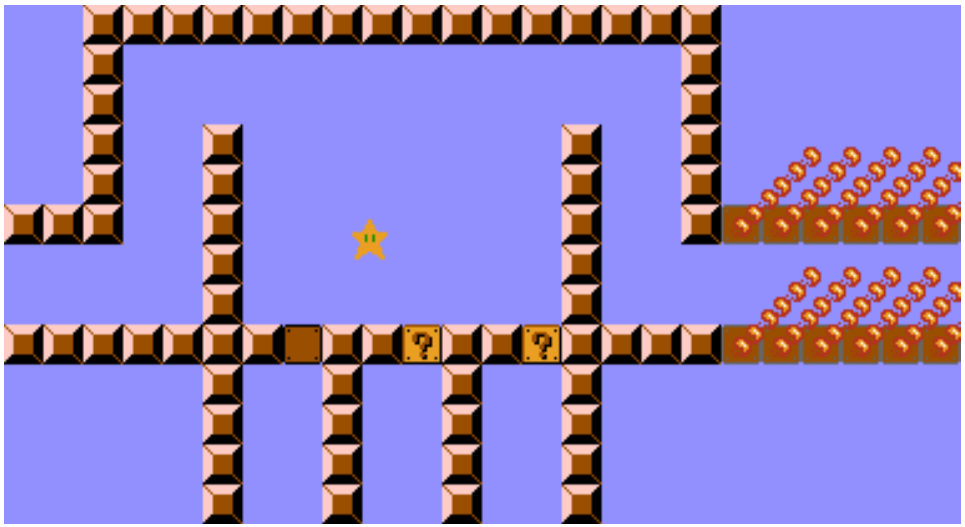
Downsides:

- Only NP-hardness of SMB was proved (PSPACE-completeness was left open)

State of the art: FUN 2014

"Classic Nintendo games are (computationally) hard" [Aloupis-Demaine-Guo-Viglietta]



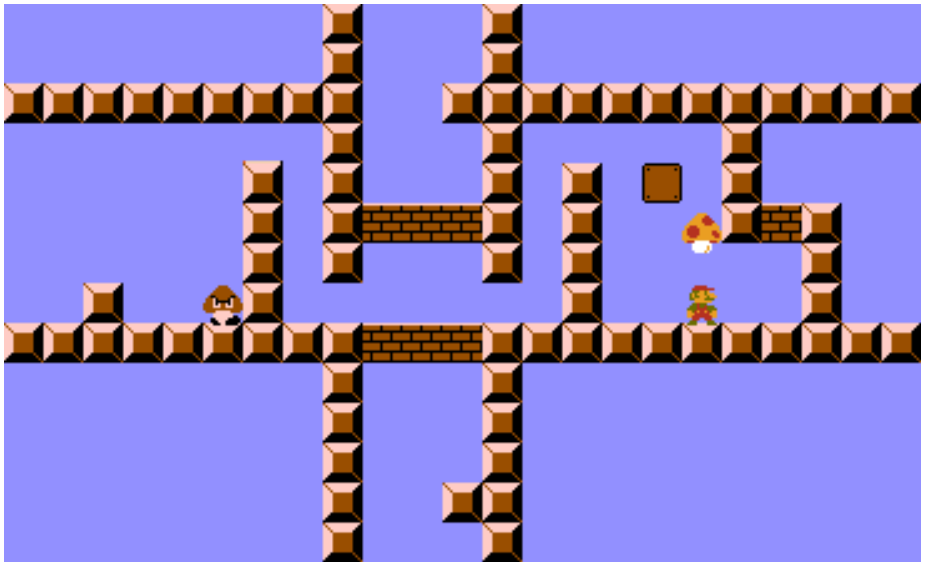


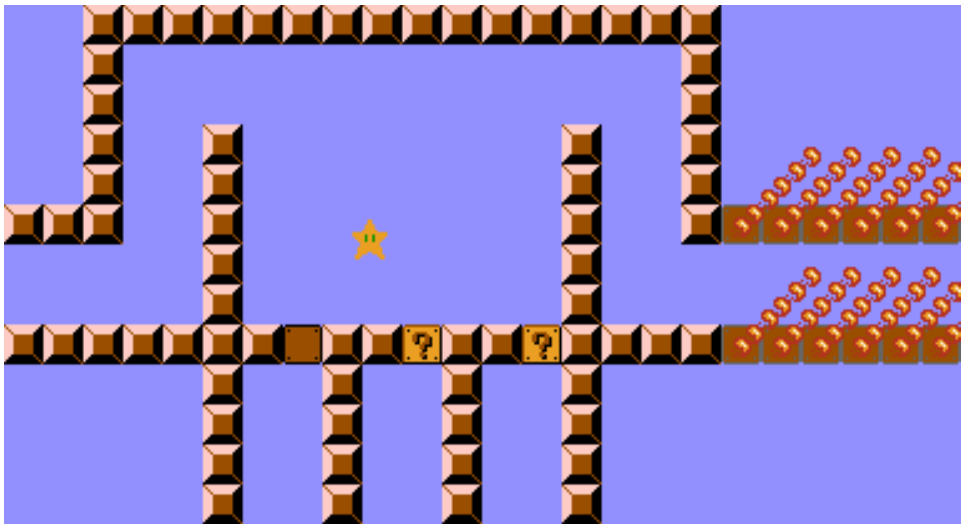
Downsides:

- Only NP-hardness of SMB was proved
  (PSPACE-completeness was left open)
- Levels were arbitrarily high
  (in SMB they have a fixed height)

State of the art: FUN 2014

"Classic Nintendo games are (computationally) hard" [Aloupis-Demaine-Guo-Viglietta]



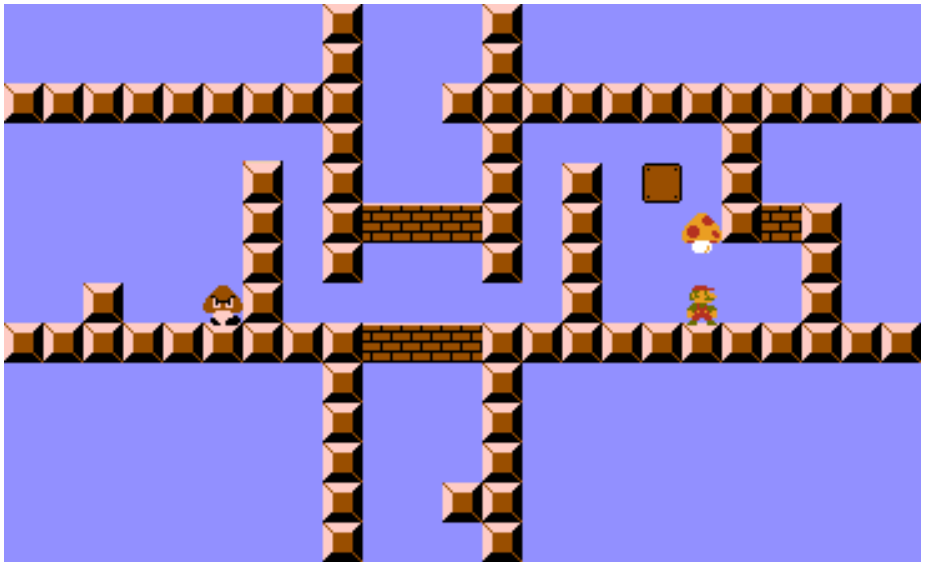


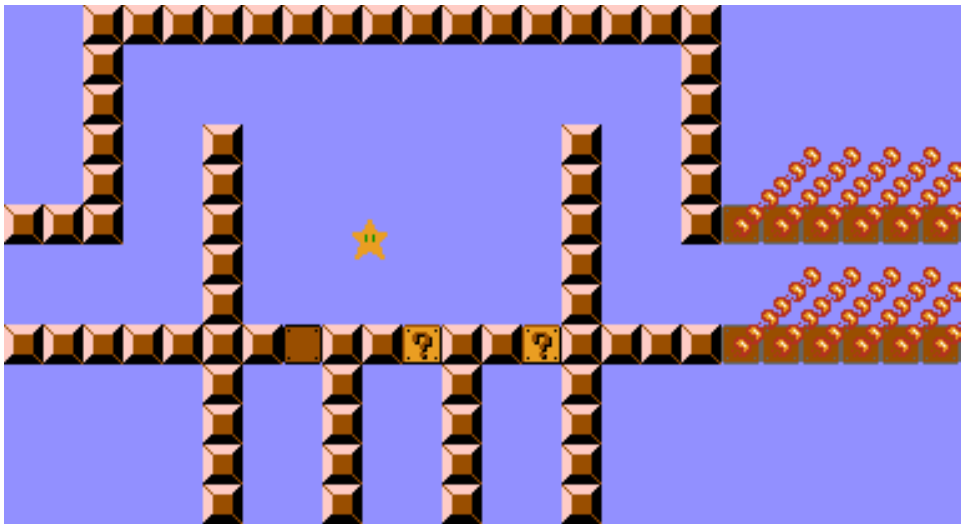
Downsides:

- Only NP-hardness of SMB was proved
  (PSPACE-completeness was left open)
- Levels were arbitrarily high
  (in SMB they have a fixed height)
- Off-screen objects were not reset
  (in SMB the "active screen" has fixed size)

State of the art: FUN 2014

"Classic Nintendo games are (computationally) hard" [Aloupis-Demaine-Guo-Viglietta]



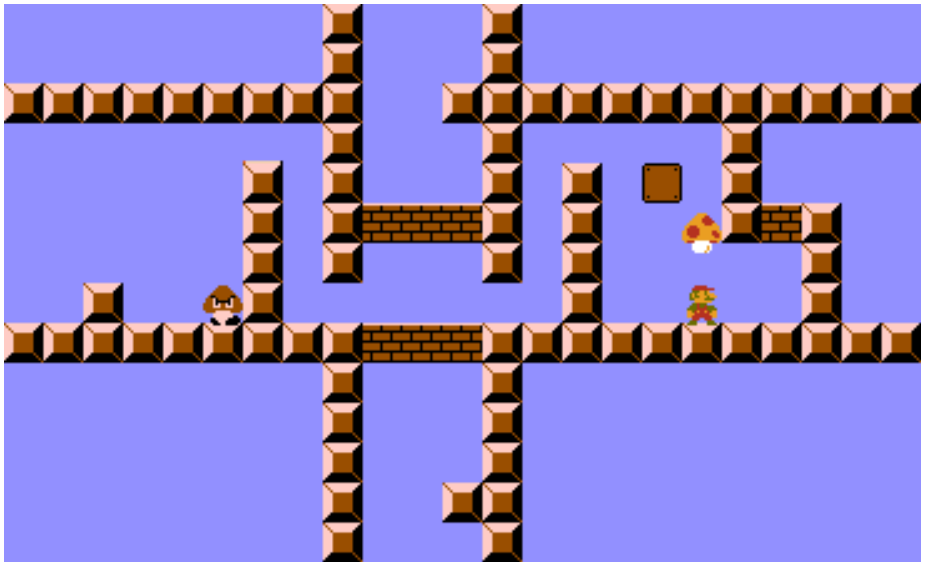


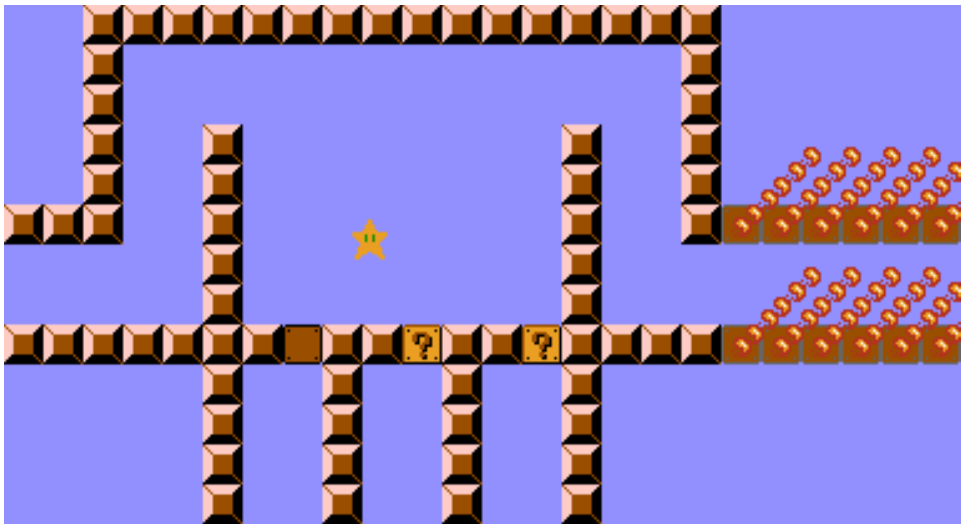
Downsides:

- Only NP-hardness of SMB was proved
  (PSPACE-completeness was left open)
- Levels were arbitrarily high
  (in SMB they have a fixed height)
- Off-screen objects were not reset
  (in SMB the "active screen" has fixed size)
- Mario was allowed to go back and forth
  (in SMB the screen can only move rightward)

State of the art: FUN 2014

"Classic Nintendo games are (computationally) hard" [Aloupis-Demaine-Guo-Viglietta]



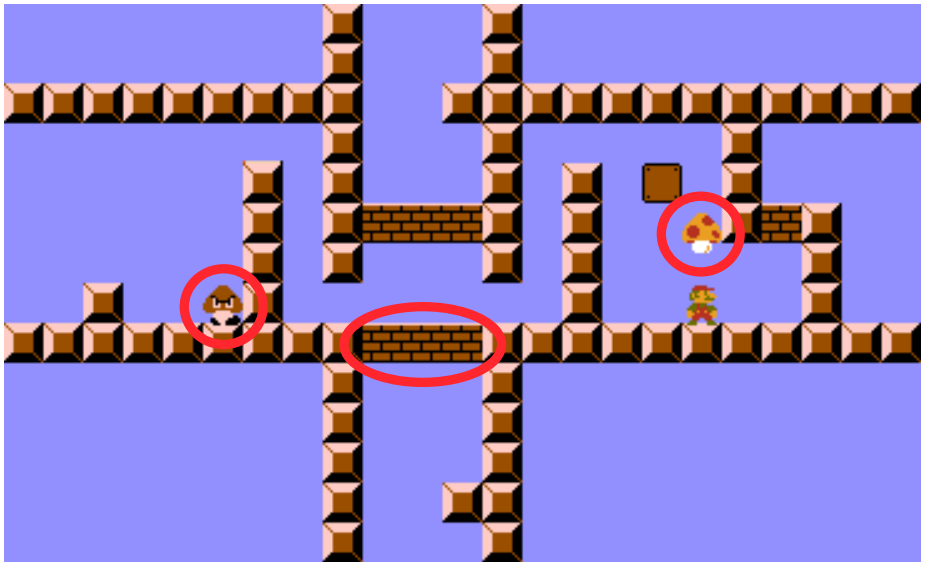


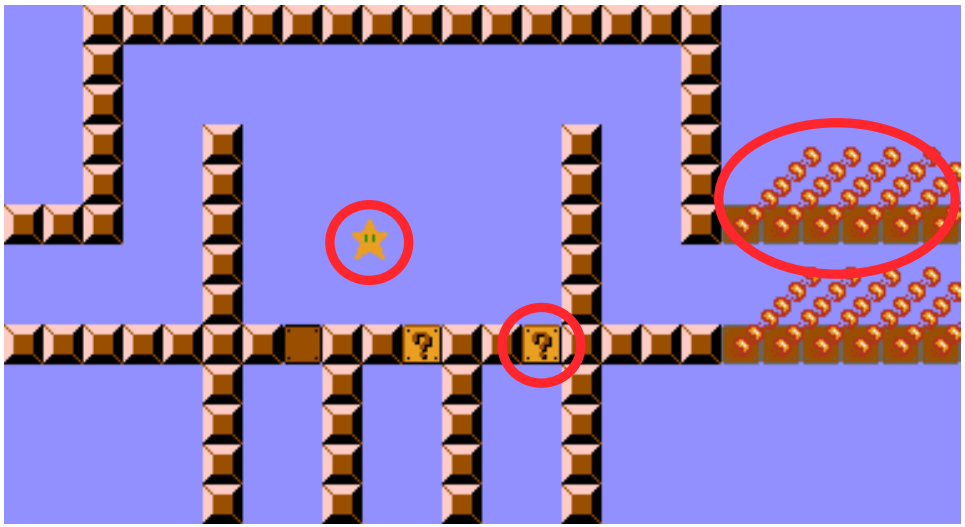
Downsides:

- Only NP-hardness of SMB was proved
  (PSPACE-completeness was left open)
- Levels were arbitrarily high
  (in SMB they have a fixed height)
- Off-screen objects were not reset
  (in SMB the "active screen" has fixed size)
- Mario was allowed to go back and forth
  (in SMB the screen can only move rightward)
- Lots of game elements were used
  (we would like to be more parsimonious)

State of the art: FUN 2014

"Classic Nintendo games are (computationally) hard" [Aloupis-Demaine-Guo-Viglietta]



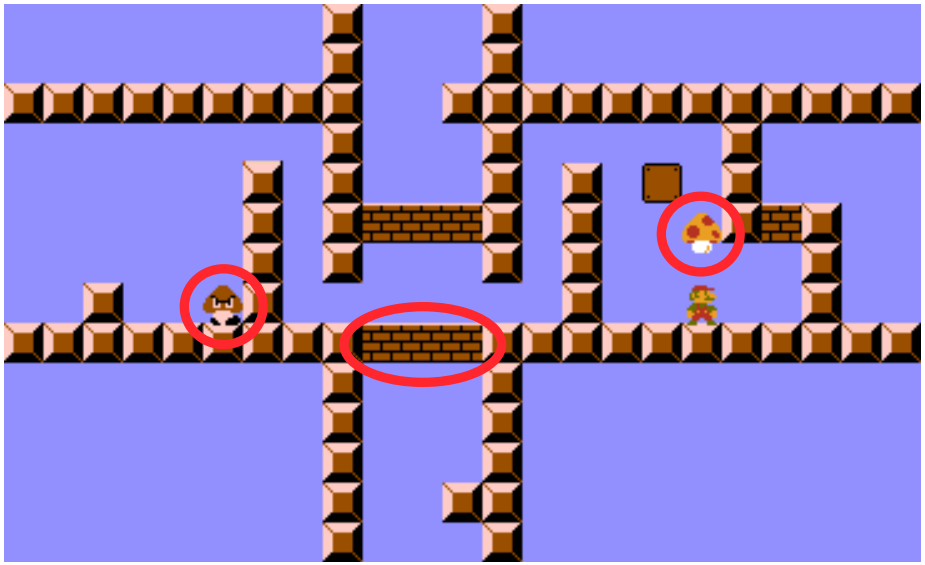


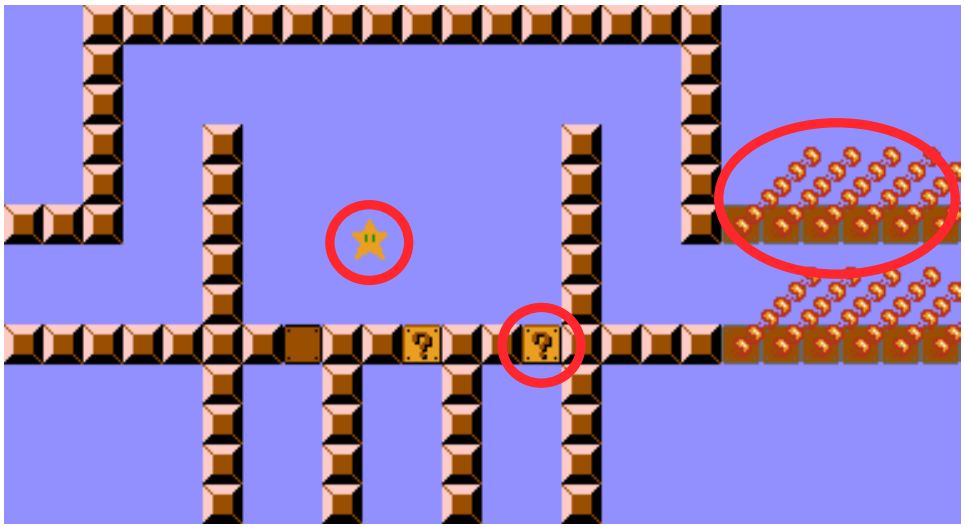
Downsides:

