logicpuzzle.sty

v2.5

A style file for typesetting logic puzzles

2014/06/15

Package author:
Josef Kleber
# Roll out your own grid-based logic puzzle

## User documentation

### PGF Layers

### Environments

#### Puzzle environments

#### Supporting environments

### Commands

#### In the grid

#### Presentation

#### Puzzle specific commands

##### 2D-Sudoku

##### Battleship

##### Bokkusu

##### Bridges

##### Chaos Sudoku

##### Four Winds

##### Hakyuu

##### Hitori

##### Kakuro

##### Kendoku

##### Killer Sudoku

##### Laser Beam

##### Magic Labyrinth

##### Magnets

##### Masyu

##### Minesweeper

##### Nonogram

##### Number Link

##### Resuko

##### Track tiles

##### Schatzsuche

##### Skyline

##### Slitherlink

##### Star Battle

##### Stars and Arrows

##### Arrows

##### Sudoku

##### Sun and Moon

##### Howl at the Moon

##### Tents and Trees

##### Tunnel

## Examples

### 2D-Sudoku

### Options
3.2 Battleship ................................................. 24
  3.2.1 Example ........................................... 24
  3.2.2 Options .......................................... 25
3.3 Bokkusu .................................................. 26
  3.3.1 Example ........................................... 26
  3.3.2 Options .......................................... 26
3.4 Bridges .................................................. 27
  3.4.1 Example ........................................... 27
  3.4.2 Options .......................................... 28
3.5 Chaos Sudoku ........................................... 29
  3.5.1 Example ........................................... 29
  3.5.2 Options .......................................... 30
3.6 Four Winds ............................................... 31
  3.6.1 Example ........................................... 31
  3.6.2 Options .......................................... 32
3.7 Hakyuu ................................................... 32
  3.7.1 Example ........................................... 32
  3.7.2 Options .......................................... 34
3.8 Hitori .................................................... 34
  3.8.1 Example ........................................... 34
  3.8.2 Options .......................................... 35
3.9 Kakuro .................................................... 36
  3.9.1 Example ........................................... 36
  3.9.2 Options .......................................... 37
3.10 Kendoku ................................................. 37
  3.10.1 Example ........................................... 37
  3.10.2 Options .......................................... 39
3.11 Killer Sudoku .......................................... 40
  3.11.1 Example ........................................... 40
  3.11.2 Options .......................................... 41
3.12 Laser Beam ............................................. 42
  3.12.1 Example ........................................... 42
  3.12.2 Options .......................................... 43
3.13 Magic Labyrinth ....................................... 43
  3.13.1 Example ........................................... 44
  3.13.2 Options .......................................... 44
3.14 Magnets ................................................ 45
  3.14.1 Example ........................................... 45
  3.14.2 Options .......................................... 46
3.15 Masyu ................................................... 46
  3.15.1 Example ........................................... 47
  3.15.2 Options .......................................... 47
3.16 Minesweeper ........................................... 48
  3.16.1 Example ........................................... 48
  3.16.2 Options .......................................... 49
3.17 Nonogram ............................................... 49
  3.17.1 Example ........................................... 49
4.1.5 Drawing the puzzle background . . . . . . . . . . . . . . . 76
4.1.6 In the grid . . . . . . . . . . . . . . . . . . . . . . . . . . . . 77
4.1.7 Around the grid . . . . . . . . . . . . . . . . . . . . . . . . . 89
4.1.8 Presentation . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
4.1.9 Misc macros . . . . . . . . . . . . . . . . . . . . . . . . . . . 92
4.2 lpenv.sty . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 97

5 References 157

6 Change History 160

7 Index 162
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.

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.

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.

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

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

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

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

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

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 skeleton and change or add some code for your specific puzzle.

## 2 User documentation

### 2.1 PGF Layers

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

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

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.
\texttt{color} \texttt{[]} specifies the color for coloring the cells.

\texttt{bgcolor} \texttt{[]} sets the background color of the grid.

\texttt{counterstyle} \texttt{[none]} defines the counter style. Predefined styles: none, left, right

\texttt{cvoffset} \texttt{[-23pt]} sets the vertical offset of the counters in the margin.

2.2.2 Supporting environments

\texttt{puzzlebackground} The \texttt{puzzlebackground} environment allows you to place elements behind the main layer on the LP\texttt{background} layer. This is for example useful for the \texttt{fillarea} command.

\texttt{puzzleforeground} The \texttt{puzzleforeground} environment allows you to place elements in front of the main layer on the LP\texttt{foreground} layer. This is for example useful for the \texttt{framearea} command.

2.3 Commands

2.3.1 In the grid

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

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

\texttt{\setbigcell} \texttt{\{fontsize\}\{column\}\{row\}\{element\}}
sets \texttt{(element)} into a big (2 x 2) cell \texttt{(column)\{row\}} as central node. The optional argument \texttt{(fontsize)} is set to 'Huge' by default.

\texttt{\setrow} \texttt{\{row\}\{csv list\}}
sets the contents of a \texttt{(row)}. These may be numbers or letters.

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

\texttt{\setcolumn} \texttt{\{column\}\{csv list\}}
sets the contents of a \texttt{(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@cc@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 environment 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.  

\footnote{Therefore, you must define an option color in the style file you want to use fill commands}
frames the grid (thicker line) with the color specified with the optional argument \([\text{color}]\) (default: black).

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

\begin{itemize}
  \item 7: up left
  \item 8: up
  \item 9: up right
  \item 4: left
  \item 5: no change
  \item 6: right
  \item 1: down left
  \item 2: down
  \item 3: down right
\end{itemize}

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

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

\xtikzpath\{\text{column}\}\{\text{row}\}\{\text{csv list}\}
is an evolution of the \tikzpath command with slightly different input syntax. In the \{\text{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.

