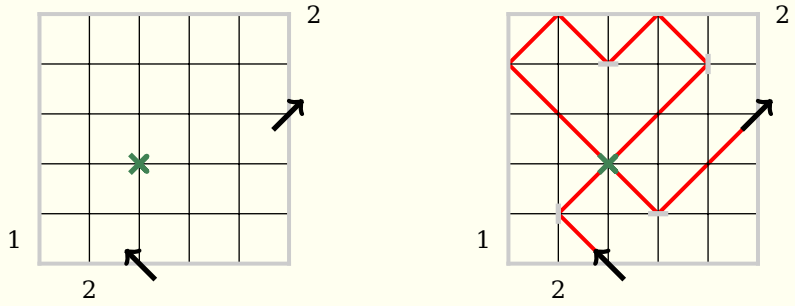
logicpuzzle.sty

v2.5

A style file for typesetting logic puzzles



2014/06/15

Package author:
Josef Kleber

josef.kleber@gmx.de
https://bitbucket.org/kleberj/logicpuzzle/

1 Roll out your own grid-based logic puzzle

As an example we take a look at the former `bokkusu.sty` package. First, we ignore the LPPL license stuff.

```
\ProvidesPackage{bokkusu}[2013/03/25 bokkusu.sty v1.2 - Josef Kleber (C) 2013]%
\RequirePackage{logicpuzzle}%
```

We wrote a package `bokkusu.sty` with version number v1.2 and date 2013/03/25 and added a copyright remark. We need to load the code base package `logicpuzzle.sty`.

```
\newcommand*\LP@BK@init@prefix{LP@BK}%
\newcommand*\LP@BK@init@package{bokkusu}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{rows}{5}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{columns}{5}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{scale}{1}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{counterstyle}{none}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{color}{black}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{bgcolor}{}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{width}{6.7cm}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{cvoffset}{-38pt}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{title}{}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{titleindent}{0.75cm}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{titlewidth}{5.85cm}%
\LP@define@choicekey@fontsize{\LP@BK@init@prefix}{\LP@BK@init@package}{Large}%
\ExecuteOptionsX{rows,columns,width,fontsize,scale,color,bgcolor,cvoffset,
                 counterstyle,title,titleindent,titlewidth}%
\ProcessOptionsX\relax%
```

We save the package prefix and name in a macro for easy change. Then we define the options for package `bokkusu.sty` and the environment `bokkusu`, which are executed afterwards to create the macros for the option values.

```
\let\valueH\LP@bottomrow%
\let\valueV\LP@leftcolumn%
\let\sumH\LP@toprow%
\let\sumV\LP@rightcolumn%
```

We need numbers around the grid. Therefore, we define some aliases for the existing generic commands.

```
\newcommand*\bokkususetup[1]%
{%
  \setkeys{bokkusu.sty}{#1}%
}%
```

We define `\bokkususetup` for resetting the global package options.

Finally, we define the bokkusu environment.

```
\newenvironment{bokkusu}[1][]%
{%
  \setkeys{bokkusu}{#1}%
  \LP@set@package{bokkusu}%
  \LP@set@env@prefix{LP@BK}%
  \setcounter{LP@rows}{\LP@BK@rows}%
  \setcounter{LP@columns}{\LP@BK@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
```

We locally set the environment options and the prefix and name of the current puzzle environment. We need to reset the counters for rows and columns, as they might have been altered.

```
  \begin{minipage}[t]{\LP@BK@width}%
    \ifthenelse{\equal{\LP@BK@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@BK@titleindent}\parbox{\LP@BK@titlewidth}
      {\strut\LP@titleformat\LP@BK@title}\vspace{3mm}\par}%
    \begin{tikzpicture}[scale=\LP@BK@scale]%
      \LP@drawbackground{1}{1}{\LP@BK@columns}{\LP@BK@rows}{\LP@BK@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@BK@columns}{\LP@BK@rows}{1cm}%
}%
```

We start a minipage with width {⟨*width*⟩}. If the user defined a title, we typeset the title and add a vertical space. Then, we draw the puzzle with the help of tikz.sty. We start drawing the background and the grid.

```
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@BK@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%
```

Finally, we just end the picture for the puzzle. We draw and step the counter. As last action, we need to close the minipage environment. That's it. Easy, isn't it? You just need to copy this skelton and change or add some code for your specific puzzle.

2 User documentation

2.1 PGF Layers

The logicpuzzle.sty package defines the PGF layers: LPdump, LPbgcolor, LPbackgroundtwo, LPbackground, LPforeground and LPforegroundtwo

Without specifying a special layer, the standard main layer is used. The LPbackground and LPforeground layers can be accessed with the puzzlebackground

and puzzleforeground environments. The LPbgcolor is and should only be used for the background color of the grid.

All layers can also be accessed with the generic PGF method:

```
\begin{pgfonlayer}{layer}
  ...
\end{pgfonlayer}
```

Order: LPdump → LPbgcolor → LPbackgroundtwo → LPbackground → main → LPforeground → LPforegroundtwo

So, if you are in the need to place something behind LPbackground or in front of LPforeground, you can use the LPbackgroundtwo and LPforegroundtwo layers. You can hide elements like help nodes behind the background color on the LPdump layer.

2.2 Environments

2.2.1 Puzzle environments

logicpuzzle The logicpuzzle environment is the generic environment for typesetting logic puzzles. With the optional argument of the environment, you can reset the options with local scope. Here, a blank grid is created. Furthermore, there are the other puzzle environments. They have their own set of options, that is also different option values and defaults! These can be changed with the \puzzlesetup commands with global scope or in the optional argument of the environment with local scope.

2.2.1.1 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: tiny, scriptsize, footnotesize, small, normalsize, large, Large, LARGE, huge, Huge

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.1cm] specifies the width of the box the title is set in.

color [] specifies the color for coloring the cells.

bgcolor [] sets the background color of the grid.

counterstyle [none] defines the counter style. Predefined styles: none, left, right

cvoffset [-23pt] sets the vertical offset of the counters in the margin.

2.2.2 Supporting environments

puzzlebackground The puzzlebackground environment allows you to place elements behind the main layer on the LPbackground layer. This is for example usefull for the \fillarea command.

puzzleforeground The puzzleforeground environment allows you to place elements in front of the main layer on the LPforeground layer. This is for example usefull for the \framearea command.

2.3 Commands

2.3.1 In the grid

\setcell \setcell{⟨*column*⟩}{⟨*row*⟩}{⟨*element*⟩}
sets an {⟨*element*⟩} into cell {⟨*column*⟩}{⟨*row*⟩} as central node. It is aware of the current values of the surrounding environment options rows, columns, scale and fontsize. Furthermore, a check if {⟨*element*⟩} is within the grid is applied.

\setcells \setcells{⟨*csv list*⟩}{⟨*element*⟩}
sets {⟨*element*⟩} into several cells by using the column/row format in {⟨*csv list*⟩}. It works for numbers, letters and most graphical objects, with the exception of commands like \KKR, which is not a graphic itself, but drawing something into the grid.

\setbigcell \setbigcell[⟨*fontsize*⟩]{⟨*column*⟩}{⟨*row*⟩}{⟨*element*⟩}
sets {⟨*element*⟩} into a big (2×2) cell {⟨*column*⟩}{⟨*row*⟩} as central node. The optional argument [⟨*fontsize*⟩] is set to 'Huge' by default.

\setrow \setrow{⟨*row*⟩}{⟨*csv list*⟩}
sets the contents of a {⟨*row*⟩}. These may be numbers or letters.

\setcolorrow \setcolorrow{⟨*row*⟩}{⟨*csv list*⟩}
sets the contents of a {⟨*row*⟩}. Furthermore, the background of the cell is filled with color LP@c@romannumber. With the number 0, you can black out the grid cell.

\setcolumn \setcolumn{⟨*column*⟩}{⟨*csv list*⟩}
sets the contents of a {⟨*column*⟩}. These may be numbers or letters.

`\setcolorcolumn` `\setcolorcolumn{⟨column⟩}{⟨csv list⟩}`
sets the contents of a {⟨column⟩}. Furthermore, the background of the cell is filled with color `LP@c@romannumber`.

`\setrule` `\setrule{⟨column⟩}{⟨row⟩}{⟨rule⟩}`
sets a calculation rule {⟨rule⟩} into the top left corner of cell {⟨column⟩}{⟨row⟩}. The rule is typeset in inline math mode. You might consider using the `\times` and `\div` commands.

`\fillcell` `\fillcell{⟨column⟩}{⟨row⟩}`
fills cell {⟨column⟩}{⟨row⟩} with the color defined with environment option `color`[1]. It is aware of the current values of the surrounding envionment options `rows`, `columns`, `scale` and `color`. Furthermore, a check if the cell is within the grid is applied.

`\fillrow` `\fillrow{⟨row⟩}{⟨csv list⟩}`
fills a {⟨row⟩}. In {⟨csv list⟩} '1' means 'fill' and '0' means 'don't fill'. Internally, `\fillrow` uses `\fillcell`.

`\fillcolumn` `\fillcolumn{⟨column⟩}{⟨csv list⟩}`
fills a {⟨column⟩}. In {⟨csv list⟩} '1' means 'fill' and '0' means 'don't fill'. Internally, `\fillcolumn` uses `\fillcell`.

`\filldiagonals` `\filldiagonals[⟨color⟩]`
fills the diagonals with the color specified with the optional argument [⟨color⟩] (default: yellow!20). Furthermore, it checks for a quadratic grid, otherwise an error message is issued.

`\framearea` `\framearea{⟨color⟩}{⟨TikZ path⟩}`
frames the area given by {⟨TikZ path⟩} with color {⟨color⟩}. The reference for coordinates is the bottom left corner of the cell.

```
\framearea{green}{(2,2)--(2,3)--(3,3)--(3,2)--(2,2)}
```

This command will color the frame of the grid cell `(2,2)` green. You should consider using this command in the `puzzleforeground` environment.

`\fillarea` `\fillarea{⟨color⟩}{⟨TikZ path⟩}`
fills the area given by {⟨TikZ path⟩} with color {⟨color⟩}. The reference for coordinates is the bottom left corner of the cell. You should consider using this command in the `puzzlebackground` environment.

`\colorarea` `\colorarea{⟨color⟩}{⟨TikZ path⟩}`
fills the area given by {⟨TikZ path⟩} with color {⟨color⟩} – just like `\framearea` without frame.

[1] Therefore, you must define an option `color` in the style file you want to use fill commands

\framepuzzle `\framepuzzle[⟨color⟩]`
frames the grid (thicker line) with the color specified with the optional argument [⟨color⟩] (default: black).

\tikzzpath `\tikzpath{⟨column⟩}{⟨row⟩}{⟨csv list⟩}`
does easily construct a TikZ path. You just need to define a starting point {⟨column⟩}{⟨row⟩} (bottom left corner) and a {⟨csv list⟩} with direction indicators relative to the current position.

7:	up left	8:	up	9:	up right
4:	left	5:	no change	6:	right
1:	down left	2:	down	3:	down right

```
\framearea{green}{\tikzpath{2}{2}{8,6,2,4}}
```

This command will frame grid cell (2,2) green.

\xtikzpath `\xtikzpath{⟨column⟩}{⟨row⟩}{⟨csv list⟩}`
is an evolution of the \tikzpath command with slightly different input syntax. In the {⟨csv list⟩} argument, it expects pairs in the form direction/length. Therefore, you can easily define paths from corner to corner.

```
\framearea{green}{\xtikzpath{2}{2}{8/2,6/2,2/2,4/2}}
```

This command will frame an area defined by the grid cells (2,2) and (3,3) green.

2.3.2 Presentation

\titleformat `\titleformat{⟨format⟩}`
defines the {⟨format⟩} of the title. By default, the definition is as follows:

```
\titleformat{\centering\Large\color{blue}}
```

\puzzlecounter `\puzzlecounter`
provides the puzzle counter in textual form to use it in \definecounterstyle.

\setpuzzlecounter `\setpuzzlecounter{⟨number⟩}`
resets the puzzle counter, for example before the solutions.

\definecounterstyle `\definecounterstyle{⟨name⟩}{⟨definition⟩}`
allows you to define your own styles. For example, the style left is defined as follows:

```
\definecounterstyle{left}{
  \begingroup\reversemarginpar\marginnote{
  \tikz\node[shape=rectangle,fill=yellow!40,inner sep=7pt,
```

```
            draw,rounded corners=3pt,thick]
  {\Huge\puzzlecounter};}[\LP@cvoffset]\endgroup}
}
```

To typeset the counter into the margin we use the command `\marginnote`. We need to use the command `\reversemarginpar` to set the counter into the left margin. Of course, we must use this command in a group for local scope. Finally we use `\puzzlecounter` in a `\tikz` node with a vertical offset set with the option `cvoffset`.

`\setgridlinestyle` `\setgridlinestyle{⟨style⟩}`
sets the style of lines used in the grid. By default, the style is set to `solid`, whereas `slitherlink` uses `dashed`.

`\setnormallinewidth` `\setnormallinewidth{⟨dimension⟩}`
sets the width of the standard lines (default: `0.5pt`)

`\setthicklinewidth` `\setthicklinewidth{⟨dimension⟩}`
sets the width of the 'thicker' lines (default: `1.5pt`)

2.3.3 Puzzle specific commands

2.3.3.1 2D-Sudoku

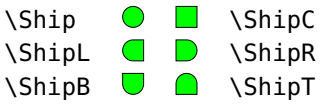
`\ddsudokucell` `\ddsudokucell{⟨column⟩}{⟨row⟩}{⟨number⟩}`
sets {⟨*number*⟩} into grid cell {⟨*column*⟩}{⟨*row*⟩}.

`\ddsudokusetup` `\ddsudokusetup{⟨options⟩}`
resets the options with global scope.

2.3.3.2 Battleship

`\placeship` `\placeship{⟨direction⟩}{⟨column⟩}{⟨row⟩}{⟨length⟩}`
places complete ships in the grid. It expects the specification of the direction as horizontal (H) or vertical (V). Furthermore, it requires the starting coordinates and the length of the ship.

`\placesegment` `\placesegment{⟨column⟩}{⟨row⟩}{⟨ship segment⟩}`
is used for the placement of ship segments in the grid. In the mandatory argument {⟨*ship segment*⟩}, you can use the following commands:



`\ship` `\ship`
The command `\ship` was replaced by the `\placesegment` command. The command `\ship` is deprecated and should not be used longer. It may still be used, but it is not recommended.

`\placewater` `\placewater{⟨column⟩}{⟨row⟩}`
places water markers (•) in the grid.

`\placeisland` `\placeisland{⟨column⟩}{⟨row⟩}`
places islands ( ) in the grid. The island outlines are created randomly: , , , ...

`\shipH` `\shipH{⟨csv list⟩}`
typesets the horizontal numbers above the grid. It expects a comma-separated list as an argument.

`\shipV` `\shipV{⟨csv list⟩}`
typesets the vertical numbers beside the grid. It also expects a comma separated list.

`\shipbox` `\shipbox{⟨csv list⟩}`
defines the number and size of the ships, which are typeset under the grid.

`\battleshipsetup` `\battleshipsetup{⟨options⟩}`
resets the options with global scope.

`\classicgame` `\classicgame{⟨csv list⟩}`
typesets a game sheet for playing classic Battleship. It expects a comma separated list with the number and sizes of the ships.

2.3.3.3 Bokkusu

`\valueH` `\valueH{⟨csv list⟩}`
typesets the numbers left to the grid indicating the values of the cells. It expects a comma-separated list as an argument.

`\valueV` `\valueV{⟨csv list⟩}`
typesets the numbers below the grid specifying the values of the cells. It also expects a comma separated list.

`\sumH` `\sumH{⟨csv list⟩}`
typesets the numbers right to the grid indicating the sums of the values of the colored cells. It expects a comma-separated list.

`\sumV` `\sumV{⟨csv list⟩}`
typesets the numbers above the grid specifying the sums of the values of the colored cells. It expects a comma separated list.

`\bokkususetup` `\bokkususetup{⟨options⟩}`
resets the options with global scope.

2.3.3.4 Bridges

`\bridgesrow` `\bridgesrow{⟨row⟩}{⟨csv list⟩}`
sets the contents of a bridges {⟨*row*⟩}. These are the numbers indicating how many bridges originate from this specific island.

`\bridgescolumn` `\bridgescolumn{⟨column⟩}{⟨csv list⟩}`
sets the contents of a bridges {⟨*column*⟩}.

`\bridge` `\bridge[⟨optional arguments⟩]margTikZ path`
draws the bridges between islands. With the optional argument [⟨*double*⟩] you can draw a double bridge. Furthermore, you can set the color of the bridge with the option [⟨*color*⟩].

`\bridgessetup` `\bridgessetup{⟨options⟩}`
resets the options with global scope.

2.3.3.5 Chaos Sudoku

`\chaossudokucell` `\chaossudokucell{⟨column⟩}{⟨row⟩}{⟨number⟩}`
sets {⟨*number*⟩} into grid cell {⟨*column*⟩}{⟨*row*⟩}.

`\chaossudokusetup` `\chaossudokusetup{⟨options⟩}`
resets the options with global scope.

2.3.3.6 Four Winds

`\fourwindscell` `\fourwindscell{⟨column⟩}{⟨row⟩}{⟨number⟩}`
sets {⟨*number*⟩} into grid cell {⟨*column*⟩}{⟨*row*⟩}. Furthermore, it draws lines specified in {⟨*csv list*⟩} in the `direction/length` format.

`\fourwindssetup` `\fourwindssetup{⟨options⟩}`
resets the options with global scope.

2.3.3.7 Hakyuu

`\hakyuucell` `\hakyuucell{⟨column⟩}{⟨row⟩}{⟨number⟩}`
sets {⟨*number*⟩} into grid cell {⟨*column*⟩}{⟨*row*⟩}.

`\hakyuusetup` `\hakyuusetup{⟨options⟩}`
reset the options with global scope.

2.3.3.8 Hitori

`\hitorisetup` `\hitorisetup{`⟨*options*⟩`}`
resets the options with global scope.

2.3.3.9 Kakuro

`\kakurorow` `\kakurorow{`⟨*row*⟩`}{`⟨*csv list*⟩`}`
sets the contents of a kakuro {⟨*row*⟩}. These may be numbers and the commands `\KKR` or `\Black`.

`\kakurocolumn` `\kakurocolumn{`⟨*column*⟩`}{`⟨*csv list*⟩`}`
sets the contents of a kakuro {⟨*column*⟩}.

`\KKR` `\KKR{`⟨*sumV*⟩`}{`⟨*sumH*⟩`}`
sets the contents of a kakuro cell.

`\Black` `\Black`
blacks out a cell.

`\kakurosetup` `\kakurosetup{`⟨*options*⟩`}`
resets the options with global scope.

2.3.3.10 Kendoku

`\kendokucell` `\kendokucell{`⟨*column*⟩`}{`⟨*row*⟩`}{`⟨*number*⟩`}`
sets {⟨*number*⟩} into grid cell {⟨*column*⟩}{⟨*row*⟩}.

`\kendokusetup` `\kendokusetup{`⟨*options*⟩`}`
resets the options with global scope.

2.3.3.11 Killer Sudoku

`\killersudokucell` `\killersudokucell{`⟨*column*⟩`}{`⟨*row*⟩`}{`⟨*number*⟩`}`
sets {⟨*number*⟩} into grid cell {⟨*column*⟩} {⟨*row*⟩}.

`\killersudokusetup` `\killersudokusetup{`⟨*options*⟩`}`
resets the options with global scope.

2.3.3.12 Laser Beam

`\laserH` `\laserH{⟨csv list⟩}`
typesets the numbers above the grid indicating how many cells are traversed by the laser beam. It expects a comma-separated list as an argument.

`\laserV` `\laserV{⟨csv list⟩}`
typesets the numbers left to the grid.

`\mirrorH` `\mirrorH{⟨csv list⟩}`
typesets the numbers below the grid indicating how many mirrors are placed in the intersections of this column.

`\mirrorV` `\mirrorV{⟨csv list⟩}`
typesets the numbers right to the grid.

`\placearrow` `\placearrow{⟨column⟩}{⟨row⟩}`
is used for the placement of arrows at the grid frame. The reference for coordinates is the bottom left corner of the cell. In the mandatory argument {⟨*direction*⟩}, you can use the following indicators: LeftUp, LeftDown, RightUp, RightDown

`\placecross` `\placecross{⟨column⟩}{⟨row⟩}`
places a cross in the intersections of the grid.

`\placemirror` `\placemirror{⟨column⟩}{⟨row⟩}{⟨direction⟩}`
places mirrors in the intersections of the grid. In the mandatory argument {⟨*direction*⟩}, you can use the following indicators: H, V

`\laser` `\laser[⟨color⟩]{⟨TikZ path⟩}`
draws the laser beam given by {⟨*TikZ path*⟩} with color [⟨*color*⟩] (default: red). The reference for coordinates is the bottom left corner of the cell.

```
\laser[green]{(1,2)--(2,3)--(1,4)}
```

You should consider using this command in the `puzzlebackground` environment.

`\laserbeamsetup` `\laserbeamsetup{⟨options⟩}`
resets the options with global scope.

2.3.3.13 Magic Labyrinth

`\magiclabyrinthcell` `\magiclabyrinthcell{⟨column⟩}{⟨row⟩}{⟨number⟩}`
sets a number into grid cell {⟨*column*⟩} {⟨*row*⟩}.

\mlline `\mlline{⟨TikZ path⟩}`
draws a line given by {⟨*TikZ path*⟩}.

\magiclabyrinthsetup `\magiclabyrinthsetup{⟨options⟩}`
resets the options with global scope.

2.3.3.14 Magnets

\plusH `\plusH{⟨csv list⟩}`
typesets the numbers above the grid indicating how many positive poles are in the respective column. It expects a comma-separated list as an argument.

\minusH `\minusH{⟨csv list⟩}`
typesets the numbers above the grid indicating how many negative poles are in the respective column.

\plusV `\plusV{⟨csv list⟩}`
typesets the numbers left to the grid indicating how many positive poles are in the respective row.

\minusV `\minusV{⟨csv list⟩}`
typesets the numbers left to the grid indicating how many negative pole ares in the respective row.

\magnetsH `\magnetsH{⟨csv list⟩}`
typesets non-magnetic horizontal plates by using the `column/row` format in {⟨*csv list*⟩}.

\magnetsV `\magnetsV{⟨csv list⟩}`
typesets non-magnetic vertical plates by using the `column/row` format in {⟨*csv list*⟩}.

\PMH `\PMH{⟨csv list⟩}`
\MPH draws horizontal magnetic plates with $\boxed{+\,-}$ arrangement. It expects the
\PMV `column/row` format in {⟨*csv list*⟩}. You can typeset the three other magnetic
\MPV arrangements by using the `\MPH`, `\PMV` and `\MPV` commands.

\magnetssetup `\magnetssetup{⟨options⟩}`
resets the options with global scope.

2.3.3.15 Masyu

\masyucell `\masyucell{⟨column⟩}{⟨row⟩}{⟨element⟩}`
sets an element into grid cell {⟨*column*⟩} {⟨*row*⟩}.

\MasyuW `\MasyuW`
draws an empty (white) circle.

\MasyuB `\MasyuB`
draws a black circle.

\masyuline `\masyuline{⟨TikZ path⟩}`
draws a line given by {⟨TikZ path⟩}.

\masyusetup `\masyusetup{⟨options⟩}`
resets the options with global scope.

2.3.3.16 Minesweeper

\Mine `\Mine`
draws a mine. It can be used in commands like `\setcell` or `\setrow`!

\minesweepersetup `\minesweepersetup{⟨options⟩}`
resets the options with global scope.

2.3.3.17 Nonogram

\nonogramrow `\nonogramrow{⟨row⟩}{⟨csv list⟩}`
sets the contents of row {⟨row⟩}. In {⟨csv list⟩} it expects the `column/length` format.

\nonogramcolumn `\nonogramcolumn{⟨column⟩}{⟨csv list⟩}`
sets the contents of column {⟨column⟩}. In {⟨csv list⟩} it expects the `row/length` format.

\nonogramV `\nonogramV{⟨csv list⟩}`
sets the contents of the extra cells left to the grid. By definition, the first number is always typeset next to the grid!

\nonogramH `\nonogramH{⟨csv list⟩}`
sets the contents of the extra cells on top of the grid.

\puzzlestrut `\puzzlestrut`
serves the height adjustment depending on option `extracells` when you want to typeset puzzle and solution (without extra cells) next to each other.

\nonogramsetup `\nonogramsetup{⟨options⟩}`
resets the options with global scope.

2.3.3.18 Number Link

\numberlinkcell `\numberlinkcell{⟨column⟩}{⟨row⟩}{⟨element⟩}`
sets a number or letter into grid cell {⟨column⟩}{⟨row⟩}.

\link `\link{⟨{⟨TikZ path⟩}⟩}`
draws a line given by {⟨*TikZ path*⟩}.

\numberlinksetup `\numberlinksetup{⟨*options*⟩}`
resets the options with global scope.

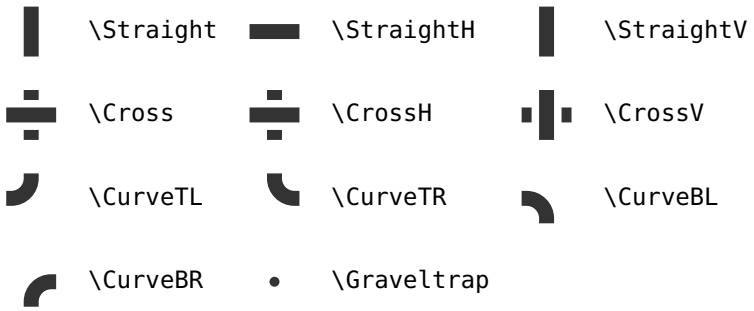
2.3.3.19 Resuko

\resukocell `\resukocell{⟨*column*⟩}{⟨*row*⟩}{⟨*element*⟩}`
sets the {⟨*element*⟩} into grid cell {⟨*column*⟩}{⟨*row*⟩}.

\Straight
\StraightH
\StraightV
\Cross
\CrossH
\CrossV
\CurveTL
\CurveTR
\CurveBL
\CurveBR
\Graveltrap

2.3.3.19.1 Track tiles

You can use the following commands to draw different track tiles, e.g. with the `\resukocell` command:



\pitlane `\pitlane{⟨*column*⟩}{⟨*row*⟩}{⟨*direction*⟩}`
draws the pit lane in grid cell {{⟨*column*⟩}}{{⟨*row*⟩}} with {{⟨*direction*⟩}} V or H.

\parkinglot `\parkinglot{⟨*column*⟩}{⟨*row*⟩}`
draws the parking lot in grid cell {⟨*column*⟩}{⟨*row*⟩}.

\trackH `\trackH{⟨*csv list*⟩}`
typesets the track tiles below the grid indicating how many different tiles are in the respective column. It expects a comma-separated list as an argument with the format `straights/curves/intersections`.

\trackV `\trackV{⟨*csv list*⟩}`
typesets the track tiles left to the grid.

\track `\track{⟨*TikZ path*⟩}`
draws the race track given by {⟨*TikZ path*⟩}. The design of the race track is based on Frédéric's answer to this question on TEX.sx. The design with auto-generated bridges will only work, if the path is not constructed with an intersection point. It's recommended to start the path on a standard straight and define the path from corner to corner with `\xtikzpath`.

\resukosetup `\resukosetup{⟨*options*⟩}`
resets the options with global scope.

2.3.3.20 Schatzsuche

`\Diamond` `\Diamond`
draws a diamond. It can be used in commands like `\setcell` or `\setrow`!

`\schatzsuchesetup` `\schatzsuchesetup{`⟨*options*⟩`}`
resets the options with global scope.

2.3.3.21 Skyline

`\skylineT` `\skylineT{`⟨*csv list*⟩`}`
typesets the numbers above the grid indicating how many skycrapers are visible. It expects a comma-separated list as an argument.

`\skylineB` `\skylineB{`⟨*csv list*⟩`}`
typesets the numbers below the grid.

`\skylineL` `\skylineL{`⟨*csv list*⟩`}`
typesets the numbers left to the grid.

`\skylineR` `\skylineR{`⟨*csv list*⟩`}`
typesets the numbers right to the grid.

`\skylinecell` `\skylinecell{`⟨*column*⟩`}{`⟨*row*⟩`}}{`⟨*height*⟩`}`
sets {⟨*height*⟩} into grid cell {⟨*column*⟩}{⟨*row*⟩}.

`\skylinesetup` `\skylinesetup{`⟨*options*⟩`}`
resets the options with global scope.

2.3.3.22 Slitherlink

`\slitherlinkcell` `\slitherlinkcell{`⟨*column*⟩`}{`⟨*row*⟩`}{`⟨*number*⟩`}`
sets {⟨*number*⟩} into grid cell {⟨*column*⟩}{⟨*row*⟩}.

`\slitherlinksetup` `\slitherlinksetup{`⟨*options*⟩`}`
resets the options with global scope.

2.3.3.23 Star Battle

`\starbattlecell` `\starbattlecell{`⟨*column*⟩`}{`⟨*row*⟩`}{`⟨*element*⟩`}`
sets {⟨*element*⟩} into grid cell {⟨*column*⟩}{⟨*row*⟩}, e.g. the `\Star` command.

`\starbattlesetup` `\starbattlesetup{`⟨*options*⟩`}`
resets the options with global scope.

2.3.3.24 Stars and Arrows

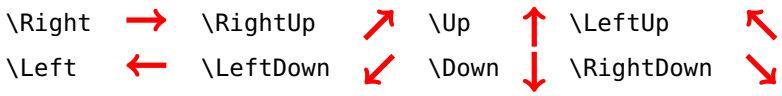
\starsH `\starsH{`⟨*csv list*⟩`}`
typesets the numbers above the grid indicating how many stars are in the respective column. It expects a comma-separated list as an argument.

\starsV `\starsV{`⟨*csv list*⟩`}`
typesets the numbers left to the grid.

\Star `\Star`
draws a star. It can be used in commands like `\setcell` or `\setrow`!

2.3.3.24.1 Arrows

\Right
\RightUp
\Up
\LeftUp
\Left
\LeftDown
\Down
\RightDown
You can use the following commands to draw different arrows:



\starsandarrowssetup `\starsandarrowssetup{`⟨*options*⟩`}`
resets the options with global scope.

2.3.3.25 Sudoku

\lpsudokucell `\lpsudokucell{`⟨*column*⟩`}{`⟨*row*⟩`}{`⟨*number*⟩`}`
sets {⟨*number*⟩} into grid cell {⟨*column*⟩}{⟨*row*⟩}.

\lpsudokusetup `\lpsudokusetup{`⟨*options*⟩`}`
resets the options with global scope.

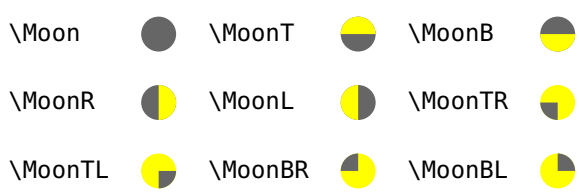
2.3.3.26 Sun and Moon

\Star `\Star`
draws a star. It can be used in commands like `\setcell` or `\setrow`!

\Cloud `\Cloud`
draws a dark cloud.

2.3.3.26.1 Howl at the Moon

\Moon
\MoonT
\MoonB
\MoonR
\MoonL
\MoonTR
\MoonTL
\MoonBR
\MoonBL
You can use the following commands to draw different iluminated moons:



\sunandmoonssetup `\sunandmoonssetup{`⟨*options*⟩`}`
resets the options with global scope.

2.3.3.27 Tents and Trees

`\tentH` `\tentH{`⟨*csv list*⟩`}`
typesets the numbers above the grid indicating how many tents are in the respective column. It expects a comma-separated list as an argument.

`\tentV` `\tentV{`⟨*csv list*⟩`}`
typesets the numbers left to the grid.

`\Tree` `\Tree`
draws a tree. It can be used in commands like `\setcell` or `\setrow`! The design of the tree is based on Alain Matthes' answer to this question on TEX.sx.

`\Tent` `\Tent`
draws a tent.

`\tentsandtreessetup` `\tentsandtreessetup{`⟨*options*⟩`}`
resets the options with global scope.

2.3.3.28 Tunnel

`\tunnelH` `\tunnelH{`⟨*csv list*⟩`}`
typesets the numbers above the grid indicating how many tube segments are in the respective column. It expects a comma-separated list as an argument.

`\tunnelV` `\tunnelV{`⟨*csv list*⟩`}`
typesets the numbers left to the grid.

`\portal` `\portal{`⟨*column*⟩`}}{{`⟨*row*⟩`}`
is used for the placement of tunnel portals in the grid.

`\tube` `\tube{`⟨*TikZ path*⟩`}`
draws the tunnel tube given by `{`⟨*TikZ path*⟩`}`. The reference for coordinates is the center of the cell. The design of the tube is based on Xoff's answer to this question on TEX.sx.

```
\tube{(1.5,2.5)--(3.5,2.5)--(3.5,4.5)}
```

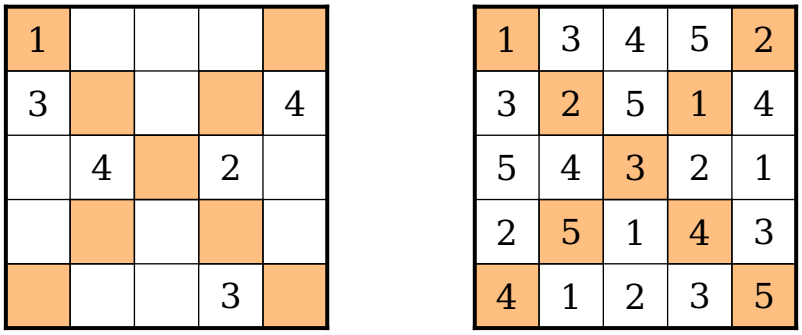
`\tunnelsetup` `\tunnelsetup{`⟨*options*⟩`}`
resets the options with global scope.

3 Examples

3.1 2D-Sudoku

Fill every row, every column and each of the two diagonals – if indicated – with numbers from 1 to SIZE of the grid.

3.1.1 Example




```
\begin{center}
  \begin{ddsudoku}
    \framepuzzle
    \filldiagonals[orange!50]
    \ddsudokucell{1}{5}{1}
    \ddsudokucell{1}{4}{3}
    \ddsudokucell{2}{3}{4}
    \ddsudokucell{4}{1}{3}
    \ddsudokucell{4}{3}{2}
    \ddsudokucell{5}{4}{4}
  \end{ddsudoku}
  \hspace{1.5cm}
  \begin{ddsudoku}
    \framepuzzle
    \filldiagonals[orange!50]
    \setrow{5}{1,3,4,5,2}
    \setrow{4}{3,2,5,1,4}
    \setrow{3}{5,4,3,2,1}
    \setrow{2}{2,5,1,4,3}
    \setrow{1}{4,1,2,3,5}
  \end{ddsudoku}
\end{center}
```

3.1.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.1cm] specifies the width of the box the title is set in.

bgcolor [] sets the background color of the grid.

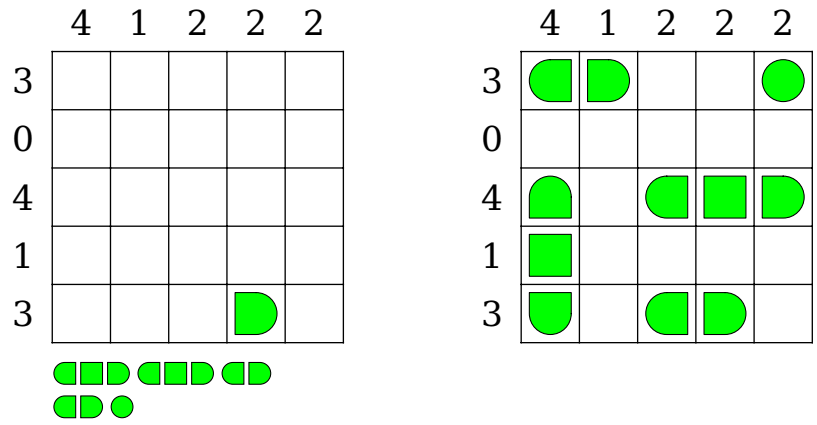
counterstyle [none] defines the counter style. Predefined styles: none, left, right

cvoffset [-23pt] sets the vertical offset of the counters in the margin.

3.2 Battleship

Try to find the positions of the ships listed below the puzzle. The numbers on the side of the puzzle reveals how many ship segments can be found in the rows and columns. All remaining fields indicate 'water'. Consider the following rules: The ships are arranged horizontally and vertically. No ship touches another ship at any point, not even diagonally.

3.2.1 Example




```
\begin{center}
  \begin{battleship}
    \placesegment{4}{1}{\ShipR}
    \shipH{4,1,2,2,2}
    \shipV{3,1,4,0,3}
```

```
    \shipbox{3,3,2,2,1}
  \end{battleship}
  \hspace{1.5cm}
  \begin{battleship}
    \placeship{V}{1}{1}{3}
    \placeship{H}{1}{5}{2}
    \placeship{H}{3}{1}{2}
    \placeship{H}{3}{3}{3}
    \placeship{H}{5}{5}{1}
    \shipH{4,1,2,2,2}
    \shipV{3,1,4,0,3}
  \end{battleship}
\end{center}
```

3.2.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid.

shipcolor [green] sets the color of the ship segments.

width [6cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0.75cm] defines the indent of the title.

titlewidth [5.15cm] specifies the width of the box the title is set in.

sbindent [0.75cm] defines the indent of the ship box below the grid.

sbwidth [5.15cm] specifies the width of the minipage, in which the ships are typeset.

sbshipscale [1] scales the size of the ships in the ship box.

bgcolor [] sets the background color of the grid.