- Only NP-hardness of SMB was proved
  (PSPACE-completeness was left open)
- Levels were arbitrarily high
  (in SMB they have a fixed height)
- Off-screen objects were not reset
  (in SMB the "active screen" has fixed size)
- Mario was allowed to go back and forth
  (in SMB the screen can only move rightward)
- Lots of game elements were used
  (we would like to be more parsimonious)

State of the art: FUN 2014

"Classic Nintendo games are (computationally) hard" [Aloupis-Demaine-Guo-Viglietta]





Downsides:

- Only NP-hardness of SMB was proved
  (PSPACE-completeness was left open)
- Levels were arbitrarily high
  (in SMB they have a fixed height)
- Off-screen objects were not reset
  (in SMB the "active screen" has fixed size)
- Mario was allowed to go back and forth
  (in SMB the screen can only move rightward)
- Lots of game elements were used
  (we would like to be more parsimonious)
- Levels were not entirely glitch-proof
  (in SMB there are glitches that make
  the old crossover gadget ineffective)

Our results

- Rescuing the princess is in **P** if the standard rules of SMB apply
- Rescuing the princess is weakly **NP-hard** if the level format allows run-length encoding, even if
  - levels have height 2
  - only coins and pipes are used
  - only buttons  and  are used
- Reaching the flagpole in a level without losing lives is **PSPACE-complete** if objects off-screen are not reset

(All constructions are glitch-proof)

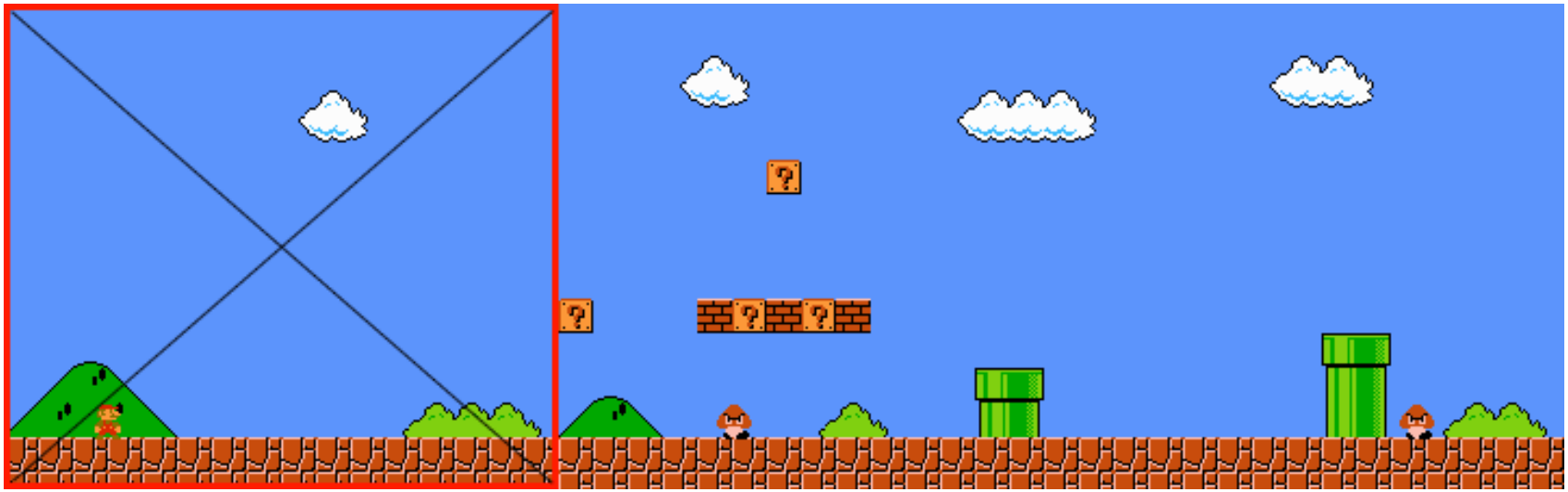
Reaching the flagpole is in P

Sketch of the proof:

Reaching the flagpole is in P

Sketch of the proof:

- The "active screen" has fixed size; it can contain at most 5 sprites (plus Mario) and everything off-screen is reset



Reaching the flagpole is in P

Sketch of the proof:

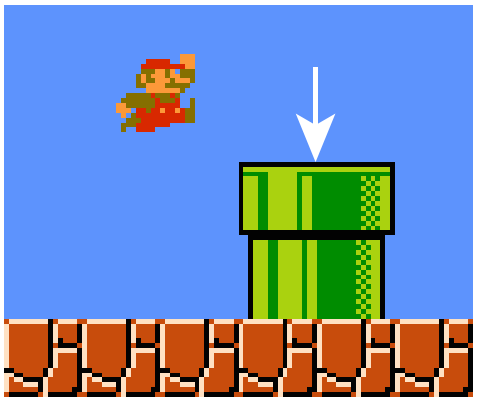
- The "active screen" has fixed size; it can contain at most 5 sprites (plus Mario) and everything off-screen is reset
- Pseudorandom numbers are generated at every frame, starting from a fixed initial seed

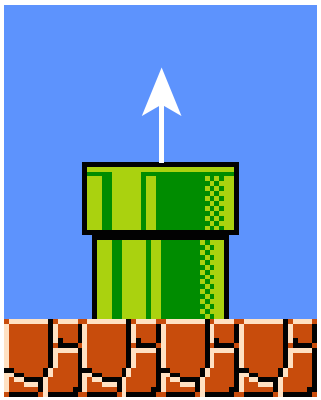
```
WBootCheck:   lda TopScoreDisplay,x       ;check each score digit in the top score
              cmp #10                     ;to see if we have a valid digit
              bcs ColdBoot                ;if not, give up and proceed with cold boot
              dex
              bpl WBootCheck
              lda WarmBootValidation      ;second checkpoint, check to see if
              cmp #$a5                    ;another location has a specific value
              bne ColdBoot
              ldy #WarmBootOffset         ;if passed both, load warm boot pointer
ColdBoot:     jsr InitializeMemory        ;clear memory using pointer in Y
              sta SND_DELTA_REG+1         ;reset delta counter load register
              sta OperMode                ;reset primary mode of operation
              lda #$a5                    ;set warm boot flag
              sta WarmBootValidation
              sta PseudoRandomBitReg      ;set seed for pseudorandom register
              lda #%00001111
              sta SND_MASTERCTRL_REG      ;enable all sound channels except dmc
```

Reaching the flagpole is in P

Sketch of the proof:

- The "active screen" has fixed size; it can contain at most 5 sprites (plus Mario) and everything off-screen is reset
- Pseudorandom numbers are generated at every frame, starting from a fixed initial seed
- Mario can instantly move to different locations through pipes

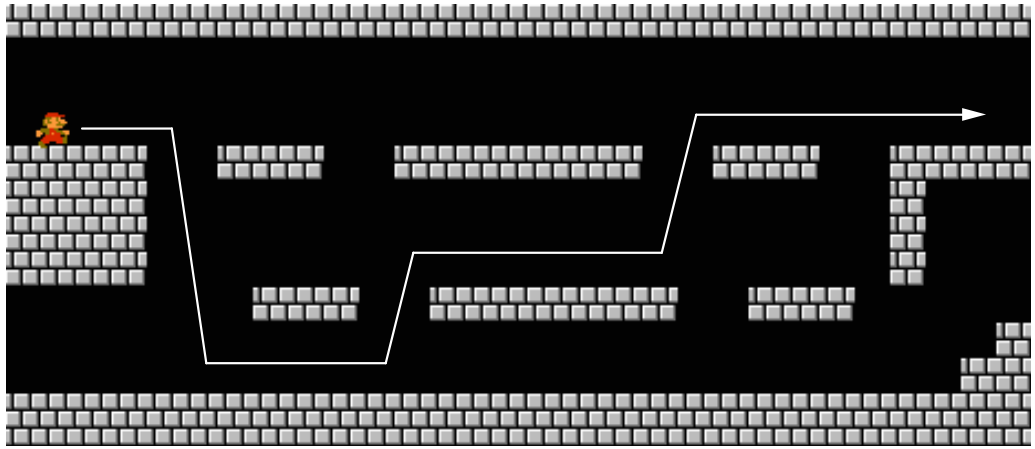




Reaching the flagpole is in P

Sketch of the proof:

- The "active screen" has fixed size; it can contain at most 5 sprites (plus Mario) and everything off-screen is reset
- Pseudorandom numbers are generated at every frame, starting from a fixed initial seed
- Mario can instantly move to different locations through pipes
- "Loop Command" objects send Mario back by 4 screens if a certain path is not followed



Reaching the flagpole is in P

Sketch of the proof:

- The "active screen" has fixed size; it can contain at most 5 sprites (plus Mario) and everything off-screen is reset
- Pseudorandom numbers are generated at every frame, starting from a fixed initial seed
- Mario can instantly move to different locations through pipes
- "Loop Command" objects send Mario back by 4 screens if a certain path is not followed

$\Longrightarrow$ Solving a level is reduced to exploring a polynomial-size "configuration graph"

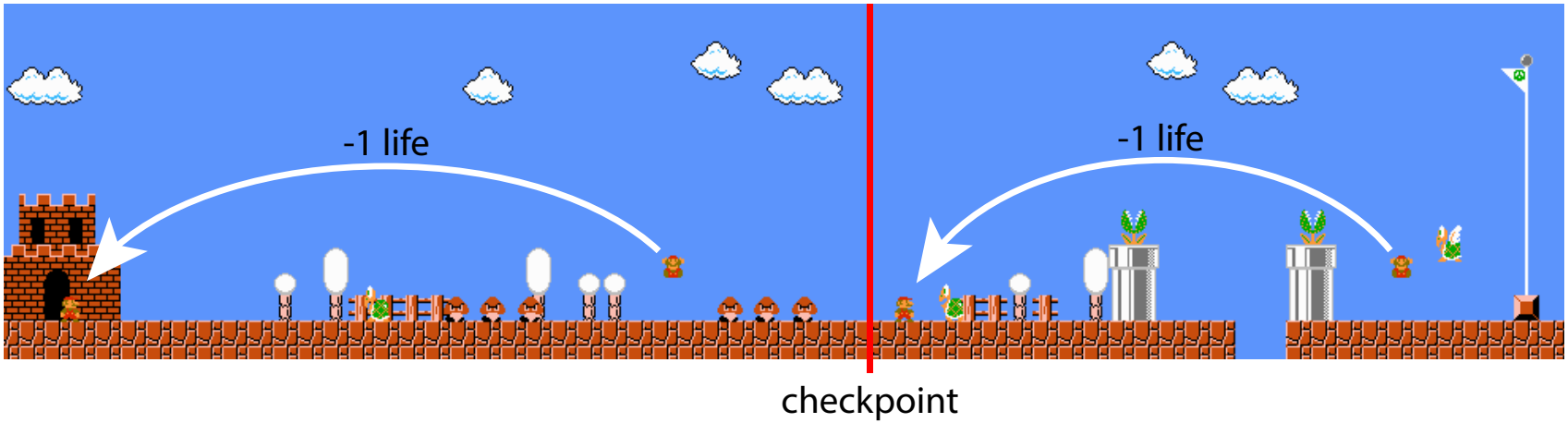
Rescuing the princess is in P

Rescuing the princess involves solving several levels,
and losing some lives in the process may be unavoidable

Rescuing the princess is in P

Rescuing the princess involves solving several levels,
and losing some lives in the process may be unavoidable

- Each level has a checkpoint



Rescuing the princess is in P

Rescuing the princess involves solving several levels,
and losing some lives in the process may be unavoidable

- Each level has a checkpoint
- Collecting coins grants extra lives:  x 100 = 

Rescuing the princess is in P

Rescuing the princess involves solving several levels,
and losing some lives in the process may be unavoidable

- Each level has a checkpoint
- Collecting coins grants extra lives:  x 100 = 
  - Note: repeatedly collecting more than 100 coins and losing a life grants Mario arbitrarily many lives

Rescuing the princess is in P

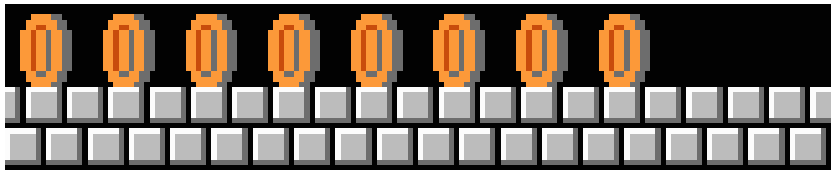
Rescuing the princess involves solving several levels,
and losing some lives in the process may be unavoidable

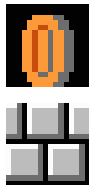
- Each level has a checkpoint
- Collecting coins grants extra lives:  x 100 = 
  - Note: repeatedly collecting more than 100 coins and losing a life grants Mario arbitrarily many lives

$\Longrightarrow$ Solving a sequence of levels is reduced to exploring a polynomial-size "macro-graph"

Run-length encoding

In SMB's level format, a row of up to 16 equal objects can be encoded as a single object:

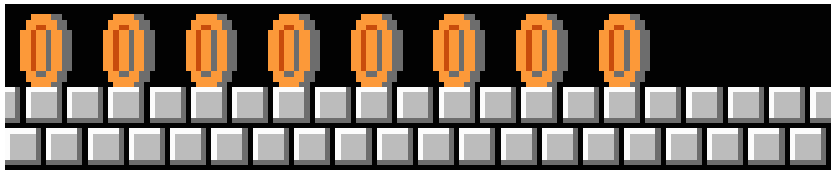
the fragment 

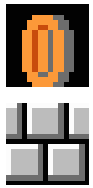
can be encoded as  x 8

x 10

Run-length encoding

In SMB's level format, a row of up to 16 equal objects can be encoded as a single object:

the fragment 

can be encoded as  x 8

x 10

Let us generalize this concept to arbitrarily long rows

NP-hardness: reduction from Knapsack

Knapsack problem. Given a set of n items with values $v_1, \ldots, v_n$ and weights $w_1, \ldots, w_n$, is there a subset with total value at least V and total weight at most W?

NP-hardness: reduction from Knapsack

Knapsack problem. Given a set of n items with values $v_1, \ldots, v_n$ and weights $w_1, \ldots, w_n$, is there a subset with total value at least V and total weight at most W?

Reduction. Construct a SMB game with the following levels:

- a *choice level* in which Mario can collect $100 \cdot V$ coins if and only if the given Knapsack instance is solvable in W seconds

NP-hardness: reduction from Knapsack

Knapsack problem. Given a set of n items with values $v_1, \ldots, v_n$ and weights $w_1, \ldots, w_n$, is there a subset with total value at least V and total weight at most W?

Reduction. Construct a SMB game with the following levels:

- a *choice level* in which Mario can collect $100 \cdot V$ coins if and only if the given Knapsack instance is solvable in W seconds
- $V + 2$ *kill levels* in which Mario has to lose a life
  (note: Mario starts the game with 3 lives)

NP-hardness: reduction from Knapsack

Knapsack problem. Given a set of n items with values $v_1, \ldots, v_n$ and weights $w_1, \ldots, w_n$, is there a subset with total value at least V and total weight at most W?