\subsection{Presentation}

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

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

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

\setpuzzlecounter{\text{number}}
resets the puzzle counter, for example before the solutions.

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

\definecounterstyle{left}{\begin{itemgroup}reversemarginpar\marginnote{\tikz\node[shape=rectangle,fill=yellow!40,inner sep=7pt,}
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 \ShipC
\ShipL \ShipR
\ShipB \ShipT

\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 \{\{column\}\}\{\{row\}\}
places water markers (\texttt{*}) in the grid.

\placeisland \{\{column\}\}\{\{row\}\}
places islands (\texttt{\[\]}) in the grid. The island outlines are created randomly: \texttt{\[\]}, \texttt{\[\]}, \texttt{\[\]}, ...

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

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

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

\battleshipsetup \{\{options\}\}
resets the options with global scope.

\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.

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

\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 \{\{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 \{\{options\}\}
resets the options with global scope.

\chapter{2.3.3 Bokkusu}

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

\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 \{\{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 \{\{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.\n
\bridgescolumn \bridgescolumn\{\column\}\{\csv\ list\}\ sets\ the\ contents\ of\ a\ bridges\ \{\column\}\.\n
\bridge \bridge\{\optional\ arguments\}\ marg\TikZ\ 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\}\.\n
\bridgessetup \bridgessetup\{\options\}\ resets\ the\ options\ with\ global\ scope.\n
2.3.3.5 Chaos Sudoku

\chaossudokucell \chaossudokucell\{\column\}\{\row\}\{\number\}\ sets\ \{\number\}\ into\ grid\ cell\ \{\column\}\{\row\}\.\n
\chaossudokusetup \chaossudokusetup\{\options\}\ resets\ the\ options\ with\ global\ scope.\n
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.\n
\fourwindssetup \fourwindssetup\{\options\}\ resets\ the\ options\ with\ global\ scope.\n
2.3.3.7 Hakyuu

\hakyuucell \hakyuucell\{\column\}\{\row\}\{\number\]\ sets\ \{\number\}\ into\ grid\ cell\ \{\column\}\{\row\}\.\n
\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.
\section*{2.3.3.12 Laser Beam}

\texttt{\laserH{\langle csv list\rangle}}
\begin{itemize}
\item 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.
\end{itemize}

\texttt{\laserV{\langle csv list\rangle}}
\begin{itemize}
\item typesets the numbers left to the grid.
\end{itemize}

\texttt{\mirrorH{\langle csv list\rangle}}
\begin{itemize}
\item typesets the numbers below the grid indicating how many mirrors are placed in the intersections of this column.
\end{itemize}

\texttt{\mirrorV{\langle csv list\rangle}}
\begin{itemize}
\item typesets the numbers right to the grid.
\end{itemize}

\texttt{\placearrow{\langle column\rangle}{\langle row\rangle}}
\begin{itemize}
\item 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 \texttt{\{direction\}}, you can use the following indicators: \texttt{LeftUp}, \texttt{LeftDown}, \texttt{RightUp}, \texttt{RightDown}
\end{itemize}

\texttt{\placecross{\langle column\rangle}{\langle row\rangle}}
\begin{itemize}
\item places a cross in the intersections of the grid.
\end{itemize}

\texttt{\placemirror{\langle column\rangle}{\langle row\rangle}{\langle direction\rangle}}
\begin{itemize}
\item places mirrors in the intersections of the grid. In the mandatory argument \texttt{\{direction\}}, you can use the following indicators: \texttt{H}, \texttt{V}
\end{itemize}

\texttt{\laser{\langle color\rangle}{\{TikZ path\}}}
\begin{itemize}
\item draws the laser beam given by \texttt{\{TikZ path\}} with color \texttt{\{color\}} (default: red). The reference for coordinates is the bottom left corner of the cell.
\end{itemize}

\begin{verbatim}
\laser[green]{(1,2)--(2,3)--(1,4)}
\end{verbatim}

You should consider using this command in the \texttt{puzzlebackground} environment.

\texttt{\laserbeamsetup{\{options\}}}
\begin{itemize}
\item resets the options with global scope.
\end{itemize}

\section*{2.3.3.13 Magic Labyrinth}

\texttt{\magiclabyrinthcell{\langle column\rangle}{\langle row\rangle}{\langle number\rangle}}
\begin{itemize}
\item sets a number into grid cell \texttt{\{column\} \{row\}}.
\end{itemize}
\mlline \mlline{\textit{TikZ path}}
draws a line given by \{\textit{TikZ path}\}.

\magiclabyrinthsetup \magiclabyrinthsetup{\{options\}}
resets the options with global scope.

### 2.3.3.14 Magnets

\plusH \plusH{\langle csv list \rangle}
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{\langle csv list \rangle}
typesets the numbers above the grid indicating how many negative poles are in the respective column.

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

\minusV \minusV{\langle csv list \rangle}
typesets the numbers left to the grid indicating how many negative poles are in the respective row.

\magnetsH \magnetsH{\langle csv list \rangle}
typesets non-magnetic horizontal plates by using the column/row format in \{\langle csv list \rangle\}.

\magnetsV \magnetsV{\langle csv list \rangle}
typesets non-magnetic vertical plates by using the column/row format in \{\langle csv list \rangle\}.

\MPH \MPH{\langle csv list \rangle}
draws horizontal magnetic plates with $\text{+} - \text{-}$ arrangement. It expects the column/row format in \{\langle csv list \rangle\}. You can typeset the three other magnetic arrangements by using the \MPH, \MPM and \MPV commands.