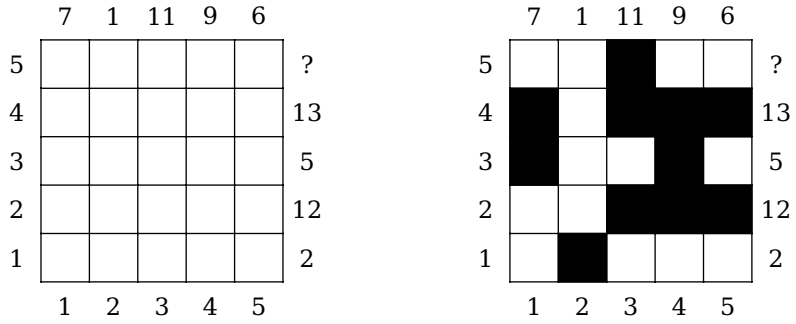
counterstyle [none] defines the counter style. Predefined styles: none, left, right

cvoffset [-23pt] sets the vertical offset of the counters in the margin.

3.3 Bokkusu

Black out some of the grid cells. The numbers on the left and the bottom edge of the grid indicate the values of the cells for adding up. The numbers on the right and the top edge of the grid specify the sums of the values of the colored cells.

3.3.1 Example




```
\begin{center}
  \begin{bokkusu}
    \valueH{1,2,3,4,5}
    \valueV{1,2,3,4,5}
    \sumH{7,1,11,9,6}
    \sumV{2,12,5,13,?}
  \end{bokkusu}
  \hspace{1.5cm}
  \begin{bokkusu}
    \valueH{1,2,3,4,5}
    \valueV{1,2,3,4,5}
    \sumH{7,1,11,9,6}
    \sumV{2,12,5,13,?}
    \fillrow{5}{0,0,1,0,0}
    \fillrow{4}{1,0,1,1,1}
    \fillrow{3}{1,0,0,1,0}
    \fillrow{2}{0,0,1,1,1}
    \fillrow{1}{0,1,0,0,0}
  \end{bokkusu}
\end{center}
```

3.3.2 Options

`rows` [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [6.7cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0.75cm] defines the indent of the title.

titlewidth [5.85cm] specifies the width of the box the title is set in.

color [black] specifies the color for coloring the cells.

bgcolor [] sets the background color of the grid.

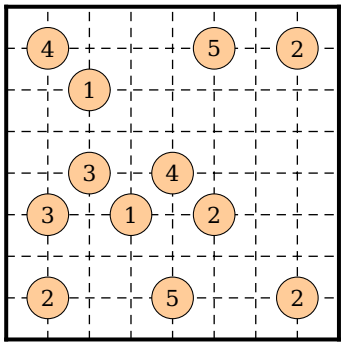
counterstyle [none] defines the counter style. Predefined styles: none, left, right

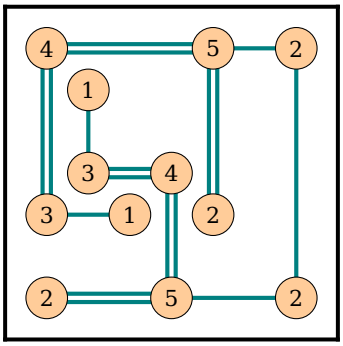
cvoffset [-38pt] sets the vertical offset of the counters in the margin.

3.4 Bridges

Connect all the islands (circles) located in the grid by bridges. The bridges may only be routed horizontally and vertically. Islands may be connected by a maximum of two bridges. The bridges must neither overlap nor cross. They may also not be built over islands. The numbers in the islands indicate how many bridges originate from this island. All islands must be fully connected.

3.4.1 Example





```
\colorlet{LP@c@bridge}{Teal}
\begin{center}
  \begin{bridges}
    \framepuzzle
    \bridgesrow{8}{{},4,{},{},{},5,{},2}
    \bridgesrow{7}{{},{},1}
    \bridgesrow{5}{{},{},3,{},4}
    \bridgesrow{4}{{},3,{},1,{},2}
    \bridgesrow{3}{{},{},{},{},{},{},{}}
    \bridgesrow{2}{{},2,{},{},5,{},{},2}
  \end{bridges}
  \hspace{1.5cm}
  \begin{bridges}[grid=none]
    \framepuzzle
    \bridgesrow{8}{{},4,{},{},{},5,{},2}
    \bridgesrow{7}{{},{},1}
    \bridgesrow{5}{{},{},3,{},4}
    \bridgesrow{4}{{},3,{},1,{},2}
    \bridgesrow{3}{{},{},{},{},{},{},{}}
    \bridgesrow{2}{{},2,{},{},5,{},{},2}
    \bridge[double]{\tikzpath{2}{4}{8,8,8,8,6,6,6,6,2,2,2,2}}
    \bridge[double]{\tikzpath{2}{2}{6,6,6,8,8,8,4,4}}
    \bridge{\tikzpath{2}{4}{6,6}}
    \bridge{\tikzpath{3}{5}{8,8}}
    \bridge{\tikzpath{5}{2}{6,6,6,8,8,8,8,8,8,4,4}}
  \end{bridges}
\end{center}
```

3.4.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [6.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [6.1cm] specifies the width of the box the title is set in.

color [green] specifies the color for coloring the islands.

bgcolor [] sets the background color of the grid.

counterstyle [none] defines the counter style. Predefined styles: none, left, right

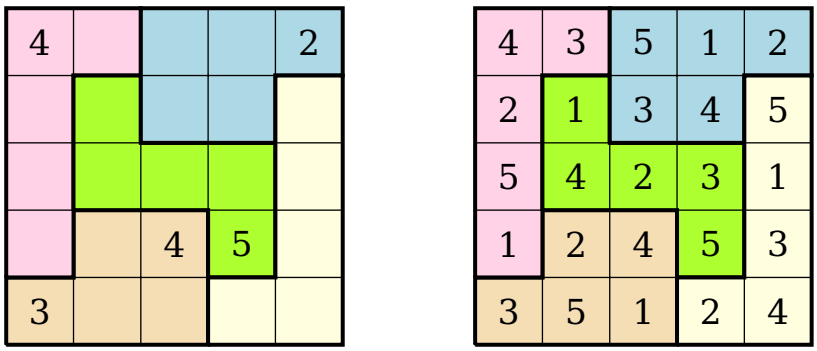
cvoffset [-23pt] sets the vertical offset of the counters in the margin.

grid [dashed] sets the style of the grid. Possible values: dashed, none, solid

3.5 Chaos Sudoku

Fill the cells of an area with numbers from 1 to N of the N*N grid. Each number can appear only once - in each area, column, row or diagonal if indicated.

3.5.1 Example




```
\begin{center}
  \begin{chaossudoku}
    \chaossudokucell{1}{1}{3}
    \chaossudokucell{1}{5}{4}
    \chaossudokucell{3}{2}{4}
    \chaossudokucell{4}{2}{5}
    \chaossudokucell{5}{5}{2}
    \begin{puzzlebackground}
      \fillarea{Wheat}{(1,1)--(1,2)--(2,2)--(2,3)--(4,3)--(4,1)
                       --(1,1)}
      \fillarea{HotPink!30}{(1,2)--(1,6)--(3,6)--(3,5)--(2,5)
                            --(2,2)--(1,2)}
      \fillarea{GreenYellow}{(2,3)--(2,5)--(3,5)--(3,4)--(5,4)
                             --(5,2)--(4,2)--(4,3)--(2,3)}
      \fillarea{LightBlue}{(3,4)--(3,6)--(6,6)--(6,5)--(5,5)
                           --(5,4)--(3,4)}
      \fillarea{LightYellow}{(4,1)--(4,2)--(5,2)--(5,5)--(6,5)
                             --(6,1)--(4,1)}
```

```
    \end{puzzlebackground}
  \end{chaossudoku}
  \hspace{1.5cm}
  \begin{chaossudoku}
    \setrow{5}{4,3,5,1,2}
    \setrow{4}{2,1,3,4,5}
    \setrow{3}{5,4,2,3,1}
    \setrow{2}{1,2,4,5,3}
    \setrow{1}{3,5,1,2,4}
    \begin{puzzlebackground}
      \fillarea{Wheat}{(1,1)--(1,2)--(2,2)--(2,3)--(4,3)--(4,1)
                       --(1,1)}
      \fillarea{HotPink!30}{(1,2)--(1,6)--(3,6)--(3,5)--(2,5)
                            --(2,2)--(1,2)}
      \fillarea{GreenYellow}{(2,3)--(2,5)--(3,5)--(3,4)--(5,4)
                             --(5,2)--(4,2)--(4,3)--(2,3)}
      \fillarea{LightBlue}{(3,4)--(3,6)--(6,6)--(6,5)--(5,5)
                           --(5,4)--(3,4)}
      \fillarea{LightYellow}{(4,1)--(4,2)--(5,2)--(5,5)--(6,5)
                             --(6,1)--(4,1)}
    \end{puzzlebackground}
  \end{chaossudoku}
\end{center}
```

3.5.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.1cm] specifies the width of the box the title is set in.

bgcolor [] sets the background color of the grid.

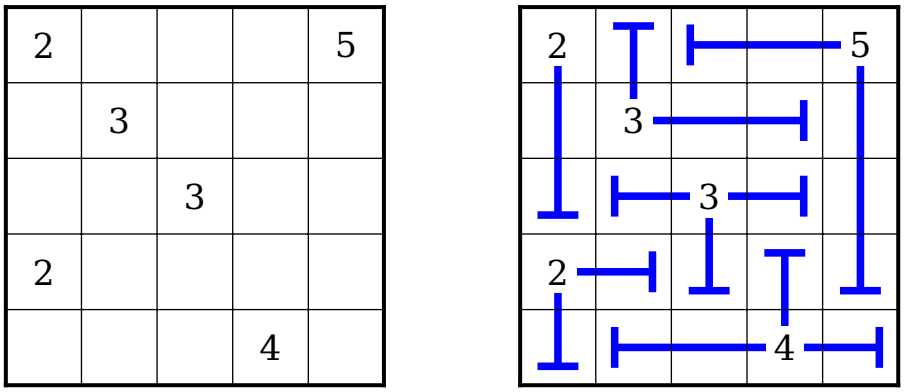
counterstyle [none] defines the counter style. Predefined styles: none, left, right

`cvoffset [-23pt]` sets the vertical offset of the counters in the margin.

3.6 Four Winds

Fill all cells with light rays. These may not intersect. Cells with numbers represent the lighting system that lits horizontally and vertically. The number indicates how many cells are illuminated. Cells with numbers do not count. No cell must remain empty.

3.6.1 Example




```
\begin{fourwinds}
  \framepuzzle
  \fourwindscell{1}{2}{2}{}
  \fourwindscell{1}{5}{2}{}
  \fourwindscell{2}{4}{3}{}
  \fourwindscell{3}{3}{3}{}
  \fourwindscell{4}{1}{4}{}
  \fourwindscell{5}{5}{5}{}
\end{fourwinds}
\hspace{1.5cm}
\begin{fourwinds}
  \framepuzzle
  \fourwindscell{1}{2}{2}{2/1,6/1}
  \fourwindscell{1}{5}{2}{2/2}
  \fourwindscell{2}{4}{3}{8/1,6/2}
  \fourwindscell{3}{3}{3}{4/1,6/1,2/1}
  \fourwindscell{4}{1}{4}{4/2,6/1,8/1}
  \fourwindscell{5}{5}{5}{4/2,2/3}
\end{fourwinds}
```

3.6.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.1cm] specifies the width of the box the title is set in.

color [blue] sets the color of the lines.

bgcolor [] sets the background color of the grid.

counterstyle [none] defines the counter style. Predefined styles: none, left, right

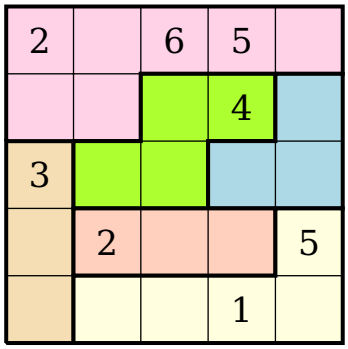
cvoffset [-23pt] sets the vertical offset of the counters in the margin.

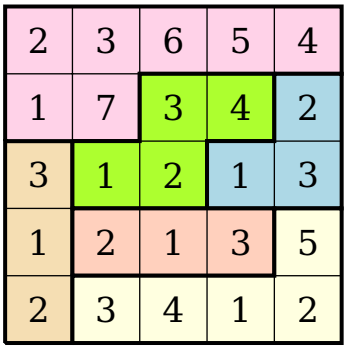
3.7 Hakyuu

Fill the cells of an area with numbers from 1 to SIZE of the area. If there are two cells with the same number N in a row or a column, there must be at least N cells between those two cells.

3.7.1 Example

```
\begin{center}
  \begin{hakyuu}
    \hakyuucell{1}{5}{2}
    \hakyuucell{3}{5}{6}
    \hakyuucell{4}{5}{5}
    \hakyuucell{4}{4}{4}
    \hakyuucell{1}{3}{3}
    \hakyuucell{2}{2}{2}
    \hakyuucell{5}{2}{5}
    \hakyuucell{4}{1}{1}
    \begin{puzzlebackground}
```






```
    \fillarea{Wheat}{(1,1)--(1,4)--(2,4)--(2,1)--(1,1)}
    \fillarea{HotPink!30}{(1,4)--(1,6)--(6,6)--(6,5)--(3,5)
                          --(3,4)--(1,4)}
    \fillarea{GreenYellow}{(2,4)--(3,4)--(3,5)--(5,5)--(5,4)
                           --(4,4)--(4,3)--(2,3)--(2,4)}
    \fillarea{LightBlue}{(5,5)--(6,5)--(6,3)--(4,3)--(4,4)
                         --(5,4)--(5,5)}
    \fillarea{LightSalmon!50}{(2,2)--(2,3)--(5,3)--(5,2)
                              --(2,2)}
    \fillarea{LightYellow}{(2,1)--(2,2)--(5,2)--(5,3)--(6,3)
                           --(6,1)--(2,1)}
  \end{puzzlebackground}
\end{hakyuu}
\hspace{1.5cm}
\begin{hakyuu}
  \setrow{5}{2,3,6,5,4}
  \setrow{4}{1,7,3,4,2}
  \setrow{3}{3,1,2,1,3}
  \setrow{2}{1,2,1,3,5}
  \setrow{1}{2,3,4,1,2}
  \begin{puzzlebackground}
    \fillarea{Wheat}{(1,1)--(1,4)--(2,4)--(2,1)--(1,1)}
    \fillarea{HotPink!30}{(1,4)--(1,6)--(6,6)--(6,5)--(3,5)
                          --(3,4)--(1,4)}
    \fillarea{GreenYellow}{(2,4)--(3,4)--(3,5)--(5,5)--(5,4)
                           --(4,4)--(4,3)--(2,3)--(2,4)}
    \fillarea{LightBlue}{(5,5)--(6,5)--(6,3)--(4,3)--(4,4)
                         --(5,4)--(5,5)}
    \fillarea{LightSalmon!50}{(2,2)--(2,3)--(5,3)--(5,2)
                              --(2,2)}
    \fillarea{LightYellow}{(2,1)--(2,2)--(5,2)--(5,3)--(6,3)
                           --(6,1)--(2,1)}
  \end{puzzlebackground}
\end{hakyuu}
```

```
\end{center}
```

3.7.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.1cm] specifies the width of the box the title is set in.

bgcolor [] sets the background color of the grid.

counterstyle [none] defines the counter style. Predefined styles: none, left, right

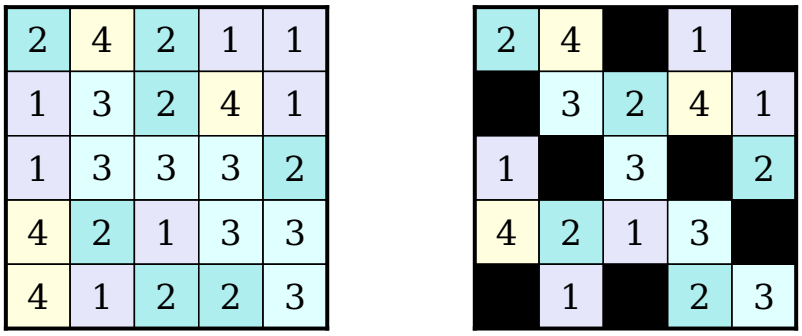
cvoffset [-23pt] sets the vertical offset of the counters in the margin.

3.8 Hitori

Black out some cells according to these specifications: In each row and each column a number may only occur once or can be completely blackened. The blackened cells can touch neither horizontal nor vertical. All non blackened cells must remain connected. Each number has its own color, which otherwise has no meaning.

3.8.1 Example

```
\begin{center}
  \begin{hitori}
    \framepuzzle
    \setcolorrow{5}{2,4,2,1,1}
    \setcolorrow{4}{1,3,2,4,1}
    \setcolorrow{3}{1,3,3,3,2}
    \setcolorrow{2}{4,2,1,3,3}
```




```
    \setcolorrow{1}{4,1,2,2,3}
  \end{hitori}
  \hspace{1.5cm}
  \begin{hitori}
    \framepuzzle
    \setcolorrow{5}{2,4,0,1,0}
    \setcolorrow{4}{0,3,2,4,1}
    \setcolorrow{3}{1,0,3,0,2}
    \setcolorrow{2}{4,2,1,3,0}
    \setcolorrow{1}{0,1,0,2,3}
  \end{hitori}
\end{center}
```

3.8.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.1cm] specifies the width of the box the title is set in.

bgcolor [] sets the background color of the grid.

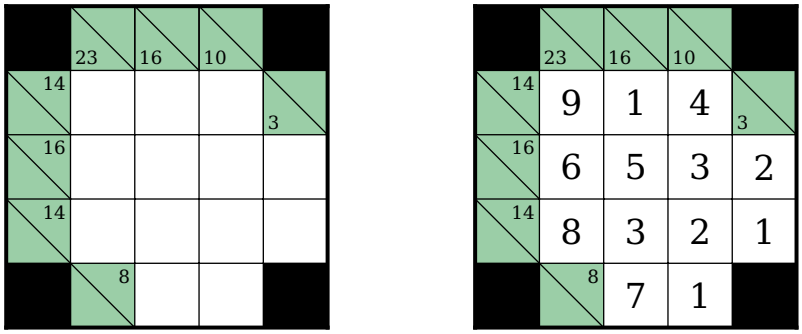
`counterstyle` [`none`] defines the counter style. Predefined styles: none, left, right

`cvoffset` [`-23pt`] sets the vertical offset of the counters in the margin.

3.9 Kakuro

Enter numbers from 1 to 9 in any order into the blank cells. Here, the given horizontal and vertical sums should result. The zero does not occur. Within a summation, no number can be repeated.

3.9.1 Example




```
\definecolor{kakuro}{RGB}{155,206,167}
\kakurosetup{color=kakuro}
\begin{center}
  \begin{kakuro}
    \framepuzzle
    \kakurorow{5}{\Black,\KKR{23}{},\KKR{16}{},\KKR{10}{},\Black}
    \kakurorow{4}{\KKR{}{14},9,1,4,\KKR{3}{}}
    \kakurorow{3}{\KKR{}{16},6,5,3,2}
    \kakurorow{2}{\KKR{}{14},8,3,2,1}
    \kakurorow{1}{\Black,\KKR{}{8},7,1,\Black}
  \end{kakuro}
  \hspace{1.5cm}
  \begin{kakuro}[solution]
    \framepuzzle
    \kakurorow{5}{\Black,\KKR{23}{},\KKR{16}{},\KKR{10}{},\Black}
    \kakurorow{4}{\KKR{}{14},9,1,4,\KKR{3}{}}
    \kakurorow{3}{\KKR{}{16},6,5,3,2}
    \kakurorow{2}{\KKR{}{14},8,3,2,1}
    \kakurorow{1}{\Black,\KKR{}{8},7,1,\Black}
  \end{kakuro}
```

```
\end{center}
```

3.9.2 Options

`rows` [5] defines the number of rows in the grid.

`columns` [5] specifies the number of columns in the grid

`width` [5.1cm] sets the width of the minipage, in which the grid is typeset.

`scale` [1] scales the size of the grid in the minipage.

`fontsize` [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

`title` [] sets the title of a puzzle.

`titleindent` [0cm] defines the indent of the title.

`titlewidth` [5.1cm] specifies the width of the box the title is set in.

`color` [green] specifies the color of the kakuro cells.

`bgcolor` [] sets the background color of the grid.

`counterstyle` [none] defines the counter style. Predefined styles: none, left, right

`cvoffset` [-23pt] sets the vertical offset of the counters in the margin.

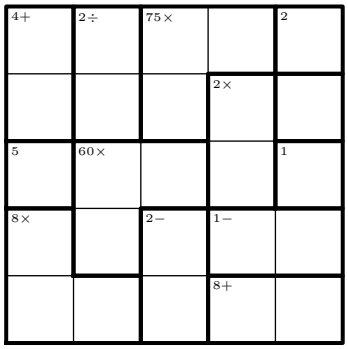
`solution` [false] You can use the solution also for the puzzle, as the numbers in the cells are only typeset with option `solution=true`.

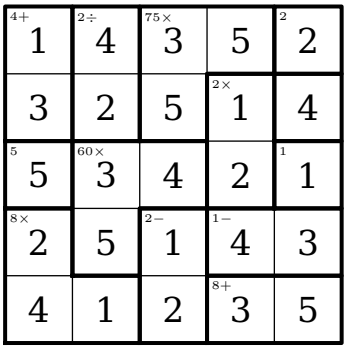
3.10 Kendoku

Fill the cells with the numbers from 1 to SIZE of the puzzle. In the top left corner of a framed area, you will find the result of the specified arithmetic function, which is applied on the entered numbers. The numbers may occur only once in each row and column. The numbers of an area may not necessarily be different when they are in different rows or columns.

3.10.1 Example

```
\begin{center}
  \begin{kendoku}
```






```
    \framearea{black}{\tikzpath{1}{1}{8,8,6,2,6,2,4,4}}
    \framearea{black}{\tikzpath{1}{3}{8,6,2,4}}
    \framearea{black}{\tikzpath{1}{4}{8,8,6,2,2,4}}
    \framearea{black}{\tikzpath{2}{2}{8,8,6,6,2,4,2,4}}
    \framearea{black}{\tikzpath{2}{4}{8,8,6,2,2,4}}
    \framearea{black}{\tikzpath{3}{1}{8,8,6,2,2,4}}
    \framearea{black}{\tikzpath{3}{4}{8,8,6,6,2,4,2,4}}
    \framearea{black}{\tikzpath{4}{1}{8,6,6,2,4,4}}
    \framearea{black}{\tikzpath{4}{2}{8,6,6,2,4,4}}
    \framearea{black}{\tikzpath{4}{3}{8,8,6,2,2,4}}
    \framearea{black}{\tikzpath{5}{3}{8,6,2,4}}
    \framearea{black}{\tikzpath{5}{4}{8,6,2,4}}
    \framearea{black}{\tikzpath{5}{5}{8,6,2,4}}
    \setrule{1}{2}{8\times}
    \setrule{1}{3}{5}
    \setrule{1}{5}{4+}
    \setrule{2}{3}{60\times}
    \setrule{2}{5}{2\div}
    \setrule{3}{2}{2-}
    \setrule{3}{5}{75\times}
    \setrule{4}{1}{8+}
    \setrule{4}{2}{1-}
    \setrule{4}{4}{2\times}
    \setrule{5}{3}{1}
    \setrule{5}{5}{2}
  \end{kendoku}
  \hspace{1.5cm}
  \begin{kendoku}
    \framearea{black}{\tikzpath{1}{1}{8,8,6,2,6,2,4,4}}
    \framearea{black}{\tikzpath{1}{3}{8,6,2,4}}
    \framearea{black}{\tikzpath{1}{4}{8,8,6,2,2,4}}
    \framearea{black}{\tikzpath{2}{2}{8,8,6,6,2,4,2,4}}
    \framearea{black}{\tikzpath{2}{4}{8,8,6,2,2,4}}
    \framearea{black}{\tikzpath{3}{1}{8,8,6,2,2,4}}
```

```
    \framearea{black}{\tikzpath{3}{4}{8,8,6,6,2,4,2,4}}
    \framearea{black}{\tikzpath{4}{1}{8,6,6,2,4,4}}
    \framearea{black}{\tikzpath{4}{2}{8,6,6,2,4,4}}
    \framearea{black}{\tikzpath{4}{3}{8,8,6,2,2,4}}
    \framearea{black}{\tikzpath{5}{3}{8,6,2,4}}
    \framearea{black}{\tikzpath{5}{4}{8,6,2,4}}
    \framearea{black}{\tikzpath{5}{5}{8,6,2,4}}
    \setrule{1}{2}{8\times}
    \setrule{1}{3}{5}
    \setrule{1}{5}{4+}
    \setrule{2}{3}{60\times}
    \setrule{2}{5}{2\div}
    \setrule{3}{2}{2-}
    \setrule{3}{5}{75\times}
    \setrule{4}{1}{8+}
    \setrule{4}{2}{1-}
    \setrule{4}{4}{2\times}
    \setrule{5}{3}{1}
    \setrule{5}{5}{2}
    \setrow{5}{1,4,3,5,2}
    \setrow{4}{3,2,5,1,4}
    \setrow{3}{5,3,4,2,1}
    \setrow{2}{2,5,1,4,3}
    \setrow{1}{4,1,2,3,5}
  \end{kendoku}
\end{center}
```

3.10.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.1cm] specifies the width of the box the title is set in.

bgcolor [] sets the background color of the grid.

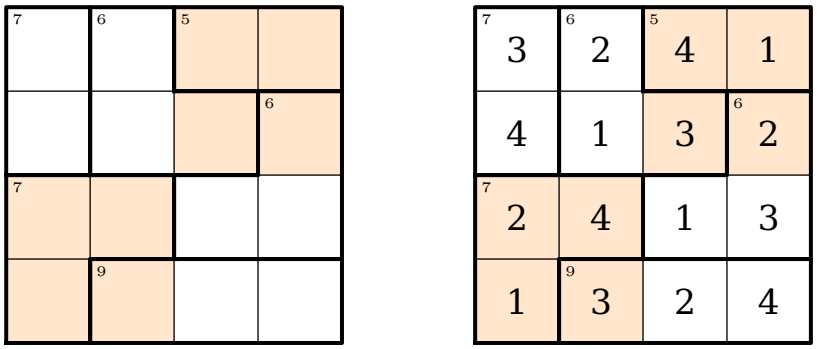
`counterstyle` [`none`] defines the counter style. Predefined styles: none, left, right

`cvoffset` [`-23pt`] sets the vertical offset of the counters in the margin.

3.11 Killer Sudoku

Fill the cells with the numbers from 1 to SIZE of the puzzle. The numbers may occur only once in each row, column and colored area if specified. In the top left corner of a framed area, you will find the sum of the entered numbers. The numbers of an area may not necessarily be different, when they are in different rows or columns. But they must be different, when additional colored areas are specified.

3.11.1 Example



```
\begin{killersudoku}
    \framearea{black}{\tikzpath{1}{1}{8,8,6,6,2,4,2,4}}
    \framearea{black}{\tikzpath{1}{3}{8,8,6,2,2,4}}
    \framearea{black}{\tikzpath{2}{1}{8,6,6,6,2,4,4,4}}
    \framearea{black}{\tikzpath{2}{3}{8,8,6,2,6,2,4,4}}
    \framearea{black}{\tikzpath{3}{2}{8,6,8,6,2,2,4,4}}
    \framearea{black}{\tikzpath{3}{4}{8,6,6,2,4,4}}
    \begin{puzzlebackground}
      \colorarea{orange!20}{\tikzpath{1}{1}{8,8,6,6,2,2,4,4}}
      \colorarea{orange!20}{\tikzpath{3}{3}{8,8,6,6,2,2,4,4}}
    \end{puzzlebackground}
    \setrule{1}{2}{7}
    \setrule{1}{4}{7}
    \setrule{2}{1}{9}
    \setrule{2}{4}{6}
    \setrule{3}{4}{5}
    \setrule{4}{3}{6}
```

```
\end{killersudoku}
\hspace{1.5cm}
\begin{killersudoku}
  \framearea{black}{\tikzpath{1}{1}{8,8,6,6,2,4,2,4}}
  \framearea{black}{\tikzpath{1}{3}{8,8,6,2,2,4}}
  \framearea{black}{\tikzpath{2}{1}{8,6,6,6,2,4,4,4}}
  \framearea{black}{\tikzpath{2}{3}{8,8,6,2,6,2,4,4}}
  \framearea{black}{\tikzpath{3}{2}{8,6,8,6,2,2,4,4}}
  \framearea{black}{\tikzpath{3}{4}{8,6,6,2,4,4}}
  \begin{puzzlebackground}
    \colorarea{orange!20}{\tikzpath{1}{1}{8,8,6,6,2,2,4,4}}
    \colorarea{orange!20}{\tikzpath{3}{3}{8,8,6,6,2,2,4,4}}
  \end{puzzlebackground}
  \setrule{1}{2}{7}
  \setrule{1}{4}{7}
  \setrule{2}{1}{9}
  \setrule{2}{4}{6}
  \setrule{3}{4}{5}
  \setrule{4}{3}{6}
  \setrow{4}{3,2,4,1}
  \setrow{3}{4,1,3,2}
  \setrow{2}{2,4,1,3}
  \setrow{1}{1,3,2,4}
\end{killersudoku}
```

3.11.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.1cm] specifies the width of the box the title is set in.

bgcolor [] sets the background color of the grid.

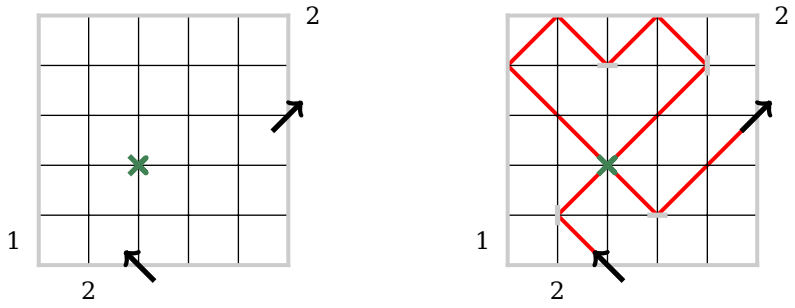
`counterstyle` [`none`] defines the counter style. Predefined styles: none, left, right

`cvoffset` [`-23pt`] sets the vertical offset of the counters in the margin.

3.12 Laser Beam

Draw a laser beam in each grid according to the following guidelines. The beam has to enter or to leave the grid at the arrows. At each intersection, a mirror, on which the laser beam must reflect on one side, can be placed horizontally or vertically. The other side must not be touched by the beam. All locations where the laser crosses are given. The numbers to the left and above the grid indicate how many cells are traversed by the beam in the corresponding row or column. The numbers to the right and below reveal, how many mirrors are found in the intersection of the corresponding row or column.

3.12.1 Example




```
\begin{center}
  \begin{laserbeam}
    \laserV{1}
    \laserH{{}}
    \mirrorH{{},2}
    \mirrorV{{},{},{},{},{},2}
    \framepuzzle[LP@c@mirror]
    \placearrow{3}{1}{LeftUp}
    \placearrow{6}{4}{RightUp}
    \placecross{3}{3}
  \end{laserbeam}
  \hspace{1cm}
  \begin{laserbeam}
    \laserV{1}
    \laserH{{}}
```

```
    \mirrorH{{},2}
    \mirrorV{{},{},{},{},{},2}
    \framepuzzle[LP@c@mirror]
    \placearrow{3}{1}{LeftUp}
    \placearrow{6}{4}{RightUp}
    \placecross{3}{3}
    \placemirror{2}{2}{V}
    \placemirror{4}{2}{H}
    \placemirror{5}{5}{V}
    \placemirror{3}{5}{H}
    \begin{puzzlebackground}
      \laser{\tikzpath{3}{1}{7,9,9,9,7,1,7,1,3,3,3,9,9}}
    \end{puzzlebackground}
  \end{laserbeam}
\end{center}
```

3.12.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [6.5cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [6.5cm] specifies the width of the box the title is set in.

bgcolor [] sets the background color of the grid.

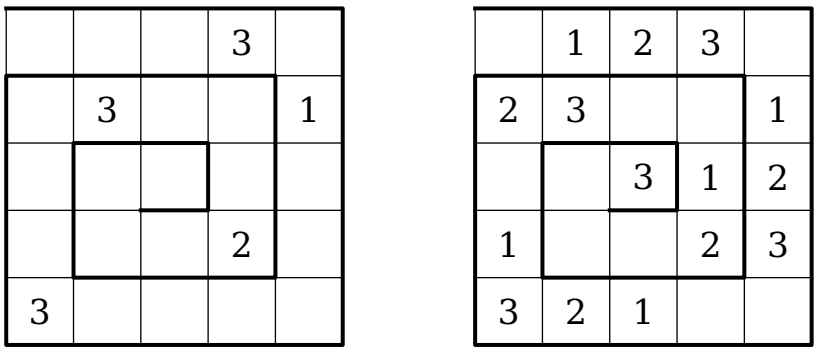
counterstyle [none] defines the counter style. Predefined styles: none, left, right

cvoffset [-38pt] sets the vertical offset of the counters in the margin.

3.13 Magic Labyrinth

Enter the numbers 1 to N into the grid. Each number can appear only once in each column and row. Following the labyrinth from the outside inwards, then the given number sequence must be repeated continuously.

3.13.1 Example




```
\begin{magiclabyrinth}
  \mlline{\xtikzpath{1}{6}{6/5,2/5,4/5,8/4,6/4,2/3,4/3,8/2,6/2,
                         2/1,4/1}}
  \setcells{1/1,2/4,4/5}{3}
  \magiclabyrinthcell{4}{2}{2}
  \magiclabyrinthcell{5}{4}{1}
\end{magiclabyrinth}
\hspace{1.5cm}
\begin{magiclabyrinth}
  \mlline{\xtikzpath{1}{6}{6/5,2/5,4/5,8/4,6/4,2/3,4/3,8/2,6/2,
                         2/1,4/1}}
  \setrow{5}{{},1,2,3}
  \setrow{4}{2,3,{},{},1}
  \setrow{3}{{},{},3,1,2}
  \setrow{2}{1,{},{},2,3}
  \setrow{1}{3,2,1}
\end{magiclabyrinth}
```

3.13.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.1cm] specifies the width of the box the title is set in.

bgcolor [] sets the background color of the grid.

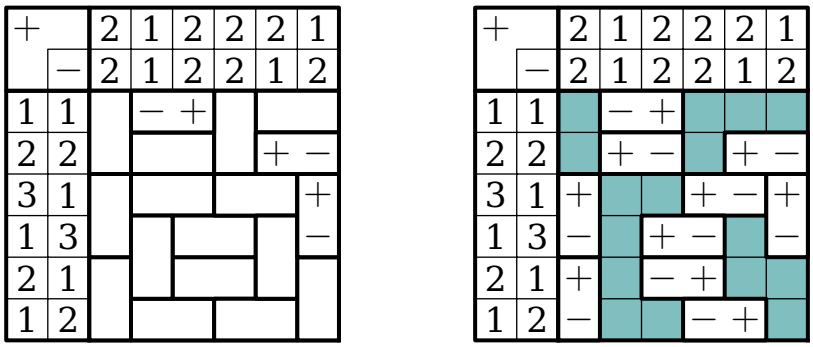
counterstyle [none] defines the counter style. Predefined styles: none, left, right

cvoffset [-23pt] sets the vertical offset of the counters in the margin.

3.14 Magnets

Draw magnetic and neutral plates into the grid. The magnetic plates have a positive and a negative pole, neutral plates do not. The same poles must not touch neither horizontal nor vertical. Neutral plates may touch. The numbers to the left and above the grid indicate how many plus or minus poles are present in the respective column or row.

3.14.1 Example



```
\magnetssetup{bgcolor=Teal!50}
\begin{magnets}
  \minusH{2,1,2,2,1,2}
  \minusV{2,1,3,1,2,1}
  \plusH{2,1,2,2,2,1}
  \plusV{1,2,1,3,2,1}
  \magnetsH{2/1,2/4,2/5,2/6,3/2,3/3,4/1,4/4,5/5,5/6}
  \magnetsV{1/1,1/3,1/5,2/2,4/5,5/2,6/1,6/3}
  \MPH{2/6}
  \PMH{5/5}
  \MPV{6/3}
```

```
\end{magnets}
\hspace{1.5cm}
\begin{magnets}
  \minusH{2,1,2,2,1,2}
  \minusV{2,1,3,1,2,1}
  \plusH{2,1,2,2,2,1}
  \plusV{1,2,1,3,2,1}
  \MPH{2/6,3/2,4/1}
  \PMH{2/5,5/5,4/4,3/3}
  \MPV{1/1,1/3,6/3}
\end{magnets}
```

3.14.2 Options

rows [6] defines the number of rows in the grid.

columns [6] specifies the number of columns in the grid

width [8.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [8.1cm] specifies the width of the box the title is set in.

bgcolor [] sets the background color of the grid for indicating the neutral areas.