Reduction. Construct a SMB game with the following levels:

- a *choice level* in which Mario can collect $100 \cdot V$ coins if and only if the given Knapsack instance is solvable in W seconds
- $V + 2$ *kill levels* in which Mario has to lose a life
  (note: Mario starts the game with 3 lives)

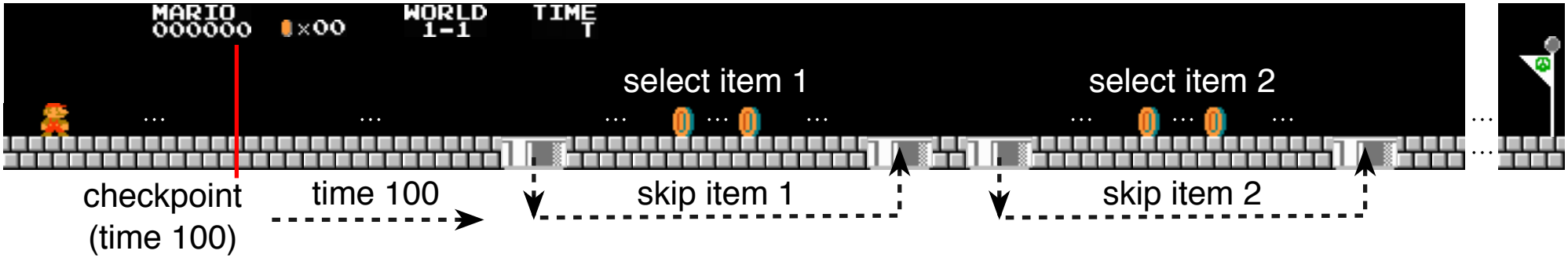
The Knapsack problem is weakly NP-complete

$\Downarrow$

SMB with run-length encoding is weakly NP-hard

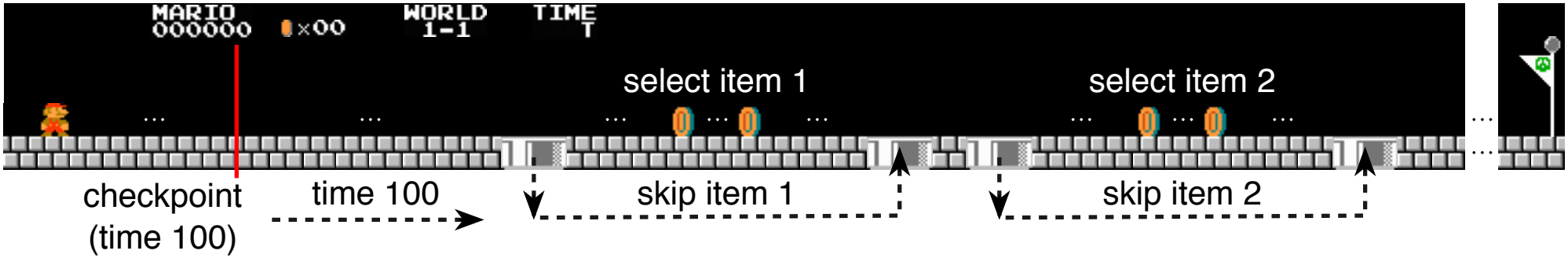
NP-hardness: reduction from Knapsack

Choice level:

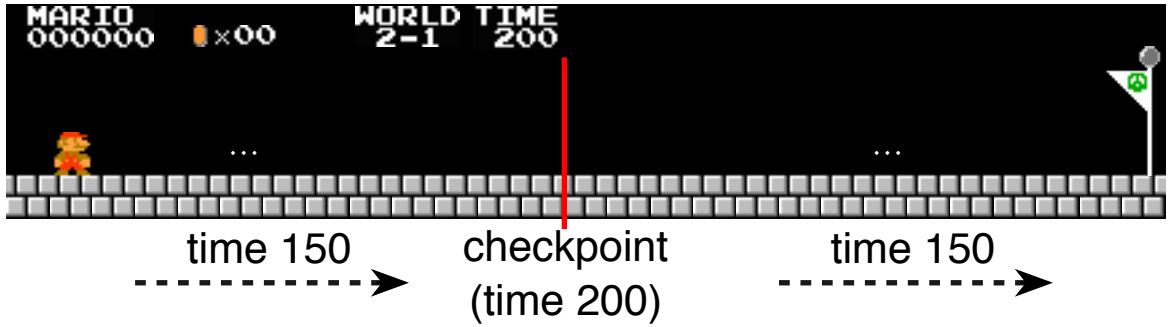


NP-hardness: reduction from Knapsack

Choice level:



Kill level:

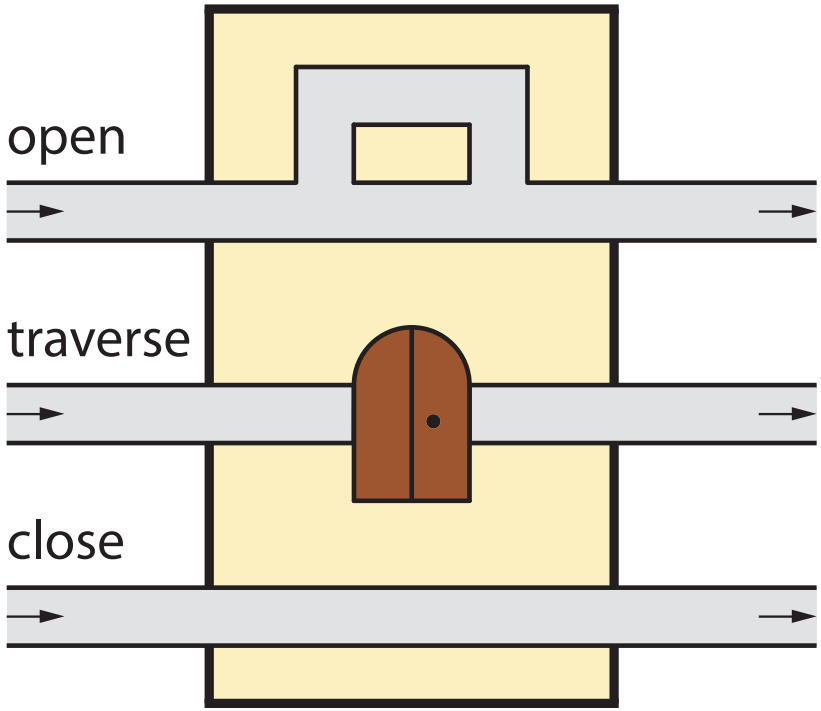


PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open

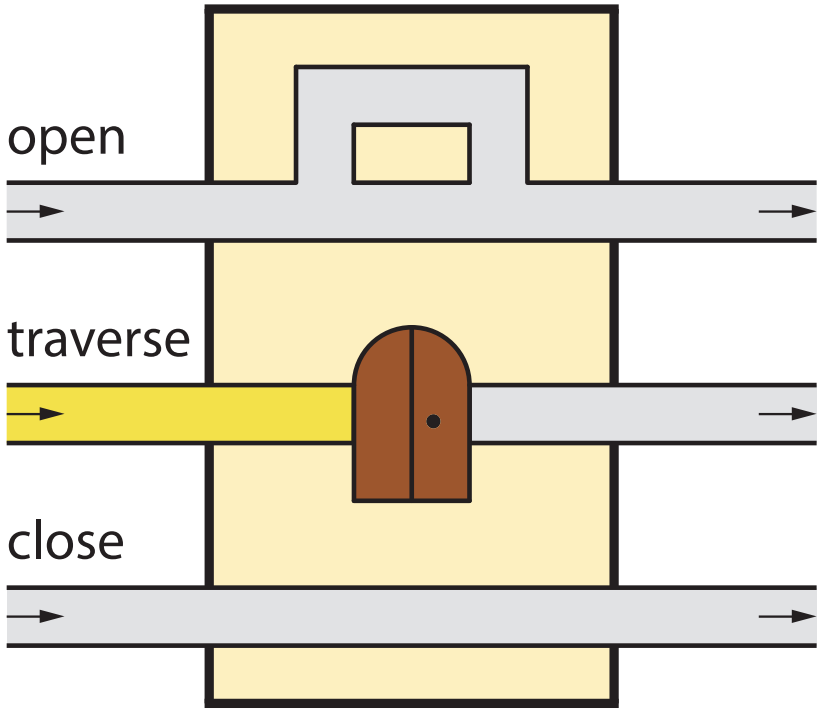
PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open



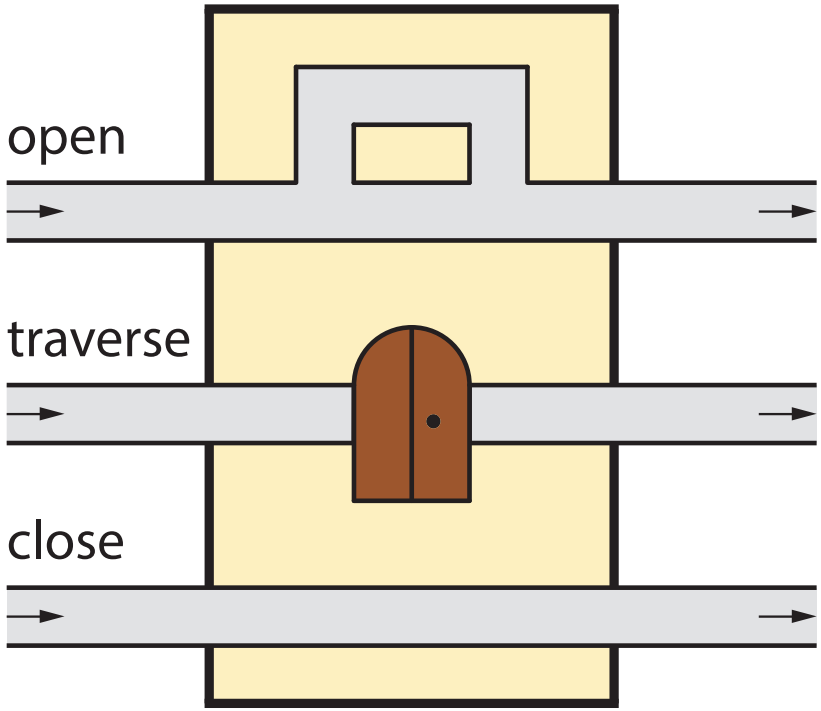
PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open



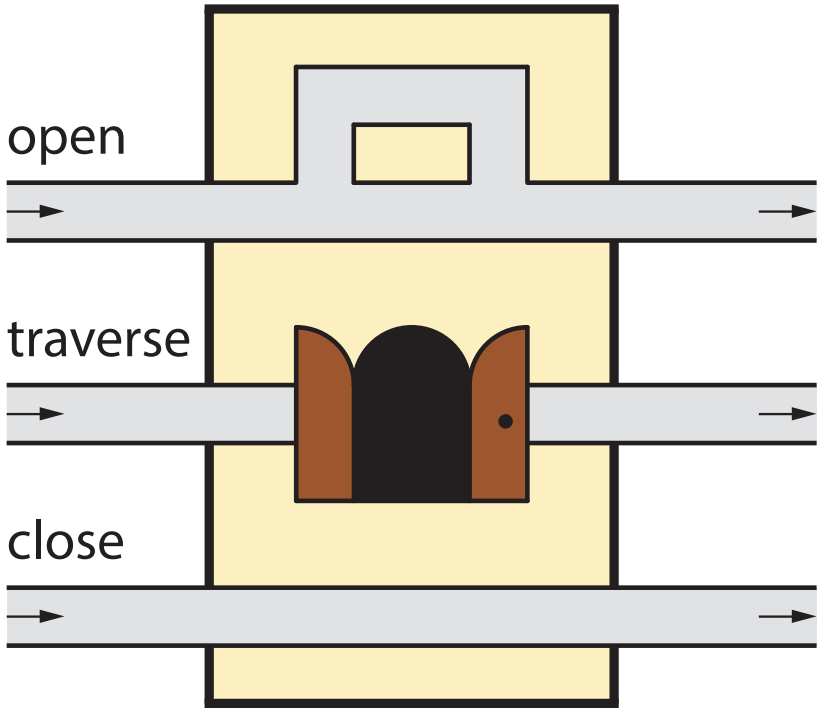
PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open



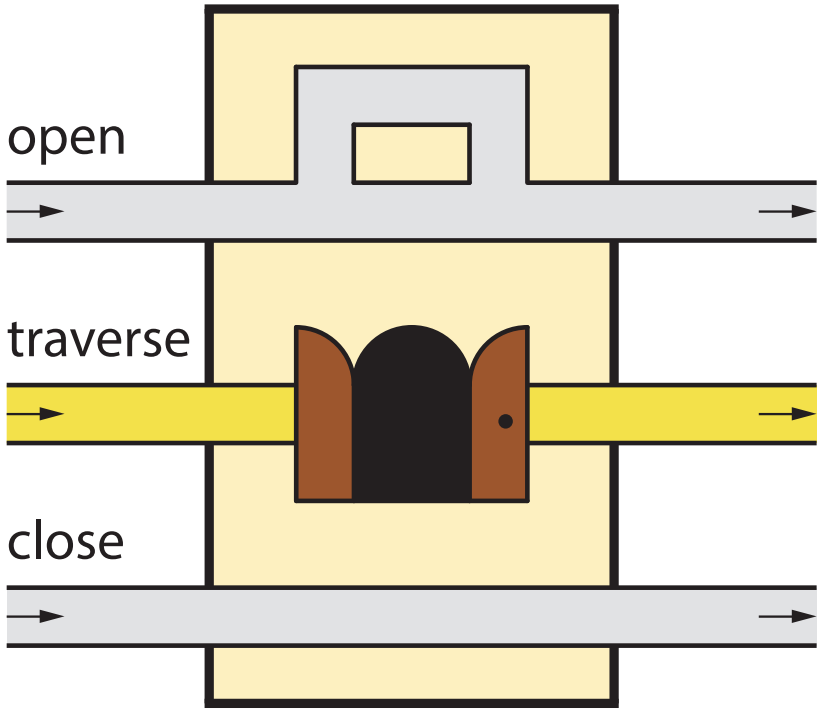
PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open



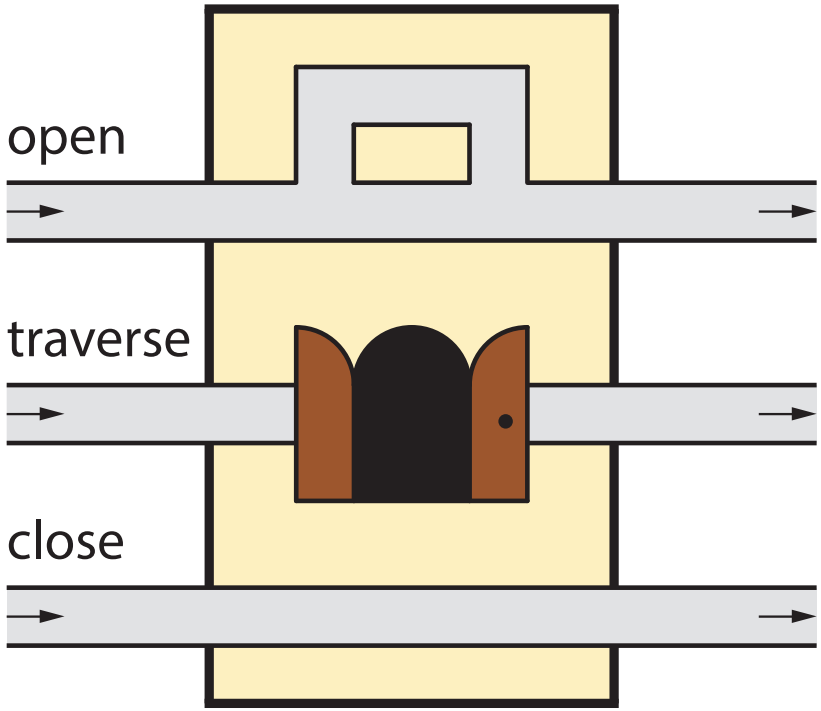
PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open



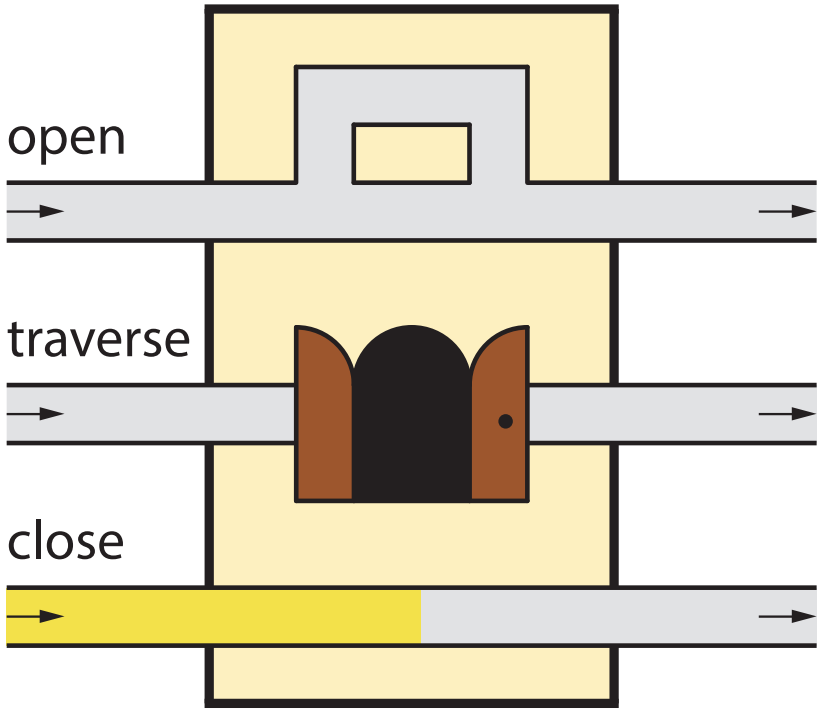
PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open



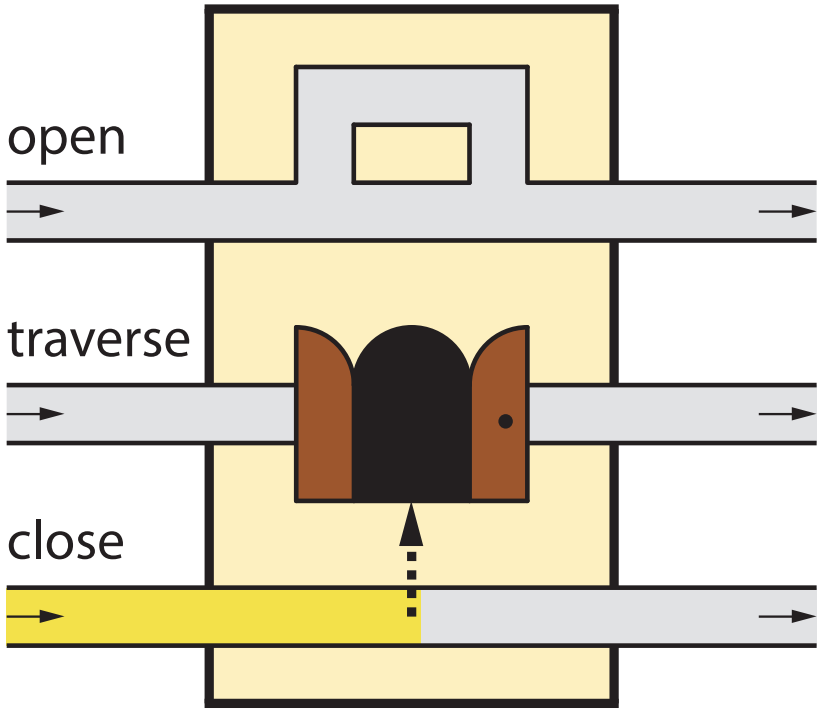
PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open



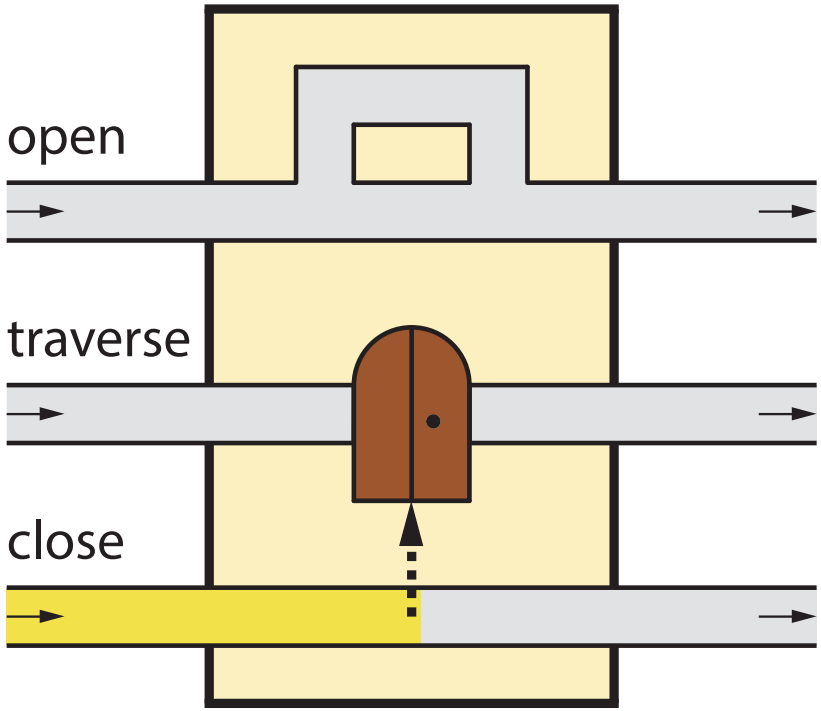
PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open