\magnetssetup \magnetssetup{\{options\}}
resets the options with global scope.

### 2.3.3.15 Masyu

\masyucell \masyucell{\langle column \rangle}{\langle row \rangle}{\langle element \rangle}
sets an element into grid cell \{\langle column \rangle \} \{\langle row \rangle\}.

\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)\}. 18
\link \link{\langle TikZ path\rangle}
draws a line given by \langle TikZ path\rangle.

\numberlinksetup \numberlinksetup{\langle options\rangle}
resets the options with global scope.

### 2.3.3.19 Resuko

\resukocell \resukocell{\langle column\rangle}{\langle row\rangle}{\langle element\rangle}
sets the \langle element\rangle into grid cell \langle column\rangle{\langle row\rangle}.

### 2.3.3.19.1 Track tiles

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

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

\pitlane \pitlane{\langle column\rangle}{\langle row\rangle}{\langle direction\rangle}
draws the pit lane in grid cell \langle column\rangle{\langle row\rangle} with \langle direction\rangle V or H.

\parkinglot \parkinglot{\langle column\rangle}{\langle row\rangle}
draws the parking lot in grid cell \langle column\rangle{\langle row\rangle}.

\trackH \trackH{\langle csv list\rangle}
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{\langle csv list\rangle}
typesets the track tiles left to the grid.

\track \track{\langle TikZ path\rangle}
draws the race track given by \langle TikZ path\rangle. 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{\langle options\rangle}
resets the options with global scope.
2.3.3.20 Schatzsuche

\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 skyscrapers 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{\langle csv list \rangle}
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{\langle csv list \rangle}
typesets the numbers left to the grid.

\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
\starsandarrowssetup{\langle options \rangle}
resets the options with global scope.

2.3.3.25 Sudoku

\lpsudokucell{\langle column \rangle}{\langle row \rangle}{\langle number \rangle}
sets \langle number \rangle into grid cell \langle column \rangle{\langle row \rangle}.

\lpsudokusetup\lpsudokusetup{\langle options \rangle}
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
\sunandmoonssetup{\langle options \rangle}
resets the options with global scope.
2.3.3.27 Tents and Trees

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

\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 draws a tent.

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

2.3.3.28 Tunnel

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

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

\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{⟨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.
**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

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

```
\begin{center}
\begin{battleship}
\placesegment{4}{1}{} \ShipR
\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 \[6.7\text{cm}\] sets the width of the minipage, in which the grid is typeset.

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

fontsize \[\text{Large}\] 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, \text{Large}, LARGE, huge, Huge

title [] sets the title of a puzzle.

titleindent \[0.75\text{cm}\] defines the indent of the title.

titlewidth \[5.85\text{cm}\] specifies the width of the box the title is set in.

color \[\text{black}\] specifies the color for coloring the cells.

bgcolor [] sets the background color of the grid.

counterstyle \[\text{none}\] defines the counter style. Predefined styles: none, left, right

cvoffset \[-38\text{pt}\] 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

![Bridges Example Image](image-url)
\begin{center}
\begin{bridges}
\framepuzzle
\bridgesrow{8}{{},4,{},5,{},2}
\bridgesrow{7}{{},1}
\bridgesrow{5}{{},3,4}
\bridgesrow{4}{{},1,{},2}
\bridgesrow{3}{{},{}\ldots{}}
\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}{{},1,{},2}
\bridgesrow{3}{{},{}\ldots{}}
\bridgesrow{2}{{},2,5,2}
\bridge[double]{\tikzpath{2}{4}{8,8,8,6,6,2,2,2}}
\bridge[double]{\tikzpath{2}{2}{6,6,8,8}}
\bridge{\tikzpath{2}{4}{6,6}}
\bridge{\tikzpath{3}{5}{8,8}}
\bridge{\tikzpath{5}{2}{6,6,8,8,8,8,4,4}}
\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)--(4,2)--(4,3)--(2,3)}
\fillarea{LightBlue}{(3,4)--(3,6)--(6,6)--(6,5)--(5,5)--(5,2)--(4,2)--(4,3)}
\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 \texttt{LaTeX} sizes are used. Possible values: tiny, scriptsize, footnotesize, small, normalsize, large, \texttt{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
**3.6 Four Winds**

Fill all cells with light rays. These may not intersect. Cells with numbers represent the lighting system that lights 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

```latex
\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}
\end{hakyuu}
\end{center}
```
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

\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}
\end{hakyuu}

```latex
\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)-
\quad(4,4)-(4,3)-(2,3)-(2,4)}
\fillarea{LightBlue}\{(5,5)-(6,5)-(6,3)-(4,3)-(4,4)-
\quad(5,4)-(5,5)}
\fillarea{LightSalmon!50}\{(2,2)-(2,3)-(5,3)-(5,2)-
\quad(2,2)}
\fillarea{LightYellow}\{(2,1)-(2,2)-(5,2)-(5,3)-(6,3)-
\quad(6,1)-(2,1)}
\end{puzzlebackground}
\end{hakyuu}
```

33
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, \texttt{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}
\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, KK{23}{\}, KK{16}{\}, KK{10}{\}, Black}
\kakurorow{4}{KK{14}{\}, 9, 1, 4, KK{3}{\}}
\kakurorow{3}{KK{16}{\}, 6, 5, 3, 2}
\kakurorow{2}{KK{14}{\}, 8, 3, 2, 1}
\kakurorow{1}{Black, KK{8}{\}, 7, 1, Black}
\end{kakuro}
\hspace{1.5cm}
\begin{kakuro}[solution]
\framepuzzle
\kakurorow{5}{\Black, KK{23}{\}, KK{16}{\}, KK{10}{\}, Black}
\kakurorow{4}{KK{14}{\}, 9, 1, 4, KK{3}{\}}
\kakurorow{3}{KK{16}{\}, 6, 5, 3, 2}
\kakurorow{2}{KK{14}{\}, 8, 3, 2, 1}
\kakurorow{1}{Black, KK{8}{\}, 7, 1, Black}
\end{kakuro}
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 \TeX\ 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}
\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}
```

40
\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}{8,8,6,2,2,4,4}}
\framearea{black}{\tikzpath{2}{1}{8,6,6,6,2,4,4,4}}
\framearea{black}{\tikzpath{2}{3}{8,6,8,6,2,2,4,4}}
\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{ }
\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. 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.