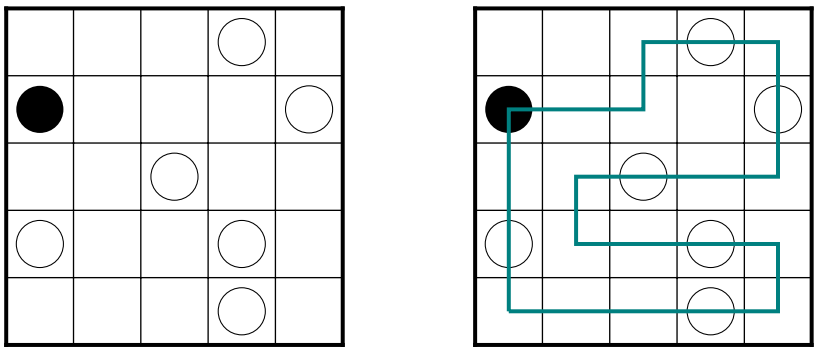
counterstyle [none] defines the counter style. Predefined styles: none, left, right

cvoffset [-23pt] sets the vertical offset of the counters in the margin.

3.15 Masyu

Draw a line into the grid. The line can only run horizontally and vertically and must pass through all the circles. In cells with a black circle it have to turn in a 90 degree angle and go straight on for at least another cell. The line must go straight through empty circles, but turn left or right in at least one of the neighboring cells. There is no need to go through all cells.

3.15.1 Example



```
\masyusetup{color=Teal}
\begin{masyu}
  \framepuzzle
  \setcells{1/2,3/3,4/1,4/2,4/5,5/4}{\MasyuW}
  \masyucell{1}{4}{\MasyuB}
\end{masyu}
\hspace{1.5cm}
\begin{masyu}
  \framepuzzle
  \setcells{1/2,3/3,4/1,4/2,4/5,5/4}{\MasyuW}
  \masyucell{1}{4}{\MasyuB}
  \masyuline{\xtikzpath{1}{1}{8/3,6/2,8/1,6/2,2/2,4/3,2/1,6/3,
                           2/1,4/4}}
\end{masyu}
```

3.15.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.1cm] specifies the width of the box the title is set in.

color [green] sets the color of the line.

bgcolor [] sets the background color of the grid.

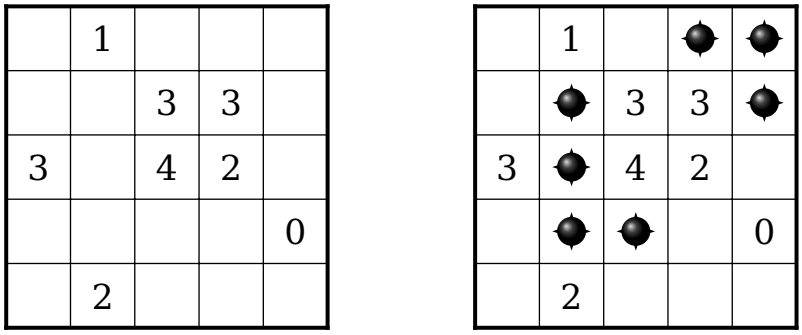
counterstyle [none] defines the counter style. Predefined styles: none, left, right

cvoffset [-23pt] sets the vertical offset of the counters in the margin.

3.16 Minesweeper

Draw a mine in some cells of the grid. The number in a cell indicates how many of the eight neighboring cells contain a mine. A numbered cell does not contain a mine.

3.16.1 Example




```
\begin{center}
  \begin{minesweeper}
    \framepuzzle
    \setrow{5}{{},1}
    \setrow{4}{{},{},3,3}
    \setrow{3}{3,{},4,2}
    \setrow{2}{{},{},{},{},0}
    \setrow{1}{{},2}
  \end{minesweeper}
  \hspace{1.5cm}
  \begin{minesweeper}
    \framepuzzle
    \setrow{5}{{},1,{},\Mine,\Mine}
    \setrow{4}{{},\Mine,3,3,\Mine}
    \setrow{3}{3,\Mine,4,2}
```

```
    \setrow{2}{{},\Mine,\Mine,{},0}
    \setrow{1}{{},2}
  \end{minesweeper}
\end{center}
```

3.16.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.1cm] specifies the width of the box the title is set in.

bgcolor [] sets the background color of the grid.

counterstyle [none] defines the counter style. Predefined styles: none, left, right

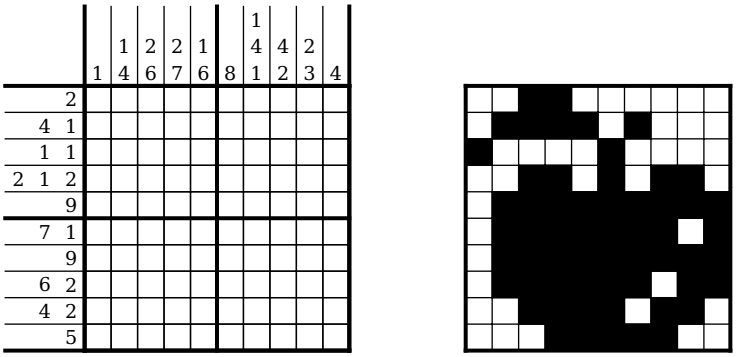
cvoffset [-23pt] sets the vertical offset of the counters in the margin.

3.17 Nonogram

Black out some cells of the grid. The black squares form stripes. The number, order, and length is defined by the number sequences on the top and left edge of the grid. Each number represents the length of the strip of black squares in the corresponding row or column. Two stripes are separated by at least one white square.

3.17.1 Example

```
\begin{center}
  \begin{nonogram}[rows=10,columns=10,scale=0.35,width=4.8cm,
                   fontsize=footnotesize,helplines=5,
                   extracells=3]
```



```
    \nonogramV{{5},{2,4},{2,6},{9},{1,7},{9},{2,1,2},{1,1},
               {1,4},{2}}
    \nonogramH{{1},{4,1},{6,2},{7,2},{6,1},{8},{1,4,1},{2,4},
               {3,2},{4}}
  \end{nonogram}
  \hspace{1cm}
  \begin{nonogram}[rows=10,columns=10,scale=0.35,width=3.8cm,
                   solution=true,extracells=3]
    \nonogramrow{10}{3/2}
    \nonogramrow{9}{2/4,7/1}
    \nonogramrow{8}{1/1,6/1}
    \nonogramrow{7}{3/2,6/1,8/2}
    \nonogramrow{6}{2/9}
    \nonogramrow{5}{2/7,10/1}
    \nonogramrow{4}{2/9}
    \nonogramrow{3}{2/6,9/2}
    \nonogramrow{2}{3/4,8/2}
    \nonogramrow{1}{4/5}
    \puzzlestrut
  \end{nonogram}
\end{center}
```

3.17.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid.

Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

`title` `[]` sets the title of a puzzle.

`titleindent` `[0cm]` defines the indent of the title.

`titlewidth` `[5.1cm]` specifies the width of the box the title is set in.

`color` `[black]` sets the color of the lines.

`bgcolor` `[]` sets the background color of the grid.

`counterstyle` `[none]` defines the counter style. Predefined styles: none, left, right

`cvoffset` `[-23pt]` sets the vertical offset of the counters in the margin.

`extracells` `[5]` sets the number of extra cells for the grid.

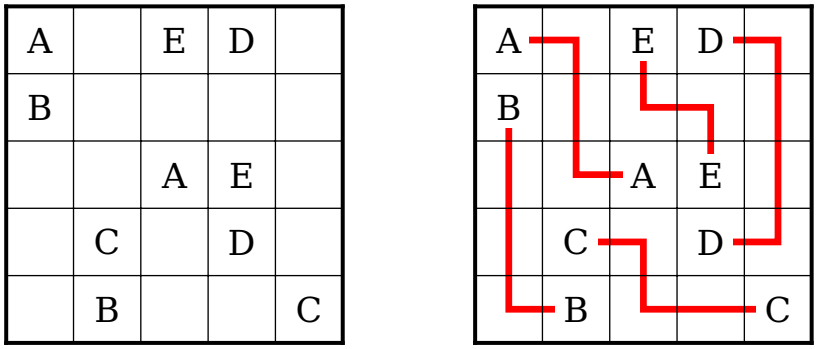
`helplines` `[5]` sets the space between thicker help lines. You can switch off help lines with `0`!

`solution` `[false]` will switch off extra cells if set to `true`.

3.18 Number Link

Link all the same numbers or letters in each case by a line. The lines can run horizontally, vertically and in 90 degree angles. Each cell must be traversed by exactly one line. The lines must not intersect.

3.18.1 Example




```
\begin{numberlink}
  \framepuzzle
```

```
  \numberlinkcell{2}{1}{B}
  \numberlinkcell{5}{1}{C}
  \numberlinkcell{2}{2}{C}
  \numberlinkcell{4}{2}{D}
  \numberlinkcell{3}{3}{A}
  \numberlinkcell{4}{3}{E}
  \numberlinkcell{1}{4}{B}
  \numberlinkcell{1}{5}{A}
  \numberlinkcell{3}{5}{E}
  \numberlinkcell{4}{5}{D}
\end{numberlink}
\hspace{1.5cm}
\begin{numberlink}
  \framepuzzle
  \numberlinkcell{2}{1}{B}
  \numberlinkcell{5}{1}{C}
  \numberlinkcell{2}{2}{C}
  \numberlinkcell{4}{2}{D}
  \numberlinkcell{3}{3}{A}
  \numberlinkcell{4}{3}{E}
  \numberlinkcell{1}{4}{B}
  \numberlinkcell{1}{5}{A}
  \numberlinkcell{3}{5}{E}
  \numberlinkcell{4}{5}{D}
  \link{\tikzpath{4}{3}{8,4,8}}
  \link{\tikzpath{2}{1}{4,8,8,8}}
  \link{\tikzpath{2}{2}{6,2,6,6}}
  \link{\tikzpath{1}{5}{6,2,2,6}}
  \link{\tikzpath{4}{2}{6,8,8,8,4}}
\end{numberlink}
```

3.18.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.1cm] specifies the width of the box the title is set in.

color [red] sets the color of the lines.

bgcolor [] sets the background color of the grid.

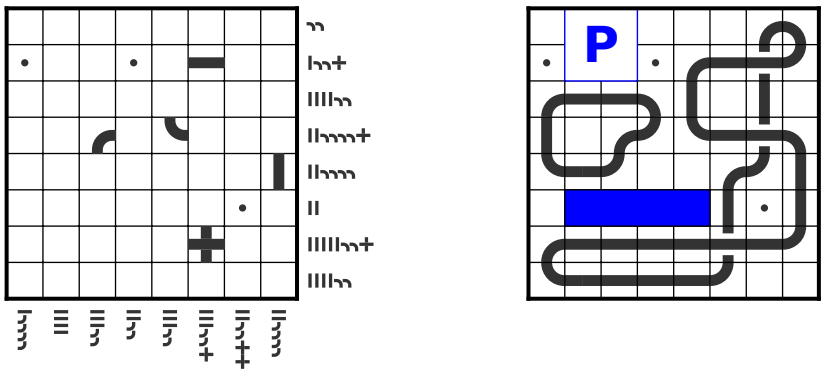
counterstyle [none] defines the counter style. Predefined styles: none, left, right

cvoffset [-23pt] sets the vertical offset of the counters in the margin.

3.19 Resuko

Complete the given elements in the grid to two race tracks (a race track and a much shorter test track) with pitlane and parking lot. The elements below and to the left of the grid indicate how many straights, curves and intersections are located in the respective columns and rows. The pit lane is always located next to four straights of the circuit on a free area of 1x4 cells. At the end, the parking lot is located on the only free area of 2x2 cells. Both can not be built on a gravel trap.

3.19.1 Example




```
\resukosetup{rows=8,columns=8,width=5.8cm,fontsize=Huge,
             scale=.708}
\begin{resuko}[width=7.4cm]
  \resukocell{1}{7}{\Graveltrap}
  \resukocell{4}{7}{\Graveltrap}
  \resukocell{7}{3}{\Graveltrap}
  \resukocell{6}{2}{\Cross}
  \resukocell{8}{4}{\StraightV}
  \resukocell{6}{7}{\StraightH}
```

```
  \resukocell{3}{5}{\CurveBR}
  \resukocell{5}{5}{\CurveTR}
  \trackH{1/4/0,4/0/0,3/2/0,2/2/0,3/2/0,3/2/1,2/2/2,2/4/0}
  \trackV{4/2/0,5/2/1,2/0/0,2/4/0,2/4/1,4/2/0,1/2/1,0/2/0}
  \framepuzzle
\end{resuko}
\hspace{1.5cm}
\begin{resuko}
  \resukocell{1}{7}{\Graveltrap}
  \resukocell{4}{7}{\Graveltrap}
  \resukocell{7}{3}{\Graveltrap}
  \parkinglot{2}{7}
  \pitlane{2}{3}{H}
  \track{\tikzpath{2}{4}{6,8,6,8,4,4,4,2,2,6}}
  \track{\xtikzpath{2}{1}{6/4,8/3,6/1,8/4,6/1,2/1,4/3,2/2,6/3,
                          2/3,4/7,2/1,6/1}}
  \framepuzzle
\end{resuko}
```

3.19.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.1cm] specifies the width of the box the title is set in.

color [blue] sets the color of the pitlane and parking lot.

bgcolor [] sets the background color of the grid.

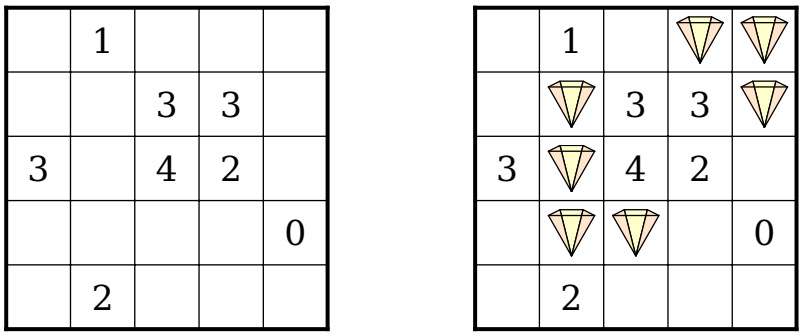
counterstyle [none] defines the counter style. Predefined styles: none, left, right

cvoffset [-23pt] sets the vertical offset of the counters in the margin.

3.20 Schatzsuche

It's a variant of Minesweeper, just with diamonds instead of mines. Draw a diamond in some cells of the grid. The number in a cell indicates how many of the eight neighboring cells contain a diamond. A numbered cell does not contain a diamond.

3.20.1 Example



```
\begin{center}
  \begin{schatzsuche}
    \framepuzzle
    \setrow{5}{{},1}
    \setrow{4}{{},{},3,3}
    \setrow{3}{3,{},4,2}
    \setrow{2}{{},{},{},{},0}
    \setrow{1}{{},2}
  \end{schatzsuche}
  \hspace{1.5cm}
  \begin{schatzsuche}
    \framepuzzle
    \setrow{5}{{},1,{},\Diamond,\Diamond}
    \setrow{4}{{},\Diamond,3,3,\Diamond}
    \setrow{3}{3,\Diamond,4,2}
    \setrow{2}{{},\Diamond,\Diamond,{},0}
    \setrow{1}{{},2}
  \end{schatzsuche}
\end{center}
```

3.20.2 Options

`rows` [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.1cm] specifies the width of the box the title is set in.

bgcolor [] sets the background color of the grid.

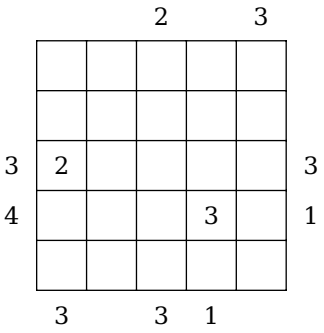
counterstyle [none] defines the counter style. Predefined styles: none, left, right

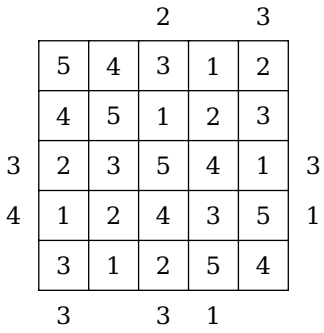
cvoffset [-23pt] sets the vertical offset of the counters in the margin.

3.21 Skyline

There are skyscrapers located in each cell. Try to find out the height of the skyscraper in the respective cell. There are heights of 1 to MAX in every row, every column, and in each of the two diagonals if indicated. Some cells may be empty (parks). The numbers around the grid indicate how many buildings you can see from this position when you look at the skyscraper lineup. Bear in mind that only those skyscrapers are visible which are higher than the ones in front.

3.21.1 Example

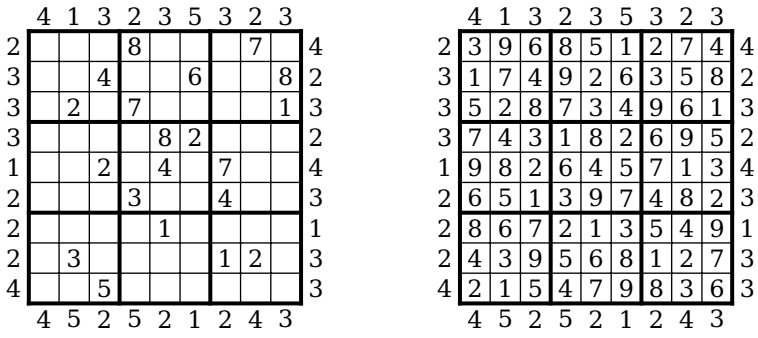




```
\begin{center}
  \begin{skyline}
    \skylineB{3,{},3,1,{}}
    \skylineL{{},4,3,{},{}}
    \skylineT{{},{},2,{},3}
    \skylineR{{},1,3,{},{}}
    \skylinecell{1}{3}{2}
    \skylinecell{4}{2}{3}
  \end{skyline}
  \hspace{1cm}
  \begin{skyline}
    \skylineB{3,{},3,1,{}}
    \skylineL{{},4,3,{},{}}
    \skylineT{{},{},2,{},3}
    \skylineR{{},1,3,{},{}}
    \setrow{5}{5,4,3,1,2}
    \setrow{4}{4,5,1,2,3}
    \setrow{3}{2,3,5,4,1}
    \setrow{2}{1,2,4,3,5}
    \setrow{1}{3,1,2,5,4}
  \end{skyline}
\end{center}
```

3.21.1.1 Variants

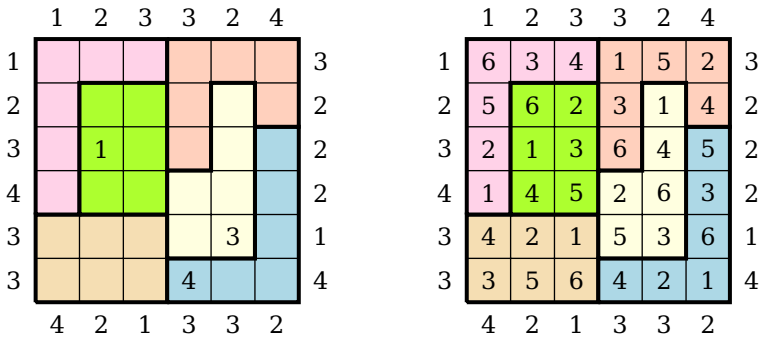
3.21.1.1.1 Skyline Sudoku




```
\begin{center}
  \begin{skyline}[sudoku,scale=.4]
    \skylineB{4,5,2,5,2,1,2,4,3}
    \skylineL{4,2,2,2,1,3,3,3,2}
    \skylineT{4,1,3,2,3,5,3,2,3}
    \skylineR{3,3,1,3,4,2,3,2,4}
```

```
    \setrow{9}{{},{},{},8,{},{},{},7}
    \setrow{8}{{},{},4,{},{},6,{},{},8}
    \setrow{7}{{},2,{},7,{},{},{},{},1}
    \setrow{6}{{},{},{},{},8,2}
    \setrow{5}{{},{},2,{},4,{},7}
    \setrow{4}{{},{},{},3,{},{},4}
    \setrow{3}{{},{},{},{},1}
    \setrow{2}{{},3,{},{},{},{},1,2}
    \setrow{1}{{},{},5}
  \end{skyline}
  \hspace{1cm}
  \begin{skyline}[sudoku,scale=.4]
    \skylineB{4,5,2,5,2,1,2,4,3}
    \skylineL{4,2,2,2,1,3,3,3,2}
    \skylineT{4,1,3,2,3,5,3,2,3}
    \skylineR{3,3,1,3,4,2,3,2,4}
    \setrow{9}{3,9,6,8,5,1,2,7,4}
    \setrow{8}{1,7,4,9,2,6,3,5,8}
    \setrow{7}{5,2,8,7,3,4,9,6,1}
    \setrow{6}{7,4,3,1,8,2,6,9,5}
    \setrow{5}{9,8,2,6,4,5,7,1,3}
    \setrow{4}{6,5,1,3,9,7,4,8,2}
    \setrow{3}{8,6,7,2,1,3,5,4,9}
    \setrow{2}{4,3,9,5,6,8,1,2,7}
    \setrow{1}{2,1,5,4,7,9,8,3,6}
  \end{skyline}
\end{center}
```

3.21.1.1.2 Skyline Sudoku (N*N)




```
\begin{center}
  \begin{skyline}[rows=6,columns=6,scale=.58]
    \skylineB{4,2,1,3,3,2}
    \skylineL{3,3,4,3,2,1}
```

```
    \skylineT{1,2,3,3,2,4}
    \skylineR{4,1,2,2,2,3}
    \skylinecell{2}{4}{1}
    \skylinecell{4}{1}{4}
    \skylinecell{5}{2}{3}
    \begin{puzzlebackground}
      \fillarea{Wheat}{(1,1)--(1,3)--(4,3)--(4,1)--(1,1)}
      \fillarea{HotPink!30}{(1,3)--(1,7)--(4,7)--(4,6)--(2,6)
                            --(2,3)--(1,3)}
      \fillarea{GreenYellow}{(2,3)--(2,6)--(4,6)--(4,3)--(2,3)}
      \fillarea{LightBlue}{(4,1)--(7,1)--(7,5)--(6,5)--(6,2)
                           --(4,2)--(4,1)}
      \fillarea{LightSalmon!50}{(4,7)--(4,4)--(5,4)--(5,6)--(6,6)
                                --(6,5)--(7,5)--(7,7)--(4,7)}
      \fillarea{LightYellow}{(4,2)--(4,4)--(5,4)--(5,6)--(6,6)
                             --(6,2)--(4,2)}
    \end{puzzlebackground}
  \end{skyline}
  \hspace{1cm}
  \begin{skyline}[rows=6,columns=6,scale=.58]
    \skylineB{4,2,1,3,3,2}
    \skylineL{3,3,4,3,2,1}
    \skylineT{1,2,3,3,2,4}
    \skylineR{4,1,2,2,2,3}
    \setrow{6}{6,3,4,1,5,2}
    \setrow{5}{5,6,2,3,1,4}
    \setrow{4}{2,1,3,6,4,5}
    \setrow{3}{1,4,5,2,6,3}
    \setrow{2}{4,2,1,5,3,6}
    \setrow{1}{3,5,6,4,2,1}
    \begin{puzzlebackground}
      \fillarea{Wheat}{(1,1)--(1,3)--(4,3)--(4,1)--(1,1)}
      \fillarea{HotPink!30}{(1,3)--(1,7)--(4,7)--(4,6)--(2,6)
                            --(2,3)--(1,3)}
      \fillarea{GreenYellow}{(2,3)--(2,6)--(4,6)--(4,3)--(2,3)}
      \fillarea{LightBlue}{(4,1)--(7,1)--(7,5)--(6,5)--(6,2)
                           --(4,2)--(4,1)}
      \fillarea{LightSalmon!50}{(4,7)--(4,4)--(5,4)--(5,6)--(6,6)
                                --(6,5)--(7,5)--(7,7)--(4,7)}
      \fillarea{LightYellow}{(4,2)--(4,4)--(5,4)--(5,6)--(6,6)
                             --(6,2)--(4,2)}
    \end{puzzlebackground}
  \end{skyline}
\end{center}
```

3.21.2 Options

`rows` [5] defines the number of rows in the grid.

`columns` [5] specifies the number of columns in the grid

`sudoku` [false] sets `rows` and `columns` to 9, in case of `true` is specified. Additionally the classic Sudoku grid is drawn.

`width` [6.7cm] sets the width of the minipage, in which the grid is typeset.

`scale` [1] scales the size of the grid in the minipage.

`fontsize` [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

`title` [] sets the title of a puzzle.

`titleindent` [0.75cm] defines the indent of the title.

`titlewidth` [5.85cm] specifies the width of the box the title is set in.

`bgcolor` [] sets the background color of the grid.

`counterstyle` [none] defines the counter style. Predefined styles: none, left, right

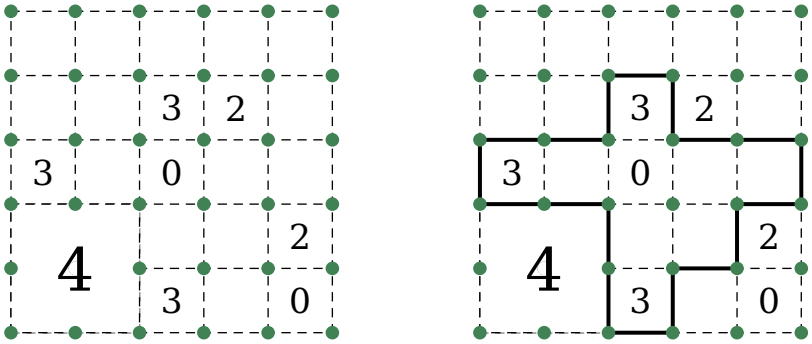
`cvoffset` [-38pt] sets the vertical offset of the counters in the margin.

3.22 Slitherlink

Draw a closed line into the grid. This line must be on the existing dashed lines, but do not have to go through all grid points. If numbers are present in the grid cells, they indicate how many sides of the cell are touched by the line. The line must not touch or cross itself.

3.22.1 Example

```
\begin{center}
  \begin{slitherlink}
    \setbigcell{1}{1}{4}
    \slitherlinkcell{1}{3}{3}
    \slitherlinkcell{3}{1}{3}
    \slitherlinkcell{3}{3}{0}
    \slitherlinkcell{3}{4}{3}
    \slitherlinkcell{4}{4}{2}
    \slitherlinkcell{5}{1}{0}
```




```
    \slitherlinkcell{5}{2}{2}
  \end{slitherlink}
  \hspace{1.5cm}
  \begin{slitherlink}
    \setbigcell{1}{1}{4}
    \slitherlinkcell{1}{3}{3}
    \slitherlinkcell{3}{1}{3}
    \slitherlinkcell{3}{3}{0}
    \slitherlinkcell{3}{4}{3}
    \slitherlinkcell{4}{4}{2}
    \slitherlinkcell{5}{1}{0}
    \slitherlinkcell{5}{2}{2}
    \framearea{black}{\tikzpath{3}{1}{8,8,4,4,8,6,6,8,6,2,
                                     6,6,2,4,2,4,2,4}}
  \end{slitherlink}
\end{center}
```

3.22.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.2cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

`titlewidth` [5.2cm] specifies the width of the box the title is set in.

`bgcolor` [] sets the background color of the grid.

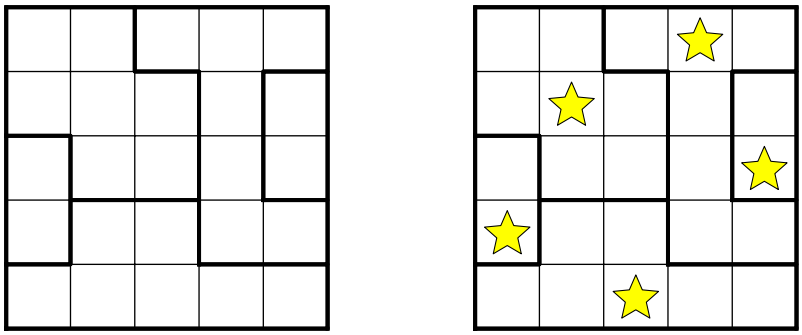
`counterstyle` [none] defines the counter style. Predefined styles: none, left, right

`cvoffset` [-23pt] sets the vertical offset of the counters in the margin.

3.23 Star Battle

Enter exactly one star in each row, each column and each area of the grid. Cells with stars must not touch each other orthogonally or diagonally.

3.23.1 Example



```
\begin{center}
  \begin{starbattle}
    \framepuzzle
    \framearea{black}{\tikzpath{1}{1}{8,6,8,6,6,2,6,6,2,4,4,4,
                                      4,4}}
    \framearea{black}{\tikzpath{1}{2}{8,8,6,2,2,4}}
    \framearea{black}{\tikzpath{1}{4}{8,8,6,6,2,6,2,2,4,4,8,4}}
    \framearea{black}{\tikzpath{4}{2}{8,8,8,4,8,6,6,6,2,4,2,2,6,
                                      2,4,4}}
    \framearea{black}{\tikzpath{5}{3}{8,8,6,2,2,4}}
  \end{starbattle}
  \hspace{1.5cm}
  \begin{starbattle}
    \framepuzzle
    \framearea{black}{\tikzpath{1}{1}{8,6,8,6,6,2,6,6,2,4,4,4,
                                      4,4}}
    \framearea{black}{\tikzpath{1}{2}{8,8,6,2,2,4}}
    \framearea{black}{\tikzpath{1}{4}{8,8,6,6,2,6,2,2,4,4,8,4}}
```

```
    \framearea{black}{\tikzpath{4}{2}{8,8,8,4,8,6,6,6,2,4,2,2,6,
                                      2,4,4}}
    \framearea{black}{\tikzpath{5}{3}{8,8,6,2,2,4}}
    \starbattlecell{1}{2}{\Star}
    \starbattlecell{2}{4}{\Star}
    \starbattlecell{3}{1}{\Star}
    \starbattlecell{4}{5}{\Star}
    \starbattlecell{5}{3}{\Star}
  \end{starbattle}
\end{center}
```

3.23.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.1cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.1cm] specifies the width of the box the title is set in.

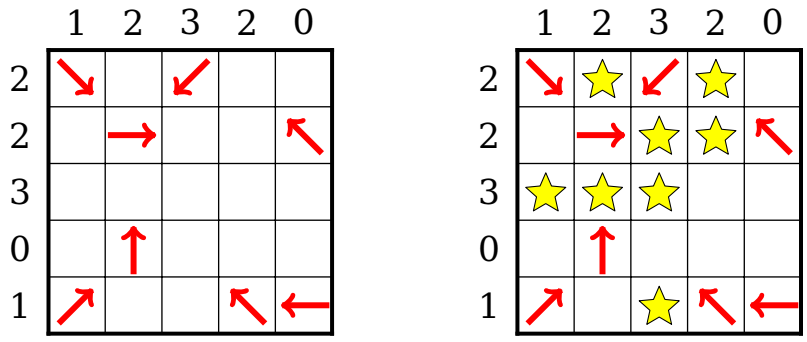
bgcolor [] sets the background color of the grid.

counterstyle [none] defines the counter style. Predefined styles: none, left, right

cvoffset [-23pt] sets the vertical offset of the counters in the margin.

3.24 Stars and Arrows

Enter a star in some empty cells of the grid. Each arrow points to at least one star and every star is referenced by at least one arrow. Arrows point to a whole row, column or diagonal, also through other stars and arrows. The numbers on the left and top of the grid indicate how many stars are located in the row or column.



3.24.1 Example

```
\begin{center}
  \begin{starsandarrows}
    \framepuzzle
    \starsH{1,2,3,2,0}
    \starsV{1,0,3,2,2}
    \setrow{5}{\RightDown,{},\LeftDown}
    \setrow{4}{{},\Right,{},{},\LeftUp}
    \setrow{2}{{},\Up,{},{},{}}
    \setrow{1}{\RightUp,{},{},\LeftUp,\Left}
  \end{starsandarrows}
  \hspace{1.5cm}
  \begin{starsandarrows}
    \framepuzzle
    \starsH{1,2,3,2,0}
    \starsV{1,0,3,2,2}
    \setrow{5}{\RightDown,\Star,\LeftDown,\Star}
    \setrow{4}{{},\Right,\Star,\Star,\LeftUp}
    \setrow{3}{\Star,\Star,\Star}
    \setrow{2}{{},\Up,{},{},{}}
    \setrow{1}{\RightUp,{},\Star,\LeftUp,\Left}
  \end{starsandarrows}
\end{center}
```

3.24.2 Options

`rows` [5] defines the number of rows in the grid.

`columns` [5] specifies the number of columns in the grid

`width` [5.9cm] sets the width of the minipage, in which the grid is typeset.

`scale` [1] scales the size of the grid in the minipage.

`fontsize` [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

`title` [] sets the title of a puzzle.

`titleindent` [0cm] defines the indent of the title.

`titlewidth` [5.9cm] specifies the width of the box the title is set in.

`bgcolor` [] sets the background color of the grid.

`counterstyle` [none] defines the counter style. Predefined styles: none, left, right

`cvoffset` [-23pt] sets the vertical offset of the counters in the margin.

3.25 Sudoku

Well, it's Sudoku – nothing to explain! Fill each row and column with numbers from 1 to 9.

3.25.1 Example

	2	6						
						1	7	
		3	1		6			
	6			5		8		3
		9	2	6	1	7		
5		4		8			6	
			8		4	3		
	4	8						
						9	4	

1	2	6	5	7	8	4	3	9
4	8	5	9	3	2	1	7	6
7	9	3	1	4	6	5	8	2
2	6	1	4	5	7	8	9	3
8	3	9	2	6	1	7	5	4
5	7	4	3	8	9	2	6	1
6	5	2	8	9	4	3	1	7
9	4	8	7	1	3	6	2	5
3	1	7	6	2	5	9	4	8

```
\begin{center}
  \begin{lpsudoku}
    \setrow{9}{{},2,6,{},{},{},{},{},{}}
    \setrow{8}{{},{},{},{},{},{},1,7,{}}
    \setrow{7}{{},{},3,1,{},6,{},{},{}}
    \setrow{6}{{},6,{},{},5,{},8,{},3}
    \setrow{5}{{},{},9,2,6,1,7,{},{}}
    \setrow{4}{5,{},4,{},8,{},{},6,{}}
```

```
  \setrow{3}{{},{},{},8,{},4,3,{},{}}
  \setrow{2}{{},4,8,{},{},{},{},{},{}}
  \setrow{1}{{},{},{},{},{},{},9,4,{}}
  \end{lpsudoku}
  \hspace{1.5cm}
  \begin{lpsudoku}
  \setrow{9}{1,2,6,5,7,8,4,3,9}
  \setrow{8}{4,8,5,9,3,2,1,7,6}
  \setrow{7}{7,9,3,1,4,6,5,8,2}
  \setrow{6}{2,6,1,4,5,7,8,9,3}
  \setrow{5}{8,3,9,2,6,1,7,5,4}
  \setrow{4}{5,7,4,3,8,9,2,6,1}
  \setrow{3}{6,5,2,8,9,4,3,1,7}
  \setrow{2}{9,4,8,7,1,3,6,2,5}
  \setrow{1}{3,1,7,6,2,5,9,4,8}
\end{lpsudoku}
```

3.25.2 Options

width [9.1cm] sets the width of the minipage, in which the grid is typeset. 9 cells of width 1cm plus a litle extra for lines.

scale [1] scales the size of the grid in the minipage. To get a width of 5cm you need to scale by $5/9$

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [9.1cm] specifies the width of the box the title is set in.

bgcolor [] sets the background color of the grid.

counterstyle [none] defines the counter style. Predefined styles: none, left, right

cvoffset [-23pt] sets the vertical offset of the counters in the margin.

3.25.3 Supporting bash scripts

3.25.3.1 createlpsudoku

The `createlpsudoku` [2] `bash` script can transform Sudoku format files into `lpsudoku` environments. It can process files in the so called one line 81 format[2] (option -e (default)) and in simple sudoku format (option -s)

Usage: `createlpsudoku [options] [-o output] -i input`

It expects an input file with the option `-i`. You can specify an output file with the option `-o`. Otherwise it writes to `stdout`. Furthermore, the following options are possible:

-w write Windows line endings (CR/LF) to file
-v prints version number
-h prints help

3.25.3.2 lpsmag

With the `lpsmag` [30] `bash` script you can half automatically produce a Sudoku magazine using the `lpsudoku` environment and the `createlpsudoku` bash script.

Usage: `lpsmag configfile`

The script needs an installed QQwing [32] and a config file for defining the magazine's contents:

```
page p1 easy
page p2 easy
startpuzzles
typesetpage p1
typesetpage p2
startsolutions
typesetsolpage p1 p2 last
```

This config file will be sourced into the `lpsmag bash` script and contains calls of `lpsmag` functions. Make sure, that the config file has UNIX line endings (LF). For a detailed documentation I refer to the following wiki [30] entry.

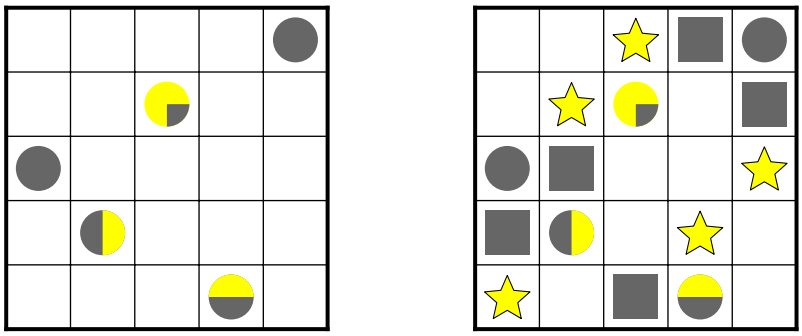
After running `lpsmag` you will find a `lpsmag.tex` in your working directory. Just run `pdflatex lpsmag.tex` twice and you finally get for example this `lpsmag.pdf`.