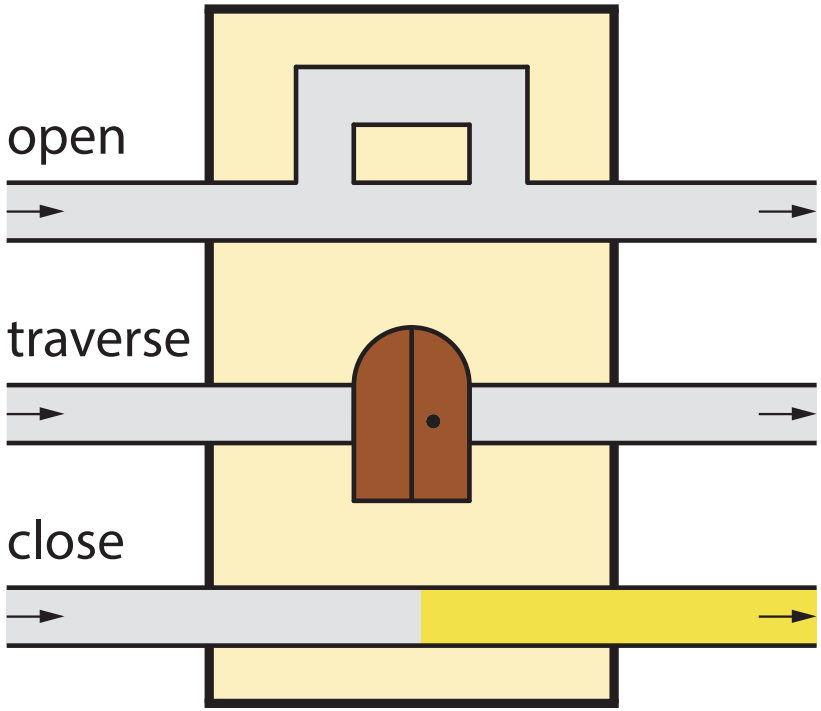
PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open



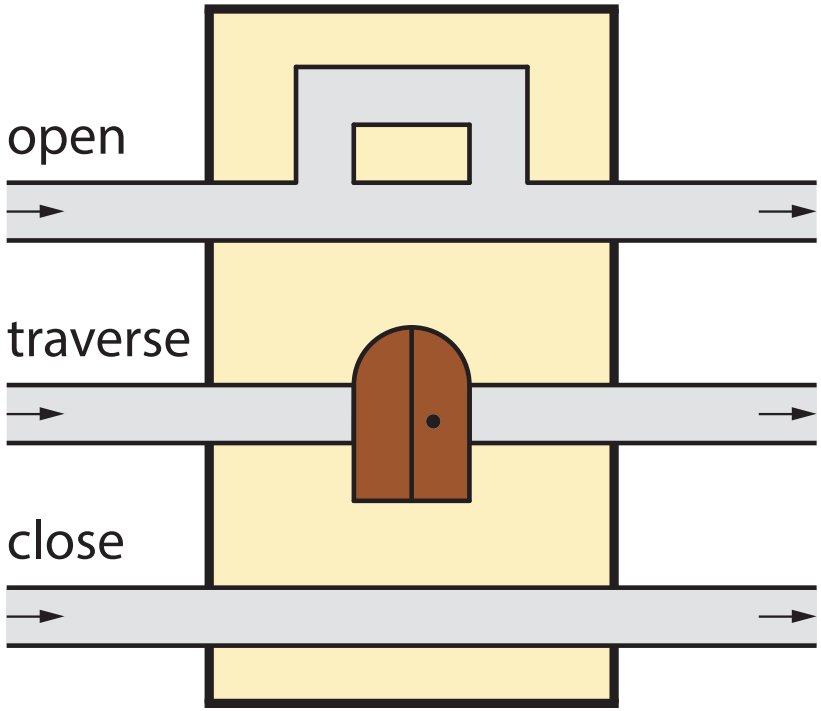
PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open



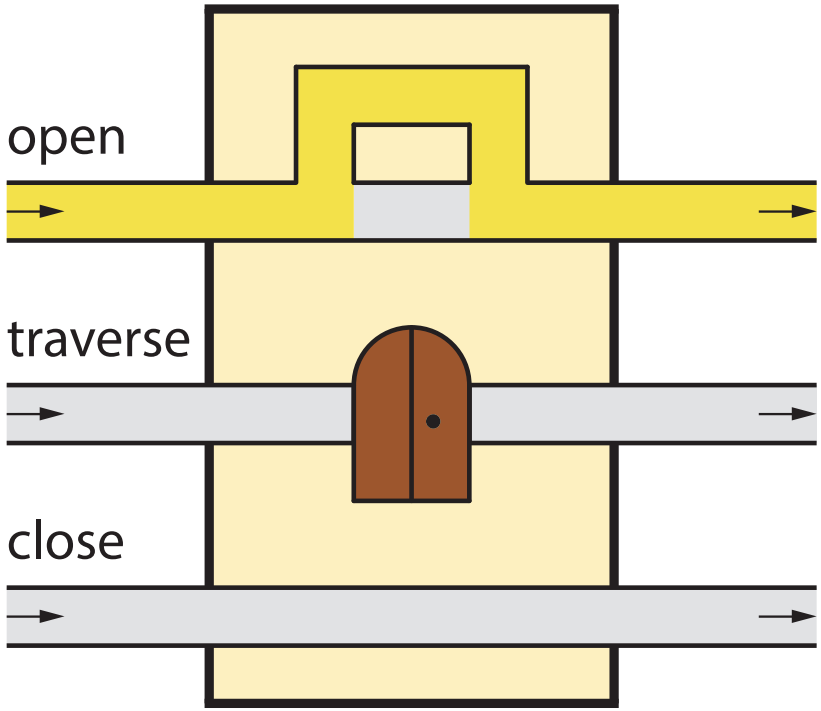
PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open



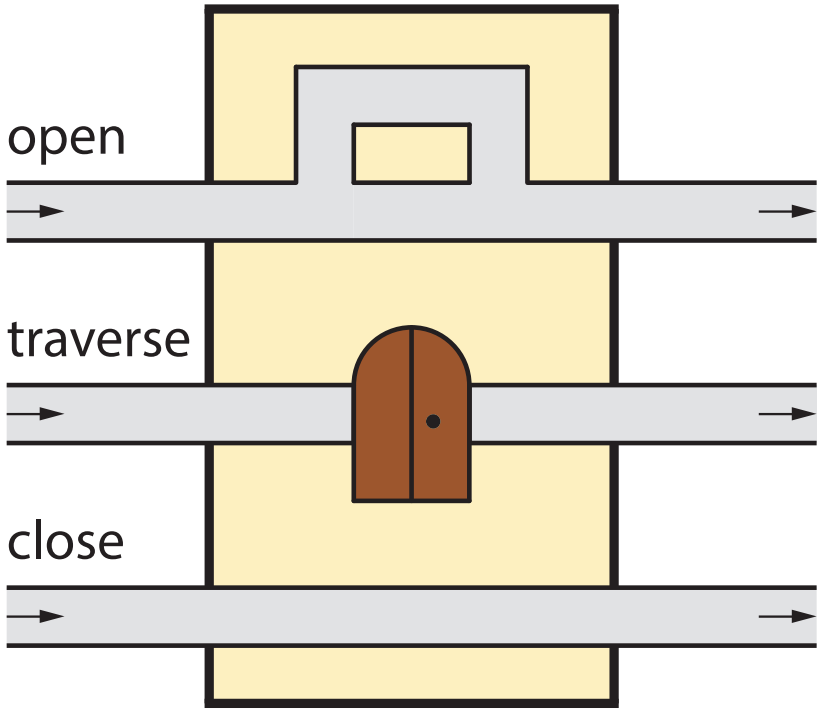
PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open



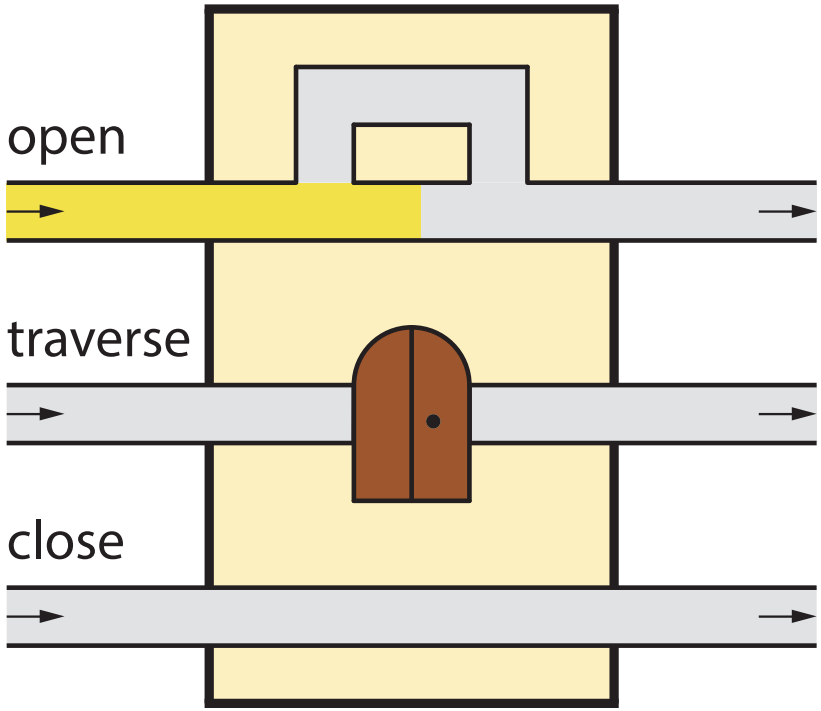
PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open



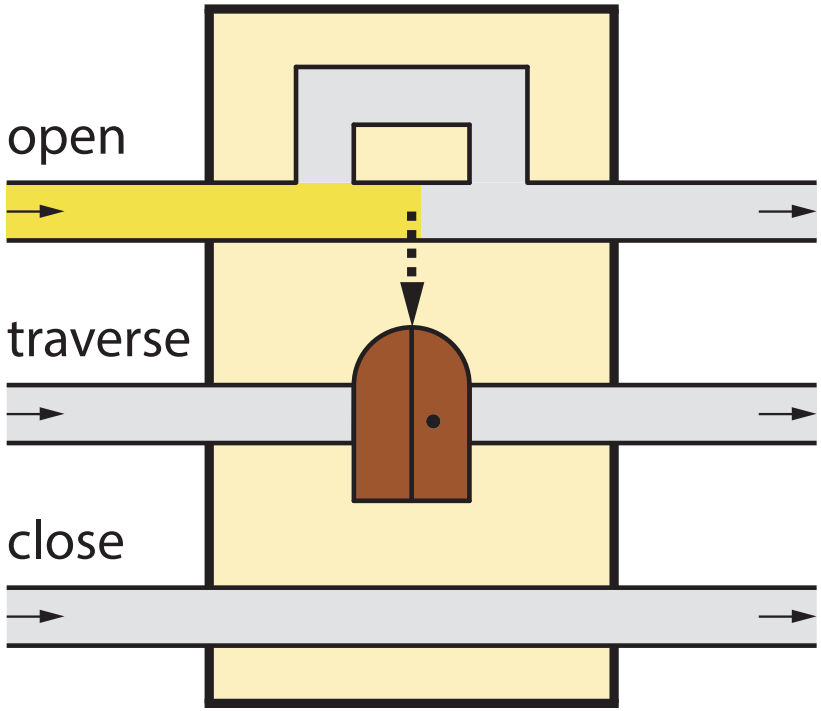
PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open



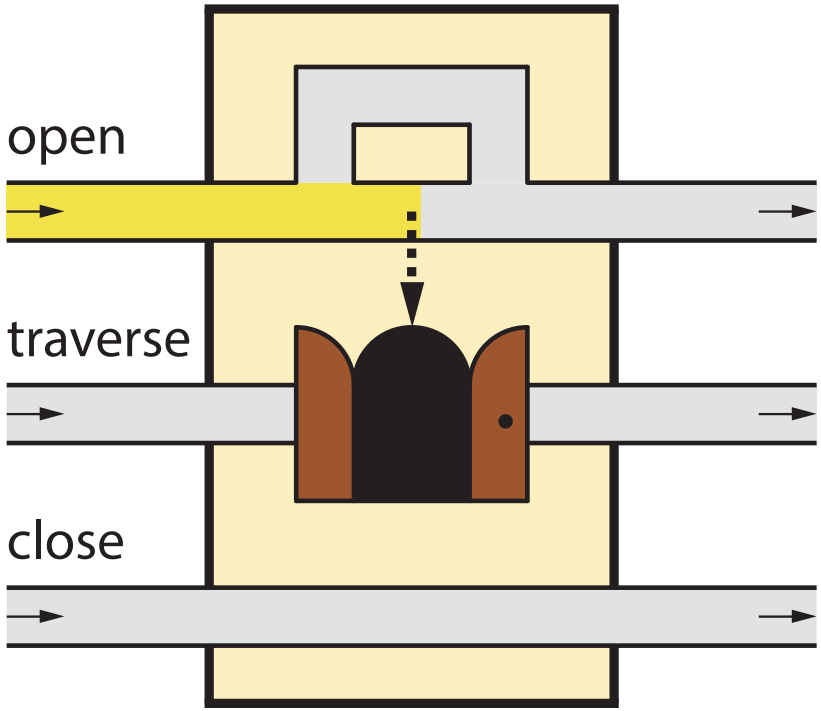
PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open