43
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, \textbf{Large}, LARGE, huge, Huge
### 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

```latex
\magnetsset{bgcolor=Teal!50}
\begin{magnets}
\minusH{2,1,2,2,2,1}
\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}
```
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

![Masyu puzzles with paths drawn](image)

\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. 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.
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}{3,3,\Mine}
\setrow{3}{\Mine,4,2}
\setrow{2}{{},{},{},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]
\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, \texttt{Large}, \texttt{LARGE}, huge, \texttt{Huge}

\texttt{title} \[\] sets the title of a puzzle.

\texttt{titleindent} \[0cm\] defines the indent of the title.

\texttt{titlewidth} \[5.1cm\] specifies the width of the box the title is set in.

\texttt{color} \[black\] sets the color of the lines.

\texttt{bgcolor} \[\] sets the background color of the grid.

\texttt{counterstyle} \[\texttt{none}\] defines the counter style. Predefined styles: none, left, right

\texttt{cvoffset} \[-23pt\] sets the vertical offset of the counters in the margin.

\texttt{extracells} \[5\] sets the number of extra cells for the grid.

\texttt{helplines} \[5\] sets the space between thicker help lines. You can switch off help lines with 0!

\texttt{solution} \[false\] will switch off extra cells if set to true.

\subsection{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.

\subsubsection{Example}

\begin{verbatim}
\begin{numberlink}
\framepuzzle
\end{numberlink}
\end{verbatim}
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 pit lane 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}
```

![Resuko example](image-url)
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

\texttt{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, \texttt{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{tabular}{c c c c c}
  &  &  &  &  \\
  &  &  &  &  \\
 3 & 2 & 3 &  &  \\
  &  &  &  &  \\
 4 &  & 3 & 1 &  \\
 3 & 3 & 1 &  &  \\
\end{tabular}
\hspace{1cm}
\begin{tabular}{c c c c c}
  &  &  &  &  \\
  &  &  &  &  \\
 5 & 4 & 3 & 1 & 2 \\
 4 & 5 & 1 & 2 & 3 \\
 2 & 3 & 5 & 4 & 1 \\
 3 & 1 & 2 & 5 & 4 \\
\end{tabular}
\end{center}
3.21.1.1 Variants

3.21.1.1.1 Skyline Sudoku

\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}

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

\begin{center}
\begin{skyline}[sudoku,scale=.4]
\skylineB{4,5,2,5,2,1,2,4,3}
\skylineL{4,2,2,1,3,3,3,2}
\skylineT{4,1,3,2,3,5,3,2,3}
\skylineR{3,3,1,3,4,2,3,2,4}
\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{4,2,1,3,3,2}
\skylineR{3,3,4,3,2,1}
\end{skyline}
\end{center}
\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{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}
\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, \texttt{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}
\end{slitherlink}
\end{center}
### Examples

\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}

#### 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.
**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

```latex
\begin{center}
\begin{starbattle}
  \framepuzzle
  \framearea{black}{\tikzpath{1}{1}{8,6,8,6,2,6,2,4,4,4,4,4}}
  \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}{8,6,8,6,2,6,2,4,4,4,4,4}}
  \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}
\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,{},\Star\}
\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,{},\Star\}
\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

\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,{} }
\end{lpsudoku}
\end{center}
\begin{lpsudoku}
\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 little 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 $\frac{5}{9}$

- **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 (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.

2processing 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

3.26.2 Options

\texttt{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 [\[
\text{Large}\]

sets 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
\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