[2] processing of several sudokus in 81 format (one in each line) is possible

3.26 Sun and Moon

Enter exactly one star and one dark cloud in each row and each column of the grid, so that the planets are illuminated as specified. The stars shine horizontally or vertically arbitrarily far, but not through a planet or a dark cloud.

3.26.1 Example



```
\begin{center}
  \begin{sunandmoon}
    \framepuzzle
    \setrow{5}{{},{},{},{},\Moon}
    \setrow{4}{{},{},\MoonTL}
    \setrow{3}{\Moon}
    \setrow{2}{{},\MoonR}
    \setrow{1}{{},{},{},\MoonT}
  \end{sunandmoon}
  \hspace{1.5cm}
  \begin{sunandmoon}
    \framepuzzle
    \setrow{5}{{},{},\Star,\Cloud,\Moon}
    \setrow{4}{{},\Star,\MoonTL,{},\Cloud}
    \setrow{3}{\Moon,\Cloud,{},{},\Star}
    \setrow{2}{\Cloud,\MoonR,{},\Star}
    \setrow{1}{\Star,{},\Cloud,\MoonT}
  \end{sunandmoon}
\end{center}
```

3.26.2 Options

`rows` [5] defines the number of rows in the grid.

`columns` [5] specifies the number of columns in the grid

`width` [5.1cm] sets the width of the minipage, in which the grid is typeset.

`scale` [1] scales the size of the grid in the minipage.

`fontsize` [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

`title` [] sets the title of a puzzle.

`titleindent` [0cm] defines the indent of the title.

`titlewidth` [5.1cm] specifies the width of the box the title is set in.

`bgcolor` [] sets the background color of the grid.

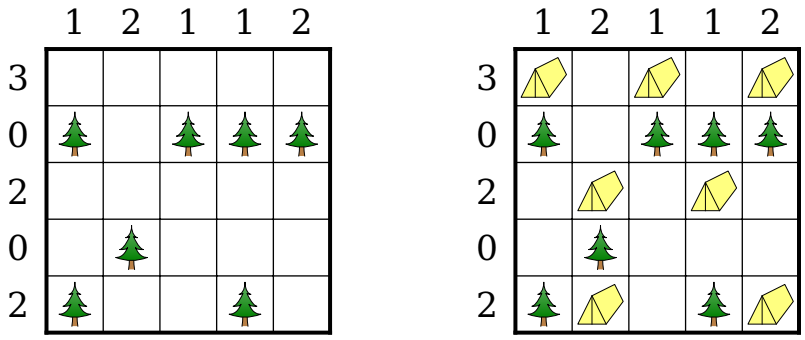
`counterstyle` [none] defines the counter style. Predefined styles: none, left, right

`cvoffset` [-23pt] sets the vertical offset of the counters in the margin.

3.27 Tents and Trees

Draw tents in the grid. Next to each tree, a tent must be entered in a horizontally or vertically adjacent cell, which is associated with this tree. The numbers next to the grid indicate the quantity of tents in each row or column. No tent can stand directly next to another one, not even diagonally.

3.27.1 Example




```
\begin{center}
  \begin{tentsandtrees}
    \framepuzzle
```

```
    \tentH{1,2,1,1,2}
    \tentV{2,0,2,0,3}
    \setrow{4}{\Tree,{},\Tree,\Tree,\Tree}
    \setrow{2}{{},\Tree}
    \setrow{1}{\Tree,{},{},\Tree}
  \end{tentsandtrees}
  \hspace{1.5cm}
  \begin{tentsandtrees}
    \framepuzzle
    \tentH{1,2,1,1,2}
    \tentV{2,0,2,0,3}
    \setrow{5}{\Tent,{},\Tent,{},\Tent,}
    \setrow{4}{\Tree,{},\Tree,\Tree,\Tree}
    \setrow{3}{{},\Tent,{},\Tent}
    \setrow{2}{{},\Tree}
    \setrow{1}{\Tree,\Tent,{},\Tree,\Tent}
  \end{tentsandtrees}
\end{center}
```

3.27.2 Options

rows [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.9cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.9cm] specifies the width of the box the title is set in.

bgcolor [] sets the background color of the grid.

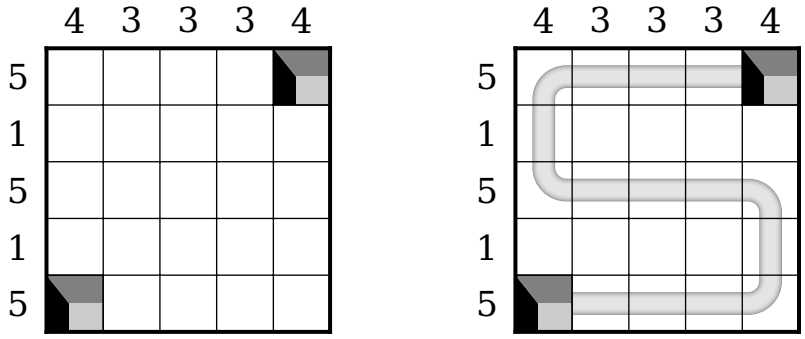
counterstyle [none] defines the counter style. Predefined styles: none, left, right

cvoffset [-23pt] sets the vertical offset of the counters in the margin.

3.28 Tunnel

Determine the course of the tube. Draw the only possible connection. from the beginning to the end. The numbers indicate how many tube segments (including portals) are present in the corresponding rows and columns. The tube is one cell wide, and does not cross or touch itself!

3.28.1 Example




```
\begin{center}
  \begin{tunnel}
    \framepuzzle
    \tunnelH{4,3,3,3,4}
    \tunnelV{5,1,5,1,5}
    \portal{1}{1}
    \portal{5}{5}
  \end{tunnel}
  \hspace{1.5cm}
  \begin{tunnel}
    \framepuzzle
    \tunnelH{4,3,3,3,4}
    \tunnelV{5,1,5,1,5}
    \portal{1}{1}
    \portal{5}{5}
    \tube{\tikzpath{1}{1}{6,6,6,6,8,8,4,4,4,4,8,8,6,6,6,6}}
  \end{tunnel}
\end{center}
```

3.28.2 Options

`rows` [5] defines the number of rows in the grid.

columns [5] specifies the number of columns in the grid

width [5.9cm] sets the width of the minipage, in which the grid is typeset.

scale [1] scales the size of the grid in the minipage.

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [5.9cm] specifies the width of the box the title is set in.

bgcolor [] sets the background color of the grid.

counterstyle [none] defines the counter style. Predefined styles: none, left, right

cvoffset [-23pt] sets the vertical offset of the counters in the margin.

You can download application examples and their solutions from the project page. The puzzles are originally licensed under CC BY NC SA.

4 Implemenation

4.1 logicpuzzle.sty

```
1 ⟨*package⟩
2 \NeedsTeXFormat{LaTeX2e}%
3 \ProvidesPackage{logicpuzzle}[2014/06/15 v2.5 logicpuzzle.sty%
4                 - Josef Kleber (C) 2013-2014]%
```

4.1.1 Package initialization

First, we load the packages needed by `logicpuzzle.sty` and the Ti*k*Z libraries we need.

```
5 \RequirePackage{xkeyval}%
6 \RequirePackage{ifthen}%
7 \RequirePackage{ragged2e}%
8 \RequirePackage{marginnote}%
9 \RequirePackage{tikz}%
10 \usetikzlibrary{decorations.pathmorphing,decorations.pathreplacing,%
11                 calc,shapes.geometric}%
```

We define a set of PGF layers for placing material on them and their order.

```
\pgfdeclarelayer{LPdump}%
\pgfdeclarelayer{LPbgcolor}%
\pgfdeclarelayer{LPbackgroundtwo}%
\pgfdeclarelayer{LPbackground}%
\pgfdeclarelayer{LPforeground}%
\pgfdeclarelayer{LPforegroundtwo}%
\pgfsetlayers{LPdump,LPbgcolor,LPbackgroundtwo,LPbackground,main,%
              LPforeground,LPforegroundtwo}%
%
```

We also need some counters and lengths

```
\newcounter{LP@rows}%
\newcounter{LP@columns}%
\newcounter{LP@counter@unique}%
\newcounter{LP@counti}%
\newcounter{LP@countii}%
\newcounter{LP@countiii}%
\newcounter{LP@whiledo@i}%
\newcounter{LP@whiledo@ii}%
\newcounter{LP@count@ig@i}%
\newcounter{LP@count@ig@ii}%
\newcounter{LP@count@ig@iii}%
\newcounter{LP@count@ig@iv}%
\newcounter{LP@puzzlecounter}%
\setcounter{LP@puzzlecounter}{1}%
%
\newlength\LP@length%
```

We define generic macros for puzzle options and add some defaults. Furthermore we define some macros, we will use all over the package.

```
\gdef\LP@rows{}%
\gdef\LP@columns{}%
\gdef\LP@scale{1}%
\gdef\LP@color{black}%
\gdef\LP@bgcolor{}%
\gdef\LP@fontsize\Large%
\gdef\LP@cvoffset{0pt}%
%
\newcommand*\LP@counterstyle{none}%
\newcommand*\LP@titleformat{}%
\gdef\LP@env@prefix{}%
\gdef\LP@package{}%
\def\LP@normallines{0.5pt}%
\def\LP@thicklines{1.5pt}%
\def\LP@grid@linestyle{}%
\def\LP@draw@opacity{1}%
\def\LP@rel@tikzpath{}%
\def\LP@tracks@scale{.3}%
\def\LP@fw@linestyle{-|}%
\newcommand*\LP@Pfive{.5}%
```

4.1.2 Defining options

\LP@define@key

We define a generic command for the definition of puzzle options with both global (for the \puzzlesetup commands) and local scope for the optional argument of the puzzles!

\LP@define@key{⟨*puzzle prefix*⟩}{⟨*puzzle*⟩}{⟨*option*⟩}{⟨*default*⟩}

```
\newcommand*\LP@define@key[4]%
{%
  \expandafter\gdef\csname#1@#3\endcsname{#4}%
  \define@key{#2.sty}{#3}[#4]%
  {%
    \expandafter\gdef\csname#1@#3\endcsname{##1}%
  }%
  \define@key{#2}{#3}%
  {%
    \expandafter\def\csname#1@#3\endcsname{##1}%
  }%
}%
```

\LP@define@choicekey@fontsize

We do the same for the more complicated choicekey for the fontsize option.

\LP@define@choicekey@fontsize{⟨*puzzle prefix*⟩}{⟨*puzzle*⟩}{⟨*default*⟩}

```
\newcommand*\LP@define@choicekey@fontsize[3]%
{%
  \expandafter\gdef\csname#1@fontsize\endcsname{\Large}%
  \define@choicekey*{#2.sty}{fontsize}[\LP@dck@fontsize\nr]{%
                    tiny,scriptsize,footnotesize,small,normalsize,%
                    large,Large,LARGE,huge,Huge}[#3]%
  {%
    \ifcase\nr\relax%
      \expandafter\gdef\csname#1@fontsize\endcsname{\tiny}%
    \or%
      \expandafter\gdef\csname#1@fontsize\endcsname{\scriptsize}%
    \or%
      \expandafter\gdef\csname#1@fontsize\endcsname{\footnotesize}%
    \or%
      \expandafter\gdef\csname#1@fontsize\endcsname{\small}%
    \or%
      \expandafter\gdef\csname#1@fontsize\endcsname{\normalsize}%
    \or%
      \expandafter\gdef\csname#1@fontsize\endcsname{\large}%
    \or%
      \expandafter\gdef\csname#1@fontsize\endcsname{\Large}%
    \or%
      \expandafter\gdef\csname#1@fontsize\endcsname{\LARGE}%
    \or%
      \expandafter\gdef\csname#1@fontsize\endcsname{\huge}%
    \or%
      \expandafter\gdef\csname#1@fontsize\endcsname{\Huge}%
```

```
    \fi%
  }%
  \define@choicekey*{#2}{fontsize}[\LP@dck@fontsize\nr]{%
                    tiny,scriptsize,footnotesize,small,normalsize,%
                    large,Large,LARGE,huge,Huge}[#3]%
  {%
    \ifcase\nr\relax%
      \expandafter\def\csname#1@fontsize\endcsname{\tiny}%
    \or%
      \expandafter\def\csname#1@fontsize\endcsname{\scriptsize}%
    \or%
      \expandafter\def\csname#1@fontsize\endcsname{\footnotesize}%
    \or%
      \expandafter\def\csname#1@fontsize\endcsname{\small}%
    \or%
      \expandafter\def\csname#1@fontsize\endcsname{\normalsize}%
    \or%
      \expandafter\def\csname#1@fontsize\endcsname{\large}%
    \or%
      \expandafter\def\csname#1@fontsize\endcsname{\Large}%
    \or%
      \expandafter\def\csname#1@fontsize\endcsname{\LARGE}%
    \or%
      \expandafter\def\csname#1@fontsize\endcsname{\huge}%
    \or%
      \expandafter\def\csname#1@fontsize\endcsname{\Huge}%
    \fi%
  }%
}%
```

4.1.3 Defining colors

`\LP@definecolor` We define a command for defining (rgb) colors. For other color models, use `xcolor`'s `\definecolor` command.

`\LP@definecolor{`⟨*name*⟩`}{`⟨*rgb color*⟩`}`

```
\newcommand*\LP@definecolor[2]%
{%
  \definecolor{#1}{rgb}{#2}%
}%
```

Predefined colors:

```
\LP@definecolor{LP@c@i}{.9,.9,.98}%
\LP@definecolor{LP@c@ii}{.688,.932,.932}%
\LP@definecolor{LP@c@iii}{.88,1,1}%
\LP@definecolor{LP@c@iv}{1,1,.88}%
\LP@definecolor{LP@c@v}{1,.855,.725}%
\LP@definecolor{LP@c@vi}{.498,1,0}%
\LP@definecolor{LP@c@vii}{.53,.808,.98}%
```

```
\LP@definecolor{LP@c@viii}{.196,.804,.196}%
\LP@definecolor{LP@c@ix}{1,.96,.932}%
\LP@definecolor{LP@c@griddots}{.25,.51,.33}%
\LP@definecolor{LP@c@cross}{.25,.51,.33}%
\LP@definecolor{LP@c@mirror}{.8,.8,.8}%
\LP@definecolor{LP@c@tunnel}{.7,.7,.7}%
\LP@definecolor{LP@c@tunnel@i}{.8,.8,.8}%
\LP@definecolor{LP@c@tunnel@ii}{.5,.5,.5}%
\LP@definecolor{LP@c@bridge}{1,0,0}%
\colorlet{LP@c@track}{black!80}%
```

4.1.4 Drawing grids

`\LP@drawgrid`

We define a command for drawing the standard grid used by all puzzles. In general, this should be a grid with a step of 1cm and thin lines with size (1,1) to (`columns` + 1,`rows` + 1). You can influence the grid by redefining the `\LP@grid@linestyle` (default: `solid` – maybe you want `dashed`) and `\LP@draw@opacity` (0 (transparent) – 1 (opaque)). They should be changed only within groups, like puzzle environment definitions.

`\LP@drawgrid{`⟨*xmin*⟩`}{`⟨*ymin*⟩`}{`⟨*xmax*⟩`}{`⟨*ymax*⟩`}{`⟨*step*⟩`}`

```
\newcommand*\LP@drawgrid[5]%
{%
  \setcounter{LP@counti}{#3}% max column
  \setcounter{LP@countii}{#4}% max row
  \stepcounter{LP@counti}%
  \stepcounter{LP@countii}%
  \draw[step=#5,line width=\LP@normallines,\LP@grid@linestyle,%
        draw opacity=\LP@draw@opacity]%
        (#1,#2) grid (\value{LP@counti},\value{LP@countii});%
}%
```

4.1.5 Drawing the puzzle background

`\LP@drawbackground`

For drawing the puzzle background, we simply draw a rectangle with the size of the puzzle on the `LPbgcolor` layer and fill it with {⟨*bgcolor*⟩}.

`\LP@drawbackground{`⟨*xmin*⟩`}{`⟨*ymin*⟩`}{`⟨*xmax*⟩`}{`⟨*ymax*⟩`}{`⟨*bgcolor*⟩`}`

```
\newcommand*\LP@drawbackground[5]%
{%
```

Of course, we only draw a background, if {⟨*bgcolor*⟩} is not empty!

```
  \ifthenelse{\equal{#5}{}}%
  {}% no bgcolor
  {%
    \setcounter{LP@counti}{#3}% max column
    \setcounter{LP@countii}{#4}% max row
```

```
    \stepcounter{LP@counti}%
    \stepcounter{LP@countii}%
    \begin{pgfonlayer}{LPbgcolor}%
      \fill[color=#5] (#1,#2) rectangle%
                      (\value{LP@counti},\value{LP@countii});%
    \end{pgfonlayer}%
  }%
}%
```

4.1.6 In the grid

`\LP@ingrid` With this macro, we can check if the specified `column` and `row` is within the puzzle borders. Otherwise we issue en error message. This macro can be used by higher level commands, which try to place something in the grid.

`\LP@ingrid{`⟨*column*⟩`}{`⟨*row*⟩`}{`⟨*max column*⟩`}{`⟨*max row*⟩`}{`⟨*package*⟩`}`

```
\newcommand*\LP@ingrid[5]%
{%
```

First, we define some counters to store the arguments.

```
  \setcounter{LP@count@ig@i}{#1}% column
  \setcounter{LP@count@ig@ii}{#2}% row
  \setcounter{LP@count@ig@iii}{#3}% max column
  \setcounter{LP@count@ig@iv}{#4}% max row
```

Then, we can check, if the specified `coordinate` is within the borders of the puzzle.

```
  \ifnum\value{LP@count@ig@i}<1%
    \PackageError{#5}%
      {element outside of the grid}%
      {You tried to set an element at (#1,#2),\MessageBreak%
       which is outside the grid (1,1) .. (#3,#4)}%
  \fi%
  \ifnum\value{LP@count@ig@ii}<1%
    \PackageError{#5}%
      {element outside of the grid}%
      {You tried to set an element at (#1,#2),\MessageBreak%
       which is outside the grid (1,1) .. (#3,#4)}%
  \fi%
  \ifnum\value{LP@count@ig@i}>\value{LP@count@ig@iii}%
    \PackageError{#5}%
      {element outside of the grid}%
      {You tried to set an element at (#1,#2),\MessageBreak%
       which is outside the grid (1,1) .. (#3,#4)}%
  \fi%
  \ifnum\value{LP@count@ig@ii}>\value{LP@count@ig@iv}%
    \PackageError{#5}%
      {element outside of the grid}%
      {You tried to set an element at (#1,#2),\MessageBreak%
```

```
199       which is outside the grid (1,1) .. (#3,#4)}%
200   \fi%
201 }%
```

\setrule For the kendoku and killersudoku environments, we need a command to place a calculation rule in the top left corner of the specified cell.

\setrule{⟨*column*⟩}{⟨*row*⟩}{⟨*rule*⟩}

```
202 \newcommand*\setrule[3]%
203 {%
```

First, we copy the scale and bgcolor values from the current environment.

```
204   \LP@set@LP@scale{\LP@env@prefix}%
205   \LP@set@LP@bgcolor{\LP@env@prefix}%
```

If no bgcolor is specified, we use white for drawing our helper rectangle. We also step our unique node counter, we need for referencing nodes between different layers.

```
206   \ifthenelse{\equal{\LP@bgcolor}{}}%
207   {\gdef\LP@sr@bgcolor{white}}%
208   {\gdef\LP@sr@bgcolor{\LP@bgcolor}}%
209   \stepcounter{LP@counter@unique}%
```

First, we draw a (invisible) helper rectangle on the LPdump layer (behind the LPbgcolor layer) in the node A_\theLP@counter@unique

```
210   \begin{pgfonlayer}{LPdump}%
211     \node [shape=rectangle,inner sep=0pt] (A_\theLP@counter@unique)%
212     at (#1\LP@Pfive,#2\LP@Pfive)%
213        {\tikz\draw[scale=\LP@scale,color=\LP@sr@bgcolor]%
214                    (0.08,0.1) rectangle (0.92,0.9);};%
215   \end{pgfonlayer}%
```

Then, we can place a visible node on the main layer in the top left corner of the invisible helper rectangle with the rule (size: \tiny × scale).

```
216   \node [shape=rectangle,inner sep=0pt,anchor=north west,%
217          scale=\LP@scale,font=\tiny] at%
218         (A_\theLP@counter@unique.north west) {$#3$};%
219 }%
```

\LP@G@setcellcontent Here, we define a generic macro for placing material into nodes placed in the bottom left corner of the grid cell. You can use the options vcenter and hcenter in the optional argument [⟨*options*⟩] of the macro to center the content horizontally and/or vertically.

\LP@G@setcellcontent[⟨*options*⟩]{⟨*column*⟩}{⟨*row*⟩}{⟨*content*⟩}

```
220 \define@key{LP@G@setcellcontent}{vcenter}[\LP@Pfive]%
221 {%
```

```
  \def\LP@scc@v{\LP@Pfive}%
}%
\define@key{LP@G@setcellcontent}{hcenter}[\LP@Pfive]%
{%
  \def\LP@scc@h{\LP@Pfive}%
}%
%
\newcommand*\LP@G@setcellcontent[4][]%
{%
  \def\LP@scc@h{}%
  \def\LP@scc@v{}%
  \setkeys{LP@G@setcellcontent}{#1}%
  \node at (#2\LP@scc@h,#3\LP@scc@v){#4};%
}%
```

`\LP@setcellcontent` For compatibility, we still provide the old `\LP@setcellcontentXY` macros.

```
\newcommand*\LP@setcellcontent[3]%
{%
  \LP@G@setcellcontent[hcenter,vcenter]{#1}{#2}{#3}%
}%
```

`\LP@setcellcontentC`

```
\newcommand*\LP@setcellcontentC[3]%
{%
  \LP@G@setcellcontent{#1}{#2}{#3}%
}%
```

`\LP@setcellcontentVE`

```
\newcommand*\LP@setcellcontentVE[3]%
{%
  \LP@G@setcellcontent[vcenter]{#1}{#2}{#3}%
}%
```

`\LP@setcellcontentHE`

```
\newcommand*\LP@setcellcontentHE[3]%
{%
  \LP@G@setcellcontent[hcenter]{#1}{#2}{#3}%
}%
```

`\LP@setrowcontents` We want also be able to set the contents of a complete row.

`\LP@setrowcontents{`⟨*csv list*⟩`}{`⟨*column*⟩`}{`⟨*row*⟩`}`

```
\newcommand*\LP@setrowcontents[3]%
{%
```

We copy `fontsize` from the current environment, if we want to typeset numbers or letters.

```
  \LP@set@LP@fontsize{\LP@env@prefix}%
  \setcounter{LP@counti}{#2}%
  \setcounter{LP@countii}{#3}%
```

We loop through the list and create a centered node in cell (`column`,`row`). Finally, we step the `column` counter.

```
  \foreach \LP@element in {#1}%
  {%
    \LP@G@setcellcontent[vcenter,hcenter]%
      {\arabic{LP@counti}}{\arabic{LP@countii}}%
      {\LP@fontsize\LP@element}%
    \stepcounter{LP@counti}%
  }%
}%
```

`\LP@setcolumncontents` Of course, we want to do the same for columns.

`\LP@setcolumncontens{`⟨*csv list*⟩`}{`⟨*column*⟩`}{`⟨*row*⟩`}`

```
\newcommand*\LP@setcolumncontents[3]%
{%
  \LP@set@LP@fontsize{\LP@env@prefix}%
  \setcounter{LP@counti}{#2}%
  \setcounter{LP@countii}{#3}%
  \foreach \LP@element in {#1}%
  {%
    \LP@G@setcellcontent[vcenter,hcenter]%
      {\arabic{LP@counti}}{\arabic{LP@countii}}%
      {\LP@fontsize\LP@element}%
    \stepcounter{LP@countii}%
  }%
}%
```

`\LP@setrowcontents@edge` For environments like `laserbeam`, we need to typeset `row` contents on the left border of the cell, instead of centered. Therfore, we only use `vcenter`.

`\LP@setrowcontents@edge{`⟨*csv list*⟩`}{`⟨*column*⟩`}{`⟨*row*⟩`}`

```
\newcommand*\LP@setrowcontents@edge[3]%
{%
  \LP@set@LP@fontsize{\LP@env@prefix}%
  \setcounter{LP@counti}{#2}%
  \setcounter{LP@countii}{#3}%
  \foreach \LP@element in {#1}%
  {%
    \LP@G@setcellcontent[vcenter]%
      {\arabic{LP@counti}}{\arabic{LP@countii}}%
      {\LP@fontsize\LP@element}%
    \stepcounter{LP@counti}%
  }%
}%
```

\LP@setcolumncontents@edge Furthermore, we need the ability to typeset a column on the bottom border of a cell (hcenter).

\LP@setcolumncontents@edge{⟨*csv list*⟩}{⟨*column*⟩}{⟨*row*⟩}

```
\newcommand*\LP@setcolumncontents@edge[3]%
{%
  \LP@set@LP@fontsize{\LP@env@prefix}%
  \setcounter{LP@counti}{#2}%
  \setcounter{LP@countii}{#3}%
  \foreach \LP@element in {#1}%
  {%
    \LP@G@setcellcontent[hcenter]%
      {\arabic{LP@counti}}{\arabic{LP@countii}}%
      {\LP@fontsize\LP@element}%
    \stepcounter{LP@countii}%
  }%
}%
```

\setrow The user command for typesetting row contents.

\setrow{⟨*row*⟩}{⟨*csv list*⟩}

```
\newcommand*\setrow[2]%
{%
```

First, we copy scale, which might be necessary for graphical objects, like \Mine. Then, we hand over the arguments to the generic macro for typesetting row contents and start with column 1.

```
  \LP@set@LP@scale{\LP@env@prefix}%
  \LP@setrowcontents{#2}{1}{#1}%
}%
```

\setcolumn Again, we do the same for columns.

\setcolumn{⟨*column*⟩}{⟨*csv list*⟩}

```
\newcommand*\setcolumn[2]%
{%
  \LP@set@LP@scale{\LP@env@prefix}%
  \LP@setcolumncontents{#2}{#1}{1}%
}%
```

\setcell We need to set numbers, letters or a graphical object into a central node in grid cell (column,row)

\setcell{⟨*column*⟩}{⟨*row*⟩}{⟨*element*⟩}

```
\newcommand*\setcell[3]%
{%
```

First, we copy `scale`, `fontsize`, `rows` and `columns` from the current environment.

```
\LP@set@LP@scale{\LP@env@prefix}%
\LP@set@LP@fontsize{\LP@env@prefix}%
\LP@set@LP@rows{\LP@env@prefix}%
\LP@set@LP@columns{\LP@env@prefix}%
```

Then, we test if ({⟨*column*⟩},{⟨*row*⟩}) is within the borders of the puzzle. Finally, we typeset {⟨*element*⟩} into a central node.

```
\LP@ingrid{#1}{#2}{\LP@columns}{\LP@rows}{\LP@package}%
\LP@G@setcellcontent[hcenter,vcenter]{#1}{#2}{\LP@fontsize#3}%
}%
```

`\setcells` We also want to typeset the same element into several grid cells. Therefore, we use a {⟨*csv list*⟩} with the format: {⟨*column/row, ...*⟩}

`\setcells{`⟨*csv list*⟩`}{`⟨*element*⟩`}`

```
\newcommand*\setcells[2]%
{%
  \LP@set@LP@scale{\LP@env@prefix}%
  \LP@set@LP@fontsize{\LP@env@prefix}%
  \LP@set@LP@rows{\LP@env@prefix}%
  \LP@set@LP@columns{\LP@env@prefix}%
  \foreach \LP@sc@column/\LP@sc@row in {#1}%
  {%
    \LP@ingrid{\LP@sc@column}{\LP@sc@row}%
      {\LP@columns}{\LP@rows}{\LP@package}%
    \LP@G@setcellcontent[hcenter,vcenter]%
      {\LP@sc@column}{\LP@sc@row}{\LP@fontsize#2}%
  };%
}%
```

`\setbigcell` For the `slitherlink` environment, we need to typeset a huge (2×2) grid cell.

`\setbigcell[`⟨*fontsize*⟩`]{`⟨*column*⟩`}{`⟨*row*⟩`}{`⟨*element*⟩`}`

```
\newcommand*\setbigcell[4][Huge]%
{%
```

First, we copy `scale` and `bgcolor`. If `bgcolor` is not specified, we assume `bgcolor` is white!

```
  \LP@set@LP@scale{\LP@env@prefix}%
  \LP@set@LP@bgcolor{\LP@env@prefix}%
  \ifthenelse{\equal{\LP@bgcolor}{}}%
  {\gdef\LP@sbc@bgcolor{white}}%
  {\gdef\LP@sbc@bgcolor{\LP@bgcolor}}%
```

The center of (2×2) cell is the bottom left corner of (`column` $+ 1$, `row` $+ 1$)

```
  \setcounter{LP@counti}{#2}%
  \setcounter{LP@countii}{#3}%
  \stepcounter{LP@counti}%
  \stepcounter{LP@countii}%
```

First, we 'clear' (2×2) area with `bgcolor` and respect the grid line style.

```
  \draw[line width=\LP@normallines,fill=\LP@sbc@bgcolor,%
        \LP@grid@linestyle]%
       (#2,#3) rectangle ++(2,2);%
```

As the grid dots were drawn on the `LPforeground` layer, we have to 'overpaint' the center dot on the `LPforegroundtwo` layer with `bgcolor`. Finally, we can typeset {⟨*element*⟩}.

```
  \begin{pgfonlayer}{LPforegroundtwo}%
    \fill[color=\LP@sbc@bgcolor]%
      (\arabic{LP@counti},\arabic{LP@countii})%
      circle [radius=3.5pt*\LP@scale];%
    \node at (\arabic{LP@counti},\arabic{LP@countii})%
      {\csname#1\endcsname#4};%
  \end{pgfonlayer}%
}%
```

`\setcolorrow` For the `hitori` environment, we need to typeset row contents with numbers associated to background colors.

`\setcolorrow{`⟨*row*⟩`}{`⟨*csv list*⟩`}`

```
\newcommand*\setcolorrow[2]%
{%
```

We start at `column` 1 and loop through {⟨*csv list*⟩}

```
  \setcounter{LP@counti}{1}%
  \setcounter{LP@countii}{#1}%
  \foreach \LP@element in {#2}%
  {%
```

If the list element is 0, we fill the cell `black` on the `LPbackground` layer.

```
    \ifthenelse{\equal{\LP@element}{0}}%
    {%
      \gdef\LP@HT@color{black}%
      \begin{puzzlebackground}%
        \fillcell{\arabic{LP@counti}}{\arabic{LP@countii}}%
      \end{puzzlebackground}%
    }%
    {%
```

Otherwise, we fill the cell with predefined color `LP@c@romannumber` on the `LPbackground` layer and typeset the list element into the grid cell.

```
      \expandafter\gdef\expandafter\LP@HT@color%
        {LP@c@\romannumeral\LP@element}%
      \begin{puzzlebackground}%
        \fillcell{\arabic{LP@counti}}{\arabic{LP@countii}}%
      \end{puzzlebackground}%
      \setcell{\arabic{LP@counti}}{\arabic{LP@countii}}{\LP@element}%
    }%
    \stepcounter{LP@counti}%
  }%
}%
```

`\setcolorcolumn` Again, the same for columns.

```
\newcommand*\setcolorcolumn[2]%
{%
  \setcounter{LP@counti}{#1}%
  \setcounter{LP@countii}{1}%
  \foreach \LP@element in {#2}%
  {%
    \ifthenelse{\equal{\LP@element}{0}}%
    {%
      \gdef\LP@HT@color{black}%
      \begin{puzzlebackground}%
        \fillcell{\arabic{LP@counti}}{\arabic{LP@countii}}%
      \end{puzzlebackground}%
    }%
    {%
      \expandafter\gdef\expandafter\LP@HT@color%
        {LP@c@\romannumeral\LP@element}%
      \begin{puzzlebackground}%
        \fillcell{\arabic{LP@counti}}{\arabic{LP@countii}}%
      \end{puzzlebackground}%
      \setcell{\arabic{LP@counti}}{\arabic{LP@countii}}{\LP@element}%
    }%
    \stepcounter{LP@countii}%
  }%
}%
```

`\fillcell` Sometimes, we need to 'blacken' a grid cell.

`\fillcell{`⟨*column*⟩`}{`⟨*row*⟩`}`

```
\newcommand*\fillcell[2]%
{%
  \LP@set@LP@scale{\LP@env@prefix}%
  \LP@set@LP@color{\LP@env@prefix}%
  \LP@set@LP@rows{\LP@env@prefix}%
  \LP@set@LP@columns{\LP@env@prefix}%
```

After a border check, we typeset a black block (`\LP@Block`) into the grid cell.

```
  \LP@ingrid{#1}{#2}{\LP@columns}{\LP@rows}{\LP@package}%
```

```
  \LP@G@setcellcontent[hcenter,vcenter]{#1}{#2}{\LP@Block}%
}%
```

`\fillrow` We also want to allow the filling of (parts of a) row.

`\fillrow{`⟨*row*⟩`}{`⟨*csv list*⟩`}`

```
\newcommand*\fillrow[2]%
{%
  \setcounter{LP@counti}{1}%
  \setcounter{LP@countii}{#1}%
  \foreach \LP@element in {#2}%
  {%
```

We loop through the list and if `element` is 1, we fill this grid cell.

```
    \ifthenelse{\equal{\LP@element}{1}}%
    {\fillcell{\arabic{LP@counti}}{\arabic{LP@countii}}}%
    {}%
    \stepcounter{LP@counti}%
  }%
}%
```

`\fillcolumn` Again the same, for columns!

`\fillcolumn{`⟨*column*⟩`}{`⟨*csv list*⟩`}`

```
%
\newcommand*\fillcolumn[2]%
{%
  \setcounter{LP@counti}{#1}%
  \setcounter{LP@countii}{1}%
  \foreach \LP@element in {#2}%
  {%
    \ifthenelse{\equal{\LP@element}{1}}%
    {\fillcell{\arabic{LP@counti}}{\arabic{LP@countii}}}%
    {}%
    \stepcounter{LP@countii}%
  }%
}%
```

`\framearea` Sometimes, we need to frame a specified area.

`\framearea{`⟨*color*⟩`}{`⟨*TikZ path*⟩`}`

```
\newcommand*\framearea[2]%
{%
  \draw[line width=\LP@thicklines,color=#1] #2;%
}%
```

`\fillarea` Sometimes, we even want to fill the area.

`\fillarea{`⟨*color*⟩`}{`⟨*TikZ path*⟩`}`

```
\newcommand*\fillarea[2]%
{%
  \draw[line width=\LP@thicklines,fill=#1] #2;%
}%
```

`\colorarea` In some cases it might be better just to fill the area without drawing a frame.

`\colorarea{`⟨*color*⟩`}{`⟨*TikZ path*⟩`}`

```
\newcommand*\colorarea[2]%
{%
  \fill[color=#1] #2 ;%
}%
```

`\tikzpath` Using a `{`⟨*Tikz path*⟩`}` can be cumbersome. `\tikzpath` construct a path starting at the bottom left corner of grid cell (`column,row`). If want to start in the center of the cell, redefine `\LP@rel@tikzpath` to `.5` inside a group! Inside `{`⟨*csv list of relative moves*⟩`}`, you can specify relative movements from one grid cell to the next based on num pad (4 → one cell right, 2 → one cell down and 9 → one cell right up).

`\tikzpath{`⟨*column*⟩`}{`⟨*row*⟩`}{`⟨*csv list of relative movement*⟩`}`

```
\newcommand*\tikzpath[3]%
{%
```

starting point

```
  (#1\LP@rel@tikzpath,#2\LP@rel@tikzpath)%
```

We loop through the list and add a relative path segment based on the direction indicator.

```
  \foreach \LP@direction in {#3}%
  {%
    \ifnum\LP@direction=1%
    --++(-1,-1)%
    \fi%
    \ifnum\LP@direction=2%
    --++(0,-1)%
    \fi%
    \ifnum\LP@direction=3%
    --++(1,-1)%
    \fi%
    \ifnum\LP@direction=4%
    --++(-1,0)%
    \fi%
    \ifnum\LP@direction=6%
    --++(1,0)%
    \fi%
    \ifnum\LP@direction=7%
    --++(-1,1)%
```

```
    \fi%
    \ifnum\LP@direction=8%
    --++(0,1)%
    \fi%
    \ifnum\LP@direction=9%
    --++(1,1)%
    \fi%
  };%
}%
```

`\xtikzpath` `\xtikzpath` is based on `\tikzpath` with a slightly different format in {⟨*csv list of relative movements*⟩}. It allows pairs of `direction/length`, e.g. (6/2) means two cells right.