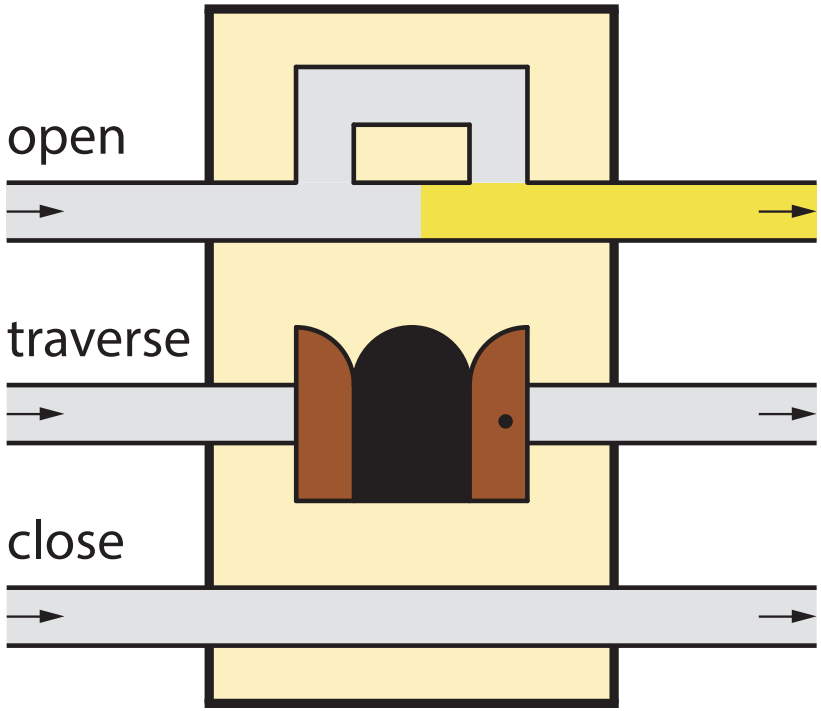
PSPACE-hardness gadgets

- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open

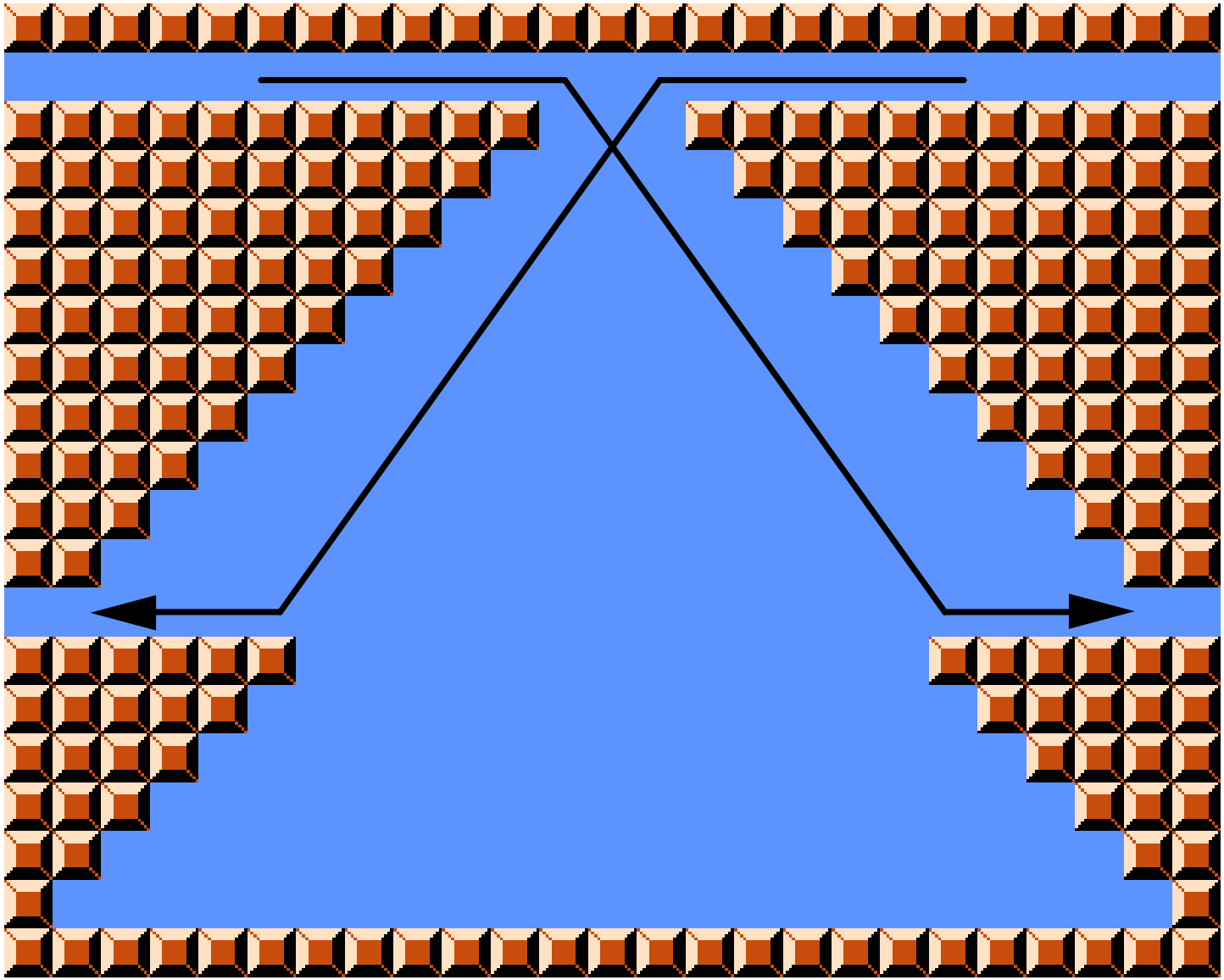


PSPACE-hardness gadgets

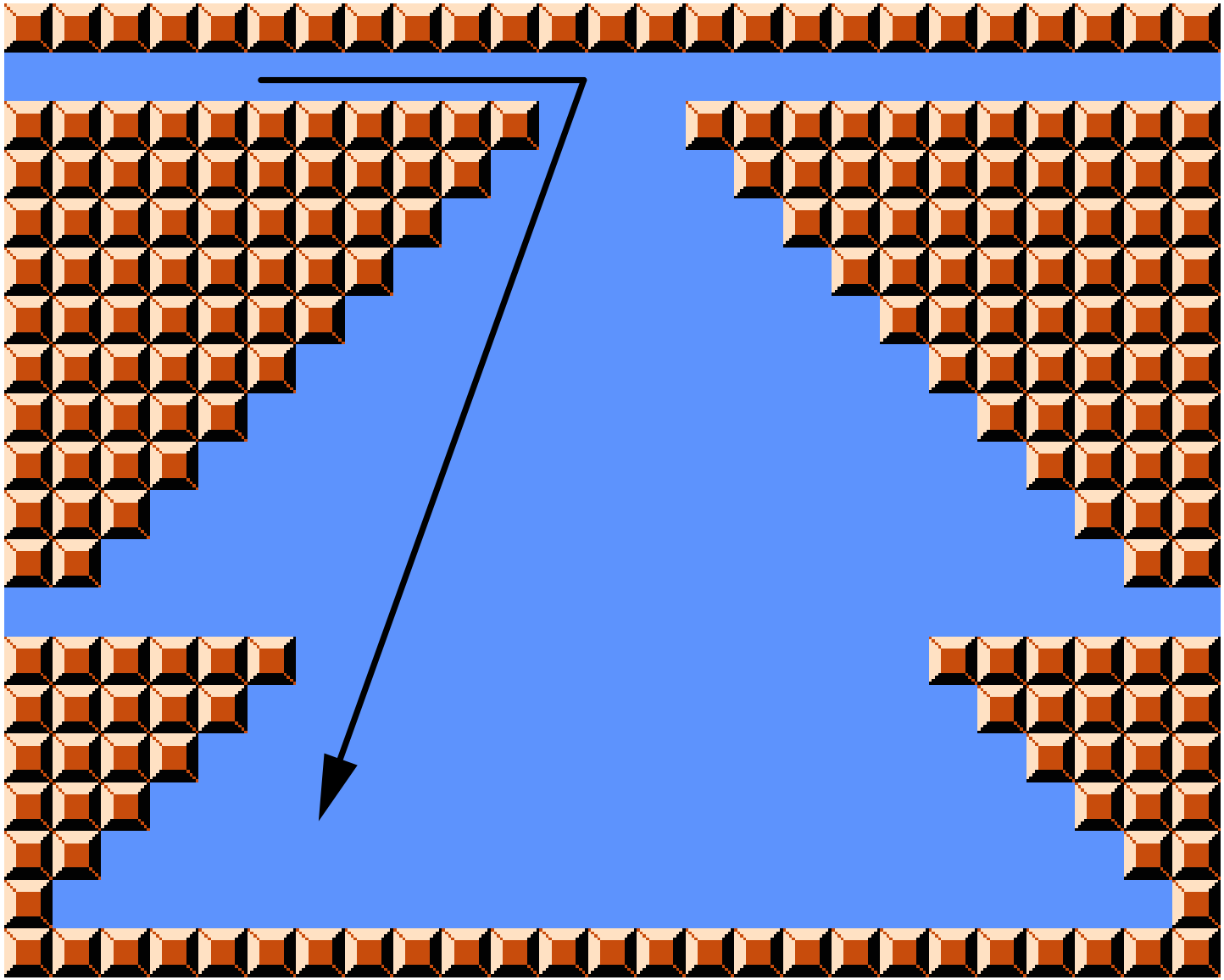
- **Start** & **Finish**
- **Crossover**, usable multiple times
- **Door**: can be opened, closed, and traversed if and only if it is open