![Example Diagram]

```latex
\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 cbna.

4 Implementation

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 TikZ 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.
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@linestyle}{} \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)}

\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}%

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

\LP@define@choicekey@fontsize

\LP@define@choicekey@fontsize{(puzzle prefix)}{(puzzle)}{(default)}

\LP@define@choicekey@fontsize[3]%
\expandafter\gdef\csname#1@fontsize\endcsname{\Large}%
\define@choicekey*[#2.sty]{fontsize}{\LP@dck@fontsize
r}{\tiny,\scriptsize,\footnotesize,\small,\normalsize,%
large,\Large,\LARGE,\huge,\Huge}{\LP@dck@fontsize\nr}{%
\expandafter\gdef\csname#1@fontsize\endcsname{\tiny}%
\expandafter\gdef\csname#1@fontsize\endcsname{\scriptsize}%
\expandafter\gdef\csname#1@fontsize\endcsname{\footnotesize}%
\expandafter\gdef\csname#1@fontsize\endcsname{\small}%
\expandafter\gdef\csname#1@fontsize\endcsname{\normalsize}%
\expandafter\gdef\csname#1@fontsize\endcsname{\large}%
\expandafter\gdef\csname#1@fontsize\endcsname{\Large}%
\expandafter\gdef\csname#1@fontsize\endcsname{\huge}%
\expandafter\gdef\csname#1@fontsize\endcsname{\Huge}%
4.1.3 Defining colors

We define a command for defining (rgb) colors. For other color models, use \texttt{xcolor}'s \texttt{definecolor} command.

\begin{verbatim}
\LP@definecolor{⟨name⟩}{⟨rgb color⟩}
\end{verbatim}

\begin{verbatim}
\newcommand*{\LP@definecolor}[2]{\definecolor{#1}{rgb}{#2}}
\end{verbatim}

Predefined colors:

\begin{verbatim}
\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}
\end{verbatim}
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 \((\text{columns} + 1, \text{rows} + 1)\). 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{(\textit{xmin})}{(\textit{ymin})}{(\textit{xmax})}{(\textit{ymax})}{(\textit{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{(\textit{xmin})}{(\textit{ymin})}{(\textit{xmax})}{(\textit{ymax})}{(bgcolor)}

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

Of course, we only draw a background, if \{(bgcolor)\} is not empty!
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 an 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)\}

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@iii}<\value{LP@count@ig@i}
\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@iv}<\value{LP@count@ig@ii}
\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%
which is outside the grid (1,1) .. (#3,#4)\%
\fi\%
\}\
\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\}}
\newcommand*{\setrule}[3]{
First, we copy the scale and bgcolor values from the current environment.
\LP@set@LP@scale{\LP@env@prefix}\%
\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.
\ifthenelse{\equal{\LP@bgcolor}{}}{
\gdef{\LP@sr@bgcolor}{white}}{
\gdef{\LP@sr@bgcolor}{\LP@bgcolor}}\%
\stepcounter{LP@counter@unique}\%
First, we draw a (invisible) helper rectangle on the LPdump layer (behind the LPbgcolor layer) in the node $A_\theLP@counter@unique$.
\begin{pgfonlayer}{LPdump}\node [shape=rectangle,inner sep=0pt]\node [shape=rectangle,inner sep=0pt,anchor=north west,\scale=\LP@scale,font=\tiny] at \node [shape=rectangle,inner sep=0pt,anchor=north west,\scale=\LP@scale,font=\tiny] at
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 $\times$ scale).
\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\}\{\LP@Pfive\}]\{\LP@Pfive\}]\{\LP@Pfive\}]\{\LP@Pfive\}
\def\LP@scc@v{\LP@Psfive}\
\define@key{LP@G@setcellcontent}{hcenter}{\LP@Psfive}\
\% \def\LP@scc@h{}% \def\LP@scc@v{}% \setkeys{LP@G@setcellcontent}{#1}% \node at (#2\LP@scc@h,#3\LP@scc@v){#4};% 

\def\LP@setcellcontent{For compatibility, we still provide the old \LP@setcellcontentXY macros.}
\newcommand*{\LP@setcellcontent}[4]{% \LP@G@setcellcontent\{hcenter,vcenter\}{#1}{#2}{#3}% 
\LP@setrowcontents{\langle csv list\rangle}{\langle column\rangle}{\langle row\rangle} 
\newcommand*{\LP@setrowcontents}[3]{% We copy fontsize from the current environment, if we want to typeset numbers or letters.
\begin{verbatim}
252 \newcommand*{\LP@setrowcontents}[3]{% 
253 {% 

79
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@setrowcontents@edge

\LP@setcolumncontents Of course, we want to do the same for columns.
\LP@setcolumncontents{\<csv list\>}{\<column\>}{\<row\>}

\LP@setrowcontents@edge For environments like laserbeam, we need to typeset row contents on the left border of the cell, instead of centered. Therefore, we only use vcenter.
\LP@setrowcontents@edge{\<csv list\>}{\<column\>}{\<row\>
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@element}
    \stepcounter{LP@countii}
  }
}

\setrow
The user command for typesetting row contents.
\setrow{\row}{\csv list}
\newcommand*{\setrow}[2]{
  \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}{\column}{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]{
  \LP@set@LP@scale{\LP@env@prefix}