`\xtikzpath{`⟨*column*⟩`}{`⟨*row*⟩`}{`⟨*csv list of relative movements*⟩`}`

```
\newcommand*\xtikzpath[3]%
{%
  (#1\LP@rel@tikzpath,#2\LP@rel@tikzpath)%
  \foreach \LP@dir/\LP@plength in {#3}%
  {%
    \ifnum\LP@dir=1%
    --++(-\LP@plength,-\LP@plength)%
    \fi%
    \ifnum\LP@dir=2%
    --++(0,-\LP@plength)%
    \fi%
    \ifnum\LP@dir=3%
    --++(\LP@plength,-\LP@plength)%
    \fi%
    \ifnum\LP@dir=4%
    --++(-\LP@plength,0)%
    \fi%
    \ifnum\LP@dir=6%
    --++(\LP@plength,0)%
    \fi%
    \ifnum\LP@dir=7%
    --++(-\LP@plength,\LP@plength)%
    \fi%
    \ifnum\LP@dir=8%
    --++(0,\LP@plength)%
    \fi%
    \ifnum\LP@dir=9%
    --++(\LP@plength,\LP@plength)%
    \fi%
  };%
}%
```

`\filldiagonals` For some puzzles we need colored diagonals indicating that also the diagonals are relevant, not just rows and columns.

`\filldiagonals[`⟨*color*⟩`]`

```
\newcommand*\filldiagonals[1][yellow!20]%
{%
```

We copy and get `scale`, `$rows+1$` and `$columns+1$`,

```
  \LP@set@LP@columns{\LP@env@prefix}%
  \LP@set@LP@rows{\LP@env@prefix}%
  \LP@set@LP@scale{\LP@env@prefix}%
  \setcounter{LP@counti}{\LP@columns}%
  \setcounter{LP@countii}{\LP@rows}%
  \stepcounter{LP@counti}%
  \stepcounter{LP@countii}%
```

We only color the diagonals, if the puzzle is quadratic. Otherwise, coloring diagonals doen't make sense and we issue an error.

```
  \ifnum\value{LP@counti}=\value{LP@countii}%
```

We color the diagonals on the `LPbackground` layer and redefine `\LP@color` (for `\LP@Block`) inside a group with local scope.

```
    \begin{puzzlebackground}%
      \def\LP@color{#1}%
```

We use two counters running from (`1 .. rows`) and (`rows .. 1`) and color the cells on the diagonals in a loop.

```
      \setcounter{LP@whiledo@i}{1}%
      \setcounter{LP@whiledo@ii}{\LP@rows}%
      \whiledo{\value{LP@whiledo@i}<\value{LP@counti}}%
      {%
        \LP@G@setcellcontent[hcenter,vcenter]%
          {\arabic{LP@whiledo@i}}{\arabic{LP@whiledo@i}}{\LP@Block}%
        \LP@G@setcellcontent[hcenter,vcenter]%
          {\arabic{LP@whiledo@i}}{\arabic{LP@whiledo@ii}}{\LP@Block}%
        \stepcounter{LP@whiledo@i}%
        \addtocounter{LP@whiledo@ii}{-1}%
      }%
    \end{puzzlebackground}%
  \else%
    \PackageError{\LP@package}%
      {non quadratic grid (\filldiagonals)}%
      {You tried to fill diagonals,\MessageBreak in a non quadratic%
       grid (1,1) .. (\arabic{LP@counti},\arabic{LP@countii})%
       \MessageBreak doesn't make sense!}%
  \fi%
}%
```

`\framepuzzle` We might want to frame the puzzle with a thicker line.

`\framepuzzle[`⟨*color*⟩`]`

```
\newcommand*\framepuzzle[1][black]%
{%
  \LP@set@LP@columns{\LP@env@prefix}%
  \LP@set@LP@rows{\LP@env@prefix}%
  \setcounter{LP@counti}{\LP@columns}%
  \setcounter{LP@countii}{\LP@rows}%
  \stepcounter{LP@counti}%
  \stepcounter{LP@countii}%
```

We copied `rows` and `columns` to get the top right corner of the puzzle. Now, we can draw the thicker lines.

```
  \draw[line width=\LP@thicklines,color=#1]%
    (1,1)--(1,\arabic{LP@countii});%
  \draw[line width=\LP@thicklines,color=#1]%
    (1,\arabic{LP@countii})--%
    (\arabic{LP@counti},\arabic{LP@countii});%
  \draw[line width=\LP@thicklines,color=#1]%
    (\arabic{LP@counti},\arabic{LP@countii})--%
    (\arabic{LP@counti},1);%
  \draw[line width=\LP@thicklines,color=#1]%
    (\arabic{LP@counti},1)--(1,1);%
}%
```

`puzzlebackground` Sometimes, we want to move material to the `LPbackground` layer.

```
\newenvironment{puzzlebackground}%
{%
  \begin{pgfonlayer}{LPbackground}%
}%
{%
  \end{pgfonlayer}%
}%
```

`puzzleforeground` Sometimes, we want to move material to the `LPforeground` layer.

```
\newenvironment{puzzleforeground}%
{%
  \begin{pgfonlayer}{LPforeground}%
}%
{%
  \end{pgfonlayer}%
}%
```

4.1.7 Around the grid

For most puzzles, we need to put numbers below, above, to the right or to the left of the puzzle.

`\LP@bottomrow` Put numbers below the puzzle.

`\LP@bottomrow{`⟨*csv list*⟩`}`

```
\newcommand*\LP@bottomrow[1]%
{%
  \LP@setrowcontents{#1}{1}{0}%
}%
```

`\LP@bottomrow@edge` Put numbers below the puzzle, but on the edge.

`\LP@bottomrow@edge{`⟨*csv list*⟩`}`

```
\newcommand*\LP@bottomrow@edge[1]%
{%
  \LP@setrowcontents@edge{#1}{1}{0}%
}%
```

`\LP@leftcolumn` Put numbers left to the puzzle.

`\LP@leftcolumn{`⟨*csv list*⟩`}`

```
\newcommand*\LP@leftcolumn[1]%
{%
  \LP@setcolumncontents{#1}{0}{1}%
}%
```

`\LP@toprow` Put number on top of the puzzle.

`\LP@toprow{`⟨*csv list*⟩`}`

```
\newcommand*\LP@toprow[1]%
{%
  \LP@setrowcontents{#1}{1}{\theLP@rows}%
}%
```

`\LP@rightcolumn` Put numbers right to the puzzle.

`\LP@rightcolumn{`⟨*csv list*⟩`}`

```
\newcommand*\LP@rightcolumn[1]%
{%
  \LP@setcolumncontents{#1}{\theLP@columns}{1}%
}%
```

`\LP@rightcolumn@edge` Put numbers right next to the puzzle, but on the edge.

`\LP@rightcolumn@rdge{`⟨*csv list*⟩`}`

```
\newcommand*\LP@rightcolumn@edge[1]%
{%
  \LP@setcolumncontents@edge{#1}{\theLP@columns}{1}%
}%
```

4.1.8 Presentation

`\titleformat` You can redefine the format of the puzzle at any time.

```
\newcommand*\titleformat[1]%
{%
  \renewcommand*\LP@titleformat{#1}%
}%
```

We define thew default title format:

```
\titleformat{\centering\Large\color{blue}}%
```

`\puzzlecounter` User command to get the current puzzle counter.

```
\newcommand*\puzzlecounter{\theLP@puzzlecounter}%
```

`\setpuzzlecounter` User command to reset the puzzle counter, e.g. before the solution section.

```
\newcommand*\setpuzzlecounter[1]%
{%
  \setcounter{LP@puzzlecounter}{#1}%
}%
```

`\LP@drawcounter` Maybe we want to draw puzzle counter, or not (`counterstyle=none`).

`\LP@drawcounter{`⟨*counterstyle*⟩`}`

```
\newcommand*\LP@drawcounter[1]%
{%
```

We copy `cvoffset` and execute the macro we defined for the respective `counterstyle`.

```
  \LP@set@LP@cvoffset{\LP@env@prefix}%
  \csname LP@cs@#1\endcsname%
}%
```

`\definecounterstyle` Maybe a user want to define his/her own counter style.

`\definecounterstyle{`⟨*counterstyle*⟩`}{`⟨*definition*⟩`}`

```
\newcommand*{\definecounterstyle}[2]%
{%
```

We define a `counterstyle` macro and store its definition.

```
  \expandafter\gdef\csname LP@cs@#1\endcsname{#2}%
}%
```

We define three predefined `counterstyles` `none`, `left` and `right`.

```
\definecounterstyle{none}{}%
\definecounterstyle{left}{%
  \begingroup\reversemarginpar%
    \marginnote{%
      \tikz\node[shape=rectangle,fill=yellow!40,inner sep=7pt,draw,%
                 rounded corners=3pt,thick]%
                  {\Huge\puzzlecounter};}[\LP@cvoffset]%
  \endgroup%
}%
\definecounterstyle{right}{%
  \marginnote{%
    \tikz\node[shape=rectangle,fill=yellow!40,inner sep=7pt,draw,%
                rounded corners=3pt,thick]%
                 {\Huge\puzzlecounter};%
  }[\LP@cvoffset]%
}%
```

4.1.9 Misc macros

`\LP@Block` We define a Ti*k*Z picture with the size of a grid cell ($1cm \times 1cm$) and fill it with `color`.

```
\newcommand*\LP@Block{%
  \tikz[scale=\LP@scale]%
    \draw[line width=\LP@normallines,fill=\LP@color]%
      (0,0) rectangle (1,1);}%
```

`\LP@Line` A generic command for drawing lines with options `double`, `color` and `linewidth`.

`\LP@Line[`⟨*options*⟩`]{`⟨*TikZ path*⟩`}`

```
\define@key{LP@Line}{double}[double]%
{%
  \def\LP@Line@double{double}%
}%
%
\define@key{LP@Line}{color}[LP@c@bridge]%
{%
  \def\LP@Line@linecolor{#1}%
}%
%
\define@key{LP@Line}{linewidth}[1mm]%
{%
  \def\LP@Line@linewidth{#1}%
}%
%
\newcommand*\LP@Line[2][]%
{%
  \def\LP@Line@double{}%
  \def\LP@Line@linecolor{LP@c@bridge}%
```

```
662  \def\LP@Line@linewidth{1mm}%
663  \setkeys{LP@Line}{#1}%
```

By default we draw a single 1mm wide line with color `LP@c@bridge`. But we can change thar with [⟨*options*⟩].

```
664  \LP@set@LP@scale{\LP@env@prefix}%
```

Then we can draw a single or double line based on the defined options or defaults.

```
665  \begin{pgfonlayer}{LPbackgroundtwo}%
666    \ifthenelse{\equal{\LP@Line@double}{double}}%
667    {\draw[double,double distance=\LP@Line@linewidth*\LP@scale,%
668           color=\LP@Line@linecolor,%
669           line width=\LP@Line@linewidth*\LP@scale] #2;}%
670    {\draw[color=\LP@Line@linecolor,%
671           line width=\LP@Line@linewidth*\LP@scale] #2;}%
672  \end{pgfonlayer}%
673 }%
```

`\LP@set@LP@scale`

```
674 \newcommand*\LP@set@LP@scale[1]%
675 {%
676   \expandafter\xdef\expandafter\LP@scale{\csname #1@scale\endcsname}%
677 }%
```

`\LP@set@LP@color`

```
678 \newcommand*\LP@set@LP@color[1]%
679 {%
680   \expandafter\xdef\expandafter\LP@color{\csname #1@color\endcsname}%
681 }%
```

`\LP@set@LP@rows`

```
682 \newcommand*\LP@set@LP@rows[1]%
683 {%
684   \expandafter\xdef\expandafter\LP@rows{\csname #1@rows\endcsname}%
685 }%
```

`\LP@set@LP@columns`

```
686 \newcommand*\LP@set@LP@columns[1]%
687 {%
688   \expandafter\xdef\expandafter\LP@columns{\csname #1@columns\endcsname}%
689 }%
```

`\LP@set@LP@cvoffset`

```
690 \newcommand*\LP@set@LP@cvoffset[1]%
691 {%
692   \expandafter\xdef\expandafter\LP@cvoffset{\csname #1@cvoffset\endcsname}%
693 }%
```

\LP@set@LP@bgcolor

```
694 \newcommand*\LP@set@LP@bgcolor[1]%
695 {%
696   \expandafter\xdef\expandafter\LP@bgcolor{\csname #1@bgcolor\endcsname}%
697 }%
```

\LP@set@LP@extracells

```
698 \newcommand*\LP@set@LP@extracells[1]%
699 {%
700   \expandafter\xdef\expandafter\LP@extracells{\csname #1@extracells\endcsname}%
701 }%
```

\LP@set@LP@fontsize

```
702 \newcommand*\LP@set@LP@fontsize[1]%
703 {%
704 %   can't expand \Large
705   \expandafter\gdef\expandafter\LP@fontsize{\csname #1@fontsize\endcsname}%
706 }%
```

\LP@set@env@prefix

```
707 \newcommand*\LP@set@env@prefix[1]%
708 {%
709   \gdef\LP@env@prefix{#1}%
710 }%
```

\LP@set@package

```
711 \newcommand*\LP@set@package[1]%
712 {%
713   \gdef\LP@package{#1}%
714 }%
```

\setgridlinestyle

```
715 \newcommand*\setgridlinestyle[1]%
716 {%
717   \def\LP@grid@linestyle{#1}%
718 }%
```

\setnormallinewidth

```
719 \newcommand*\setnormallinewidth[1]%
```

```
{%
  \def\LP@normallines{#1}%
}%
```

\setthicklinewidth

```
\newcommand*\setthicklinewidth[1]%
{%
  \def\LP@thicklines{#1}%
}%
```

\puzzlestrut

```
\newcommand*\puzzlestrut%
{%
  \LP@set@LP@rows{\LP@env@prefix}%
  \ifthenelse{\equal{\LP@package}{nonogram}}%
  {%
    \LP@set@LP@extracells{\LP@env@prefix}%
    \setcounter{LP@counti}{\LP@rows}% max row
    \stepcounter{LP@counti}%
    \addtocounter{LP@counti}{\LP@NG@extracells}%
    \draw[line width=\LP@thicklines, transparent]%
      (1,1) -- (1,\arabic{LP@counti});%
  }{}%
}%
```

\setTikZpreset

```
\newcommand*\setTikZpreset[1]%
{%
  \tikzset{%
    LPpreset/.style={#1}%
  }%
}%
%
\setTikZpreset{line cap=rect,line join=round}%
```

logicpuzzle

```
\newcommand*\LP@LP@init@prefix{LP@LP}%
\newcommand*\LP@LP@init@package{logicpuzzle}%
%
\LP@define@key{\LP@LP@init@prefix}{\LP@LP@init@package}%
{rows}{5}%
\LP@define@key{\LP@LP@init@prefix}{\LP@LP@init@package}%
{columns}{5}%
\LP@define@key{\LP@LP@init@prefix}{\LP@LP@init@package}%
{scale}{1}%
\LP@define@key{\LP@LP@init@prefix}{\LP@LP@init@package}%
{counterstyle}{none}%
```

```
\LP@define@key{\LP@LP@init@prefix}{\LP@LP@init@package}%
{color}{}%
\LP@define@key{\LP@LP@init@prefix}{\LP@LP@init@package}%
{bgcolor}{}%
\LP@define@key{\LP@LP@init@prefix}{\LP@LP@init@package}%
{width}{5.1cm}%
\LP@define@key{\LP@LP@init@prefix}{\LP@LP@init@package}%
{cvoffset}{-23pt}%
\LP@define@key{\LP@LP@init@prefix}{\LP@LP@init@package}%
{title}{}%
\LP@define@key{\LP@LP@init@prefix}{\LP@LP@init@package}%
{titleindent}{0cm}%
\LP@define@key{\LP@LP@init@prefix}{\LP@LP@init@package}%
{titlewidth}{5.1cm}%
\LP@define@choicekey@fontsize{\LP@LP@init@prefix}%
{\LP@LP@init@package}{Large}%
%
\ExecuteOptionsX{rows,columns,width,fontsize,scale,bgcolor%
,cvoffset,counterstyle,title,titleindent,titlewidth}%
%
\ProcessOptionsX\relax%
%
\let\logicpuzzlecell\setcell%
%
\newcommand*\logicpuzzlesetup[1]%
{%
  \setkeys{logicpuzzle.sty}{#1}%
}%
%
\newenvironment{logicpuzzle}[1][]%
{%
  \setkeys{logicpuzzle}{#1}%
  \LP@set@package{logicpuzzle}%
  \LP@set@env@prefix{LP@LP}%
  \setcounter{LP@rows}{\LP@LP@rows}%
  \setcounter{LP@columns}{\LP@LP@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@LP@width}%
    \ifthenelse{\equal{\LP@LP@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@LP@titleindent}%
    \parbox{\LP@LP@titlewidth}{\strut\LP@titleformat\LP@LP@title}%
    \vspace{3mm}\par}%
    \begin{tikzpicture}[scale=\LP@LP@scale]%
      \LP@drawbackground{1}{1}{\LP@LP@columns}{\LP@LP@rows}%
      {\LP@LP@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@LP@columns}{\LP@LP@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
```

```
    \LP@drawcounter{\LP@LP@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%
%
\RequirePackage{lpenv}%

⟨/package⟩
```

4.2 lpenv.sty

Here's just a flat copy of `lpenv.sty`! The code for the puzzles are more or less simple copy & paste. Take a look at section 1 for a detailed explanation of code needed for a puzzle environment.

```
⟨*lpenv⟩

%
% battleship environment and options
%
\newcommand*\LP@BS@init@prefix{LP@BS}%
\newcommand*\LP@BS@init@package{battleship}%
%
\LP@define@key{\LP@BS@init@prefix}{\LP@BS@init@package}{rows}{5}%
\LP@define@key{\LP@BS@init@prefix}{\LP@BS@init@package}{columns}{5}%
\LP@define@key{\LP@BS@init@prefix}{\LP@BS@init@package}{shipcolor}{green}%
\LP@define@key{\LP@BS@init@prefix}{\LP@BS@init@package}{scale}{1}%

\LP@define@key{\LP@BS@init@prefix}{\LP@BS@init@package}{counterstyle}{none}%

\LP@define@key{\LP@BS@init@prefix}{\LP@BS@init@package}{bgcolor}{}%
\LP@define@key{\LP@BS@init@prefix}{\LP@BS@init@package}{width}{6cm}%

\LP@define@key{\LP@BS@init@prefix}{\LP@BS@init@package}{cvoffset}{-23pt}%

\LP@define@key{\LP@BS@init@prefix}{\LP@BS@init@package}{title}{}%
\LP@define@key{\LP@BS@init@prefix}{\LP@BS@init@package}{titleindent}{0.75cm}%
\LP@define@key{\LP@BS@init@prefix}{\LP@BS@init@package}{titlewidth}{5.15cm}%
\LP@define@key{\LP@BS@init@prefix}{\LP@BS@init@package}{sbindent}{0.75cm}%
\LP@define@key{\LP@BS@init@prefix}{\LP@BS@init@package}{sbwidth}{5.15cm}%
\LP@define@key{\LP@BS@init@prefix}{\LP@BS@init@package}{sbshipscale}{1}%
\LP@define@choicekey@fontsize{\LP@BS@init@prefix}{\LP@BS@init@package}{{Large}}%
%
\gdef\LP@BS@shipbox{}%
\tikzstyle{island} = [fill=yellow!30,draw, decorate, decoration={random steps,segment len
%
\newcommand*\Ship{\tikz[scale=\LP@BS@scale]\draw[scale=.36,fill=\LP@BS@shipcolor] (0,0) c
\newcommand*\ShipC{\tikz[scale=\LP@BS@scale]\draw[scale=.36,fill=\LP@BS@shipcolor] (0,0)-
\newcommand*\ShipT{\tikz[scale=\LP@BS@scale]\draw[scale=.36,fill=\LP@BS@shipcolor](2,1)--
```

```
\newcommand*\ShipB{\tikz[scale=\LP@BS@scale]\draw[scale=.36,fill=\LP@BS@shipcolor](2,1)--
\newcommand*\ShipL{\tikz[scale=\LP@BS@scale]\draw[scale=.36,fill=\LP@BS@shipcolor](1,2)--
\newcommand*\ShipR{\tikz[scale=\LP@BS@scale]\draw[scale=.36,fill=\LP@BS@shipcolor](1,2)--
\newcommand*\Island{\tikz[scale=\LP@BS@scale]\draw[scale=.36,island] (0,0) rectangle (2,2
\newcommand*\Water{\tikz[scale=\LP@BS@scale]\draw[scale=.36,fill,blue!40] (1,1) circle (0
% versions for \shipbox without second scale
\newcommand*\@Ship{\tikz\draw[scale=.144,fill=\LP@BS@shipcolor] (0,0) circle (1);}%
\newcommand*\@ShipC{\tikz\draw[scale=.144,fill=\LP@BS@shipcolor] (0,0)--(0,2)--(2,2)--(2,
\newcommand*\@ShipT{\tikz\draw[scale=.144,fill=\LP@BS@shipcolor](2,1)--(2,0)--(0,0)--(0,1
\newcommand*\@ShipB{\tikz\draw[scale=.144,fill=\LP@BS@shipcolor](2,1)--(2,2)--(0,2)--(0,1
\newcommand*\@ShipL{\tikz\draw[scale=.144,fill=\LP@BS@shipcolor](1,2)--(2,2)--(2,0)--(1,0
\newcommand*\@ShipR{\tikz\draw[scale=.144,fill=\LP@BS@shipcolor](1,2)--(0,2)--(0,0)--(1,0
%
\newcommand*\LP@BS@printship[1]%
{%
  \ifthenelse{\equal{#1}{1}}{\scalebox{\LP@BS@sbshipscale}{\@Ship}\space\allowbreak}{}%
  \ifthenelse{\equal{#1}{2}}{\scalebox{\LP@BS@sbshipscale}{\@ShipL\,\@ShipR}\space\allowb
  \ifthenelse{\equal{#1}{3}}{\scalebox{\LP@BS@sbshipscale}{\@ShipL\,\@ShipC\,\@ShipR}\spa
  \ifthenelse{\equal{#1}{4}}{\scalebox{\LP@BS@sbshipscale}{\@ShipL\,\@ShipC\,\@ShipC\,\@S
  \ifthenelse{\equal{#1}{5}}{\scalebox{\LP@BS@sbshipscale}{\@ShipL\,\@ShipC\,\@ShipC\,\@S
  \ifthenelse{\equal{#1}{6}}{\scalebox{\LP@BS@sbshipscale}{\@ShipL\,\@ShipC\,\@ShipC\,\@S
  \ifthenelse{\equal{#1}{7}}{\scalebox{\LP@BS@sbshipscale}{\@ShipL\,\@ShipC\,\@ShipC\,\@S
  \ifthenelse{\equal{#1}{8}}{\scalebox{\LP@BS@sbshipscale}{\@ShipL\,\@ShipC\,\@ShipC\,\@S
  \ifthenelse{\equal{#1}{9}}{\scalebox{\LP@BS@sbshipscale}{\@ShipL\,\@ShipC\,\@ShipC\,\@S
  \ifthenelse{\equal{#1}{10}}{\scalebox{\LP@BS@sbshipscale}{\@ShipL\,\@ShipC\,\@ShipC\,\@
}%
%
\newcommand*\LP@BS@printshipbox[1]%
{%
%   no {#1}! to avoid expansion
  \foreach \LP@element in #1%
  {%
    \LP@BS@printship{\LP@element}%
  }%
}%
%
\newcommand*\shipH[1]%
{%
  \LP@setrowcontents{#1}{1}{\value{LP@rows}}%
}%
%
\newcommand*\shipV[1]%
{%
  \LP@setcolumncontents{#1}{0}{1}%
}%

\newcommand*\placesegment[3]%
{%
  \LP@ingrid{#1}{#2}{\LP@BS@columns}{\LP@BS@rows}{battleship}%
  \LP@G@setcellcontent[hcenter,vcenter]{#1}{#2}{#3}%
}%
```

```
%
\let\ship\placesegment%

\newcommand*\placeisland[2]%
{%
  \LP@ingrid{#1}{#2}{\LP@BS@columns}{\LP@BS@rows}{battleship}%
  \LP@G@setcellcontent[hcenter,vcenter]{#1}{#2}{\Island}%
}%

\newcommand*\placewater[2]%
{%
  \LP@ingrid{#1}{#2}{\LP@BS@columns}{\LP@BS@rows}{battleship}%
  \LP@G@setcellcontent[hcenter,vcenter]{#1}{#2}{\Water}%
}%
%
\newcommand*\shipbox[2][]%
{%
  \setkeys{shipbox}{#1}%
  \gdef\LP@BS@shipbox{#2}%
}%

\newcommand*\placeship[4]%
{%
  \setcounter{LP@counti}{#4}% length
  \ifnum\value{LP@counti}<1%
    \PackageError{battleship}%
                 {ship length < 1}%
                 {The length of your ship should be at least 1}%
  \fi%
  \ifnum\value{LP@counti}>10%
    \PackageError{battleship}%
                 {ship length > 10}%
                 {The supported max length of ships is 10!}%
  \fi%
  \ifnum\value{LP@counti}=1%
    \placesegment{#2}{#3}{\Ship}%
  \else%
    \setcounter{LP@whiledo@i}{\value{LP@counti}}%
    \addtocounter{LP@whiledo@i}{-2}% length of middle ship
    \ifthenelse{\equal{#1}{V}}%
    {%
      \placesegment{#2}{#3}{\ShipB}%
      \setcounter{LP@countii}{#3}%
      \whiledo{\value{LP@whiledo@i}>0}%
      {%
        \addtocounter{LP@whiledo@i}{-1}%
        \stepcounter{LP@countii}%
        \placesegment{#2}{\theLP@countii}{\ShipC}%
      }%
      \stepcounter{LP@countii}%
      \placesegment{#2}{\theLP@countii}{\ShipT}%
```

```
944     }%
945     {%
946       \ifthenelse{\equal{#1}{H}}%
947       {%
948         \placesegment{#2}{#3}{\ShipL}%
949         \setcounter{LP@counti}{#2}%
950         \whiledo{\value{LP@whiledo@i}>0}%
951         {%
952           \addtocounter{LP@whiledo@i}{-1}%
953           \stepcounter{LP@counti}%
954           \placesegment{\theLP@counti}{#3}{\ShipC}%
955         }%
956         \stepcounter{LP@counti}%
957         \placesegment{\theLP@counti}{#3}{\ShipR}%
958       }%
959       {\PackageError{battleship}%
960                     {invalid direction (H/V)}%
961                     {You can place your ship only\MessageBreak%
962                      horizontally (H) or vertically (V)!}%
963       }%
964     }%
965   \fi%
966 }%
967 %

968 \newcommand*\battleshipsetup[1]%
969 {%
970   \setkeys{battleship.sty}{#1}%
971 }%
972 %

973 \newcommand{\classicgame}[1]%
974 {%
975   \begin{center}%
976     \begin{battleship}[rows=10,columns=10,width=6.5cm,title=Me,sbindent=0.65cm,titleinden
977       \shipV{J,I,H,G,F,E,D,C,B,A}%
978       \shipH{1,2,3,4,5,6,7,8,9,10}%
979       \shipbox{#1}%
980     \end{battleship}%
981     \hspace{1cm}%
982     \begin{battleship}[rows=10,columns=10,width=6.5cm,title=Enemy,sbindent=0.65cm,titlein
983       \shipV{J,I,H,G,F,E,D,C,B,A}%
984       \shipH{1,2,3,4,5,6,7,8,9,10}%
985     \end{battleship}%
986   \end{center}%
987   \par\vspace{1cm}%
988   \begin{center}%
989     \begin{battleship}[rows=10,columns=10,width=6.5cm,title=Me,sbindent=0.65cm,titleinden
990       \shipV{J,I,H,G,F,E,D,C,B,A}%
991       \shipH{1,2,3,4,5,6,7,8,9,10}%
992       \shipbox{#1}%
```

```
    \end{battleship}%
    \hspace{1cm}%
    \begin{battleship}[rows=10,columns=10,width=6.5cm,title=Enemy,sbindent=0.65cm,titlein
      \shipV{J,I,H,G,F,E,D,C,B,A}%
      \shipH{1,2,3,4,5,6,7,8,9,10}%
    \end{battleship}%
  \end{center}%
}%
```

battleship

```
\newenvironment{battleship}[1][]%
{%
  \setkeys{battleship}{#1}%
  \LP@set@package{battleship}%
  \LP@set@env@prefix{LP@BS}%
  \shipbox{}% clear shipbox
  \setcounter{LP@rows}{\LP@BS@rows}%
  \setcounter{LP@columns}{\LP@BS@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@BS@width}%
    \ifthenelse{\equal{\LP@BS@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@BS@titleindent}\parbox{\LP@BS@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@BS@scale]%
      \LP@drawbackground{1}{1}{\LP@BS@columns}{\LP@BS@rows}{\LP@BS@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@BS@columns}{\LP@BS@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@BS@counterstyle}%
    \par\hbadness=10000\medskip\noindent\hspace{\LP@BS@sbindent}\begin{minipage}{\LP@BS@s
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%


%
% bokkusu environment and options
%
\newcommand*\LP@BK@init@prefix{LP@BK}%
\newcommand*\LP@BK@init@package{bokkusu}%
%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{rows}{5}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{columns}{5}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{scale}{1}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{counterstyle}{none}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{color}{black}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{bgcolor}{}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{width}{6.7cm}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{cvoffset}{-38pt}%
```

```
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{title}{}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{titleindent}{0.75cm}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{titlewidth}{5.85cm}%
\LP@define@choicekey@fontsize{\LP@BK@init@prefix}{\LP@BK@init@package}{Large}%
%
\let\valueH\LP@bottomrow%
\let\valueV\LP@leftcolumn%
\let\sumH\LP@toprow%
\let\sumV\LP@rightcolumn%
%
\newcommand*\bokkususetup[1]%
{%
  \setkeys{bokkusu.sty}{#1}%
}%
```

bokkusu

```
\newenvironment{bokkusu}[1][]%
{%
  \setkeys{bokkusu}{#1}%
  \LP@set@package{bokkusu}%
  \LP@set@env@prefix{LP@BK}%
  \setcounter{LP@rows}{\LP@BK@rows}%
  \setcounter{LP@columns}{\LP@BK@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@BK@width}%
    \ifthenelse{\equal{\LP@BK@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@BK@titleindent}\parbox{\LP@BK@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@BK@scale]%
      \LP@drawbackground{1}{1}{\LP@BK@columns}{\LP@BK@rows}{\LP@BK@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@BK@columns}{\LP@BK@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@BK@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% bridges environment and options
%
\newcommand*\LP@BG@init@prefix{LP@BG}%
\newcommand*\LP@BG@init@package{bridges}%
%
\LP@define@key{\LP@BG@init@prefix}{\LP@BG@init@package}{rows}{5}%
\LP@define@key{\LP@BG@init@prefix}{\LP@BG@init@package}{columns}{5}%
\LP@define@key{\LP@BG@init@prefix}{\LP@BG@init@package}{scale}{1}%
\LP@define@key{\LP@BG@init@prefix}{\LP@BG@init@package}{counterstyle}{none}%
\LP@define@key{\LP@BG@init@prefix}{\LP@BG@init@package}{grid}{dashed}%
```

```
\LP@define@key{\LP@BG@init@prefix}{\LP@BG@init@package}{bgcolor}{}%
\LP@define@key{\LP@BG@init@prefix}{\LP@BG@init@package}{color}{green}%
\LP@define@key{\LP@BG@init@prefix}{\LP@BG@init@package}{width}{6.1cm}%
\LP@define@key{\LP@BG@init@prefix}{\LP@BG@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@BG@init@prefix}{\LP@BG@init@package}{title}{}%
\LP@define@key{\LP@BG@init@prefix}{\LP@BG@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@BG@init@prefix}{\LP@BG@init@package}{titlewidth}{6.1cm}%
\LP@define@choicekey@fontsize{\LP@BG@init@prefix}{\LP@BG@init@package}{Large}%
%
\let\bridgescell\setcell%
%
\newcommand*\bridgessetup[1]%
{%
  \setkeys{bridges.sty}{#1}%
}%
```

`\bridgesrow` For the `bridges` environment, we need a special row command for drawing the islands with the numbers of bridges.

`\bridgesrow{`⟨*row*⟩`}{`⟨*csv list*⟩`}`

```
\newcommand*\bridgesrow[2]%
{%
  \setcounter{LP@counti}{1}%
  \setcounter{LP@countii}{#1}%
  \LP@set@LP@fontsize{\LP@env@prefix}%
  \LP@set@LP@color{\LP@env@prefix}%
  \foreach \LP@element in {#2}%
  {%
```

We loop through the list anf if `element` is not empty, we draw an island (circle filled with `bgcolor`) and typeset the number of bridges into to center of the circle.

```
    \ifthenelse{\equal{\LP@element}{}}%
    {}%
    {%
      \draw[fill=\LP@color] (\arabic{LP@counti},\arabic{LP@countii})%
        circle (0.5cm);%
      \node[scale=\LP@scale,font=\LP@fontsize] at%
        (\arabic{LP@counti},\arabic{LP@countii}){\LP@element};%
    }%
    \stepcounter{LP@counti}%
  };%
}%
```

`\bridgescolumn` And again, for columns!

```
\newcommand*\bridgescolumn[2]%
{%
  \setcounter{LP@counti}{#1}%
  \setcounter{LP@countii}{1}%
```

```
  \LP@set@LP@fontsize{\LP@env@prefix}%
  \LP@set@LP@color{\LP@env@prefix}%
  \foreach \LP@element in {#2}%
  {%
    \ifthenelse{\equal{\LP@element}{}}%
    {}%
    {%
      \draw[fill=\LP@color] (\arabic{LP@counti},\arabic{LP@countii})%
        circle (0.5cm);%
      \node[scale=\LP@scale,font=\LP@fontsize]%
        at (\arabic{LP@counti},\arabic{LP@countii})%
        {\LP@element};%
    }%
    \stepcounter{LP@countii}%
  };%
}%
```

`\bridge` For the `bridges` environment, we need to draw bridges. Keep in mind that you can influence the appearance of the brigde with the options `double`, `color` and `linewidth`.