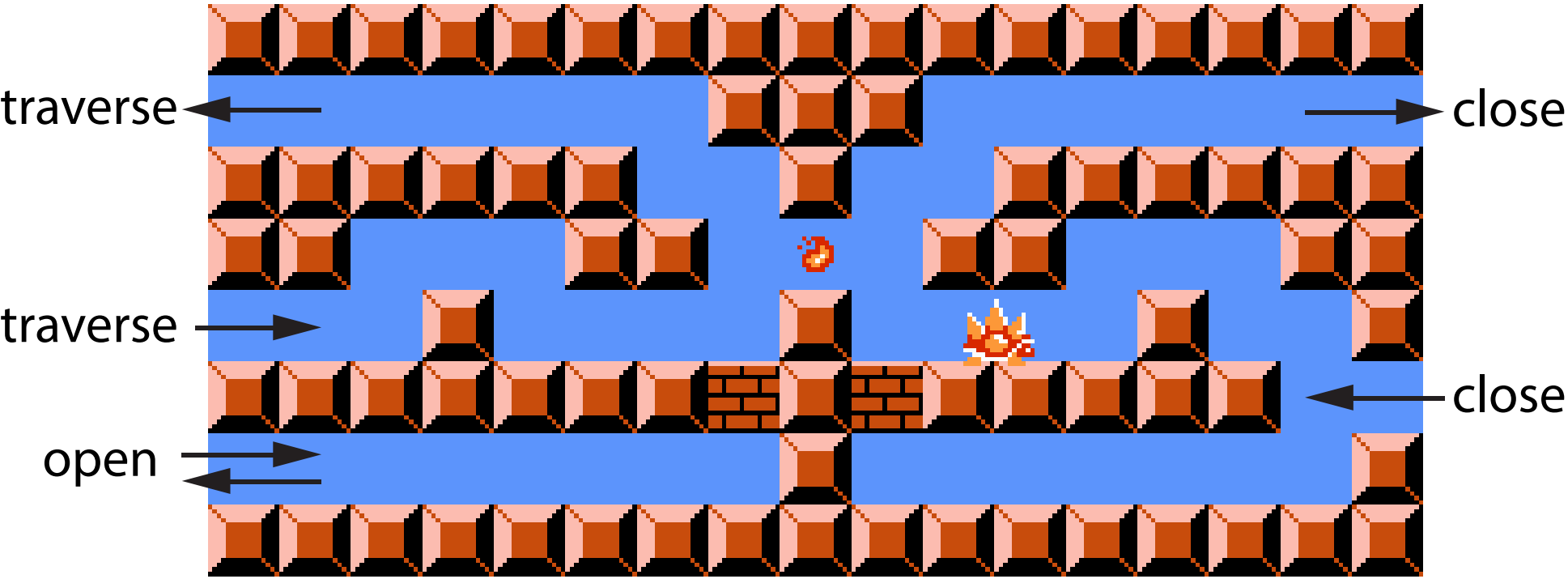
Crossover gadget



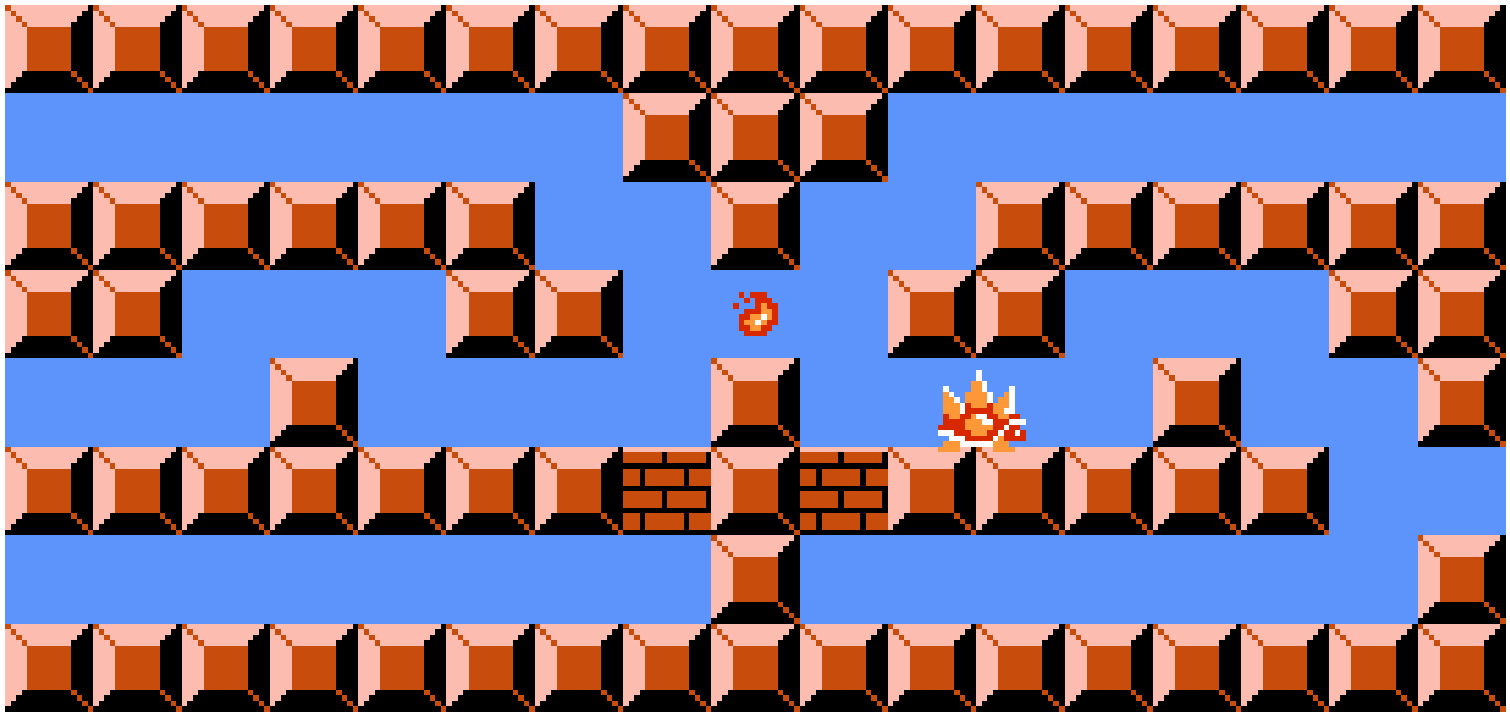
Crossover gadget



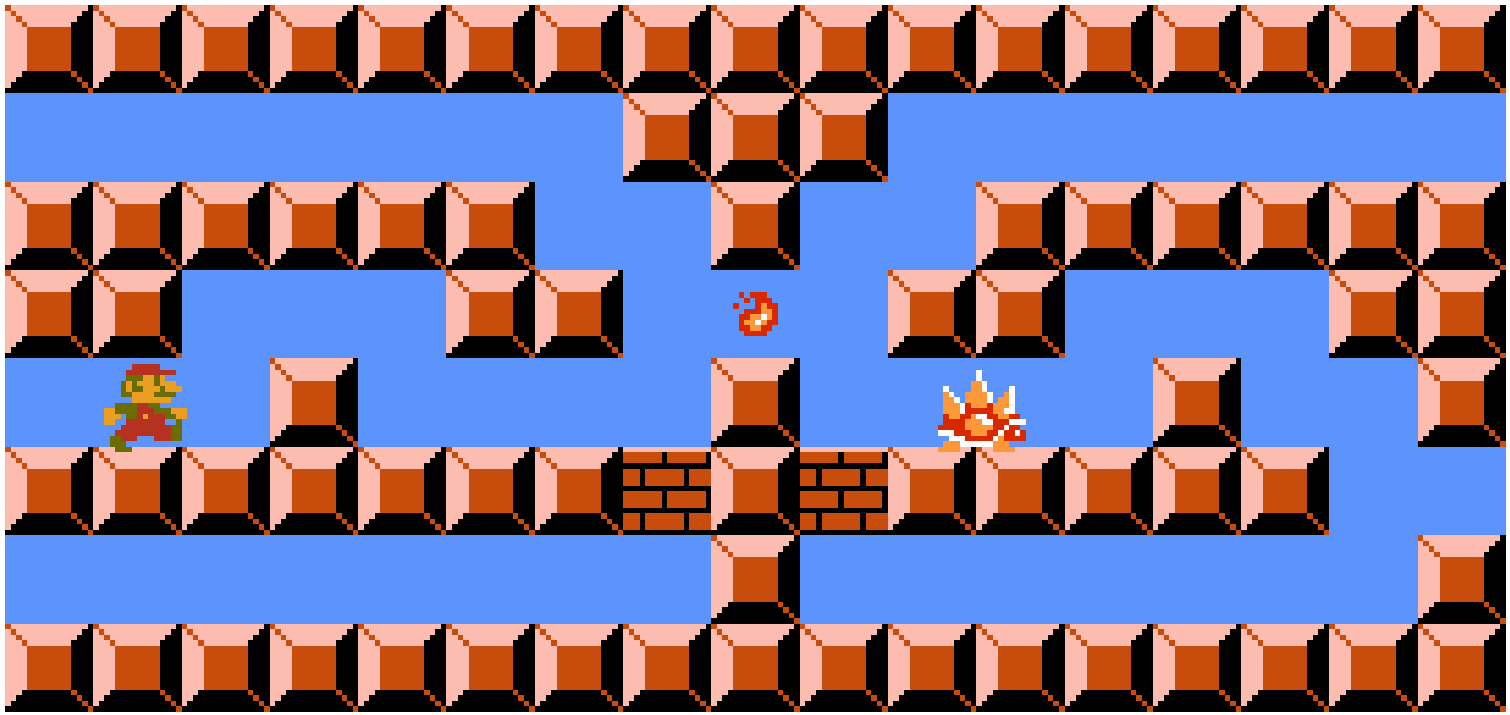
Door gadget



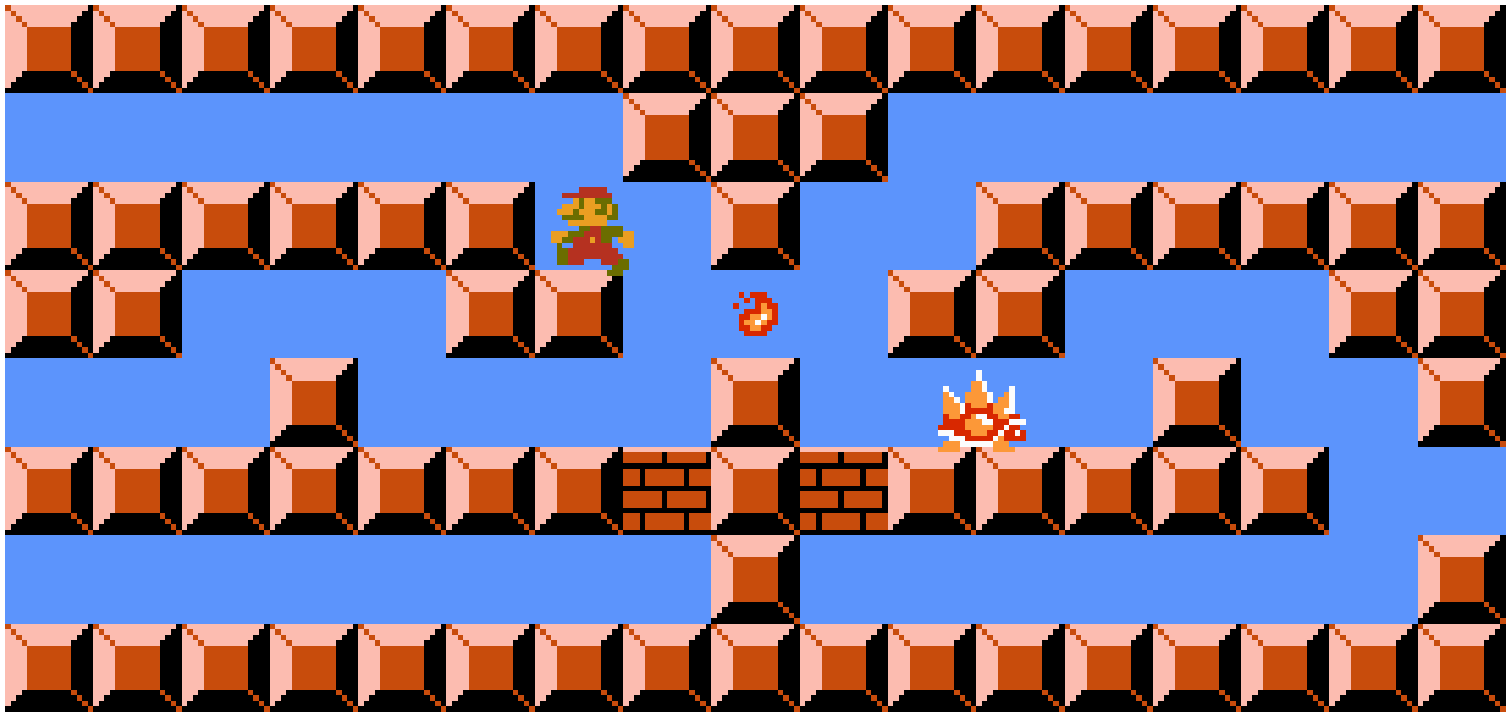
Door gadget



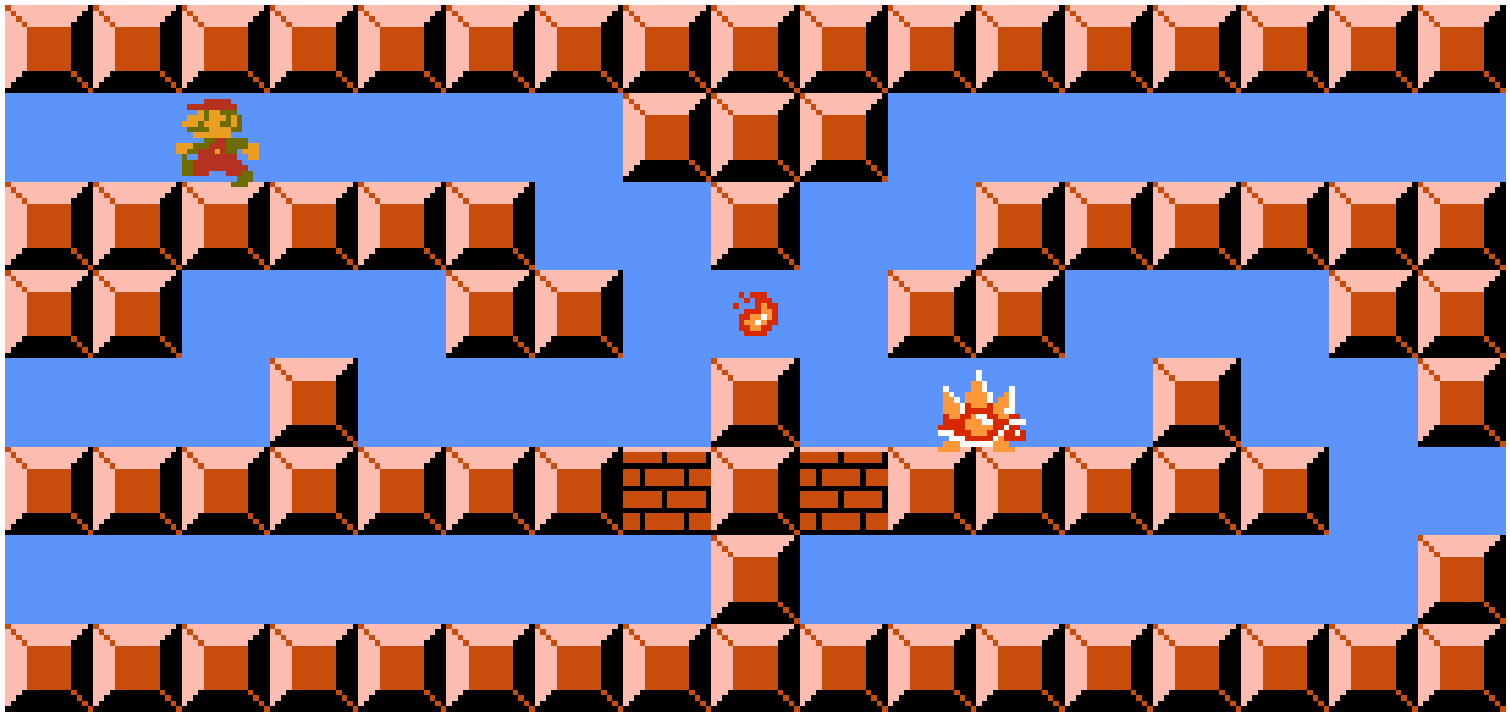
Door gadget



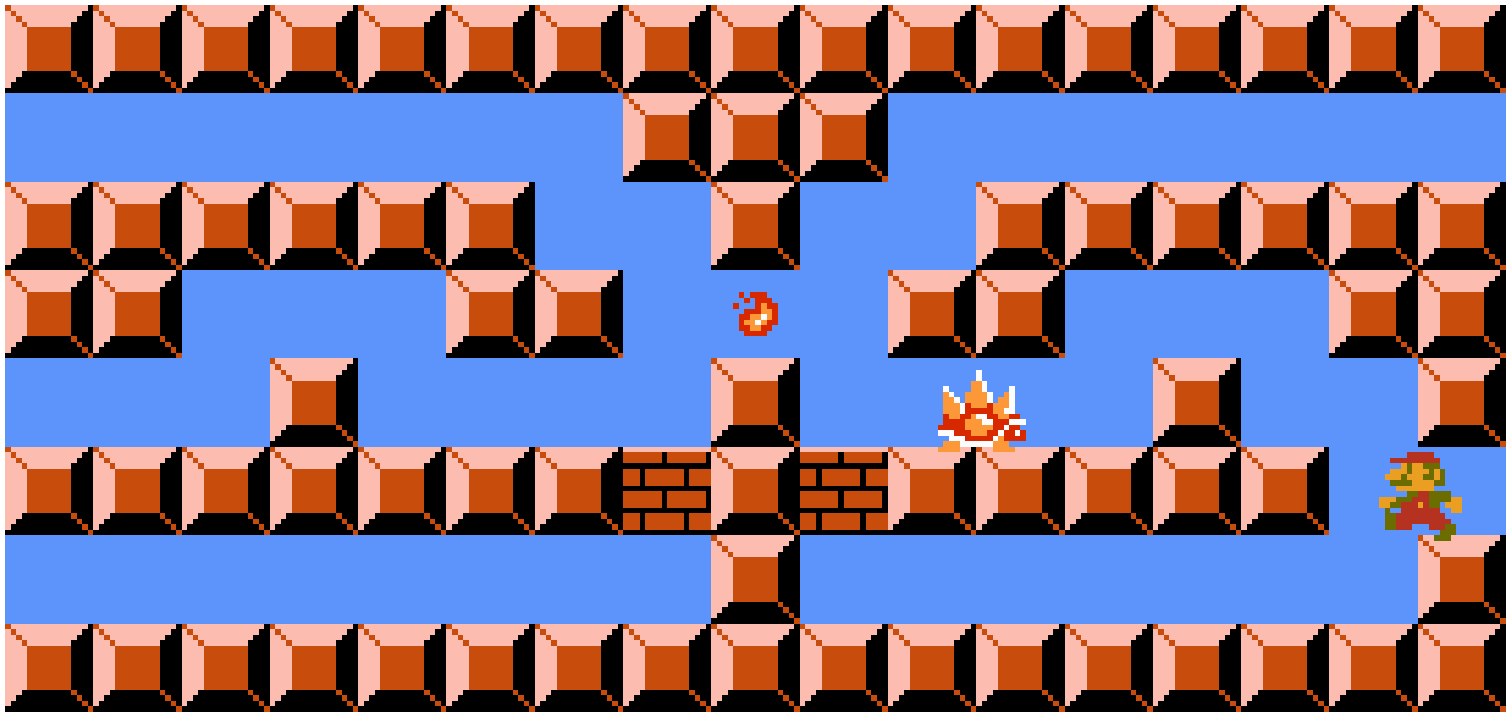
Door gadget



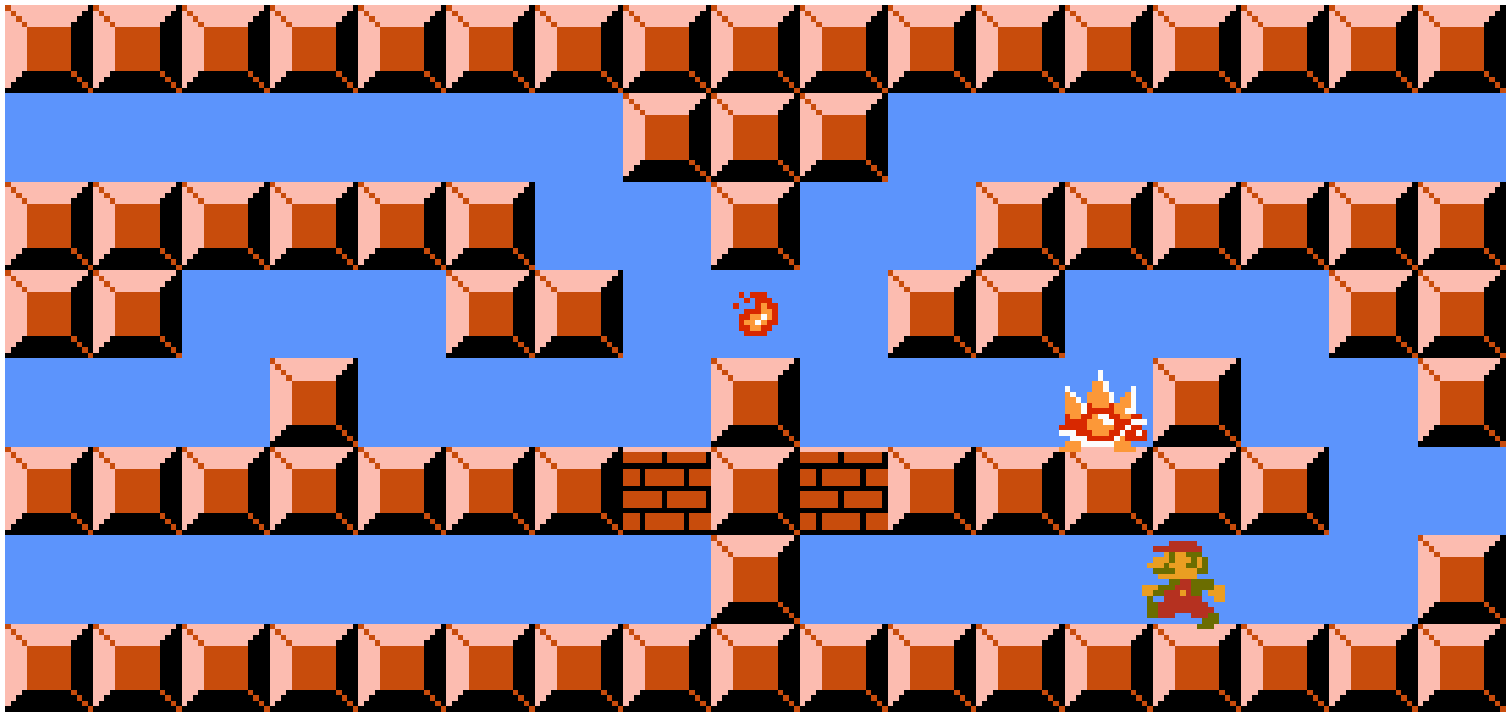
Door gadget



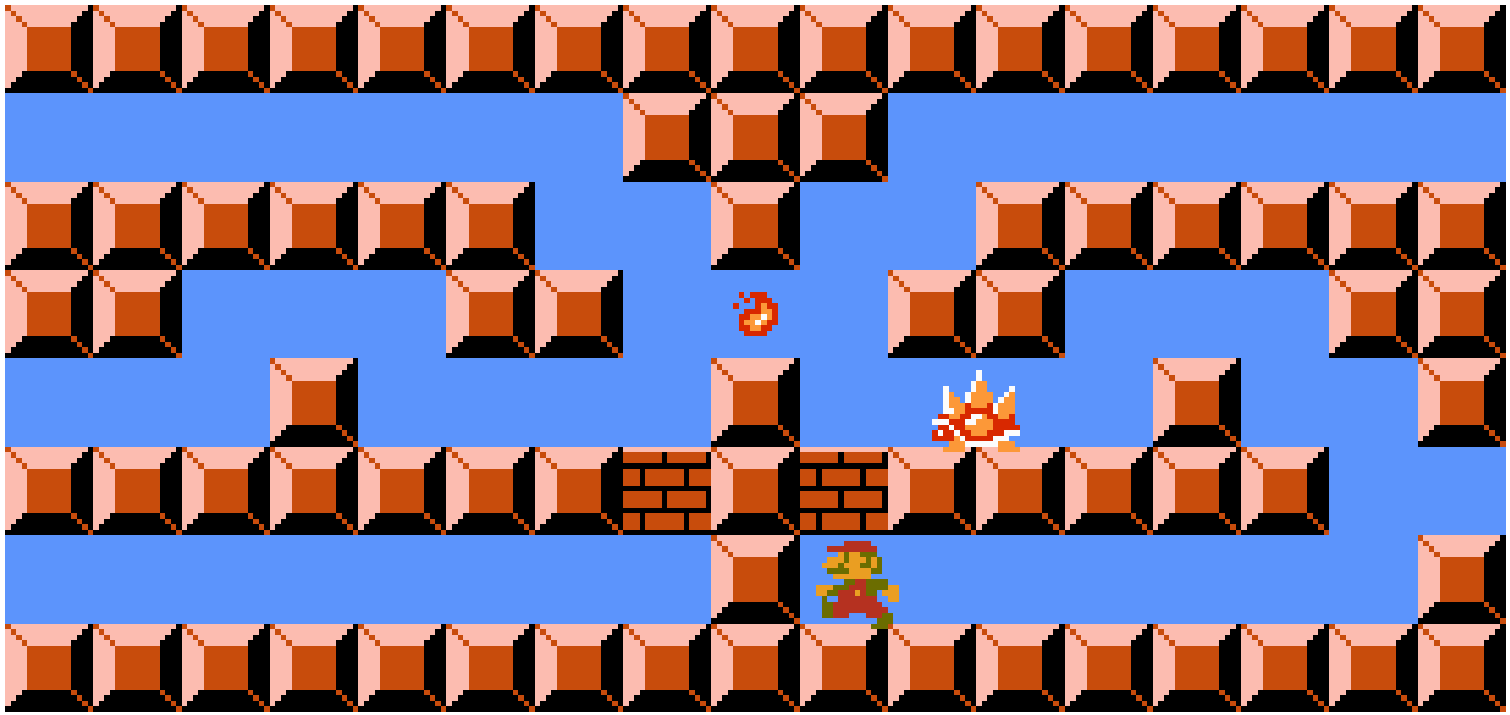
Door gadget



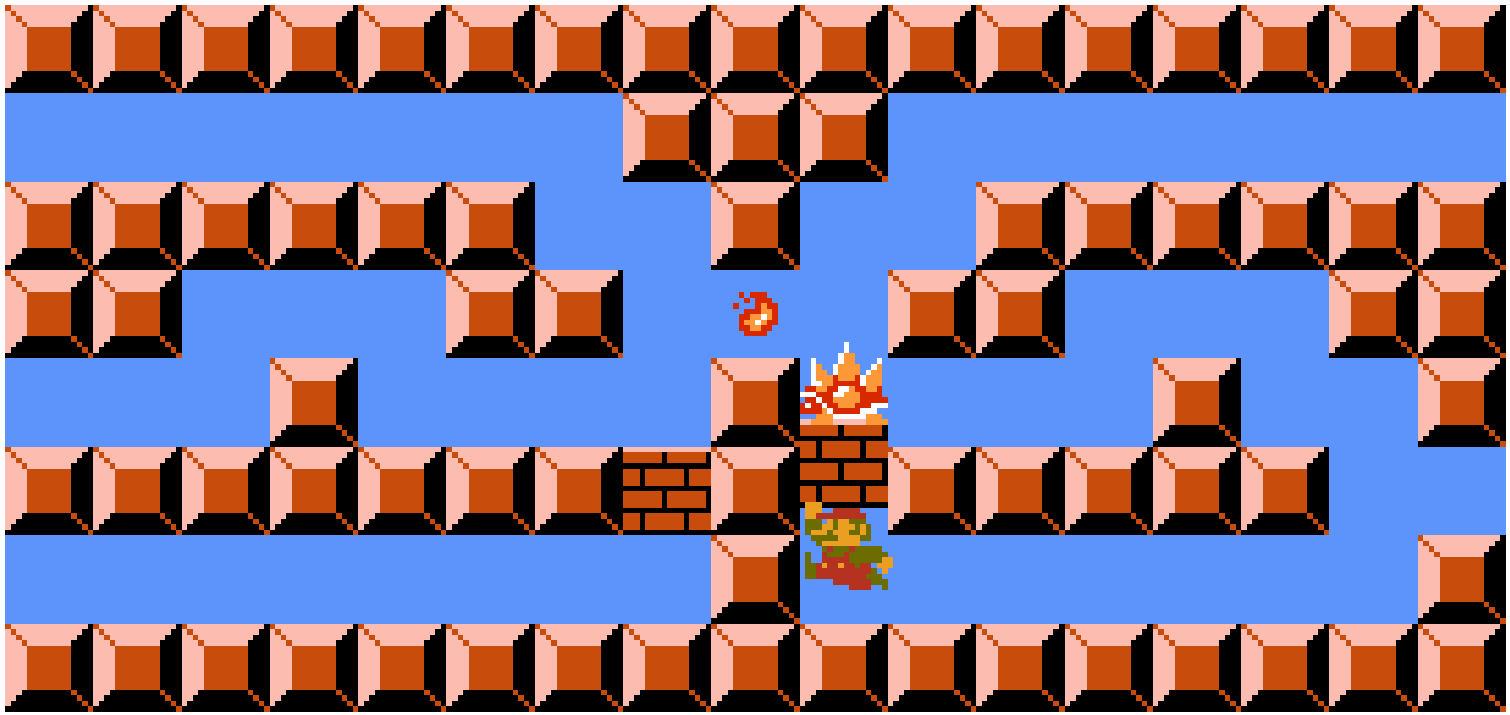
Door gadget



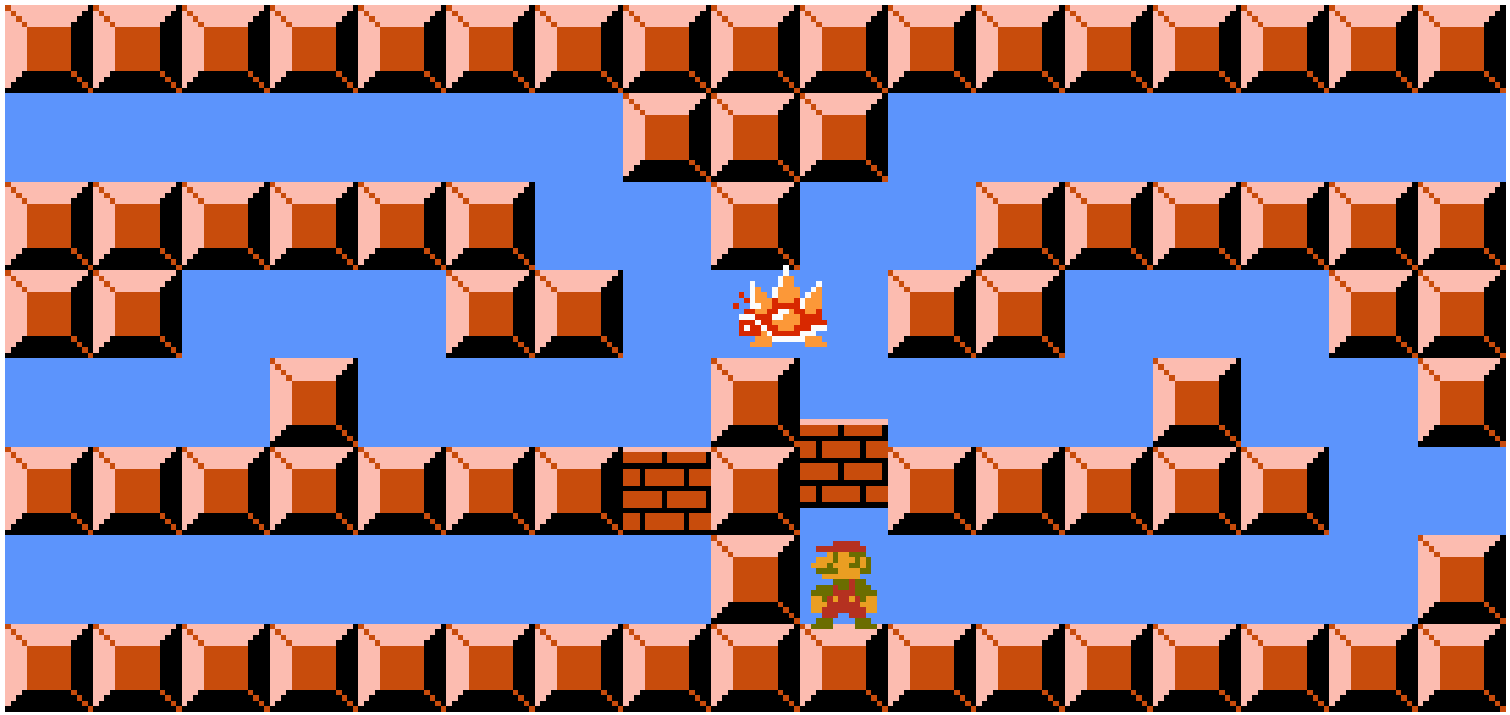