  \LP@setcolumncontents{#2}{\column}{1}{#1}{#3}
}
First, we copy \texttt{scale}, \texttt{fontsize}, \texttt{rows} and \texttt{columns} from the current environment.

\begin{verbatim}
\setLP@scale{\env@prefix}\%
\setLP@fontsize{\env@prefix}\%
\setLP@rows{\env@prefix}\%
\setLP@columns{\env@prefix}\%
\end{verbatim}

Then, we test if \((\langle column \rangle,\langle row \rangle)\) is within the borders of the puzzle. Finally, we typeset \((\langle element \rangle)\) into a central node.

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

We also want to typeset the same element into several grid cells. Therefore, we use a \texttt{\{csv list\}} with the format: \((\langle column/row, \ldots \rangle)\)

\begin{verbatim}
def setcells{\{csv list\}}{(element)}
\end{verbatim}

\begin{verbatim}
\newcommand*\setbigcell[4][Huge]\
{\setLP@scale{\env@prefix}\%
\setLP@bgcolor{\env@prefix}\%
\ifthenelse{\equal{\bgcolor}{}}{\gdef\sbc@bgcolor{white}}{\gdef\sbc@bgcolor{\bgcolor}}\%
\setLP@rows{\env@prefix}\%
\setLP@columns{\env@prefix}\%
\foreach \LP@sc@column/\LP@sc@row in {#1}\%
{\LP@ingrid{\LP@sc@column}{\LP@sc@row}{\columns}{\rows}{\package}\%
\LP@G@setcellcontent[hcenter,vcenter]{\LP@sc@column}{\LP@sc@row}{\fontsize#2}\%
};\%
}
\end{verbatim}

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

\begin{verbatim}
def setbigcell{\{fontsize\}}{(column)}{(row)}{(element)}
\end{verbatim}

\begin{verbatim}
\newcommand*\setbigcell[4][Huge]\
{\setLP@scale{\env@prefix}\%
\setLP@bgcolor{\env@prefix}\%
\ifthenelse{\equal{\bgcolor}{}}{\gdef\sbc@bgcolor{white}}{\gdef\sbc@bgcolor{\bgcolor}}\%
\setLP@rows{\env@prefix}\%
\setLP@columns{\env@prefix}\%
{\LP@G@setcellcontent[hcenter,vcenter]{\LP@sc@column}{\LP@sc@row}{\columns}{\rows}{\package}\%
};\%
}
\end{verbatim}

The center of \((2 \times 2)\) cell is the bottom left corner of \((\text{column} + 1, \text{row} + 1)\)
First, we ‘clear’ (2 × 2) area with bgcolor and respect the grid line style.

\draw[\textwidth=LP@normallines,fill=LP@sbc@bgcolor,\
\textstyle=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 \{\text{element}\}.

\begin{pgfonlayer}{LPforegroundtwo}%
\fill[\textcolor=LP@sbc@bgcolor]%
(\arabic{LP@counti},\arabic{LP@countii})% 
circle [radius=3.5pt*\textscale];% 
\node at (\arabic{LP@counti},\arabic{LP@countii})% 
{\textcolor\textname#1\textname#4};% 
\end{pgfonlayer}% 
\setcolorrow For the hitori environment, we need to typeset row contents with numbers associated to background colors.
\setcolorrow\{\text{row}\}\{\text{csv list}\}%
\newcommand\setcolorrow[2]{% 
We start at column 1 and loop through \{\text{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\textcolor\text{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.
\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} \LP@ingrid{#1}{#2}{\LP@columns}{\LP@rows}{\LP@package}}

\begin{puzzlebackground} \fillcell{\arabic{LP@counti}}{\arabic{LP@countii}} \end{puzzlebackground} \setcell{\arabic{LP@counti}}{\arabic{LP@countii}}{\LP@element} \stepcounter{LP@countii} }

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

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

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

Again the same, for columns!
\framearea\{color\}\{TikZ path\}
\newcommand*{\framearea}[2]{%\draw[line width=\LP@thicklines,color=#1] #2;\}}%

Sometimes, we need to frame a specified area.
\fillarea\{color\}\{TikZ path\}
\newcommand*{\fillarea}[2]{%\fill\draw[line width=\LP@thicklines,color=#1] #2;\}}%

Sometimes, we even want to fill the area.
\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]{\begin{tikzpicture}[scale=.5]
\draw (\LP@rel@tikzpath,\LP@rel@tikzpath) (-1,-1) grid (1,1);
\draw[thick] (0,0) circle (1cm);
\foreach \LP@direction in {#3} {
  \ifnum\LP@direction=1 \draw[thick] ++(-1,-1) \fi
  \ifnum\LP@direction=2 \draw[thick] ++(0,-1) \fi
  \ifnum\LP@direction=3 \draw[thick] ++(1,-1) \fi
  \ifnum\LP@direction=4 \draw[thick] ++(-1,0) \fi
  \ifnum\LP@direction=6 \draw[thick] ++(1,0) \fi
  \ifnum\LP@direction=7 \draw[thick] ++(-1,1) \fi
}\end{tikzpicture}}

starting point
\begin{tikzpicture}[scale=.5]
\draw (\LP@rel@tikzpath,\LP@rel@tikzpath) (-1,-1) grid (1,1);
\draw[thick] (0,0) circle (1cm);
\foreach \LP@direction in {#3} {
  \ifnum\LP@direction=1 \draw[thick] ++(-1,-1) \fi
  \ifnum\LP@direction=2 \draw[thick] ++(0,-1) \fi
  \ifnum\LP@direction=3 \draw[thick] ++(1,-1) \fi
  \ifnum\LP@direction=4 \draw[thick] ++(-1,0) \fi
  \ifnum\LP@direction=6 \draw[thick] ++(1,0) \fi
  \ifnum\LP@direction=7 \draw[thick] ++(-1,1) \fi
}\end{tikzpicture}

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

\filldiagonals For some puzzles we need colored diagonals indicating that also the diagonals are relevant, not just rows and columns.
\filldiagonals[\textcolor{color}]

We copy and get scale, $\text{rows}+1$ and $\text{columns}+1$,

\begin{framepuzzle}
\begin{puzzlebackground}
\def\LP@color{#1}
\setcounter{LP@whiledo@i}{1}
\setcounter{LP@whiledo@ii}{\text{\textbf{\textless{rows}}} \text{\textbf{\textgreater{1}}}}