`\bridge[`⟨*options*⟩`]{`⟨*Tikz path*⟩`}`

```
\newcommand*\bridge[2][]%
{%
  \LP@Line[#1]{#2}%
}%
```

`bridges`

```
\newenvironment{bridges}[1][]%
{%
  \setkeys{bridges}{#1}%
  \LP@set@package{bridges}%
  \LP@set@env@prefix{LP@BG}%
  \setcounter{LP@rows}{\LP@BG@rows}%
  \setcounter{LP@columns}{\LP@BG@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@BG@width}%
    \ifthenelse{\equal{\LP@BG@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@BG@titleindent}\parbox{\LP@BG@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@BG@scale]%
      \LP@drawbackground{1}{1}{\LP@BG@columns}{\LP@BG@rows}{\LP@BG@bgcolor}%
      \ifthenelse{\equal{\LP@BG@grid}{none}}%
      {}%
      {%
        \ifthenelse{\equal{\LP@BG@grid}{dashed}}%
        {%
          \setgridlinestyle{dashed}%
          \LP@drawgrid{1}{1}{\LP@BG@columns}{\LP@BG@rows}{1cm}%
```

```
        }%
        {%
          \LP@drawgrid{1}{1}{\LP@BG@columns}{\LP@BG@rows}{1cm}%
        }%
      }%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@BG@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% chaossudoku environment and options
%
\newcommand*\LP@CS@init@prefix{LP@CS}%
\newcommand*\LP@CS@init@package{chaossudoku}%
%
\LP@define@key{\LP@CS@init@prefix}{\LP@CS@init@package}{rows}{5}%
\LP@define@key{\LP@CS@init@prefix}{\LP@CS@init@package}{columns}{5}%
\LP@define@key{\LP@CS@init@prefix}{\LP@CS@init@package}{scale}{1}%
\LP@define@key{\LP@CS@init@prefix}{\LP@CS@init@package}{counterstyle}{none}%
\LP@define@key{\LP@CS@init@prefix}{\LP@CS@init@package}{bgcolor}{}%
\LP@define@key{\LP@CS@init@prefix}{\LP@CS@init@package}{width}{5.1cm}%
\LP@define@key{\LP@CS@init@prefix}{\LP@CS@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@CS@init@prefix}{\LP@CS@init@package}{title}{}%
\LP@define@key{\LP@CS@init@prefix}{\LP@CS@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@CS@init@prefix}{\LP@CS@init@package}{titlewidth}{5.1cm}%
\LP@define@choicekey@fontsize{\LP@CS@init@prefix}{\LP@CS@init@package}{Large}%
%
\let\chaossudokucell\setcell%
%
\newcommand*\chaossudokusetup[1]%
{%
  \setkeys{chaossudoku.sty}{#1}%
}%
```

chaossudoku

```
\newenvironment{chaossudoku}[1][]%
{%
  \setkeys{chaossudoku}{#1}%
  \LP@set@package{chaossudoku}%
  \LP@set@env@prefix{LP@CS}%
  \setcounter{LP@rows}{\LP@CS@rows}%
  \setcounter{LP@columns}{\LP@CS@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@CS@width}%
    \ifthenelse{\equal{\LP@CS@title}{}}%
    {\par\enspace\par}% empty
```

```
    {\enspace\par\noindent\hspace{\LP@CS@titleindent}\parbox{\LP@CS@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@CS@scale]%
      \LP@drawbackground{1}{1}{\LP@CS@columns}{\LP@CS@rows}{\LP@CS@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@CS@columns}{\LP@CS@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@CS@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% ddsudoku environment and options
%
\newcommand*\LP@DDS@init@prefix{LP@DDS}%
\newcommand*\LP@DDS@init@package{ddsudoku}%
%
\LP@define@key{\LP@DDS@init@prefix}{\LP@DDS@init@package}{rows}{5}%
\LP@define@key{\LP@DDS@init@prefix}{\LP@DDS@init@package}{columns}{5}%
\LP@define@key{\LP@DDS@init@prefix}{\LP@DDS@init@package}{scale}{1}%
\LP@define@key{\LP@DDS@init@prefix}{\LP@DDS@init@package}{counterstyle}{none}%
\LP@define@key{\LP@DDS@init@prefix}{\LP@DDS@init@package}{bgcolor}{}%
\LP@define@key{\LP@DDS@init@prefix}{\LP@DDS@init@package}{width}{5.1cm}%
\LP@define@key{\LP@DDS@init@prefix}{\LP@DDS@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@DDS@init@prefix}{\LP@DDS@init@package}{title}{}%
\LP@define@key{\LP@DDS@init@prefix}{\LP@DDS@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@DDS@init@prefix}{\LP@DDS@init@package}{titlewidth}{5.1cm}%
\LP@define@choicekey@fontsize{\LP@DDS@init@prefix}{\LP@DDS@init@package}{Large}%
%
\let\ddsudokucell\setcell%
%
\newcommand*\ddsudokusetup[1]%
{%
  \setkeys{ddsudoku.sty}{#1}%
}%
```

ddsudoku

```
\newenvironment{ddsudoku}[1][]%
{%
  \setkeys{ddsudoku}{#1}%
  \LP@set@package{ddsudoku}%
  \LP@set@env@prefix{LP@DDS}%
  \setcounter{LP@rows}{\LP@DDS@rows}%
  \setcounter{LP@columns}{\LP@DDS@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@DDS@width}%
    \ifthenelse{\equal{\LP@DDS@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@DDS@titleindent}\parbox{\LP@DDS@titlewidth}{\strut\
```

```
    \begin{tikzpicture}[LPpreset,scale=\LP@DDS@scale]%
      \LP@drawbackground{1}{1}{\LP@DDS@columns}{\LP@DDS@rows}{\LP@DDS@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@DDS@columns}{\LP@DDS@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@DDS@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% fourwinds environment and options
%
\newcommand*\LP@FW@init@prefix{LP@FW}%
\newcommand*\LP@FW@init@package{fourwinds}%
%
\LP@define@key{\LP@FW@init@prefix}{\LP@FW@init@package}{rows}{5}%
\LP@define@key{\LP@FW@init@prefix}{\LP@FW@init@package}{columns}{5}%
\LP@define@key{\LP@FW@init@prefix}{\LP@FW@init@package}{scale}{1}%
\LP@define@key{\LP@FW@init@prefix}{\LP@FW@init@package}{counterstyle}{none}%
\LP@define@key{\LP@FW@init@prefix}{\LP@FW@init@package}{bgcolor}{}%
\LP@define@key{\LP@FW@init@prefix}{\LP@FW@init@package}{color}{blue}%
\LP@define@key{\LP@FW@init@prefix}{\LP@FW@init@package}{width}{5.1cm}%
\LP@define@key{\LP@FW@init@prefix}{\LP@FW@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@FW@init@prefix}{\LP@FW@init@package}{title}{}%
\LP@define@key{\LP@FW@init@prefix}{\LP@FW@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@FW@init@prefix}{\LP@FW@init@package}{titlewidth}{5.1cm}%
\LP@define@choicekey@fontsize{\LP@FW@init@prefix}{\LP@FW@init@package}{Large}%
%
\newcommand*\fourwindscell[4]%
{%
  \LP@set@LP@fontsize{\LP@env@prefix}%
  \LP@set@LP@color{\LP@env@prefix}%
  \LP@set@LP@bgcolor{\LP@env@prefix}%
  \LP@set@LP@scale{\LP@env@prefix}%
  \ifthenelse{\equal{\LP@bgcolor}{}}%
  {\gdef\LP@c@bg@fw{white}}%
  {\gdef\LP@c@bg@fw{\LP@bgcolor}}%
  \foreach \LP@fw@dir/\LP@fw@length in {#4}%
  {%
    \begin{pgfonlayer}{LPbackgroundtwo}%
      \def\LP@rel@tikzpath{.5}%
      \draw[\LP@fw@linestyle,color=\LP@color,line width=.1cm*\LP@scale,shorten >=-3mm*\LP
        \xtikzpath{#1}{#2}{\LP@fw@dir/\LP@fw@length};%
    \end{pgfonlayer}%
  };%
  \begin{puzzlebackground}%
    \node[fill=\LP@c@bg@fw,font=\LP@fontsize] at (#1.5,#2.5) {#3};%
  \end{puzzlebackground}%
}%
%
```

```
\newcommand*\fourwindssetup[1]%
{%
  \setkeys{fourwinds.sty}{#1}%
}%
```

fourwinds

```
\newenvironment{fourwinds}[1][]%
{%
  \setkeys{fourwinds}{#1}%
  \LP@set@package{fourwinds}%
  \LP@set@env@prefix{LP@FW}%
  \setcounter{LP@rows}{\LP@FW@rows}%
  \setcounter{LP@columns}{\LP@FW@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@FW@width}%
    \ifthenelse{\equal{\LP@FW@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@FW@titleindent}\parbox{\LP@FW@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@FW@scale]%
      \LP@drawbackground{1}{1}{\LP@FW@columns}{\LP@FW@rows}{\LP@FW@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@FW@columns}{\LP@FW@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@FW@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% hakyuu environment and options
%
\newcommand*\LP@HY@init@prefix{LP@HY}%
\newcommand*\LP@HY@init@package{hakyuu}%
%
\LP@define@key{\LP@HY@init@prefix}{\LP@HY@init@package}{rows}{5}%
\LP@define@key{\LP@HY@init@prefix}{\LP@HY@init@package}{columns}{5}%
\LP@define@key{\LP@HY@init@prefix}{\LP@HY@init@package}{scale}{1}%
\LP@define@key{\LP@HY@init@prefix}{\LP@HY@init@package}{counterstyle}{none}%
\LP@define@key{\LP@HY@init@prefix}{\LP@HY@init@package}{bgcolor}{}%
\LP@define@key{\LP@HY@init@prefix}{\LP@HY@init@package}{width}{5.1cm}%
\LP@define@key{\LP@HY@init@prefix}{\LP@HY@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@HY@init@prefix}{\LP@HY@init@package}{title}{}%
\LP@define@key{\LP@HY@init@prefix}{\LP@HY@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@HY@init@prefix}{\LP@HY@init@package}{titlewidth}{5.1cm}%
\LP@define@choicekey@fontsize{\LP@HY@init@prefix}{\LP@HY@init@package}{Large}%
%
\let\hakyuucell\setcell%
%
\newcommand*\hakyuusetup[1]%
```

```
{%
  \setkeys{hakyuu.sty}{#1}%
}%
```

hakyuu

```
\newenvironment{hakyuu}[1][]%
{%
  \setkeys{hakyuu}{#1}%
  \LP@set@package{hakyuu}%
  \LP@set@env@prefix{LP@HY}%
  \setcounter{LP@rows}{\LP@HY@rows}%
  \setcounter{LP@columns}{\LP@HY@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@HY@width}%
    \ifthenelse{\equal{\LP@HY@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@HY@titleindent}\parbox{\LP@HY@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@HY@scale]%
      \LP@drawbackground{1}{1}{\LP@HY@columns}{\LP@HY@rows}{\LP@HY@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@HY@columns}{\LP@HY@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@HY@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% hitori environment and options
%
\newcommand*\LP@HT@init@prefix{LP@HT}%
\newcommand*\LP@HT@init@package{hitori}%
%
\LP@define@key{\LP@HT@init@prefix}{\LP@HT@init@package}{rows}{5}%
\LP@define@key{\LP@HT@init@prefix}{\LP@HT@init@package}{columns}{5}%
\LP@define@key{\LP@HT@init@prefix}{\LP@HT@init@package}{scale}{1}%
\LP@define@key{\LP@HT@init@prefix}{\LP@HT@init@package}{counterstyle}{none}%
\LP@define@key{\LP@HT@init@prefix}{\LP@HT@init@package}{bgcolor}{}%
\LP@define@key{\LP@HT@init@prefix}{\LP@HT@init@package}{width}{5.1cm}%
\LP@define@key{\LP@HT@init@prefix}{\LP@HT@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@HT@init@prefix}{\LP@HT@init@package}{title}{}%
\LP@define@key{\LP@HT@init@prefix}{\LP@HT@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@HT@init@prefix}{\LP@HT@init@package}{titlewidth}{5.1cm}%
\LP@define@choicekey@fontsize{\LP@HT@init@prefix}{\LP@HT@init@package}{Large}%
%
\let\hitoricell\setcell%
%
\newcommand*\hitorisetup[1]%
{%
```

```
1411   \setkeys{hitori.sty}{#1}%
1412 }%
```

hitori

```
1413 \newenvironment{hitori}[1][]%
1414 {%
1415   \setkeys{hitori}{#1}%
1416   \LP@set@package{hitori}%
1417   \LP@set@env@prefix{LP@HT}%
1418   \setcounter{LP@rows}{\LP@HT@rows}%
1419   \setcounter{LP@columns}{\LP@HT@columns}%
1420   \stepcounter{LP@rows}%
1421   \stepcounter{LP@columns}%
1422   \begin{minipage}[t]{\LP@HT@width}%
1423     \ifthenelse{\equal{\LP@HT@title}{}}%
1424     {\par\enspace\par}% empty
1425     {\enspace\par\noindent\hspace{\LP@HT@titleindent}\parbox{\LP@HT@titlewidth}{\strut\LP
1426     \begin{tikzpicture}[LPpreset,scale=\LP@HT@scale]%
1427       \LP@drawbackground{1}{1}{\LP@HT@columns}{\LP@HT@rows}{\LP@HT@bgcolor}%
1428       \LP@drawgrid{1}{1}{\LP@HT@columns}{\LP@HT@rows}{1cm}%
1429 }%
1430 {%
1431     \end{tikzpicture}%
1432     \LP@drawcounter{\LP@HT@counterstyle}%
1433     \stepcounter{LP@puzzlecounter}%
1434   \end{minipage}%
1435 }%

1436 %
1437 % kakuro environment and options
1438 %
1439 \newcommand*\LP@KKR@init@prefix{LP@KKR}%
1440 \newcommand*\LP@KKR@init@package{kakuro}%
1441 %
1442 \LP@define@key{\LP@KKR@init@prefix}{\LP@KKR@init@package}{rows}{5}%
1443 \LP@define@key{\LP@KKR@init@prefix}{\LP@KKR@init@package}{columns}{5}%
1444 \LP@define@key{\LP@KKR@init@prefix}{\LP@KKR@init@package}{scale}{1}%
1445 \LP@define@key{\LP@KKR@init@prefix}{\LP@KKR@init@package}{counterstyle}{none}%
1446 \LP@define@key{\LP@KKR@init@prefix}{\LP@KKR@init@package}{bgcolor}{}%
1447 \LP@define@key{\LP@KKR@init@prefix}{\LP@KKR@init@package}{color}{green}%
1448 \LP@define@key{\LP@KKR@init@prefix}{\LP@KKR@init@package}{width}{5.1cm}%
1449 \LP@define@key{\LP@KKR@init@prefix}{\LP@KKR@init@package}{cvoffset}{-23pt}%
1450 \LP@define@key{\LP@KKR@init@prefix}{\LP@KKR@init@package}{title}{}%
1451 \LP@define@key{\LP@KKR@init@prefix}{\LP@KKR@init@package}{titleindent}{0cm}%
1452 \LP@define@key{\LP@KKR@init@prefix}{\LP@KKR@init@package}{titlewidth}{5.1cm}%
1453 \LP@define@choicekey@fontsize{\LP@KKR@init@prefix}{\LP@KKR@init@package}{Large}%
1454 %
1455 \newif\ifLP@KKR@solution\LP@KKR@solutionfalse%
1456 %
1457 \define@choicekey*{kakuro.sty}{solution}[\LP@KKR@solution\nr]{true,false}[true]%
1458 {%
```

```
  \ifcase\nr\relax%
    \renewcommand*\LP@KKR@solution{true}%
    \LP@KKR@solutiontrue%
  \or%
    \renewcommand*\LP@KKR@solution{false}%
    \LP@KKR@solutionfalse%
  \fi%
}%
%
\define@choicekey*{kakuro}{solution}[\LP@KKR@solution\nr]{true,false}[true]%
{%
  \ifcase\nr\relax%
    \renewcommand*\LP@KKR@solution{true}%
    \LP@KKR@solutiontrue%
  \or%
    \renewcommand*\LP@KKR@solution{false}%
    \LP@KKR@solutionfalse%
  \fi%
}%
%
\let\kakurocell\setcell%
%
\newcommand*\kakurosetup[1]%
{%
  \setkeys{kakuro.sty}{#1}%
}%
```

`\LP@KKR@print@element` For the `kakuro` environment, we need to typeset numbers and special Kakuro cells (`\KKR`).

`\LP@KKR@print@element{`⟨*element*⟩`}`

```
\def\LP@KKR@print@element#1%
{%
```

First, we test if {⟨*element*⟩} is a number and typeset {⟨*element*⟩} into the grid cell, if we are in solution mode. Otherwise, we execute the `\KKR` command.

```
  \if!\ifnum9<1#1!\else_\fi%
    \ifLP@KKR@solution%
      \setcell{\arabic{LP@counti}}{\arabic{LP@countii}}{#1}%
    \else%
    \fi%
  \else%
    #1%
  \fi%
}%
```

`\kakurorow` For the `kakuro` environment, we need special row and column commands, which can also handle the `\KKR` commands.

```
\newcommand*\kakurorow[2]%
```

```
{%
  \setcounter{LP@counti}{1}%
  \setcounter{LP@countii}{#1}%
  \foreach \LP@element in {#2}%
  {%
    \LP@KKR@print@element{\LP@element}%
    \stepcounter{LP@counti}%
  };%
}%
```

\kakurocolumn

```
\newcommand*\kakurocolumn[2]%
{%
  \setcounter{LP@counti}{#1}%
  \setcounter{LP@countii}{1}%
  \foreach \LP@element in {#2}%
  {%
    \LP@KKR@print@element{\LP@element}%
    \stepcounter{LP@countii}%
  };%
}%
```

\KKR For the kakuro environment, we need to draw complex kakuro cells. They consist of a diagonally divided cell with the sums of the cells below and right of the current cell. They also have a special background color.

\KKR{⟨*vertical sum*⟩}{⟨*horizontal sum*⟩}

```
\newcommand*\KKR[2]%
{%
```

First of all, we copy scale, bgcolor and color.

```
  \LP@set@LP@scale{\LP@env@prefix}%
  \LP@set@LP@bgcolor{\LP@env@prefix}%
  \LP@set@LP@color{\LP@env@prefix}%
```

If bgcolor is undefined, we assume white.

```
  \ifthenelse{\equal{\LP@bgcolor}{}}%
  {\gdef\LP@sr@bgcolor{white}}%
  {\gdef\LP@sr@bgcolor{\LP@bgcolor}}%
```

To get unique node names, we step our unique counter.

```
  \stepcounter{LP@counter@unique}%
```

In a first step, we draw a rectangular helper node in bgcolor color on the LPdump layer.

```
  \begin{pgfonlayer}{LPdump}%
    \node [shape=rectangle,inner sep=0pt] (A_\theLP@counter@unique)%
```

```
      at (\arabic{LP@counti}\LP@Pfive,\arabic{LP@countii}\LP@Pfive)%
      {\tikz\draw[scale=\LP@scale,color=\LP@sr@bgcolor]%
        (0.08,0.1) rectangle (0.92,0.9);};%
  \end{pgfonlayer}%
```

In the second step, we fill the cell with `color` color and draw a diagonal line.

```
  \begin{puzzlebackground}%
    \fill[color=\LP@color]%
      (\arabic{LP@counti},\arabic{LP@countii}) rectangle ++(1,1);%
    \draw[line width=\LP@normallines]%
      (\arabic{LP@counti},\arabic{LP@countii}) -- ++(0,1) -- ++(1,-1);%
  \end{puzzlebackground}%
```

Finally, we use the corners of the helper node to place the sums.

```
  \node [shape=rectangle,inner sep=0pt,anchor=south west,%
         scale=\LP@scale,font=\small]%
           at (A_\theLP@counter@unique.south west) {#1};%
  \node [shape=rectangle,inner sep=0pt,anchor=north east,%
         scale=\LP@scale,font=\small]%
           at (A_\theLP@counter@unique.north east) {#2};%
}%
```

`\Black` Sometimes, we need black cells.

```
\newcommand*\Black%
{%
  \LP@set@LP@scale{\LP@env@prefix}%
  \begin{puzzlebackground}%
    \fill[color=black]%
      (\arabic{LP@counti},\arabic{LP@countii}) rectangle ++(1,1);%
  \end{puzzlebackground}%
}%
```

`kakuro`

```
\newenvironment{kakuro}[1][]%
{%
  \setkeys{kakuro}{#1}%
  \LP@set@package{kakuro}%
  \LP@set@env@prefix{LP@KKR}%
  \setcounter{LP@rows}{\LP@KKR@rows}%
  \setcounter{LP@columns}{\LP@KKR@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@KKR@width}%
    \ifthenelse{\equal{\LP@KKR@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@KKR@titleindent}\parbox{\LP@KKR@titlewidth}{\strut\
    \begin{tikzpicture}[LPpreset,scale=\LP@KKR@scale]%
      \LP@drawbackground{1}{1}{\LP@KKR@columns}{\LP@KKR@rows}{\LP@KKR@bgcolor}%
```

```
      \LP@drawgrid{1}{1}{\LP@KKR@columns}{\LP@KKR@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@KKR@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% kendoku environment and options
%
\newcommand*\LP@KD@init@prefix{LP@KD}%
\newcommand*\LP@KD@init@package{kendoku}%
%
\LP@define@key{\LP@KD@init@prefix}{\LP@KD@init@package}{rows}{5}%
\LP@define@key{\LP@KD@init@prefix}{\LP@KD@init@package}{columns}{5}%
\LP@define@key{\LP@KD@init@prefix}{\LP@KD@init@package}{scale}{1}%
\LP@define@key{\LP@KD@init@prefix}{\LP@KD@init@package}{counterstyle}{none}%
\LP@define@key{\LP@KD@init@prefix}{\LP@KD@init@package}{bgcolor}{}%
\LP@define@key{\LP@KD@init@prefix}{\LP@KD@init@package}{width}{5.1cm}%
\LP@define@key{\LP@KD@init@prefix}{\LP@KD@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@KD@init@prefix}{\LP@KD@init@package}{title}{}%
\LP@define@key{\LP@KD@init@prefix}{\LP@KD@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@KD@init@prefix}{\LP@KD@init@package}{titlewidth}{5.1cm}%
\LP@define@choicekey@fontsize{\LP@KD@init@prefix}{\LP@KD@init@package}{Large}%
%
\let\kendokucell\setcell%
%
\newcommand*\kendokusetup[1]%
{%
  \setkeys{kendoku.sty}{#1}%
}%
```

kendoku

```
\newenvironment{kendoku}[1][]%
{%
  \setkeys{kendoku}{#1}%
  \LP@set@package{kendoku}%
  \LP@set@env@prefix{LP@KD}%
  \setcounter{LP@rows}{\LP@KD@rows}%
  \setcounter{LP@columns}{\LP@KD@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@KD@width}%
    \ifthenelse{\equal{\LP@KD@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@KD@titleindent}\parbox{\LP@KD@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@KD@scale]%
      \LP@drawbackground{1}{1}{\LP@KD@columns}{\LP@KD@rows}{\LP@KD@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@KD@columns}{\LP@KD@rows}{1cm}%
```

```
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@KD@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% killersudoku environment and options
%
\newcommand*\LP@KSDK@init@prefix{LP@KSDK}%
\newcommand*\LP@KSDK@init@package{killersudoku}%
%
\LP@define@key{\LP@KSDK@init@prefix}{\LP@KSDK@init@package}{rows}{5}%
\LP@define@key{\LP@KSDK@init@prefix}{\LP@KSDK@init@package}{columns}{5}%
\LP@define@key{\LP@KSDK@init@prefix}{\LP@KSDK@init@package}{scale}{1}%
\LP@define@key{\LP@KSDK@init@prefix}{\LP@KSDK@init@package}{counterstyle}{none}%
\LP@define@key{\LP@KSDK@init@prefix}{\LP@KSDK@init@package}{bgcolor}{}%
\LP@define@key{\LP@KSDK@init@prefix}{\LP@KSDK@init@package}{width}{5.1cm}%
\LP@define@key{\LP@KSDK@init@prefix}{\LP@KSDK@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@KSDK@init@prefix}{\LP@KSDK@init@package}{title}{}%
\LP@define@key{\LP@KSDK@init@prefix}{\LP@KSDK@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@KSDK@init@prefix}{\LP@KSDK@init@package}{titlewidth}{5.1cm}%
\LP@define@choicekey@fontsize{\LP@KSDK@init@prefix}{\LP@KSDK@init@package}{Large}%
%
\let\killersudokucell\setcell%
%
\newcommand*\killersudokusetup[1]%
{%
  \setkeys{killersudoku.sty}{#1}%
}%
```

killersudoku

```
\newenvironment{killersudoku}[1][]%
{%
  \setkeys{killersudoku}{#1}%
  \LP@set@package{killersudoku}%
  \LP@set@env@prefix{LP@KSDK}%
  \setcounter{LP@rows}{\LP@KSDK@rows}%
  \setcounter{LP@columns}{\LP@KSDK@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@KSDK@width}%
    \ifthenelse{\equal{\LP@KSDK@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@KSDK@titleindent}\parbox{\LP@KSDK@titlewidth}{\stru
    \begin{tikzpicture}[LPpreset,scale=\LP@KSDK@scale]%
      \LP@drawbackground{1}{1}{\LP@KSDK@columns}{\LP@KSDK@rows}{\LP@KSDK@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@KSDK@columns}{\LP@KSDK@rows}{1cm}%
}%
```

```
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@KSDK@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% laserbeam environment and options
%
\newcommand*\LP@LB@init@prefix{LP@LB}%
\newcommand*\LP@LB@init@package{laserbeam}%
%
\LP@define@key{\LP@LB@init@prefix}{\LP@LB@init@package}{rows}{5}%
\LP@define@key{\LP@LB@init@prefix}{\LP@LB@init@package}{columns}{5}%
\LP@define@key{\LP@LB@init@prefix}{\LP@LB@init@package}{scale}{1}%
\LP@define@key{\LP@LB@init@prefix}{\LP@LB@init@package}{counterstyle}{none}%
\LP@define@key{\LP@LB@init@prefix}{\LP@LB@init@package}{bgcolor}{}%
\LP@define@key{\LP@LB@init@prefix}{\LP@LB@init@package}{width}{6.5cm}%
\LP@define@key{\LP@LB@init@prefix}{\LP@LB@init@package}{cvoffset}{-38pt}%
\LP@define@key{\LP@LB@init@prefix}{\LP@LB@init@package}{title}{}%
\LP@define@key{\LP@LB@init@prefix}{\LP@LB@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@LB@init@prefix}{\LP@LB@init@package}{titlewidth}{6.5cm}%
\LP@define@choicekey@fontsize{\LP@LB@init@prefix}{\LP@LB@init@package}{Large}%
%
\let\laserH\LP@toprow%
\let\laserV\LP@leftcolumn%
\let\mirrorH\LP@bottomrow@edge%
\let\mirrorV\LP@rightcolumn@edge%
%
\newcommand*\laser[2][red]%
{%
%   avoid little laser peaks outside grid
  \clip (1,1) rectangle (\arabic{LP@columns},\arabic{LP@rows});%
  \framearea{#1}{#2}%
}%
%
\newcommand*\laserbeamsetup[1]%
{%
  \setkeys{laserbeam.sty}{#1}%
}%
```

`\placecross` Place a cross in the bottom left corner of the grid cell.

`\placecross{`⟨*column*⟩`}{`⟨*row*⟩`}`

```
\newcommand*\placecross[2]%
{%
  \LP@G@setcellcontent{#1}{#2}{\LP@Cross}%
}%
```

`\placemirror` Place a mirror in the bottom left corner of the grid cell.

`\placemirror{`⟨*column*⟩`}{`⟨*row*⟩`}`

```
\newcommand*\placemirror[3]%
{%
  \LP@G@setcellcontent{#1}{#2}{\LP@Mirror{#3}}%
}%
```

`\placearrow` Place an arrow in the bottom left corner of the grid cell. `{`⟨*direction*⟩`}` may be: `RightUp`, `LeftUp`, `LeftDown` or `RightDown`

`\placearrow{`⟨*column*⟩`}{`⟨*row*⟩`}{`⟨*direction*⟩`}`

```
\newcommand*\placearrow[3]%
{%
  \LP@G@setcellcontent{#1}{#2}{\LP@Arrow{#3}}%
}%
```

`\LP@Arrow` We define a Ti*k*Z picture for an arrow in four directions: `RightUp`, `LeftUp`, `LeftDown` or `RightDown`

```
\newcommand*\LP@Arrow[1]%
{%
  \LP@set@LP@scale{\LP@env@prefix}%
  \def\LP@rotate{45}%
  \ifthenelse{\equal{#1}{LeftUp}}{\def\LP@rotate{135}}{}%
  \ifthenelse{\equal{#1}{LeftDown}}{\def\LP@rotate{225}}{}%
  \ifthenelse{\equal{#1}{RightDown}}{\def\LP@rotate{315}}{}%
```

We define the line width of the arrow base on `scale`. Line widths are absolut and not influenced by a `scale` factor of the picture.

```
  \pgfmathsetmacro{\LPlinewidth}{3pt*\LP@scale}%
  \tikz\draw[->,line width=\LPlinewidth,%
             rotate=\LP@rotate,scale=\LP@scale]%
              (0.1,.5) -- (0.9,.5);%
}%
```

`\LP@Cross` We define a Ti*k*Z picture of a cross.

```
\newcommand*\LP@Cross%
{%
  \LP@set@LP@scale{\LP@env@prefix}%
  \begin{tikzpicture}%
```

We ensure that the cross is on top of all elements on the `main` layer.

```
    \begin{pgfonlayer}{LPforeground}%
      \pgfmathsetmacro{\LPlinewidth}{3pt*\LP@scale}%
```

For drawing the cross we defined a line width. Now we can draw the cross with the predefined color `LP@c@cross`.

```
      \draw[line width=\LPlinewidth,scale=\LP@scale,color=LP@c@cross]%
```

```
1734        (.35,.35) -- (0.65,.65) -- (.5,.5) -- (.65,.35)-- (.35,.65);%
1735    \end{pgfonlayer}%
1736  \end{tikzpicture}%
1737 }%
```

`\LP@Mirror` We define a Ti*k*Z picture for a mirror.

`\LP@Mirror{`⟨*direction*⟩`}`

```
1738 \newcommand*\LP@Mirror[1]%
1739 {%
```

Based on {⟨*direction*⟩}, we define the angle of the mirror.

```
1740   \def\LP@rotate{0}%
1741   \ifthenelse{\equal{#1}{V}}{\def\LP@rotate{90}}{}%
1742   \LP@set@LP@scale{\LP@env@prefix}%
1743   \begin{tikzpicture}%
```

Mirrors should be on to of everything, therefore we draw them on the `LPforegroundtwo` layer and use the predefined color `LP@c@mirror`.

```
1744     \begin{pgfonlayer}{LPforegroundtwo}%
1745       \pgfmathsetmacro{\LPlinewidth}{3pt*\LP@scale}%
1746       \draw[line width=\LPlinewidth,scale=\LP@scale,%
1747             color=LP@c@mirror,rotate=\LP@rotate]%
1748              (.35,.5) -- (0.65,.5);%
1749     \end{pgfonlayer}%
1750   \end{tikzpicture}%
1751 }%
```

`laserbeam`

```
1752 \newenvironment{laserbeam}[1][]%
1753 {%
1754   \setkeys{laserbeam}{#1}%
1755   \LP@set@package{laserbeam}%
1756   \LP@set@env@prefix{LP@LB}%
1757   \setcounter{LP@rows}{\LP@LB@rows}%
1758   \setcounter{LP@columns}{\LP@LB@columns}%
1759   \stepcounter{LP@rows}%
1760   \stepcounter{LP@columns}%
1761   \begin{minipage}[t]{\LP@LB@width}%
1762     \ifthenelse{\equal{\LP@LB@title}{}}%
1763     {\par\enspace\par}% empty
1764     {\enspace\par\noindent\hspace{\LP@LB@titleindent}\parbox{\LP@LB@titlewidth}{\strut\LP
1765     \begin{tikzpicture}[LPpreset,scale=\LP@LB@scale]%
1766       \LP@drawbackground{1}{1}{\LP@LB@columns}{\LP@LB@rows}{\LP@LB@bgcolor}%
1767       \LP@drawgrid{1}{1}{\LP@LB@columns}{\LP@LB@rows}{1cm}%
1768 }%
1769 {%
1770     \end{tikzpicture}%
1771     \LP@drawcounter{\LP@LB@counterstyle}%
```

```
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% lpsudoku environment and options
%
\newcommand*\LP@SDK@init@prefix{LP@SDK}%
\newcommand*\LP@SDK@init@package{lpsudoku}%
%
\LP@define@key{\LP@SDK@init@prefix}{\LP@SDK@init@package}{rows}{9}%
\LP@define@key{\LP@SDK@init@prefix}{\LP@SDK@init@package}{columns}{9}%
\LP@define@key{\LP@SDK@init@prefix}{\LP@SDK@init@package}{scale}{1}%
\LP@define@key{\LP@SDK@init@prefix}{\LP@SDK@init@package}{counterstyle}{none}%
\LP@define@key{\LP@SDK@init@prefix}{\LP@SDK@init@package}{bgcolor}{}%
\LP@define@key{\LP@SDK@init@prefix}{\LP@SDK@init@package}{width}{9.1cm}%
\LP@define@key{\LP@SDK@init@prefix}{\LP@SDK@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@SDK@init@prefix}{\LP@SDK@init@package}{title}{}%
\LP@define@key{\LP@SDK@init@prefix}{\LP@SDK@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@SDK@init@prefix}{\LP@SDK@init@package}{titlewidth}{9.1cm}%
\LP@define@choicekey@fontsize{\LP@SDK@init@prefix}{\LP@SDK@init@package}{Large}%
%
\let\lpsudokucell\setcell%
%
\newcommand*\lpsudokusetup[1]%
{%
  \setkeys{lpsudoku.sty}{#1}%
}%
```

`\LP@drawsudokugrid` For the `lpsudoku` and `skyscrapers` environments, we need to add the typical thick Sudoku lines to the standard grid.

```
\newcommand*\LP@drawsudokugrid%
{%
  \draw[line width=\LP@thicklines,draw opacity=\LP@draw@opacity]%
       (1,1) -- (1,10);%
  \draw[line width=\LP@thicklines,draw opacity=\LP@draw@opacity]%
       (4,1) -- (4,10);%
  \draw[line width=\LP@thicklines,draw opacity=\LP@draw@opacity]%
       (7,1) -- (7,10);%
  \draw[line width=\LP@thicklines,draw opacity=\LP@draw@opacity]%
       (10,1) -- (10,10);%
  \draw[line width=\LP@thicklines,draw opacity=\LP@draw@opacity]%
       (1,1) -- (10,1);%
  \draw[line width=\LP@thicklines,draw opacity=\LP@draw@opacity]%
       (1,4) -- (10,4);%
  \draw[line width=\LP@thicklines,draw opacity=\LP@draw@opacity]%
       (1,7) -- (10,7);%
  \draw[line width=\LP@thicklines,draw opacity=\LP@draw@opacity]%
       (1,10) -- (10,10);%
}%
```

\LP@magnetsgrid For the magnets environment, we need to add some lines to the standard grid.

```
\newcommand*\LP@magnetsgrid%
{%
```

First, we copy the values of fontsize, columns and rows from the environment we are in at the moment. Then, we step columns and rows to get the upper right grid coordinate.

```
  \LP@set@LP@fontsize{\LP@env@prefix}%
  \LP@set@LP@columns{\LP@env@prefix}%
  \LP@set@LP@rows{\LP@env@prefix}%
  \setcounter{LP@counti}{\LP@columns}% max column
  \setcounter{LP@countii}{\LP@rows}% max row
  \stepcounter{LP@counti}%
  \stepcounter{LP@countii}%
```

Now, we can draw the additional lines and the + and - signs.

```
  \draw[step=1cm,line width=\LP@normallines]%
       (-1,1) grid (1,\arabic{LP@countii});%
  \draw[step=1cm,line width=\LP@normallines]%
       (1,\arabic{LP@countii}) grid ++(\LP@columns,2);%
  \draw[line width=\LP@normallines]%
       (0,\arabic{LP@countii}) -- ++(0,1) -- ++(1,0);%
  \draw[line width=\LP@thicklines]%
       (-1,1) rectangle (1,\arabic{LP@countii});%
  \draw[line width=\LP@thicklines]%
       (1,\arabic{LP@countii}) rectangle ++(\LP@columns,2);%
  \draw[line width=\LP@thicklines]%
       (1,\arabic{LP@countii}) rectangle ++(-2,2);%
  \node[font=\LP@fontsize\bfseries] at (0.5,\arabic{LP@counti}.5)%
       {$-$};%
  \stepcounter{LP@counti}%
  \node[font=\LP@fontsize\bfseries] at (-0.5,\arabic{LP@counti}.5)%
       {$+$};%
}%
```

lpsudoku

```
\newenvironment{lpsudoku}[1][]%
{%
  \setkeys{lpsudoku}{#1}%
  \LP@set@package{lpsudoku}%
  \LP@set@env@prefix{LP@SDK}%
  \setcounter{LP@rows}{\LP@SDK@rows}%
  \setcounter{LP@columns}{\LP@SDK@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@SDK@width}%
    \ifthenelse{\equal{\LP@SDK@title}{}}%
    {\par\enspace\par}% empty
```

```
    {\enspace\par\noindent\hspace{\LP@SDK@titleindent}\parbox{\LP@SDK@titlewidth}{\strut\
    \begin{tikzpicture}[LPpreset,scale=\LP@SDK@scale]%
      \LP@drawbackground{1}{1}{\LP@SDK@columns}{\LP@SDK@rows}{\LP@SDK@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@SDK@columns}{\LP@SDK@rows}{1cm}%
      \LP@drawsudokugrid%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@SDK@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% magiclabyrinth environment and options
%
\newcommand*\LP@ML@init@prefix{LP@ML}%
\newcommand*\LP@ML@init@package{magiclabyrinth}%
%
\LP@define@key{\LP@ML@init@prefix}{\LP@ML@init@package}{rows}{5}%
\LP@define@key{\LP@ML@init@prefix}{\LP@ML@init@package}{columns}{5}%
\LP@define@key{\LP@ML@init@prefix}{\LP@ML@init@package}{scale}{1}%
\LP@define@key{\LP@ML@init@prefix}{\LP@ML@init@package}{counterstyle}{none}%
\LP@define@key{\LP@ML@init@prefix}{\LP@ML@init@package}{bgcolor}{}%
\LP@define@key{\LP@ML@init@prefix}{\LP@ML@init@package}{width}{5.1cm}%
\LP@define@key{\LP@ML@init@prefix}{\LP@ML@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@ML@init@prefix}{\LP@ML@init@package}{title}{}%
\LP@define@key{\LP@ML@init@prefix}{\LP@ML@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@ML@init@prefix}{\LP@ML@init@package}{titlewidth}{5.1cm}%
\LP@define@choicekey@fontsize{\LP@ML@init@prefix}{\LP@ML@init@package}{Large}%
%
\let\magiclabyrinthcell\setcell%
%
\newcommand*\magiclabyrinthsetup[1]%
{%
  \setkeys{magiclabyrinth.sty}{#1}%
}%
```

\mlline

```
\newcommand*\mlline[1]%
{%
  \draw[color=black, line width=\LP@thicklines] #1;%
}%
```