Door gadget



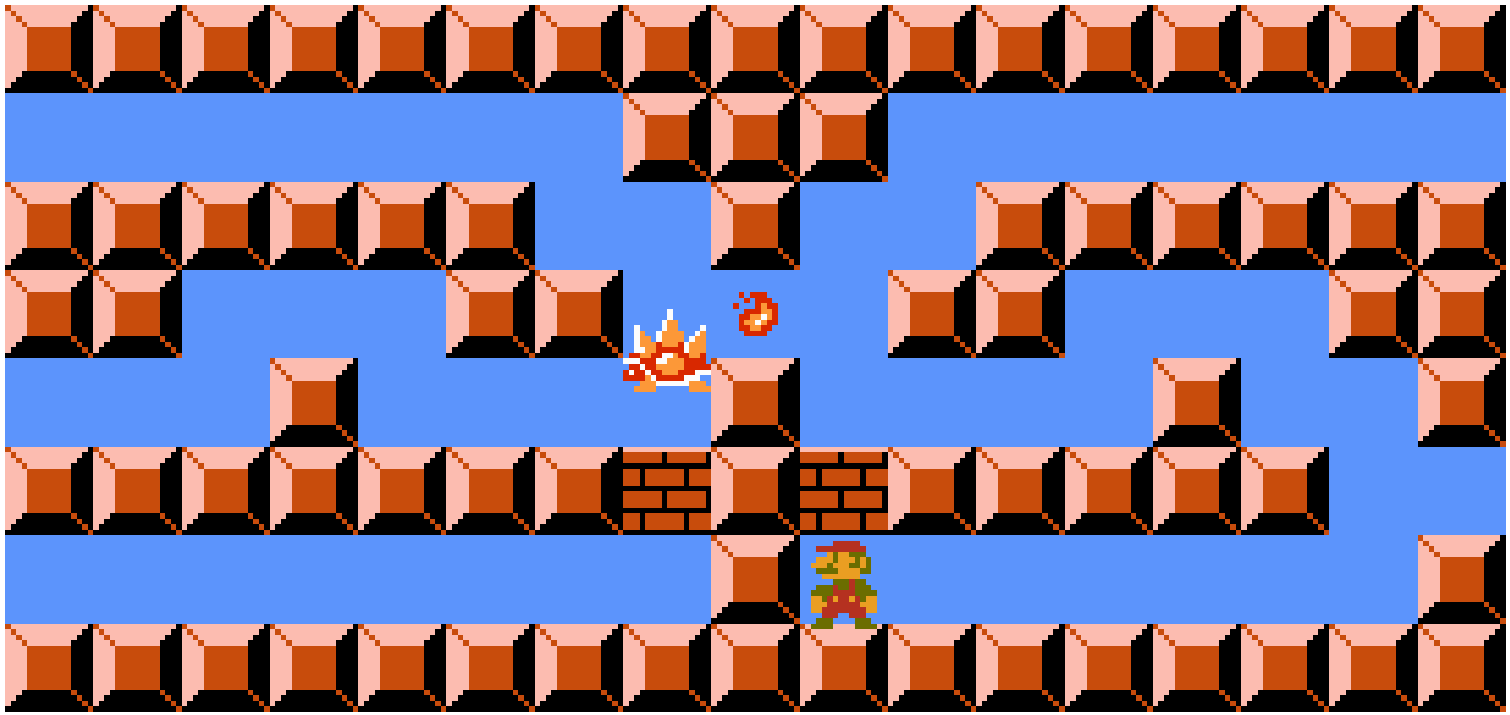
Door gadget



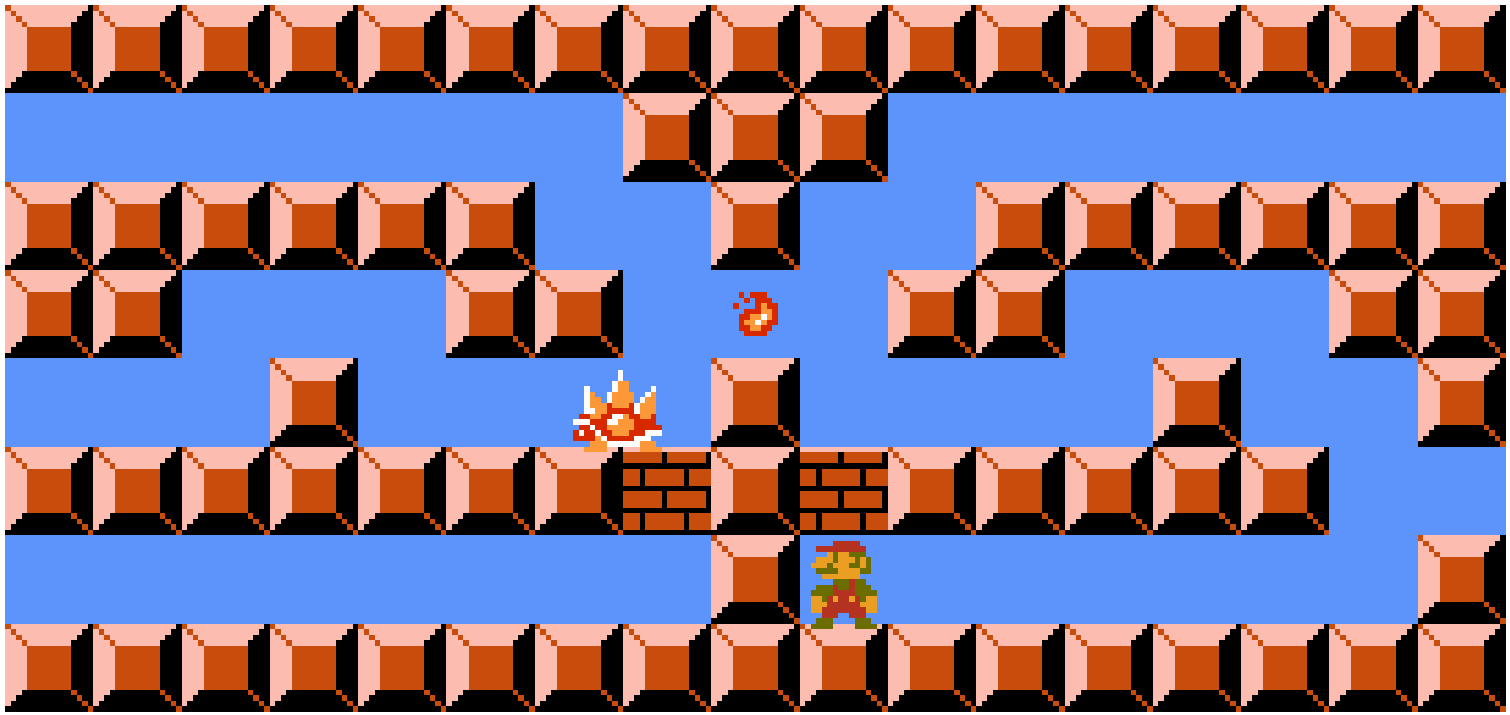
Door gadget



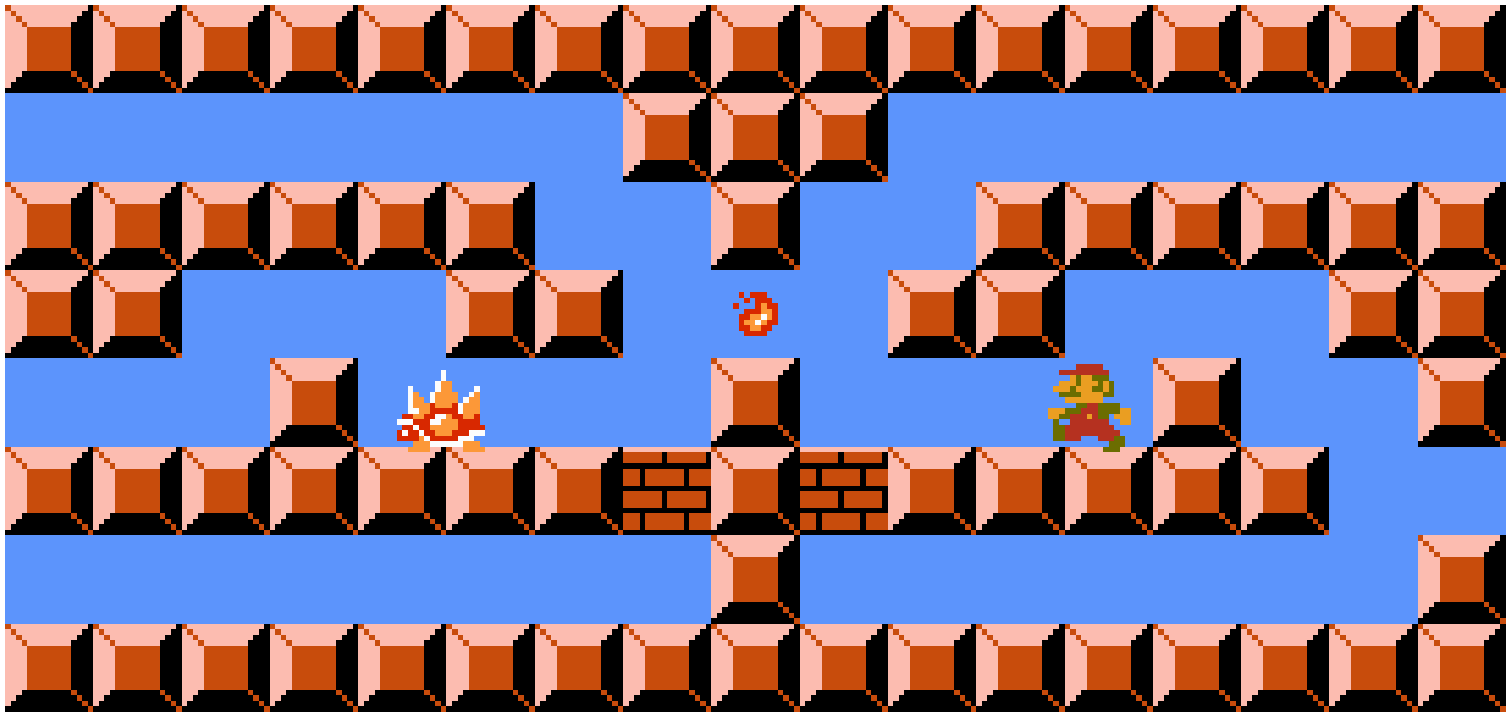
Door gadget



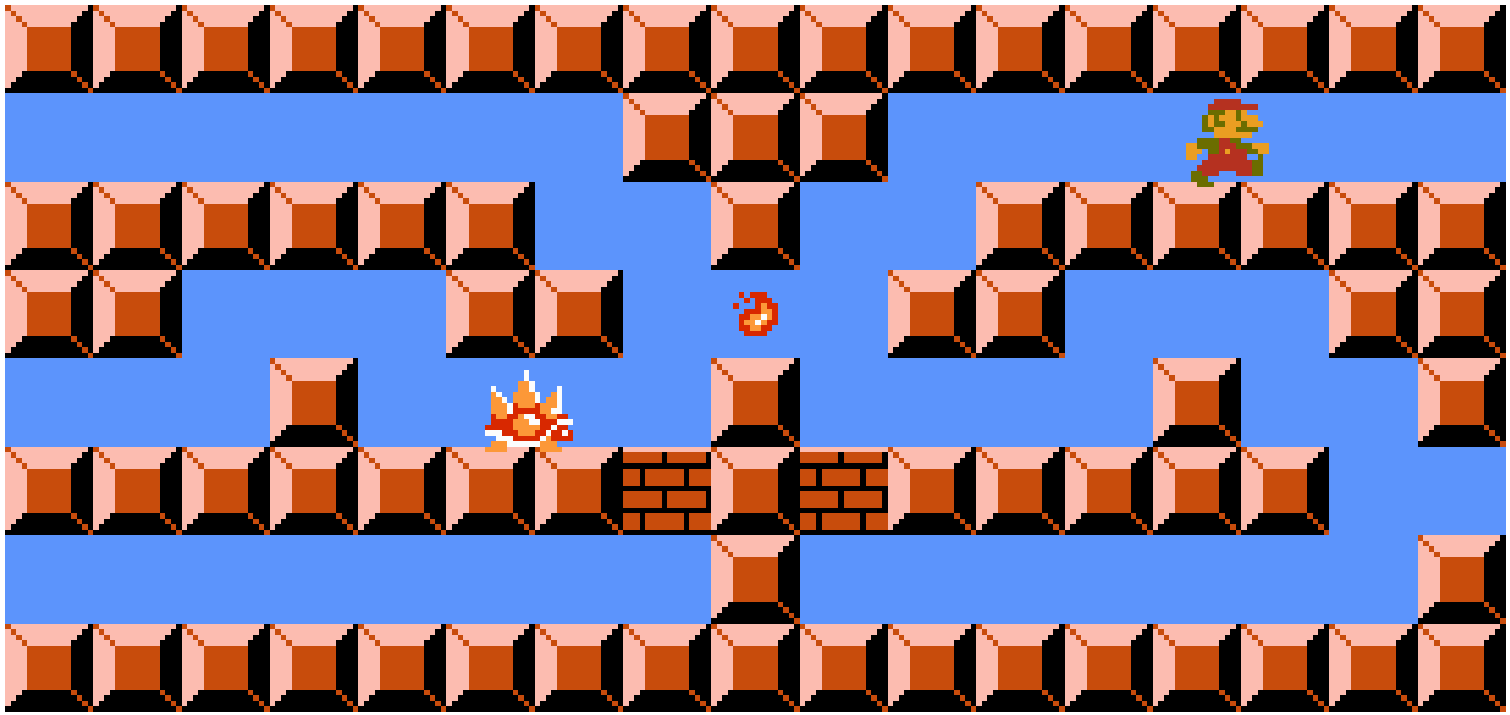
Door gadget



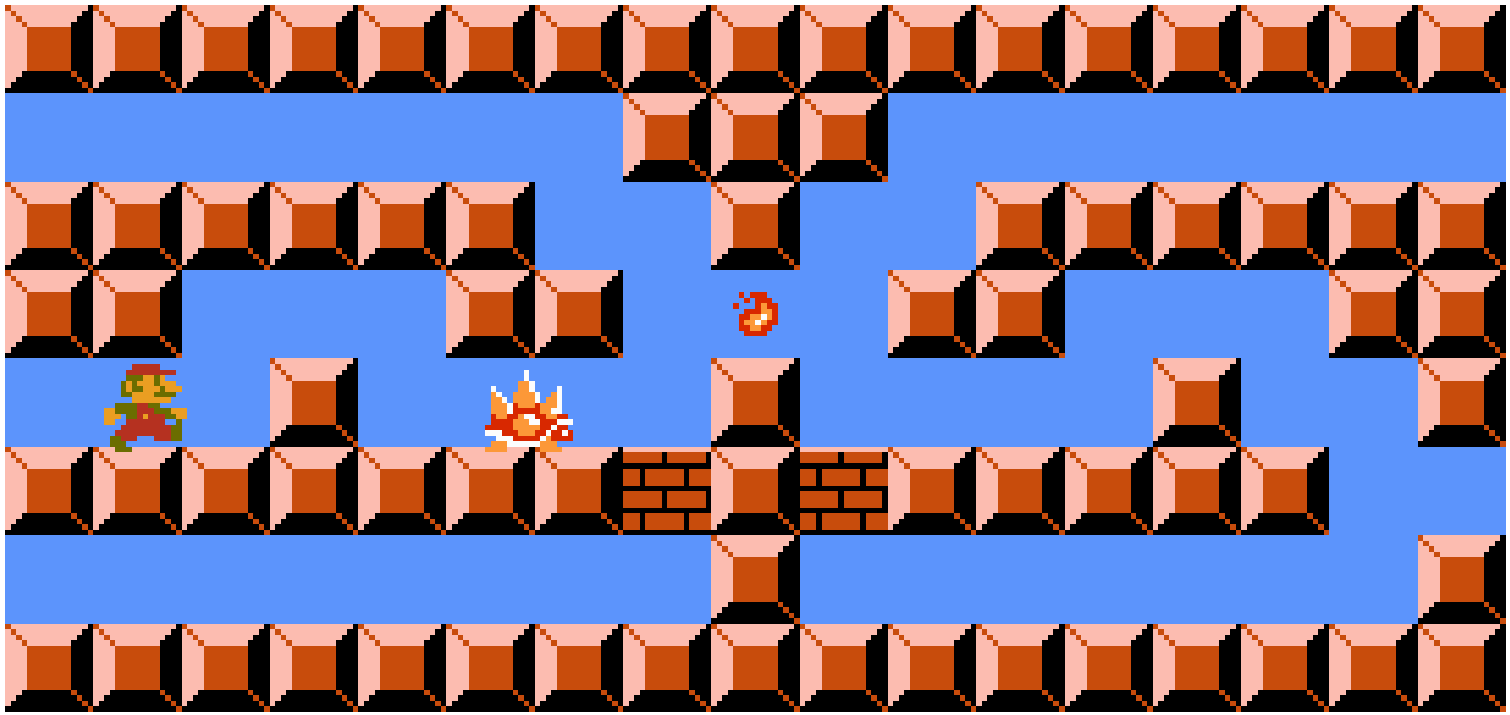
Door gadget



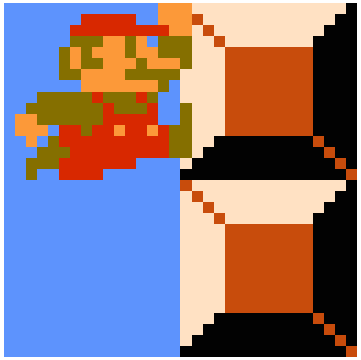
Door gadget



Door gadget

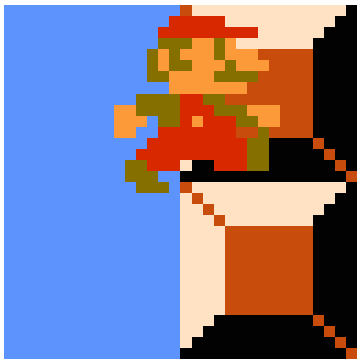


Glitches: walk through walls and walljump



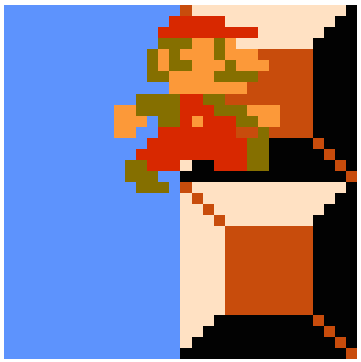
Jump toward a wall and steer in the opposite direction