\whiledo{\value{LP@whiledo@i}<\text{\textbf{\textless{columns}}} \text{\textbf{\textgreater{1}}}}{}
\end{puzzlebackground}
\end{framepuzzle}

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

\framepuzzle[\textcolor{color}]

88
We copied rows and columns to get the top right corner of the puzzle. Now, we can draw the thicker lines.

\begin{pgfonlayer}{LPbackground}
\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);
\end{pgfonlayer}

\begin{pgfonlayer}{LPforeground}
\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);
\end{pgfonlayer}

\LP@bottomrow

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

\LP@bottomrow

Sometimes, we want to move material to the LPbackground layer:

\begin{pgfonlayer}{LPbackground}
%...
\end{pgfonlayer}

Sometimes, we want to move material to the LPforeground layer:

\begin{pgfonlayer}{LPforeground}
%...
\end{pgfonlayer}

\section*{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\{\langle csv list \rangle\}
\newcommand*{\LP@bottomrow}[1]{\LP@setrowcontents{#1}{1}{0}}

\LP@bottomrow@edge Put numbers below the puzzle, but on the edge.
\LP@bottomrow@edge\{\langle csv list \rangle\}
\newcommand*{\LP@bottomrow@edge}[1]{\LP@setrowcontents@edge{#1}{1}{0}}

\LP@leftcolumn Put numbers left to the puzzle.
\LP@leftcolumn\{\langle csv list \rangle\}
\newcommand*{\LP@leftcolumn}[1]{\LP@setcolumncontents{#1}{0}{1}}

\LP@toprow Put number on top of the puzzle.
\LP@toprow\{\langle csv list \rangle\}
\newcommand*{\LP@toprow}[1]{\LP@setrowcontents{#1}{1}{\theLP@rows}}

\LP@rightcolumn Put numbers right to the puzzle.
\LP@rightcolumn\{\langle csv list \rangle\}
\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@edge\{\langle csv list \rangle\}
\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 the default title format:
\titleformat{\centering\Large\color{blue}}{\puzzlecounter}{\setpuzzlecounter}

\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}{(counterstyle)}{\LP@drawcounter}%

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}{(counterstyle)}{(definition)}%\definecounterstyle{%

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.
We define a TikZ picture with the size of a grid cell ($1cm \times 1cm$) and fill it with color.

\LP@Block

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

\LP@Line[\texttt{options}]{\texttt{(TikZ path)}}
By default we draw a single 1mm wide line with color $\text{LP@c@bridge}$. But we can change that with $\langle\text{options}\rangle$.

Then we can draw a single or double line based on the defined options or defaults.
\newcommand*{\LP@set@cvoffset}[1]{\expandafter\xdef\expandafter{\LP@cvoffset}{\csname #1@cvoffset\endcsname}}
\newcommand*{\LP@set@bgcolor}[1]{\expandafter\xdef\expandafter{\LP@bgcolor}{\csname #1@bgcolor\endcsname}}
\newcommand*{\LP@set@extracells}[1]{\expandafter\xdef\expandafter{\LP@extracells}{\csname #1@extracells\endcsname}}
\newcommand*{\LP@set@fontsize}[1]{% can’t expand \Large \expandafter\gdef\expandafter{\LP@fontsize}{\csname #1@fontsize\endcsname}}
\newcommand*{\LP@set@env@prefix}[1]{\gdef\LP@env@prefix{#1}}
\newcommand*{\LP@set@package}[1]{\gdef\LP@package{#1}}
\newcommand*{\setgridlinestyle}[1]{\def\LP@grid@linestyle{#1}}
\newcommand*{\setnormallinewidth}[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}}
}

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}
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.
\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}\setkeys{shipbox}{V}\whiledo{\value{LP@whiledo@i}>0}{\addtocounter{LP@whiledo@i}{-1}\setcounter{LP@countii}{\theLP@whiledo@i}\placesegment{#2}{\theLP@countii}{\ShipC}}\stepcounter{LP@countii}\placesegment{#2}{\theLP@countii}{\ShipT}\fi\fi\setcounter{LP@countii}{#3}\setcounter{LP@counti}{#2}}
\begin{center}
\begin{battleship}[rows=10,columns=10,width=6.5cm,title=Me,sbindent=0.65cm,titleindent=0.55cm,titlewidth=5.7cm,sbwidth=5.7cm,scale=0.59,fontsize=normalsize]
\shipV{J,I,H,G,F,E,D,C,B,A}
\shipH{1,2,3,4,5,6,7,8,9,10}
\shipbox{#1}
\end{battleship}
\hspace{1cm}
\begin{battleship}[rows=10,columns=10,width=6.5cm,title=Enemy,sbindent=0.65cm,titleindent=0.55cm,titlewidth=5.7cm,sbwidth=5.7cm,scale=0.59,fontsize=normalsize]
\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}
\par
\vspace{1cm}
\begin{center}
\begin{battleship}[rows=10,columns=10,width=6.5cm,title=Me,sbindent=0.65cm,titleindent=0.55cm,titlewidth=5.7cm,sbwidth=5.7cm,scale=0.59,fontsize=normalsize]
\shipV{J,I,H,G,F,E,D,C,B,A}
\shipH{1,2,3,4,5,6,7,8,9,10}
\shipbox{#1}
\end{battleship}
\end{center}
\begin{battleship}[rows=10, columns=10, width=6.5cm, title=Enemy, sbindent=0.65cm, titleindent=0.55cm, titlewidth=5.7cm, sbwidth=5.7cm, scale=0.59, fontsize=\normalsize]
\shipV{J, I, H, G, F, E, D, C, B, A}
\shipH{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
\end{battleship}

\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}%
}
\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}{\enspace\par
      \hspace{\LP@BK@titleindent}\parbox{\LP@BK@titlewidth}{\strut\LP@titleformat{\LP@BK@title}}\vspace{3mm}\par}%