\MasyuW

```
\newcommand*\MasyuW%
{%
  \begin{tikzpicture}[scale=0.7*\LP@scale]%
    \draw[color=black,fill=white] (0.5,0.5) circle (0.5cm);%
  \end{tikzpicture}%
```

```
}%
```

magiclabyrinth

```
\newenvironment{magiclabyrinth}[1][]%
{%
  \setkeys{magiclabyrinth}{#1}%
  \LP@set@package{magiclabyrinth}%
  \LP@set@env@prefix{LP@ML}%
  \setcounter{LP@rows}{\LP@ML@rows}%
  \setcounter{LP@columns}{\LP@ML@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@ML@width}%
    \ifthenelse{\equal{\LP@ML@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@ML@titleindent}\parbox{\LP@ML@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@ML@scale]%
      \LP@drawbackground{1}{1}{\LP@ML@columns}{\LP@ML@rows}{\LP@ML@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@ML@columns}{\LP@ML@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@ML@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% magnets environment and options
%
\newcommand*\LP@MN@init@prefix{LP@MN}%
\newcommand*\LP@MN@init@package{magnets}%
%
\LP@define@key{\LP@MN@init@prefix}{\LP@MN@init@package}{rows}{6}%
\LP@define@key{\LP@MN@init@prefix}{\LP@MN@init@package}{columns}{6}%
\LP@define@key{\LP@MN@init@prefix}{\LP@MN@init@package}{scale}{1}%
\LP@define@key{\LP@MN@init@prefix}{\LP@MN@init@package}{counterstyle}{none}%
\LP@define@key{\LP@MN@init@prefix}{\LP@MN@init@package}{bgcolor}{}%
\LP@define@key{\LP@MN@init@prefix}{\LP@MN@init@package}{width}{8.1cm}%
\LP@define@key{\LP@MN@init@prefix}{\LP@MN@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@MN@init@prefix}{\LP@MN@init@package}{title}{}%
\LP@define@key{\LP@MN@init@prefix}{\LP@MN@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@MN@init@prefix}{\LP@MN@init@package}{titlewidth}{8.1cm}%
\LP@define@choicekey@fontsize{\LP@MN@init@prefix}{\LP@MN@init@package}{Large}%
%
\let\magnetscell\setcell%
%
\let\minusH\LP@toprow%
\let\minusV\LP@leftcolumn%
%
\newcommand*\plusV[1]%
```

```
{%
  \LP@set@LP@fontsize{\LP@env@prefix}%
  \setcounter{LP@counti}{1}%
  \foreach \LP@element in{#1}%
  {%
    \node at (-0.5,\arabic{LP@counti}\LP@Pfive){\LP@fontsize\LP@element};%
    \stepcounter{LP@counti}%
  };%
}%
%
\newcommand*\plusH[1]%
{%
  \LP@set@LP@fontsize{\LP@env@prefix}%
  \LP@set@LP@rows{\LP@env@prefix}%
  \setcounter{LP@counti}{1}%
  \setcounter{LP@countii}{\LP@rows}%
  \addtocounter{LP@countii}{2}%
  \foreach \LP@element in{#1}%
  {%
    \node at (\arabic{LP@counti}\LP@Pfive,\arabic{LP@countii}\LP@Pfive){\LP@fontsize\LP@e
    \stepcounter{LP@counti}%
  };%
}%
%
\newcommand*\magnetssetup[1]%
{%
  \setkeys{magnets.sty}{#1}%
}%
```

\PMH

```
\newcommand*\PMH[1]%
{%
  \LP@set@LP@fontsize{\LP@env@prefix}%
  \foreach \LP@fe@column/\LP@fe@row in {#1}%
  {%
    \setcounter{LP@counti}{\LP@fe@column}%
    \setcounter{LP@countii}{\LP@fe@row}%
    \draw[line width=\LP@thicklines,fill=white]%
      (\arabic{LP@counti},\arabic{LP@countii}) rectangle ++(2,1);%
    \node[font=\LP@fontsize\bfseries]%
      at (\arabic{LP@counti}.5,\arabic{LP@countii}.5) {$+$};%
    \stepcounter{LP@counti}%
    \node[font=\LP@fontsize\bfseries]%
      at (\arabic{LP@counti}.5,\arabic{LP@countii}.5) {$-$};%
  };%
}%
```

\MPH

```
\newcommand*\MPH[1]%
{%
```

```
1996   \LP@set@LP@fontsize{\LP@env@prefix}%
1997   \foreach \LP@fe@column/\LP@fe@row in {#1}%
1998   {%
1999     \setcounter{LP@counti}{\LP@fe@column}%
2000     \setcounter{LP@countii}{\LP@fe@row}%
2001     \draw[line width=\LP@thicklines,fill=white]%
2002       (\arabic{LP@counti},\arabic{LP@countii}) rectangle ++(2,1);%
2003     \node[font=\LP@fontsize\bfseries]%
2004       at (\arabic{LP@counti}.5,\arabic{LP@countii}.5) {$-$};%
2005     \stepcounter{LP@counti}%
2006     \node[font=\LP@fontsize\bfseries]%
2007       at (\arabic{LP@counti}.5,\arabic{LP@countii}.5) {$+$};%
2008   };%
2009 }%
```

\PMV

```
2010 \newcommand*\PMV[1]%
2011 {%
2012   \LP@set@LP@fontsize{\LP@env@prefix}%
2013   \foreach \LP@fe@column/\LP@fe@row in {#1}%
2014   {%
2015     \setcounter{LP@counti}{\LP@fe@column}%
2016     \setcounter{LP@countii}{\LP@fe@row}%
2017     \draw[line width=\LP@thicklines,fill=white]%
2018       (\arabic{LP@counti},\arabic{LP@countii}) rectangle ++(1,2);%
2019     \node[font=\LP@fontsize\bfseries]%
2020       at (\arabic{LP@counti}.5,\arabic{LP@countii}.5) {$+$};%
2021     \stepcounter{LP@countii}%
2022     \node[font=\LP@fontsize\bfseries]%
2023       at (\arabic{LP@counti}.5,\arabic{LP@countii}.5) {$-$};%
2024   };%
2025 }%
```

\MPV

```
2026 \newcommand*\MPV[1]%
2027 {%
2028   \LP@set@LP@fontsize{\LP@env@prefix}%
2029   \foreach \LP@fe@column/\LP@fe@row in {#1}%
2030   {%
2031     \setcounter{LP@counti}{\LP@fe@column}%
2032     \setcounter{LP@countii}{\LP@fe@row}%
2033     \draw[line width=\LP@thicklines,fill=white]%
2034       (\arabic{LP@counti},\arabic{LP@countii}) rectangle ++(1,2);%
2035     \node[font=\LP@fontsize\bfseries]%
2036       at (\arabic{LP@counti}.5,\arabic{LP@countii}.5) {$-$};%
2037     \stepcounter{LP@countii}%
2038     \node[font=\LP@fontsize\bfseries]%
2039       at (\arabic{LP@counti}.5,\arabic{LP@countii}.5) {$+$};%
2040   };%
2041 }%
```

\magnetsH

```
\newcommand*\magnetsH[1]%
{%
  \foreach \LP@fe@column/\LP@fe@row in {#1}%
  {%
    \draw[line width=\LP@thicklines,fill=white]%
      (\LP@fe@column,\LP@fe@row) rectangle ++(2,1);%
  };%
}%
```

\magnetsV

```
\newcommand*\magnetsV[1]%
{%
  \foreach \LP@fe@column/\LP@fe@row in {#1}%
  {%
    \draw[line width=\LP@thicklines,fill=white]%
      (\LP@fe@column,\LP@fe@row) rectangle ++(1,2);%
  };%
}%
```

magnets

```
\newenvironment{magnets}[1][]%
{%
  \setkeys{magnets}{#1}%
  \LP@set@package{magnets}%
  \LP@set@env@prefix{LP@MN}%
  \setcounter{LP@rows}{\LP@MN@rows}%
  \setcounter{LP@columns}{\LP@MN@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@MN@width}%
    \ifthenelse{\equal{\LP@MN@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@MN@titleindent}\parbox{\LP@MN@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@MN@scale]%
      \LP@drawbackground{1}{1}{\LP@MN@columns}{\LP@MN@rows}{\LP@MN@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@MN@columns}{\LP@MN@rows}{1cm}%
      \LP@magnetsgrid%
      \framepuzzle%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@MN@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% masyu environment and options
```

```
%
\newcommand*\LP@MY@init@prefix{LP@MY}%
\newcommand*\LP@MY@init@package{masyu}%
%
\LP@define@key{\LP@MY@init@prefix}{\LP@MY@init@package}{rows}{5}%
\LP@define@key{\LP@MY@init@prefix}{\LP@MY@init@package}{columns}{5}%
\LP@define@key{\LP@MY@init@prefix}{\LP@MY@init@package}{scale}{1}%
\LP@define@key{\LP@MY@init@prefix}{\LP@MY@init@package}{counterstyle}{none}%
\LP@define@key{\LP@MY@init@prefix}{\LP@MY@init@package}{color}{green}%
\LP@define@key{\LP@MY@init@prefix}{\LP@MY@init@package}{bgcolor}{}%
\LP@define@key{\LP@MY@init@prefix}{\LP@MY@init@package}{width}{5.1cm}%
\LP@define@key{\LP@MY@init@prefix}{\LP@MY@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@MY@init@prefix}{\LP@MY@init@package}{title}{}%
\LP@define@key{\LP@MY@init@prefix}{\LP@MY@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@MY@init@prefix}{\LP@MY@init@package}{titlewidth}{5.1cm}%
\LP@define@choicekey@fontsize{\LP@MY@init@prefix}{\LP@MY@init@package}{Large}%
%
\let\masyucell\setcell%
%
\newcommand*\masyusetup[1]%
{%
  \setkeys{masyu.sty}{#1}%
}%
```

\MasyuB

```
\newcommand*\MasyuB%
{%
  \begin{tikzpicture}[scale=0.7*\LP@scale]%
    \fill[color=black] (0.5,0.5) circle (0.5cm);%
  \end{tikzpicture}%
}%
```

\masyuline

```
\newcommand*\masyuline[1]%
{%
  \LP@set@LP@color{\LP@env@prefix}%
  \begin{puzzleforeground}%
    \bgroup%
      \def\LP@rel@tikzpath{.5}%
      \draw[color=\LP@color, line width=\LP@thicklines] #1;%
    \egroup%
  \end{puzzleforeground}%
}%
```

masyu

```
\newenvironment{masyu}[1][]%
{%
  \setkeys{masyu}{#1}%
```

```
  \LP@set@package{masyu}%
  \LP@set@env@prefix{LP@MY}%
  \setcounter{LP@rows}{\LP@MY@rows}%
  \setcounter{LP@columns}{\LP@MY@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@MY@width}%
    \ifthenelse{\equal{\LP@MY@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@MY@titleindent}\parbox{\LP@MY@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@MY@scale]%
      \LP@drawbackground{1}{1}{\LP@MY@columns}{\LP@MY@rows}{\LP@MY@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@MY@columns}{\LP@MY@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@MY@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% minesweeper environment and options
%
\newcommand*\LP@MS@init@prefix{LP@MS}%
\newcommand*\LP@MS@init@package{minesweeper}%
%
\LP@define@key{\LP@MS@init@prefix}{\LP@MS@init@package}{rows}{5}%
\LP@define@key{\LP@MS@init@prefix}{\LP@MS@init@package}{columns}{5}%
\LP@define@key{\LP@MS@init@prefix}{\LP@MS@init@package}{scale}{1}%
\LP@define@key{\LP@MS@init@prefix}{\LP@MS@init@package}{counterstyle}{none}%
\LP@define@key{\LP@MS@init@prefix}{\LP@MS@init@package}{bgcolor}{}%
\LP@define@key{\LP@MS@init@prefix}{\LP@MS@init@package}{width}{5.1cm}%
\LP@define@key{\LP@MS@init@prefix}{\LP@MS@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@MS@init@prefix}{\LP@MS@init@package}{title}{}%
\LP@define@key{\LP@MS@init@prefix}{\LP@MS@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@MS@init@prefix}{\LP@MS@init@package}{titlewidth}{5.1cm}%
\LP@define@choicekey@fontsize{\LP@MS@init@prefix}{\LP@MS@init@package}{Large}%
%
\let\minesweepercell\setcell%
%
\newcommand*\minesweepersetup[1]%
{%
  \setkeys{minesweeper.sty}{#1}%
}%
```

`\LP@Mine` We define a Ti*k*Z picture for a mine.

```
\newcommand*\LP@Mine%
{%
  \begin{tikzpicture}[scale=\LP@scale]%
```

We simply draw a shaded ball with four spikes.

```
    \fill[color=black] (.2,.5) -- (.5,.6) -- (.5,.4) -- cycle;%
    \fill[color=black] (.8,.5) -- (.5,.6) -- (.5,.4) -- cycle;%
    \fill[color=black] (.5,.8) -- (.4,.5) -- (.6,.5) -- cycle;%
    \fill[color=black] (.5,.2) -- (.4,.5) -- (.6,.5) -- cycle;%
    \shade[ball color=black] (.5,.5) circle (.225cm);%
  \end{tikzpicture}%
}%
```

`\Mine` A user command for mines.

```
\let\Mine\LP@Mine%
```

`minesweeper`

```
\newenvironment{minesweeper}[1][]%
{%
  \setkeys{minesweeper}{#1}%
  \LP@set@package{minesweeper}%
  \LP@set@env@prefix{LP@MS}%
  \setcounter{LP@rows}{\LP@MS@rows}%
  \setcounter{LP@columns}{\LP@MS@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@MS@width}%
    \ifthenelse{\equal{\LP@MS@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@MS@titleindent}\parbox{\LP@MS@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@MS@scale]%
      \LP@drawbackground{1}{1}{\LP@MS@columns}{\LP@MS@rows}{\LP@MS@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@MS@columns}{\LP@MS@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@MS@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% nonogram environment and options
%
\newcommand*\LP@NG@init@prefix{LP@NG}%
\newcommand*\LP@NG@init@package{nonogram}%
\LP@define@key{\LP@NG@init@prefix}{\LP@NG@init@package}{rows}{5}%
\LP@define@key{\LP@NG@init@prefix}{\LP@NG@init@package}{columns}{5}%
\LP@define@key{\LP@NG@init@prefix}{\LP@NG@init@package}{extracells}{5}%
\LP@define@key{\LP@NG@init@prefix}{\LP@NG@init@package}{helplines}{5}%
\LP@define@key{\LP@NG@init@prefix}{\LP@NG@init@package}{solution}{false}%
\LP@define@key{\LP@NG@init@prefix}{\LP@NG@init@package}{scale}{1}%
\LP@define@key{\LP@NG@init@prefix}{\LP@NG@init@package}{counterstyle}{none}%
```

```
\LP@define@key{\LP@NG@init@prefix}{\LP@NG@init@package}{bgcolor}{}%
\LP@define@key{\LP@NG@init@prefix}{\LP@NG@init@package}{color}{black}%
\LP@define@key{\LP@NG@init@prefix}{\LP@NG@init@package}{width}{5.1cm}%
\LP@define@key{\LP@NG@init@prefix}{\LP@NG@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@NG@init@prefix}{\LP@NG@init@package}{title}{}%
\LP@define@key{\LP@NG@init@prefix}{\LP@NG@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@NG@init@prefix}{\LP@NG@init@package}{titlewidth}{5.1cm}%
\LP@define@choicekey@fontsize{\LP@NG@init@prefix}{\LP@NG@init@package}{Large}%
%
```

\nonogramrow

```
\newcommand*\nonogramrow[2]%
{%
  \foreach \LP@fe@column/\LP@fe@length in {#2}%
  {%
    \setcounter{LP@whiledo@i}{\LP@fe@length}%
    \setcounter{LP@countiii}{\LP@fe@column}%
    \whiledo{\value{LP@whiledo@i}>0}%
    {%
      \fillcell{\arabic{LP@countiii}}{#1}%
      \addtocounter{LP@countiii}{1}%
      \addtocounter{LP@whiledo@i}{-1}%
    }%
  };%
}%
```

\nonogramcolumn

```
\newcommand*\nonogramcolumn[2]%
{%
  \foreach \LP@fe@row/\LP@fe@length in {#2}%
  {%
    \setcounter{LP@whiledo@i}{\LP@fe@length}%
    \setcounter{LP@countiii}{\LP@fe@row}%
    \whiledo{\value{LP@whiledo@i}>0}%
    {%
      \fillcell{#1}{\arabic{LP@countiii}}%
      \addtocounter{LP@countiii}{1}%
      \addtocounter{LP@whiledo@i}{-1}%
    }%
  };%
}%
```

\nonogramV

```
\newcommand*\nonogramV[1]%
{%
  \LP@set@LP@fontsize{\LP@env@prefix}%
  \setcounter{LP@whiledo@i}{0}%
  \foreach \LP@line in {#1}%
```

```
2259   {%
2260     \setcounter{LP@countiii}{0}%
2261     \addtocounter{LP@whiledo@i}{1}%
2262     \foreach \LP@element in \LP@line%
2263     {%
2264       \ifthenelse{\value{LP@countiii}=0}%
2265       {%
2266         {\LP@fontsize\node at (0.5,\arabic{LP@whiledo@i}.5){\LP@element};}%
2267       }%
2268       {%
2269         \ifthenelse{\value{LP@countiii}=-1}%
2270         {%
2271           {\LP@fontsize\node at (-0.5,\arabic{LP@whiledo@i}.5){\LP@element};}%
2272         }%
2273         {%
2274           \addtocounter{LP@countiii}{1}%
2275           {\LP@fontsize\node at%
2276             (\arabic{LP@countiii}.5,\arabic{LP@whiledo@i}.5){\LP@element};}%
2277           \addtocounter{LP@countiii}{-1}%
2278         }%
2279       }%
2280       \addtocounter{LP@countiii}{-1}%
2281     }%
2282   };%
2283 }%
```

\nonogramH

```
2284 \newcommand*\nonogramH[1]%
2285 {%
2286   \setcounter{LP@whiledo@i}{0}%
2287   \foreach \LP@line in {#1}%
2288   {%
2289     \setcounter{LP@countiii}{\LP@rows}%
2290     \addtocounter{LP@countiii}{1}%
2291     \addtocounter{LP@whiledo@i}{1}%
2292     \foreach \LP@element in \LP@line%
2293     {%
2294       {\LP@fontsize\node at%
2295         (\arabic{LP@whiledo@i}.5,\arabic{LP@countiii}.5){\LP@element};}%
2296       \addtocounter{LP@countiii}{1}%
2297     }%
2298   };%
2299 }%
```

\LP@nonogramgrid

```
2300 \newcommand*\LP@nonogramgrid%
2301 {%
2302   \LP@set@LP@columns{\LP@env@prefix}%
2303   \LP@set@LP@rows{\LP@env@prefix}%
2304   \setcounter{LP@counti}{\LP@columns}% max column
```

```
\setcounter{LP@countii}{\LP@rows}% max row
\setcounter{LP@countiii}{\LP@NG@extracells}%
\stepcounter{LP@counti}%
\stepcounter{LP@countii}%
\addtocounter{LP@countiii}{-1}%
\setcounter{LP@whiledo@i}{1}%
\setcounter{LP@whiledo@ii}{\LP@NG@extracells}%
\addtocounter{LP@whiledo@ii}{\arabic{LP@countii}}%
\setcounter{LP@countiii}{\arabic{LP@countii}}%
\addtocounter{LP@countiii}{\LP@NG@extracells}%
\addtocounter{LP@counti}{1}%
\whiledo{\value{LP@whiledo@i}<\value{LP@counti}}%
{%
  \draw[line width=\LP@normallines]%
    (\arabic{LP@whiledo@i},1) --%
    (\arabic{LP@whiledo@i},\arabic{LP@countiii});%
  \addtocounter{LP@whiledo@i}{1}%
}%
\addtocounter{LP@counti}{-1}%
\setcounter{LP@whiledo@i}{1}%
\setcounter{LP@whiledo@ii}{-\LP@NG@extracells}%
\addtocounter{LP@whiledo@ii}{1}%
\addtocounter{LP@countii}{1}%
\whiledo{\value{LP@whiledo@i}<\value{LP@countii}}%
{%
  \draw[line width=\LP@normallines]%
    (\arabic{LP@whiledo@ii},\arabic{LP@whiledo@i}) --%
    (\arabic{LP@counti},\arabic{LP@whiledo@i});%
  \addtocounter{LP@whiledo@i}{1}%
}%
\addtocounter{LP@countii}{-1}%
\setcounter{LP@countiii}{\LP@NG@helplines}%
\ifthenelse{\arabic{LP@countiii}>0}%
{%
  \setcounter{LP@whiledo@i}{1}%
  \addtocounter{LP@whiledo@i}{\LP@NG@helplines}%
  \setcounter{LP@whiledo@ii}{\LP@NG@extracells}%
  \addtocounter{LP@whiledo@ii}{\arabic{LP@countii}}%
  \setcounter{LP@countiii}{\arabic{LP@countii}}%
  \addtocounter{LP@countiii}{\LP@NG@extracells}%
  \whiledo{\value{LP@whiledo@i}<\value{LP@counti}}%
  {%
    \draw[line width=\LP@thicklines]%
      (\arabic{LP@whiledo@i},1) --%
      (\arabic{LP@whiledo@i},\arabic{LP@countiii});%
    \addtocounter{LP@whiledo@i}{\LP@NG@helplines}%
  }%
  \draw[line width=\LP@thicklines] (1,1) -- (1,\arabic{LP@countiii});%
  \draw[line width=\LP@thicklines]%
  (\arabic{LP@counti},1) --%
  (\arabic{LP@counti},\arabic{LP@countiii});%
```

```
    \setcounter{LP@whiledo@i}{1}%
    \addtocounter{LP@whiledo@i}{\LP@NG@helplines}%
    \setcounter{LP@whiledo@ii}{-\LP@NG@extracells}%
    \addtocounter{LP@whiledo@ii}{1}%
    \whiledo{\value{LP@whiledo@i}<\value{LP@countii}}%
    {%
      \draw[line width=\LP@thicklines]%
        (\arabic{LP@whiledo@ii},\arabic{LP@whiledo@i}) --%
        (\arabic{LP@counti},\arabic{LP@whiledo@i});%
      \addtocounter{LP@whiledo@i}{\LP@NG@helplines}%
    }%
    \draw[line width=\LP@thicklines]%
      (\arabic{LP@whiledo@ii},1) -- (\arabic{LP@counti},1);%
    \draw[line width=\LP@thicklines]%
      (\arabic{LP@whiledo@ii},\arabic{LP@countii}) --%
      (\arabic{LP@counti},\arabic{LP@countii});%
  }%
  {}%
}%
```

\nonogramsetup

```
\newcommand*\nonogramsetup[1]%
{%
  \setkeys{nonogram.sty}{#1}%
}%
```

nonogram

```
\newenvironment{nonogram}[1][]%
{%
  \setkeys{nonogram}{#1}%
  \LP@set@package{nonogram}%
  \LP@set@env@prefix{LP@NG}%
  \setcounter{LP@rows}{\LP@NG@rows}%
  \setcounter{LP@columns}{\LP@NG@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@NG@width}%
    \ifthenelse{\equal{\LP@NG@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@NG@titleindent}\parbox{\LP@NG@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@NG@scale]%
      \LP@drawbackground{1}{1}{\LP@NG@columns}{\LP@NG@rows}{\LP@NG@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@NG@columns}{\LP@NG@rows}{1cm}%
      \framepuzzle%
      \ifthenelse{\equal{\LP@NG@solution}{false}}%
      {%
        \LP@nonogramgrid%
      }%
      {}%
}%
```

```
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@NG@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% numberlink environment and options
%
\newcommand*\LP@NL@init@prefix{LP@NL}%
\newcommand*\LP@NL@init@package{numberlink}%
%
\LP@define@key{\LP@NL@init@prefix}{\LP@NL@init@package}{rows}{5}%
\LP@define@key{\LP@NL@init@prefix}{\LP@NL@init@package}{columns}{5}%
\LP@define@key{\LP@NL@init@prefix}{\LP@NL@init@package}{scale}{1}%
\LP@define@key{\LP@NL@init@prefix}{\LP@NL@init@package}{counterstyle}{none}%
\LP@define@key{\LP@NL@init@prefix}{\LP@NL@init@package}{bgcolor}{}%
\LP@define@key{\LP@NL@init@prefix}{\LP@NL@init@package}{color}{red}%
\LP@define@key{\LP@NL@init@prefix}{\LP@NL@init@package}{width}{5.1cm}%
\LP@define@key{\LP@NL@init@prefix}{\LP@NL@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@NL@init@prefix}{\LP@NL@init@package}{title}{}%
\LP@define@key{\LP@NL@init@prefix}{\LP@NL@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@NL@init@prefix}{\LP@NL@init@package}{titlewidth}{5.1cm}%
\LP@define@choicekey@fontsize{\LP@NL@init@prefix}{\LP@NL@init@package}{Large}%
%
\newcommand*\numberlinkcell[3]%
{%
  \fourwindscell{#1}{#2}{#3}{}%
}%
%
%
\newcommand*\numberlinksetup[1]%
{%
  \setkeys{numberlink.sty}{#1}%
}%
```

\link

```
\newcommand*\link[2][]%
{%
  \LP@set@LP@color{\LP@env@prefix}%
  \bgroup%
    \def\LP@rel@tikzpath{.5}%
    \LP@Line[color=\LP@color,#1]{#2}%
  \egroup%
}%
```

numberlink

```
\newenvironment{numberlink}[1][]%
{%
```

```
  \setkeys{numberlink}{#1}%
  \LP@set@package{numberlink}%
  \LP@set@env@prefix{LP@NL}%
  \setcounter{LP@rows}{\LP@NL@rows}%
  \setcounter{LP@columns}{\LP@NL@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@NL@width}%
    \ifthenelse{\equal{\LP@NL@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@NL@titleindent}\parbox{\LP@NL@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@NL@scale]%
      \LP@drawbackground{1}{1}{\LP@NL@columns}{\LP@NL@rows}{\LP@NL@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@NL@columns}{\LP@NL@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@NL@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% resuko environment and options
%
\newcommand*\LP@RSK@init@prefix{LP@RSK}%
\newcommand*\LP@RSK@init@package{resuko}%
%
\LP@define@key{\LP@RSK@init@prefix}{\LP@RSK@init@package}{rows}{5}%
\LP@define@key{\LP@RSK@init@prefix}{\LP@RSK@init@package}{columns}{5}%
\LP@define@key{\LP@RSK@init@prefix}{\LP@RSK@init@package}{scale}{1}%
\LP@define@key{\LP@RSK@init@prefix}{\LP@RSK@init@package}{counterstyle}{none}%
\LP@define@key{\LP@RSK@init@prefix}{\LP@RSK@init@package}{bgcolor}{}%
\LP@define@key{\LP@RSK@init@prefix}{\LP@RSK@init@package}{color}{blue}%
\LP@define@key{\LP@RSK@init@prefix}{\LP@RSK@init@package}{width}{5.1cm}%
\LP@define@key{\LP@RSK@init@prefix}{\LP@RSK@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@RSK@init@prefix}{\LP@RSK@init@package}{title}{}%
\LP@define@key{\LP@RSK@init@prefix}{\LP@RSK@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@RSK@init@prefix}{\LP@RSK@init@package}{titlewidth}{5.1cm}%
\LP@define@choicekey@fontsize{\LP@RSK@init@prefix}{\LP@RSK@init@package}{Large}%
%
\let\resukocell\setcell%
%
\newcommand*\resukosetup[1]%
{%
  \setkeys{resuko.sty}{#1}%
}%
```

`\LP@trackC` For the `resuko` environment, we need to draw differnt tiles of track segments. First, we start with a curve from the bottom to the left.

```
\newcommand*\LP@trackC%
```

```
{%
  \LP@set@LP@scale{\LP@env@prefix}%
  \begin{tikzpicture}[scale=\LP@scale]%
    \pgfsetcornersarced{\pgfpoint{4mm*\LP@scale}{4mm*\LP@scale}}%
    \draw[color=LP@c@track, line width=.3cm*\LP@scale]%
      (0,.5)--(.5,.5)--(.5,0);%
  \end{tikzpicture}%
}%
```

`\LP@G@trackC` A generic command for drawing track curves with a rotation `[`⟨*angle*⟩`]`.

`\LP@G@trackC[`⟨*angle*⟩`]`

```
\newcommand*\LP@G@trackC[1][0]%
{%
  \LP@set@LP@scale{\LP@env@prefix}%
```

We draw a Ti*k*Z picture and apply a rotation.

```
  \begin{tikzpicture}[scale=\LP@scale,rotate=#1]%
```

We want tiles fitting into a cell, so we clip the picture. Furthermore, we want rounded corners for the race track, of course!

```
    \clip (0,0) rectangle (1,1);%
    \pgfsetcornersarced{\pgfpoint{4mm*\LP@scale}{4mm*\LP@scale}}%
```

Then we draw a track curve from the bottom to the left. Rotation applies for the complete picture!

```
    \draw[color=LP@c@track, line width=.3cm*\LP@scale]%
      (0,.5)--(.5,.5)--(.5,0);%
  \end{tikzpicture}%
}%
```

`\CurveBL`

```
\let\CurveBL\LP@G@trackC%
```

`\CurveBR`

```
\newcommand*\CurveBR{\LP@G@trackC[90]}%
```

`\CurveTR`

```
\newcommand*\CurveTR{\LP@G@trackC[180]}%
```

`\CurveTL`

```
\newcommand*\CurveTL{\LP@G@trackC[270]}%
```

`\LP@trackS` Of course, we also need straights.

```
\newcommand*\LP@trackS%
{%
  \LP@set@LP@scale{\LP@env@prefix}%
  \begin{tikzpicture}[scale=\LP@scale]%
    \draw[draw opacity=0] (.2,0) -- (.8,1);%
    \draw[color=LP@c@track, line width=.3cm*\LP@scale]%
      (.5,0)--(.5,1);%
  \end{tikzpicture}%
}%
```

`\LP@G@trackS` We need horizontal and vertical straights, therefore we allow an rotate angle.

`\LP@G@trackS[`⟨*angle*⟩`]`

```
\newcommand*\LP@G@trackS[1][0]%
{%
  \LP@set@LP@scale{\LP@env@prefix}%
  \begin{tikzpicture}[scale=\LP@scale,rotate=#1]%
    \clip (0,0) rectangle (1,1);%
    \draw[color=LP@c@track, line width=.3cm*\LP@scale]%
      (.5,0)--(.5,1);%
  \end{tikzpicture}%
}%
```

`\StraightV`

```
\newcommand*\StraightV{\LP@G@trackS}%
```

`\StraightH`

```
\newcommand*\StraightH{\LP@G@trackS[90]}%
```

`\Straight`

```
\let\Straight\StraightV%
```

`\LP@trackCR` Crossings are needed as well!

```
\newcommand*\LP@trackCR%
{%
  \LP@set@LP@scale{\LP@env@prefix}%
  \begin{tikzpicture}[scale=\LP@scale]%
    \draw[color=LP@c@track, line width=.3cm*\LP@scale]%
      (0,0.5)--(1,.5);%
    \draw[color=LP@c@track, line width=.3cm*\LP@scale]%
      (0.5,0)--(.5,.2);%
    \draw[color=LP@c@track, line width=.3cm*\LP@scale]%
      (0.5,1)--(.5,.8);%
  \end{tikzpicture}%
}%
```

\LP@G@trackCR Same game again, we need also rotated versions.

```
\newcommand*\LP@G@trackCR[1][0]%
{%
  \LP@set@LP@scale{\LP@env@prefix}%
  \begin{tikzpicture}[scale=\LP@scale,rotate=#1]%
    \clip (0,0) rectangle (1,1);%
    \draw[color=LP@c@track, line width=.3cm*\LP@scale]%
      (0,0.5)--(1,.5);%
    \draw[color=LP@c@track, line width=.3cm*\LP@scale]%
      (0.5,0)--(.5,.2);%
    \draw[color=LP@c@track, line width=.3cm*\LP@scale]%
      (0.5,1)--(.5,.8);%
  \end{tikzpicture}%
}%
%
\newcommand*\CrossH{\LP@G@trackCR}%
\newcommand*\CrossV{\LP@G@trackCR[90]}%
\let\Cross\CrossH%
```

\LP@graveltrap We also need a gravel trap.

```
\newcommand*\LP@graveltrap%
{%
  \LP@set@LP@scale{\LP@env@prefix}%
  \begin{tikzpicture}[scale=\LP@scale]%
    \clip (0,0) rectangle (1,1);%
    \fill[color=LP@c@track] (.5,.5) circle (.1cm);%
  \end{tikzpicture}%
}%
%
\let\Graveltrap\LP@graveltrap%
```

\parkinglot And a parking lot!

```
\newcommand*\parkinglot[2]%
{%
  \LP@set@LP@scale{\LP@env@prefix}%
  \LP@set@LP@color{\LP@env@prefix}%
  \LP@set@LP@bgcolor{\LP@env@prefix}%
  \LP@set@LP@fontsize{\LP@env@prefix}%
  \ifthenelse{\equal{\LP@color}{}}%
  {\gdef\LP@c@parkinglot{black}}%
  {\gdef\LP@c@parkinglot{\LP@color}}%
  \ifthenelse{\equal{\LP@bgcolor}{}}%
  {\gdef\LP@c@bg@parkinglot{white}}%
  {\gdef\LP@c@bg@parkinglot{\LP@bgcolor}}%
  \setcounter{LP@counti}{#1}%
  \setcounter{LP@countii}{#2}%
  \stepcounter{LP@counti}%
  \stepcounter{LP@countii}%
  \draw[color=\LP@c@parkinglot,line width=\LP@normallines,%
```

```
2593        fill=\LP@c@bg@parkinglot] (#1,#2) rectangle ++(2,2);%
2594   \node[color=\LP@c@parkinglot,font=\sffamily\bfseries\LP@fontsize]%
2595     at (\arabic{LP@counti},\arabic{LP@countii}){P};%
2596 }%
```

`\pitlane` And finally, the pitlane!

```
2597 \newcommand*\pitlane[3]%
2598 {%
2599   \LP@set@LP@scale{\LP@env@prefix}%
2600   \LP@set@LP@color{\LP@env@prefix}%
2601   \ifthenelse{\equal{\LP@color}{}}%
2602   {\gdef\LP@c@pitlane{black}}%
2603   {\gdef\LP@c@pitlane{\LP@color}}%
2604   \ifthenelse{\equal{#3}{V}}%
2605   {\draw[color=black,line width=\LP@normallines,fill=\LP@c@pitlane]%
2606     (#1,#2) rectangle ++(1,4);}%
2607   {\draw[color=black,line width=\LP@normallines,fill=\LP@c@pitlane]%
2608     (#1,#2) rectangle ++(4,1);}%
2609 }%
```

`\LP@trackline` This macro provides the "box" placed next to the grid, showing the straights, curves and crossings, which are needed in that line!

`\LP@trackline{`⟨*# straights*⟩`}{`⟨*# curves*⟩`}{`⟨*# crossings*⟩`}`

```
2610 \newcommand*\LP@trackline[3]%
2611 {%
2612   \LP@set@LP@scale{\LP@env@prefix}%
2613   \pgfmathsetlength{\LP@length}{.1cm*\LP@scale}%
2614   \scalebox{\LP@tracks@scale}%
2615   {%
2616     \hspace{\LP@length}%
2617     \setcounter{LP@whiledo@i}{#1}%
2618     \whiledo{\arabic{LP@whiledo@i}>0}%
2619     {%
2620       \LP@trackS%
2621       \addtocounter{LP@whiledo@i}{-1}%
2622     }%
2623     \setcounter{LP@whiledo@i}{#2}%
2624     \whiledo{\arabic{LP@whiledo@i}>0}%
2625     {%
2626       \LP@trackC%
2627       \addtocounter{LP@whiledo@i}{-1}%
2628     }%
2629     \hspace{\LP@length}%
2630     \setcounter{LP@whiledo@i}{#3}%
2631     \whiledo{\arabic{LP@whiledo@i}>0}%
2632     {%
2633       \LP@trackCR%
2634       \addtocounter{LP@whiledo@i}{-1}%
2635     }%
```

```
2636   }%
2637 }%
```

\trackV This macro places the vertical track lines and expects a csv list in the format straights/curves/crossings!

\trackV{⟨*csv list*⟩}

```
2638 \newcommand*\trackV[1]%
2639 {%
2640   \LP@set@LP@columns{\LP@env@prefix}%
2641   \setcounter{LP@counti}{\LP@columns}%
2642   \stepcounter{LP@counti}%
2643   \setcounter{LP@countii}{1}%
2644   \foreach \LP@c@straight/\LP@c@curve/\LP@c@cross in {#1}%
2645   {%
2646     \node[anchor=west] at (\arabic{LP@counti},\arabic{LP@countii}.5)%
2647       {\LP@trackline{\LP@c@straight}{\LP@c@curve}{\LP@c@cross}};%
2648     \stepcounter{LP@countii}%
2649   };%
2650 }%
```

\trackH The same for the horizontal track lines!

\trackH{⟨*csv list*⟩}

```
2651 \newcommand*\trackH[1]%
2652 {%
2653   \setcounter{LP@counti}{1}%
2654   \setcounter{LP@countii}{1}%
2655   \foreach \LP@c@straight/\LP@c@curve/\LP@c@cross in {#1}%
2656   {%
2657     \node[anchor=west,rotate=-90]%
2658       at (\arabic{LP@counti}.5,\arabic{LP@countii})%
2659         {\LP@trackline{\LP@c@straight}{\LP@c@curve}{\LP@c@cross}};%
2660     \stepcounter{LP@counti}%
2661   };%
2662 }%
```

\track We still need to draw the race track. The used decoration - which allows automatic over-/underbridges - was provided by Frédéric in this question!