Glitches: walk through walls and walljump



Jump toward a wall and steer in the opposite direction
If done properly, the wall sucks Mario in instead of pushing him out

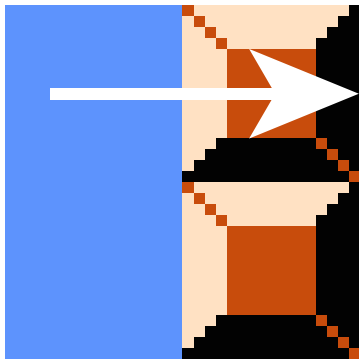
Glitches: walk through walls and walljump

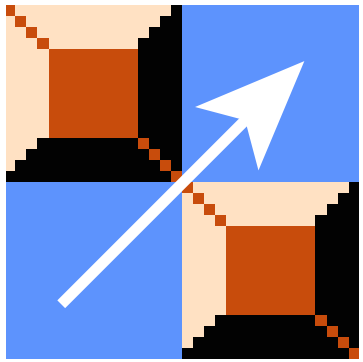


Jump toward a wall and steer in the opposite direction
If done properly, the wall sucks Mario in instead of pushing him out
This allows Mario to walk through certain walls or do a "walljump"

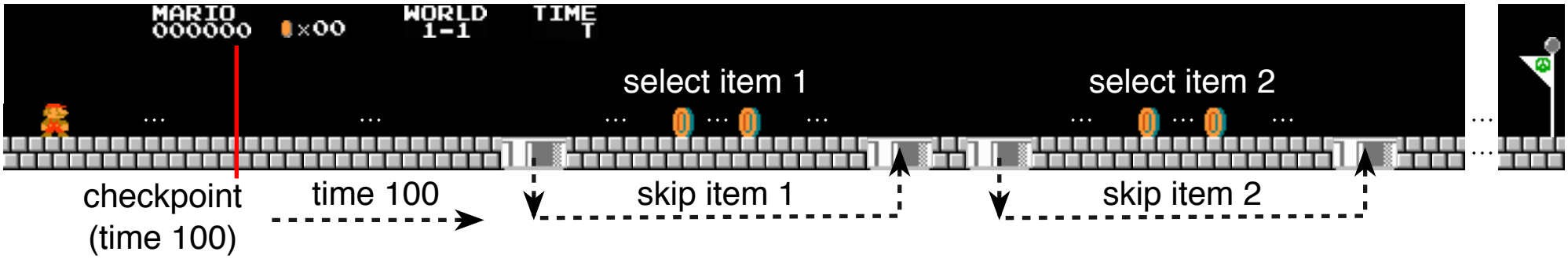
Glitches: walk through walls and walljump

Solution. Avoid the two following configurations:



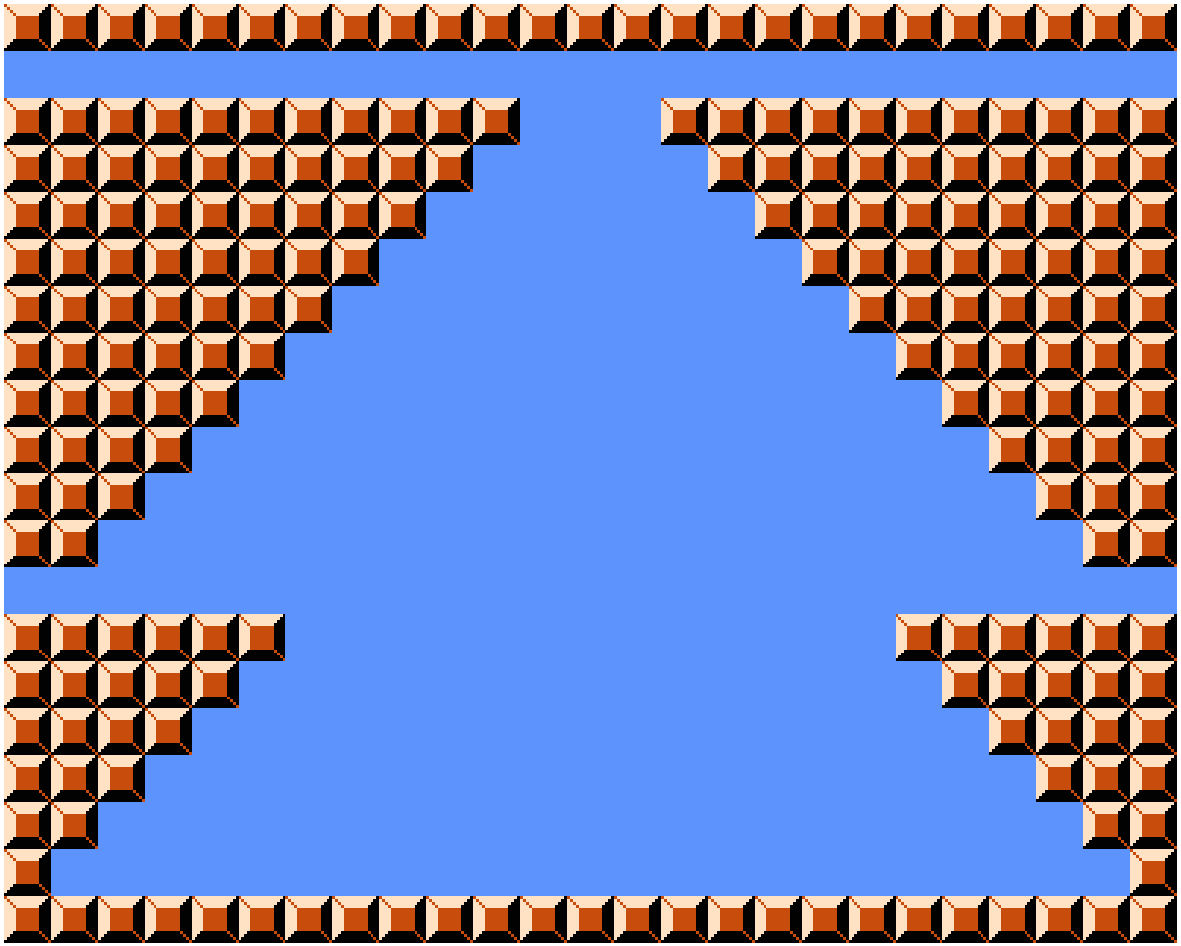


Glitches: walk through walls and walljump



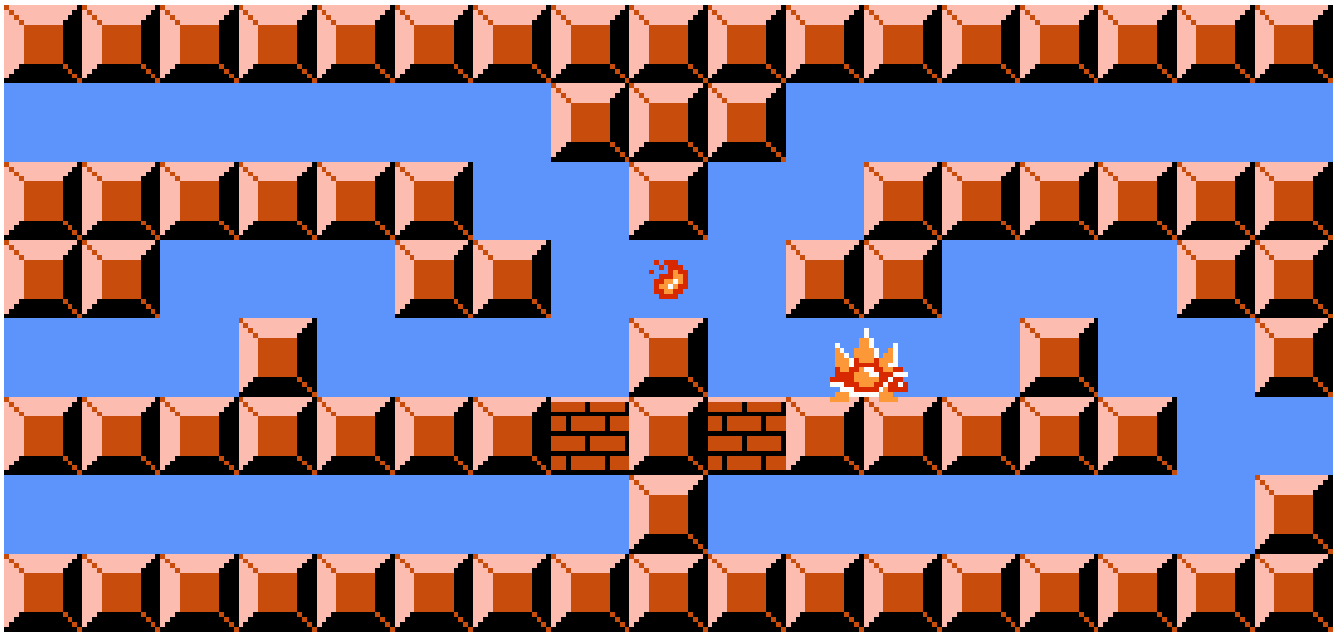
Reduction from Knapsack: OK!

Glitches: walk through walls and walljump



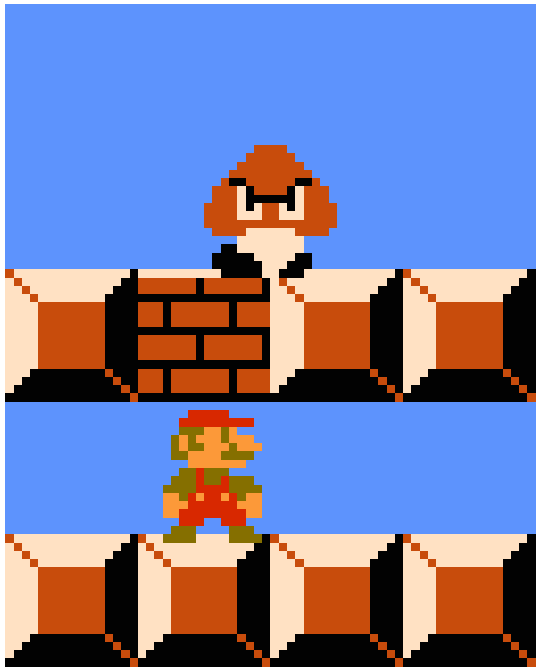
Crossover gadget: OK!

Glitches: walk through walls and walljump



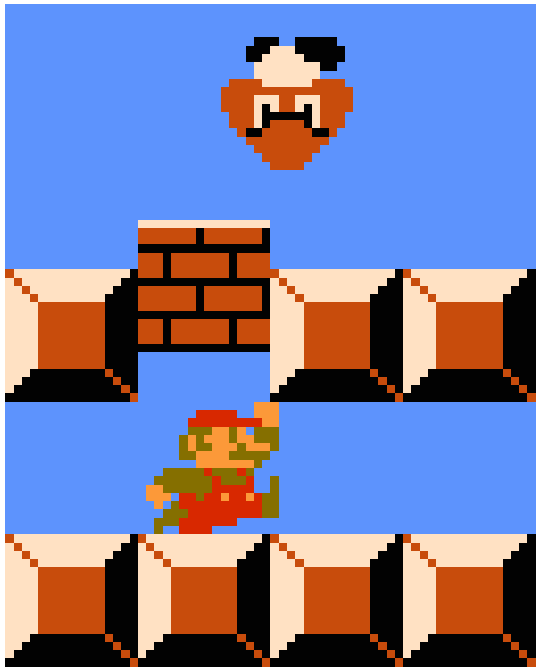
Door gadget: OK!

Glitches: walk through walls and walljump



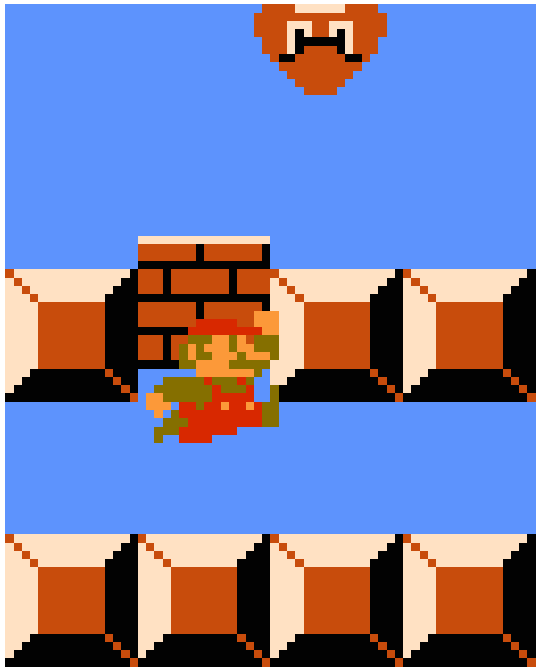
If there is a monster on top of a brick block, Mario can jump through it by quickly hitting it twice from below

Glitches: walk through walls and walljump



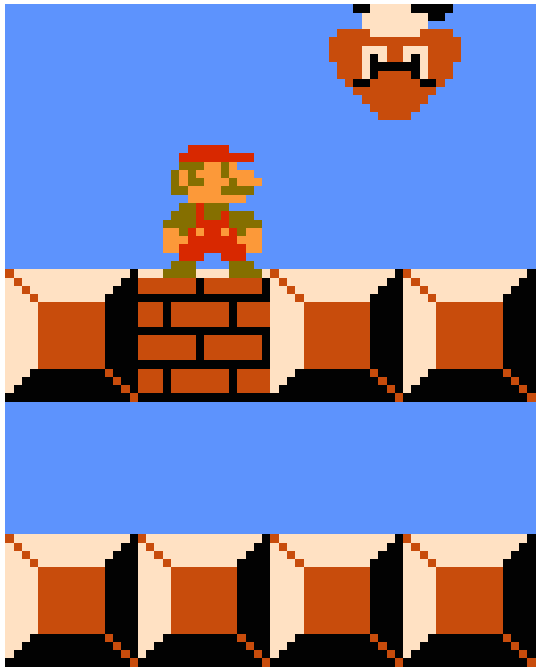
If there is a monster on top of a brick block, Mario can jump through it by quickly hitting it twice from below

Glitches: walk through walls and walljump



If there is a monster on top of a brick block, Mario can jump through it by quickly hitting it twice from below

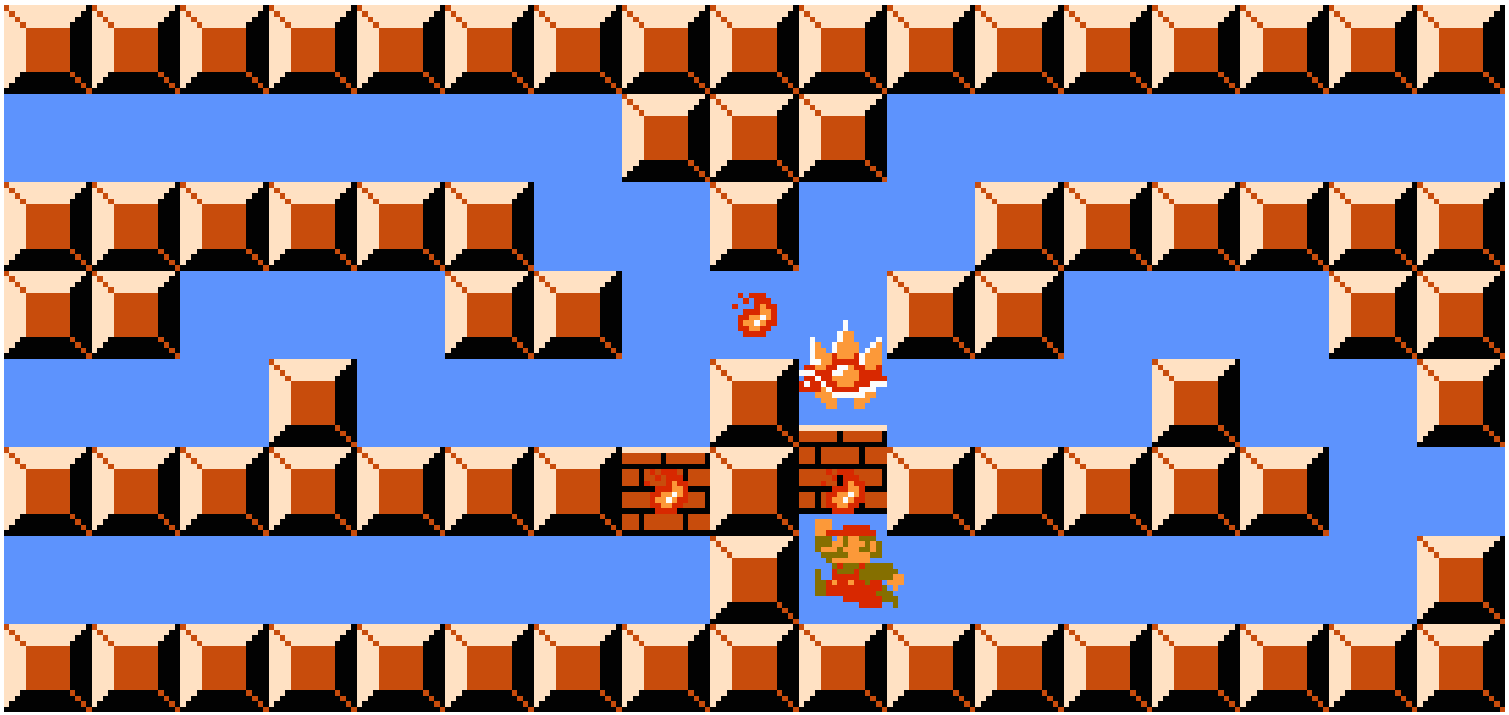
Glitches: walk through walls and walljump



If there is a monster on top of a brick block, Mario can jump through it by quickly hitting it twice from below

Glitches: walk through walls and walljump

Solution. Attach firebars to brick blocks




MARIO
094950
×36
WORLD
8-4
TIME
171
THANK YOU MARIO!
YOUR QUEST IS OVER.