\track{⟨*Tikz path*⟩}

```
2663 \newcommand*\track[1]%
2664 {%
2665   \LP@set@LP@scale{\LP@env@prefix}%
2666   \LP@set@LP@bgcolor{\LP@env@prefix}%
2667   \ifthenelse{\equal{\LP@bgcolor}{}}%
2668   {\gdef\LP@c@bg@track{white}}%
2669   {\gdef\LP@c@bg@track{\LP@bgcolor}}%
2670   \bgroup%
```

```
    \def\LP@rel@tikzpath{.5}%
    \begin{pgfonlayer}{LPbackgroundtwo}%
      \pgfsetcornersarced{\pgfpoint{4.9mm*\LP@scale}{4.9mm*\LP@scale}}%
      \draw[decorate,decoration={show path construction,%
        lineto code={%
          \draw [\LP@c@bg@track,line width=0.15cm*\LP@scale,%
                 double=LP@c@track,double distance=.3cm*\LP@scale]%
            (\tikzinputsegmentfirst) -- (\tikzinputsegmentlast);},%
        curveto code={%
          \draw [LP@c@track,line width=0.3cm*\LP@scale]%
            (\tikzinputsegmentfirst) .. controls%
            (\tikzinputsegmentsupporta) and (\tikzinputsegmentsupportb)%
            ..(\tikzinputsegmentlast);}}] #1;%
    \end{pgfonlayer}%
  \egroup%
}%
```

resuko

```
\newenvironment{resuko}[1][]%
{%
  \setkeys{resuko}{#1}%
  \LP@set@package{resuko}%
  \LP@set@env@prefix{LP@RSK}%
  \setcounter{LP@rows}{\LP@RSK@rows}%
  \setcounter{LP@columns}{\LP@RSK@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@RSK@width}%
    \ifthenelse{\equal{\LP@RSK@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@RSK@titleindent}\parbox{\LP@RSK@titlewidth}{\strut\
    \begin{tikzpicture}[LPpreset,scale=\LP@RSK@scale]%
      \LP@drawbackground{1}{1}{\LP@RSK@columns}{\LP@RSK@rows}{\LP@RSK@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@RSK@columns}{\LP@RSK@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@RSK@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% schatzsuche environment and options
%
\newcommand*\LP@SS@init@prefix{LP@SS}%
\newcommand*\LP@SS@init@package{schatzsuche}%
%
\LP@define@key{\LP@SS@init@prefix}{\LP@SS@init@package}{rows}{5}%
\LP@define@key{\LP@SS@init@prefix}{\LP@SS@init@package}{columns}{5}%
\LP@define@key{\LP@SS@init@prefix}{\LP@SS@init@package}{scale}{1}%
```

```
\LP@define@key{\LP@SS@init@prefix}{\LP@SS@init@package}{counterstyle}{none}%
\LP@define@key{\LP@SS@init@prefix}{\LP@SS@init@package}{bgcolor}{}%
\LP@define@key{\LP@SS@init@prefix}{\LP@SS@init@package}{width}{5.1cm}%
\LP@define@key{\LP@SS@init@prefix}{\LP@SS@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@SS@init@prefix}{\LP@SS@init@package}{title}{}%
\LP@define@key{\LP@SS@init@prefix}{\LP@SS@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@SS@init@prefix}{\LP@SS@init@package}{titlewidth}{5.1cm}%
\LP@define@choicekey@fontsize{\LP@SS@init@prefix}{\LP@SS@init@package}{Large}%
%
\let\schatzsuchecell\setcell%
%
\newcommand*\schatzsuchesetup[1]%
{%
  \setkeys{schatzsuche.sty}{#1}%
}%
```

`\LP@Diamond` For the `schatzsuche` environment, we need to draw diamonds.

```
\newcommand*\LP@Diamond%
{%
  \begin{tikzpicture}[scale=\LP@scale*.9]%
  \clip (.05,.905) rectangle (.95,.1);%
  \draw[line width=\LP@normallines,fill=yellow!20]%
    (.5,.1) -- (.7,.9) -- (.3,.9) -- cycle;%
  \draw[line width=\LP@normallines,fill=orange!20]%
    (.5,.1) -- (.7,.9) -- (.9,.8) -- cycle;%
  \draw[line width=\LP@normallines,fill=orange!20]%
    (.5,.1) -- (.3,.9) -- (.1,.8) -- cycle;%
  \draw[line width=\LP@normallines] (.1,.8) -- (.9,.8);%
\end{tikzpicture}%
}%
```

`\Diamond` A user command for drawing diamonds.

```
\let\Diamond\LP@Diamond%
```

`schatzsuche`

```
\newenvironment{schatzsuche}[1][]%
{%
  \setkeys{schatzsuche}{#1}%
  \LP@set@package{schatzsuche}%
  \LP@set@env@prefix{LP@SS}%
  \setcounter{LP@rows}{\LP@SS@rows}%
  \setcounter{LP@columns}{\LP@SS@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@SS@width}%
    \ifthenelse{\equal{\LP@SS@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@SS@titleindent}\parbox{\LP@SS@titlewidth}{\strut\LP
```

```
    \begin{tikzpicture}[LPpreset,scale=\LP@SS@scale]%
      \LP@drawbackground{1}{1}{\LP@SS@columns}{\LP@SS@rows}{\LP@SS@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@SS@columns}{\LP@SS@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@SS@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% skyline environment and options
%
\newcommand*\LP@SL@init@prefix{LP@SL}%
\newcommand*\LP@SL@init@package{skyline}%
%
\LP@define@key{\LP@SL@init@prefix}{\LP@SL@init@package}{rows}{5}%
\LP@define@key{\LP@SL@init@prefix}{\LP@SL@init@package}{columns}{5}%
\LP@define@key{\LP@SL@init@prefix}{\LP@SL@init@package}{scale}{1}%
\LP@define@key{\LP@SL@init@prefix}{\LP@SL@init@package}{counterstyle}{none}%
\LP@define@key{\LP@SL@init@prefix}{\LP@SL@init@package}{bgcolor}{}%
\LP@define@key{\LP@SL@init@prefix}{\LP@SL@init@package}{width}{6.7cm}%
\LP@define@key{\LP@SL@init@prefix}{\LP@SL@init@package}{cvoffset}{-38pt}%
\LP@define@key{\LP@SL@init@prefix}{\LP@SL@init@package}{title}{}%
\LP@define@key{\LP@SL@init@prefix}{\LP@SL@init@package}{titleindent}{0.75cm}%
\LP@define@key{\LP@SL@init@prefix}{\LP@SL@init@package}{titlewidth}{5.85cm}%
\LP@define@choicekey@fontsize{\LP@SL@init@prefix}{\LP@SL@init@package}{Large}%
%
\newcommand*\LP@SL@sudoku{false}%
%
\define@choicekey*{skyline.sty}{sudoku}[\LP@SL@sudoku\nr]{true,false}[true]%
{%
  \ifcase\nr\relax%
    \renewcommand*\LP@SL@sudoku{true}%
    \renewcommand*\LP@SL@rows{9}%
    \renewcommand*\LP@SL@columns{9}%
  \or%
    \renewcommand*\LP@SL@sudoku{false}%
  \fi%
}%
%
\define@choicekey*{skyline}{sudoku}[\LP@SL@sudoku\nr]{true,false}[true]%
{%
  \ifcase\nr\relax%
    \renewcommand*\LP@SL@sudoku{true}%
    \renewcommand*\LP@SL@rows{9}%
    \renewcommand*\LP@SL@columns{9}%
  \or%
    \renewcommand*\LP@SL@sudoku{false}%
  \fi%
}%
```

```
%
\let\skylineB\LP@bottomrow%
\let\skylineT\LP@toprow%
\let\skylineL\LP@leftcolumn%
\let\skylineR\LP@rightcolumn%
\let\skylinecell\setcell%
%
\newcommand*\skylinesetup[1]%
{%
  \setkeys{skyline.sty}{#1}%
}%
```

skyline

```
\newenvironment{skyline}[1][]%
{%
  \setkeys{skyline}{#1}%
  \LP@set@package{skyline}%
  \LP@set@env@prefix{LP@SL}%
  \setcounter{LP@rows}{\LP@SL@rows}%
  \setcounter{LP@columns}{\LP@SL@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@SL@width}%
    \ifthenelse{\equal{\LP@SL@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@SL@titleindent}\parbox{\LP@SL@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@SL@scale]%
      \LP@drawbackground{1}{1}{\LP@SL@columns}{\LP@SL@rows}{\LP@SL@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@SL@columns}{\LP@SL@rows}{1cm}%
      \ifthenelse{\equal{\LP@SL@sudoku}{true}}%
      {\LP@drawsudokugrid}%
      {}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@SL@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% slitherlink environment and options
%
\newcommand*\LP@SK@init@prefix{LP@SK}%
\newcommand*\LP@SK@init@package{slitherlink}%
%
\LP@define@key{\LP@SK@init@prefix}{\LP@SK@init@package}{rows}{5}%
\LP@define@key{\LP@SK@init@prefix}{\LP@SK@init@package}{columns}{5}%
\LP@define@key{\LP@SK@init@prefix}{\LP@SK@init@package}{scale}{1}%
\LP@define@key{\LP@SK@init@prefix}{\LP@SK@init@package}{counterstyle}{none}%
\LP@define@key{\LP@SK@init@prefix}{\LP@SK@init@package}{bgcolor}{}%
```

```
\LP@define@key{\LP@SK@init@prefix}{\LP@SK@init@package}{width}{5.2cm}%
\LP@define@key{\LP@SK@init@prefix}{\LP@SK@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@SK@init@prefix}{\LP@SK@init@package}{title}{}%
\LP@define@key{\LP@SK@init@prefix}{\LP@SK@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@SK@init@prefix}{\LP@SK@init@package}{titlewidth}{5.2cm}%
\LP@define@choicekey@fontsize{\LP@SK@init@prefix}{\LP@SK@init@package}{Large}%
%
\let\slitherlinkcell\setcell%
%
\newcommand*\slitherlinksetup[1]%
{%
  \setkeys{slitherlink.sty}{#1}%
}%
```

`\LP@drawgriddots` For the `slitherlink` environment, we need little dots in every cell corner. Therefore, we loop through all rows and columns and paint the dots on the `LPforeground` layer.

```
\newcommand*\LP@drawgriddots%
{%
  \pgfmathsetmacro{\START}{1}%
```

The dots are drawn immediatly after the grid, so we can recycle `LP@counti` and `LP@countii`.

```
  \pgfmathsetmacro{\ENDC}{\arabic{LP@counti}}%
  \pgfmathsetmacro{\ENDR}{\arabic{LP@countii}}%
  \begin{pgfonlayer}{LPforeground}%
    \foreach \i in {\START,...,\ENDC}%
      \foreach \j in {\START,...,\ENDR}%
        \fill[color=LP@c@griddots] (\i,\j) circle [radius=3pt];%
  \end{pgfonlayer}%
}%
```

`slitherlink`

```
\newenvironment{slitherlink}[1][]%
{%
  \setkeys{slitherlink}{#1}%
  \LP@set@package{slitherlink}%
  \LP@set@env@prefix{LP@SK}%
  \ifthenelse{\equal{\LP@grid@linestyle}{}}%
    {\setgridlinestyle{dashed}}{}%
  \setcounter{LP@rows}{\LP@SK@rows}%
  \setcounter{LP@columns}{\LP@SK@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@SK@width}%
    \ifthenelse{\equal{\LP@SK@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@SK@titleindent}\parbox{\LP@SK@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@SK@scale]%
```

```
      \LP@drawbackground{1}{1}{\LP@SK@columns}{\LP@SK@rows}{\LP@SK@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@SK@columns}{\LP@SK@rows}{1cm}%
      \LP@drawgriddots%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@SK@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% starbattle environment and options
%
\newcommand*\LP@SB@init@prefix{LP@SB}%
\newcommand*\LP@SB@init@package{starbattle}%
%
\LP@define@key{\LP@SB@init@prefix}{\LP@SB@init@package}{rows}{5}%
\LP@define@key{\LP@SB@init@prefix}{\LP@SB@init@package}{columns}{5}%
\LP@define@key{\LP@SB@init@prefix}{\LP@SB@init@package}{scale}{1}%
\LP@define@key{\LP@SB@init@prefix}{\LP@SB@init@package}{counterstyle}{none}%
\LP@define@key{\LP@SB@init@prefix}{\LP@SB@init@package}{bgcolor}{}%
\LP@define@key{\LP@SB@init@prefix}{\LP@SB@init@package}{width}{5.1cm}%
\LP@define@key{\LP@SB@init@prefix}{\LP@SB@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@SB@init@prefix}{\LP@SB@init@package}{title}{}%
\LP@define@key{\LP@SB@init@prefix}{\LP@SB@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@SB@init@prefix}{\LP@SB@init@package}{titlewidth}{5.1cm}%
\LP@define@choicekey@fontsize{\LP@SB@init@prefix}{\LP@SB@init@package}{Large}%
%
\let\starbattlecell\setcell%
%
\newcommand*\starbattlesetup[1]%
{%
  \setkeys{starbattle.sty}{#1}%
}%
```

starbattle

```
\newenvironment{starbattle}[1][]%
{%
  \setkeys{starbattle}{#1}%
  \LP@set@package{starbattle}%
  \LP@set@env@prefix{LP@SB}%
  \setcounter{LP@rows}{\LP@SB@rows}%
  \setcounter{LP@columns}{\LP@SB@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@SB@width}%
    \ifthenelse{\equal{\LP@SB@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@SB@titleindent}\parbox{\LP@SB@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@SB@scale]%
```

```
      \LP@drawbackground{1}{1}{\LP@SB@columns}{\LP@SB@rows}{\LP@SB@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@SB@columns}{\LP@SB@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@SB@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% starsandarrows environment and options
%
\newcommand*\LP@SAA@init@prefix{LP@SAA}%
\newcommand*\LP@SAA@init@package{starsandarrows}%
%
\LP@define@key{\LP@SAA@init@prefix}{\LP@SAA@init@package}{rows}{5}%
\LP@define@key{\LP@SAA@init@prefix}{\LP@SAA@init@package}{columns}{5}%
\LP@define@key{\LP@SAA@init@prefix}{\LP@SAA@init@package}{scale}{1}%
\LP@define@key{\LP@SAA@init@prefix}{\LP@SAA@init@package}{counterstyle}{none}%
\LP@define@key{\LP@SAA@init@prefix}{\LP@SAA@init@package}{bgcolor}{}%
\LP@define@key{\LP@SAA@init@prefix}{\LP@SAA@init@package}{width}{5.9cm}%
\LP@define@key{\LP@SAA@init@prefix}{\LP@SAA@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@SAA@init@prefix}{\LP@SAA@init@package}{title}{}%
\LP@define@key{\LP@SAA@init@prefix}{\LP@SAA@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@SAA@init@prefix}{\LP@SAA@init@package}{titlewidth}{5.9cm}%
\LP@define@choicekey@fontsize{\LP@SAA@init@prefix}{\LP@SAA@init@package}{Large}%
%
\let\starsH\LP@toprow%
\let\starsV\LP@leftcolumn%
%
\newcommand*\starsandarrowssetup[1]%
{%
  \setkeys{starsandarrows.sty}{#1}%
}%
```

`\LP@Star` For several environments, we need to draw stars.

```
\newcommand*\LP@Star%
{%
  \tikz\node[shape=star,fill=yellow,draw,scale=.8*\LP@scale,%
             star point height=.25cm] {};%
}%
```

A user command for drawing stars.

```
\let\Star\LP@Star%
```

`\LP@ArrowA` A generic macro for drawing arrows with different colors and angles.

`\LP@ArrowA[`⟨*color*⟩`]{`⟨*angle*⟩`}`

```
\newcommand*\LP@ArrowA[2][red]%
{%
  \LP@set@LP@scale{\LP@env@prefix}%
  \pgfmathsetmacro{\LPlinewidth}{3pt*\LP@scale}%
  \tikz\draw[->,line width=\LPlinewidth,rotate=#2,%
              scale=\LP@scale,color=#1]%
                (0.1,.5) -- (0.9,.5);%
}%
```

The we define some arrows with standard angles.

```
\newcommand*\LP@c@stararrow{red}%
```

`\Right`

```
\newcommand*\Right{\LP@ArrowA[\LP@c@stararrow]{0}}%
```

`\RightUp`

```
\newcommand*\RightUp{\LP@ArrowA[\LP@c@stararrow]{45}}%
```

`\Up`

```
\newcommand*\Up{\LP@ArrowA[\LP@c@stararrow]{90}}%
```

`\LeftUp`

```
\newcommand*\LeftUp{\LP@ArrowA[\LP@c@stararrow]{135}}%
```

`\Left`

```
\newcommand*\Left{\LP@ArrowA[\LP@c@stararrow]{180}}%
```

`\LeftDown`

```
\newcommand*\LeftDown{\LP@ArrowA[\LP@c@stararrow]{225}}%
```

`\Down`

```
\newcommand*\Down{\LP@ArrowA[\LP@c@stararrow]{270}}%
```

`\RightDown`

```
\newcommand*\RightDown{\LP@ArrowA[\LP@c@stararrow]{315}}%
```

`starsandarrows`

```
\newenvironment{starsandarrows}[1][]%
{%
  \setkeys{starsandarrows}{#1}%
  \LP@set@package{starsandarrows}%
```

```
3009   \LP@set@env@prefix{LP@SAA}%
3010   \setcounter{LP@rows}{\LP@SAA@rows}%
3011   \setcounter{LP@columns}{\LP@SAA@columns}%
3012   \stepcounter{LP@rows}%
3013   \stepcounter{LP@columns}%
3014   \begin{minipage}[t]{\LP@SAA@width}%
3015     \ifthenelse{\equal{\LP@SAA@title}{}}%
3016     {\par\enspace\par}% empty
3017     {\enspace\par\noindent\hspace{\LP@SAA@titleindent}\parbox{\LP@SAA@titlewidth}{\strut\
3018     \begin{tikzpicture}[LPpreset,scale=\LP@SAA@scale]%
3019       \LP@drawbackground{1}{1}{\LP@SAA@columns}{\LP@SAA@rows}{\LP@SAA@bgcolor}%
3020       \LP@drawgrid{1}{1}{\LP@SAA@columns}{\LP@SAA@rows}{1cm}%
3021 }%
3022 {%
3023     \end{tikzpicture}%
3024     \LP@drawcounter{\LP@SAA@counterstyle}%
3025     \stepcounter{LP@puzzlecounter}%
3026   \end{minipage}%
3027 }%

3028 %
3029 % sunandmoon environment and options
3030 %
3031 \newcommand*\LP@SAM@init@prefix{LP@SAM}%
3032 \newcommand*\LP@SAM@init@package{sunandmoon}%
3033 %
3034 \LP@define@key{\LP@SAM@init@prefix}{\LP@SAM@init@package}{rows}{5}%
3035 \LP@define@key{\LP@SAM@init@prefix}{\LP@SAM@init@package}{columns}{5}%
3036 \LP@define@key{\LP@SAM@init@prefix}{\LP@SAM@init@package}{scale}{1}%
3037 \LP@define@key{\LP@SAM@init@prefix}{\LP@SAM@init@package}{counterstyle}{none}%
3038 \LP@define@key{\LP@SAM@init@prefix}{\LP@SAM@init@package}{bgcolor}{}%
3039 \LP@define@key{\LP@SAM@init@prefix}{\LP@SAM@init@package}{width}{5.1cm}%
3040 \LP@define@key{\LP@SAM@init@prefix}{\LP@SAM@init@package}{cvoffset}{-23pt}%
3041 \LP@define@key{\LP@SAM@init@prefix}{\LP@SAM@init@package}{title}{}%
3042 \LP@define@key{\LP@SAM@init@prefix}{\LP@SAM@init@package}{titleindent}{0cm}%
3043 \LP@define@key{\LP@SAM@init@prefix}{\LP@SAM@init@package}{titlewidth}{5.1cm}%
3044 \LP@define@choicekey@fontsize{\LP@SAM@init@prefix}{\LP@SAM@init@package}{Large}%
3045 %
3046 \let\sunandmooncell\setcell%
3047 %
3048 \newcommand*\sunandmoonsetup[1]%
3049 {%
3050   \setkeys{sunandmoon.sty}{#1}%
3051 }%
```

`\LP@DarkCloud` For the `sunandmoon` environment, we need to draw a dark cloud and differently lighted moons.

```
3052 \newcommand*\LP@DarkCloud{%
3053   \tikz[scale=0.7*\LP@scale]\fill[color=black!60] (0,0)%
3054     rectangle (1,1);%
3055 }%
```

\LP@Moon

```
\newcommand*\LP@Moon{%
  \tikz[scale=0.7*\LP@scale]\fill[color=black!60]%
    (0,0) circle (0.5cm);%
}%
```

\LP@MoonR

```
\newcommand*\LP@MoonR%
{%
  \begin{tikzpicture}[scale=0.7*\LP@scale]%
    \fill[color=black!60] (0.5,0.5) circle (0.5cm);%
    \fill[color=yellow] (.5,0) arc (270:450:.5);%
  \end{tikzpicture}%
}%
```

\LP@MoonL

```
\newcommand*\LP@MoonL%
{%
  \begin{tikzpicture}[scale=0.7*\LP@scale]%
    \fill[color=black!60] (0.5,0.5) circle (0.5cm);%
    \fill[color=yellow] (.5,0) arc (270:90:.5);%
  \end{tikzpicture}%
}%
```

\LP@MoonT

```
\newcommand*\LP@MoonT%
{%
  \begin{tikzpicture}[scale=0.7*\LP@scale]%
    \fill[color=black!60] (0.5,0.5) circle (0.5cm);%
    \fill[color=yellow] (0,0.5) arc (180:0:.5);%
  \end{tikzpicture}%
}%
```

\LP@MoonB

```
\newcommand*\LP@MoonB%
{%
  \begin{tikzpicture}[scale=0.7*\LP@scale]%
    \fill[color=black!60] (0.5,0.5) circle (0.5cm);%
    \fill[color=yellow] (0,0.5) arc (180:360:.5);%
  \end{tikzpicture}%
}%
```

\LP@MoonTL

```
\newcommand*\LP@MoonTL%
{%
```

```
  \begin{tikzpicture}[scale=0.7*\LP@scale]%
    \fill[color=yellow] (0.5,0.5) circle (0.5cm);%
    \fill[color=black!60] (1,.5) -- (.5,.5) -- (.5,0)%
                          arc (270:360:.5);%
  \end{tikzpicture}%
}%
```

\LP@MoonBL

```
\newcommand*\LP@MoonBL%
{%
  \begin{tikzpicture}[scale=0.7*\LP@scale]%
    \fill[color=yellow] (0.5,0.5) circle (0.5cm);%
    \fill[color=black!60] (1,.5) -- (.5,.5) -- (.5,1)%
                          arc (90:0:.5);%
  \end{tikzpicture}%
}%
```

\LP@MoonBR

```
\newcommand*\LP@MoonBR%
{%
  \begin{tikzpicture}[scale=0.7*\LP@scale]%
    \fill[color=yellow] (0.5,0.5) circle (0.5cm);%
    \fill[color=black!60] (0,.5) -- (.5,.5) -- (.5,1)%
                          arc (90:180:.5);%
  \end{tikzpicture}%
}%
```

\LP@MoonTR

```
\newcommand*\LP@MoonTR%
{%
  \begin{tikzpicture}[scale=0.7*\LP@scale]%
    \fill[color=yellow] (0.5,0.5) circle (0.5cm);%
    \fill[color=black!60] (0,.5) -- (.5,.5) -- (.5,0)%
                          arc (270:180:.5);%
  \end{tikzpicture}%
}%
```

\Cloud

```
\let\Cloud\LP@DarkCloud%
```

\Moon

```
\let\Moon\LP@Moon%
```

\MoonR

```
\let\MoonR\LP@MoonR%
```

\MoonL

```
\let\MoonL\LP@MoonL%
```

\MoonT

```
\let\MoonT\LP@MoonT%
```

\MoonB

```
\let\MoonB\LP@MoonB%
```

\MoonTL

```
\let\MoonTL\LP@MoonTL%
```

\MoonBL

```
\let\MoonBL\LP@MoonBL%
```

\MoonBR

```
\let\MoonBR\LP@MoonBR%
```

\MoonTR

```
\let\MoonTR\LP@MoonTR%
```

sunandmoon

```
\newenvironment{sunandmoon}[1][]%
{%
  \setkeys{sunandmoon}{#1}%
  \LP@set@package{sunandmoon}%
  \LP@set@env@prefix{LP@SAM}%
  \setcounter{LP@rows}{\LP@SAM@rows}%
  \setcounter{LP@columns}{\LP@SAM@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@SAM@width}%
    \ifthenelse{\equal{\LP@SAM@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@SAM@titleindent}\parbox{\LP@SAM@titlewidth}{\strut\
    \begin{tikzpicture}[LPpreset,scale=\LP@SAM@scale]%
      \LP@drawbackground{1}{1}{\LP@SAM@columns}{\LP@SAM@rows}{\LP@SAM@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@SAM@columns}{\LP@SAM@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@SAM@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
```

```
  \end{minipage}%
}%

%
% tentsandtrees environment and options
%
\newcommand*\LP@TAT@init@prefix{LP@TAT}%
\newcommand*\LP@TAT@init@package{tentsandtrees}%
%
\LP@define@key{\LP@TAT@init@prefix}{\LP@TAT@init@package}{rows}{5}%
\LP@define@key{\LP@TAT@init@prefix}{\LP@TAT@init@package}{columns}{5}%
\LP@define@key{\LP@TAT@init@prefix}{\LP@TAT@init@package}{scale}{1}%
\LP@define@key{\LP@TAT@init@prefix}{\LP@TAT@init@package}{counterstyle}{none}%
\LP@define@key{\LP@TAT@init@prefix}{\LP@TAT@init@package}{bgcolor}{}%
\LP@define@key{\LP@TAT@init@prefix}{\LP@TAT@init@package}{width}{5.9cm}%
\LP@define@key{\LP@TAT@init@prefix}{\LP@TAT@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@TAT@init@prefix}{\LP@TAT@init@package}{title}{}%
\LP@define@key{\LP@TAT@init@prefix}{\LP@TAT@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@TAT@init@prefix}{\LP@TAT@init@package}{titlewidth}{5.9cm}%
\LP@define@choicekey@fontsize{\LP@TAT@init@prefix}{\LP@TAT@init@package}{Large}%
%
\let\tentH\LP@toprow%
\let\tentV\LP@leftcolumn%
%
\newcommand*\tentsandtreessetup[1]%
{%
  \setkeys{tentsandtrees.sty}{#1}%
}%
```

\LP@Tree For the tentsandtree environments, we need to draw trees.

```
\newcommand*\LP@Tree%
{%
  \begin{tikzpicture}[scale=.07*\LP@scale]%
    \draw[fill=brown!95!black!95,line width=.1pt*\LP@scale]%
      (.75,-1) .. controls (.5,.5) and   (.5,3) .. (0.5,4) --%
      (-0.5,4) .. controls (-.5,3) and (-.5,.5) .. (-.75,-1);%
    \shade[top color=green!50!black!60,draw=black,%
           line width=.1pt*\LP@scale,bottom color=green!50!black]%
      (0,10) .. controls (0,8)     and (1,7)    .. (1.5,7)%
             .. controls (1,7)     and (1,7)    .. (0.5,7.25)%
             .. controls (1.5,5)   and (2.5,4)  .. (3,4)%
             .. controls (2,4)     and (1.25,4) .. (1,4.5)%
             .. controls (2,2)     and (3.5,2)  .. (4,2)%
             .. controls (1,1)     and (-1,1)   .. (-4,2)%
             .. controls (-3.5,2)  and (-2,2)   .. (-1,4.5)%
             .. controls (-1.25,4) and (-2,4)   .. (-3,4)%
             .. controls (-2.5,4)  and (-1.5,5) .. (-0.5,7.25)%
             .. controls (-1,7)    and (-1,7)   .. (-1.5,7)%
             .. controls (-1,7)    and (0,8)    .. (0,10);%
  \end{tikzpicture}%
}%
```

\Tree A user command for drawing a tree.

```
\let\Tree\LP@Tree%
```

\LP@Tent We also need tents.

```
\newcommand*\LP@Tent%
{%
  \begin{tikzpicture}[scale=\LP@scale]%
    \draw[fill=yellow!50,line width=.1pt*\LP@scale]%
      (.1,.1) -- (.6,.1) -- (.9,.5) -- (.75,.8) -- (.35,.6)%
      -- cycle;%
    \draw[line width=.1pt*\LP@scale] (.35,.6) -- (.35,.1);%
    \draw[line width=.1pt*\LP@scale] (.35,.6) -- (.6,.1);%
  \end{tikzpicture}%
}%
```

\Tent A user command for drawing tents.

```
\let\Tent\LP@Tent%
```

tentsandtrees

```
\newenvironment{tentsandtrees}[1][]%
{%
  \setkeys{tentsandtrees}{#1}%
  \LP@set@package{tentsandtrees}%
  \LP@set@env@prefix{LP@TAT}%
  \setcounter{LP@rows}{\LP@TAT@rows}%
  \setcounter{LP@columns}{\LP@TAT@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@TAT@width}%
    \ifthenelse{\equal{\LP@TAT@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@TAT@titleindent}\parbox{\LP@TAT@titlewidth}{\strut\
    \begin{tikzpicture}[LPpreset,scale=\LP@TAT@scale]%
      \LP@drawbackground{1}{1}{\LP@TAT@columns}{\LP@TAT@rows}{\LP@TAT@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@TAT@columns}{\LP@TAT@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
    \LP@drawcounter{\LP@TAT@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%

%
% tunnel environment and options
%
\newcommand*\LP@TN@init@prefix{LP@TN}%
\newcommand*\LP@TN@init@package{tunnel}%
```

```
%
\LP@define@key{\LP@TN@init@prefix}{\LP@TN@init@package}{rows}{5}%
\LP@define@key{\LP@TN@init@prefix}{\LP@TN@init@package}{columns}{5}%
\LP@define@key{\LP@TN@init@prefix}{\LP@TN@init@package}{scale}{1}%
\LP@define@key{\LP@TN@init@prefix}{\LP@TN@init@package}{counterstyle}{none}%
\LP@define@key{\LP@TN@init@prefix}{\LP@TN@init@package}{bgcolor}{}%
\LP@define@key{\LP@TN@init@prefix}{\LP@TN@init@package}{width}{5.9cm}%
\LP@define@key{\LP@TN@init@prefix}{\LP@TN@init@package}{cvoffset}{-23pt}%
\LP@define@key{\LP@TN@init@prefix}{\LP@TN@init@package}{title}{}%
\LP@define@key{\LP@TN@init@prefix}{\LP@TN@init@package}{titleindent}{0cm}%
\LP@define@key{\LP@TN@init@prefix}{\LP@TN@init@package}{titlewidth}{5.9cm}%
\LP@define@choicekey@fontsize{\LP@TN@init@prefix}{\LP@TN@init@package}{Large}%
%
\let\tunnelH\LP@toprow%
\let\tunnelV\LP@leftcolumn%
%
\newcommand*\tunnelsetup[1]%
{%
  \setkeys{tunnel.sty}{#1}%
}%
```

`\LP@Portal` We define a Ti*k*Z picture that symbolizes a tunnel portal.

```
\newcommand*\LP@Portal%
{%
  \LP@set@LP@scale{\LP@env@prefix}%
  \begin{tikzpicture}[scale=\LP@scale]%
    \fill[color=black]%
      (0,0) -- (0,1) -- (.4,.5) -- (.4,0) -- cycle;%
    \fill[color=LP@c@tunnel@ii]%
      (0,1) -- (1,1) -- (1,.5) -- (.4,.5) -- cycle;%
    \fill[color=LP@c@tunnel@i]%
      (.4,0) rectangle (1,.5);%
    \draw[line width=\LP@normallines] (0,0) rectangle (1,1) ;%
  \end{tikzpicture}%
}%
```

`\portal` We define a user command to set a tunnel portal into a grid cell.

`\portal{`⟨*column*⟩`}{`⟨*row*⟩`}`

```
\newcommand*\portal[2]%
{%
```

To avoid interference with the grid lines, we use the `LPbackground` layer.

```
  \begin{puzzlebackground}%
    \LP@G@setcellcontent[hcenter,vcenter]{#1}{#2}{\LP@Portal}%
  \end{puzzlebackground}%
}%
```

`\tube` For the `tunnel` environments, we need to draw tubes.

\tube{⟨*TikZ path*⟩}

```
\newcommand*\tube[1]%
{%
  \LP@set@LP@scale{\LP@env@prefix}%
  \bgroup%
```

Tubes start in the center of a grid cell, therefore we must redefine `\LP@rel@tikzpath` to `.5` inside a group for `\tikzpath` and `\xtikzpath`.

```
    \def\LP@rel@tikzpath{.5}%
%       \end{macrocode}
% We draw on the \layer{LPbackgroundtwo} layer to get a segmented look behind
% the grid lines.
%    \begin{macrocode}
    \begin{pgfonlayer}{LPbackgroundtwo}%
      \pgfsetcornersarced{\pgfpoint{3mm}{3mm}}%
```

We draw the tube several times with slightly different colors to get a 3D effect.

```
      \draw[color=LP@c@tunnel!80!black, line width=.4cm*\LP@scale]%
        #1;%
      \draw[color=LP@c@tunnel, line width=.38cm*\LP@scale] #1;%
      \draw[color=LP@c@tunnel!85, line width=.35cm*\LP@scale] #1;%
      \draw[color=LP@c@tunnel!70, line width=.32cm*\LP@scale] #1;%
      \draw[color=LP@c@tunnel!55, line width=.29cm*\LP@scale] #1;%
      \draw[color=LP@c@tunnel!45, line width=.25cm*\LP@scale] #1;%
      \draw[color=LP@c@tunnel!35, line width=.2cm*\LP@scale] #1;%
    \end{pgfonlayer}%
  \egroup%
}%
```

`tunnel`

```
\newenvironment{tunnel}[1][]%
{%
  \setkeys{tunnel}{#1}%
  \LP@set@package{tunnel}%
  \LP@set@env@prefix{LP@TN}%
  \setcounter{LP@rows}{\LP@TN@rows}%
  \setcounter{LP@columns}{\LP@TN@columns}%
  \stepcounter{LP@rows}%
  \stepcounter{LP@columns}%
  \begin{minipage}[t]{\LP@TN@width}%
    \ifthenelse{\equal{\LP@TN@title}{}}%
    {\par\enspace\par}% empty
    {\enspace\par\noindent\hspace{\LP@TN@titleindent}\parbox{\LP@TN@titlewidth}{\strut\LP
    \begin{tikzpicture}[LPpreset,scale=\LP@TN@scale]%
      \LP@drawbackground{1}{1}{\LP@TN@columns}{\LP@TN@rows}{\LP@TN@bgcolor}%
      \LP@drawgrid{1}{1}{\LP@TN@columns}{\LP@TN@rows}{1cm}%
}%
{%
    \end{tikzpicture}%
```

```
    \LP@drawcounter{\LP@TN@counterstyle}%
    \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%
```

⟨/lpenv⟩

5 References

[1] Bernhard Hobiger. HoDoKu, 2012. (http://hodoku.sourceforge.net/en/index.php).

[2] Josef Kleber. createlpsudoku, 2013. (http://logicpuzzle.square7.de/createlpsudoku).

[3] Josef Kleber. Examples & Solutions (battleship), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-battleship.pdf).

[4] Josef Kleber. Examples & Solutions (bokkusu), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-bokkusu.pdf).

[5] Josef Kleber. Examples & Solutions (bridges), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-bridges.pdf).

[6] Josef Kleber. Examples & Solutions (chaossudoku), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-chaossudoku.pdf).

[7] Josef Kleber. Examples & Solutions (ddsudoku), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-ddsudoku.pdf).

[8] Josef Kleber. Examples & Solutions (fourwinds), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-fourwinds.pdf).

[9] Josef Kleber. Examples & Solutions (hakyuu), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-hakyuu.pdf).

[10] Josef Kleber. Examples & Solutions (hitori), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-hitori.pdf).

[11] Josef Kleber. Examples & Solutions (kakuro), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-kakuro.pdf).

[12] Josef Kleber. Examples & Solutions (kendoku), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-kendoku.pdf).

[13] Josef Kleber. Examples & Solutions (killersudoku), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-killersudoku.pdf).

[14] Josef Kleber. Examples & Solutions (laserbeam), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-laserbeam.pdf).

[15] Josef Kleber. Examples & Solutions (lpsudoku), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-lpsudoku.pdf).

[16] Josef Kleber. Examples & Solutions (magiclabyrinth), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-magiclabyrinth.pdf).

[17] Josef Kleber. Examples & Solutions (magnets), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-magnets.pdf).

[18] Josef Kleber. Examples & Solutions (masyu), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-masyu.pdf).

[19] Josef Kleber. Examples & Solutions (minesweeper), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-minesweeper.pdf).

[20] Josef Kleber. Examples & Solutions (numberlink), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-numberlink.pdf).

[21] Josef Kleber. Examples & Solutions (schatzsuche), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-schatzsuche.pdf).

[22] Josef Kleber. Examples & Solutions (skyline), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-skyline.pdf).

[23] Josef Kleber. Examples & Solutions (slitherlink), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-slitherlink.pdf).

[24] Josef Kleber. Examples & Solutions (sunandmoon), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-sunandmoon.pdf).

[25] Josef Kleber. Examples & Solutions (tentsandtrees), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-tentsandtrees.pdf).

[26] Josef Kleber. Examples & Solutions (tunnel), 2013. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-tunnel.pdf).

[27] Josef Kleber. How to create a Sudoku magazine, 2013. (http://logicpuzzle.square7.de/sudoku_magazine).

[28] Josef Kleber. logicpuzzle – A LaTeX style file for typesetting logic puzzles, 2013. (http://logicpuzzle.square7.de/).

[29] Josef Kleber. The logicpuzzle bundle, 2013. (https://bitbucket.org/kleberj/logicpuzzle/).

[30] Josef Kleber. lpsmag, 2013. (http://logicpuzzle.square7.de/lpsmag).

[31] Josef Kleber. Examples & Solutions (nonogram), 2014. (https://bitbucket.org/kleberj/logicpuzzle/downloads/example-nonogram.pdf).

[32] Stephen Ostermiller. QQwing – Sudoku Generator and Solver, 2011. (http://ostermiller.org/qqwing/).

6 Change History

M

R

S

T

U

V

W

X