    \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}%
}
\bokkususetup[1]%

bokkusu

\newenvironment{bokkusu}[1][{}]{%
  \setkeys{bokkusu}{#1}%
}
\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}{\enspace\par
      \hspace{\LP@BK@titleindent}\parbox{\LP@BK@titlewidth}{\strut\LP@titleformat{\LP@BK@title}}\vspace{3mm}\par}%
    \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}%
}
\bokkususetup[1]%
For the bridges environment, we need a special row command for drawing the islands with the numbers of bridges.

\bridgesrow\(\langle\text{row}\rangle\)\{\text{csv list}\}

We loop through the list and 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.

\bridgescolumn And again, for columns!
For the bridges environment, we need to draw bridges. Keep in mind that you can influence the appearance of the bridge with the options double, color and linewidth.

\begin{tikzpicture}
\draw[dashed] (0,0) -- (1,1);
\end{tikzpicture}
\begin{example}
% \LP@drawgrid{1}{1}\LP@BG\LP@columns\LP@BG\LP@rows\LP cm

% \LP@drawcounter\LP@BG\LP@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@package{rows}{5}
% \LP@define@key\LP@CS@init@package{columns}{5}
% \LP@define@key\LP@CS@init@package{scale}{1}
% \LP@define@key\LP@CS@init@package{counterstyle}{none}
% \LP@define@key\LP@CS@init@package{bgcolor}{}
% \LP@define@key\LP@CS@init@package{width}{5.1cm}
% \LP@define@key\LP@CS@init@package{cvoffset}{-23pt}
% \LP@define@key\LP@CS@init@package{title}{}
% \LP@define@key\LP@CS@init@package{titleindent}{0cm}
% \LP@define@key\LP@CS@init@package{titlewidth}{5.1cm}
% \LP@define@choicekey@fontsize\LP@CS@init@package{Large}

% \let\chaossudokucell\setcell
% \newcommand\chaossudokusetup[1]{\setkeys{chaossudoku}{#1}}

% chaossudoku
% \newenvironment\chaossudoku[1][]{}
% \LP@set@package\chaossudoku
% \LP@set@env@prefix\LP@CS
% \setcounter\LP@rows\LP@CS\LP@columns
% \stepcounter\LP@rows\LP@columns
% \begin{minipage}[t]{\LP@CS\LP@width}
% \ifthenelse{\equal\LP@CS\title}{\par}{\enspace\par}

\end{example}
\newcommand*\fourwindssetup[1]%
\setkeys{fourwinds.sty}{#1}%

\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@titleformat\LP@FW@title}\vspace{3mm}\par}%
\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}{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%
\hakyuusetup[1]%
\setkeys{hakyuu.sty}{#1}%
\newenvironment{hakyuu}{}{%
\setkeys{hakyuu}{#1}%
\begin{minipage}[t]{\LP@HY@width}%
\ifthenelse{\equal{\LP@HY@title}{}}{}{\enspace\par
\enspace\par}% empty
\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}%;
%
\LP@HT@init@prefix{LP@HT}%
\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}{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]{%
\setkeys{hitori.sty}{#1}\
\newenvironment{hitori}{\setkeys{hitori}{#1}\LP@set@package{hitori}\LP@set@env@prefix{LP@HT}\setcounter{LP@rows}{\LP@HT@rows}\setcounter{LP@columns}{\LP@HT@columns}\stepcounter{LP@rows}\stepcounter{LP@columns}\begin{minipage}[t]{\LP@HT@width}\ifthenelse{\equal{\LP@HT@title}{}}{\par\enspace\par}{\enspace\par\noindent\hspace{\LP@HT@titleindent}\parbox{\LP@HT@titlewidth}{\strut\LP@titleformat{\LP@HT@title}}\vspace{3mm}\par}\begin{tikzpicture}[LPpreset,scale=\LP@HT@scale]\LP@drawbackground{1}{1}{\LP@HT@columns}{\LP@HT@rows}{\LP@HT@bgcolor}\LP@drawgrid{1}{1}{\LP@HT@columns}{\LP@HT@rows}{1cm}\end{tikzpicture}\LP@drawcounter{\LP@HT@counterstyle}\stepcounter{LP@puzzlecounter}\end{minipage}}%

% kakuro environment and options
\newcommand\LP@KKR@init@prefix{LP@KKR}\newcommand\LP@KKR@init@package{kakuro}\LP@define@key\LP@KKR@init@prefix{\LP@KKR@init@package}{rows}{5}\LP@define@key\LP@KKR@init@prefix{\LP@KKR@init@package}{columns}{5}\LP@define@key\LP@KKR@init@prefix{\LP@KKR@init@package}{scale}{1}\LP@define@key\LP@KKR@init@prefix{\LP@KKR@init@package}{counterstyle}{none}\LP@define@key\LP@KKR@init@prefix{\LP@KKR@init@package}{bgcolor}{}\LP@define@key\LP@KKR@init@prefix{\LP@KKR@init@package}{color}{green}\LP@define@key\LP@KKR@init@prefix{\LP@KKR@init@package}{width}{5.1cm}\LP@define@key\LP@KKR@init@prefix{\LP@KKR@init@package}{cvoffset}{-23pt}\LP@define@key\LP@KKR@init@prefix{\LP@KKR@init@package}{titleindent}{8cm}\LP@define@key\LP@KKR@init@prefix{\LP@KKR@init@package}{titlewidth}{5.1cm}\LP@define@choicekey\LP@KKR@init@prefix{\LP@KKR@init@package}{fontsize}{\LP@KKR@init@package}{Large}\ifLP@KKR@solution\LP@KKR@solutionfalse\LP@define@choicekey\LP@KKR@solution\LP@KKR@solutionfalsefalse[true]false[true]}%
For the \texttt{kakuro} environment, we need to typeset numbers and special Kakuro cells (\KKR).

\begin{verbatim}
\def\LP@KKR@print@element#1{% 
  \ifnum9<\numexpr#1\relax% 
    \ifLP@KKR@solution% 
      \setcell{\arabic{LP@counti}}{\arabic{LP@countii}}{#1}% 
    \else% 
      \fi% 
    \else% 
      #1% 
    \fi% 
}
\end{verbatim}

For the \texttt{kakuro} environment, we need special row and column commands, which can also handle the \KKR commands.

\begin{verbatim}
\newcommand*\kakurorow[2]{% 
\end{verbatim}

\begin{verbatim}
\def\LP@KKR@print@element#1{% 
  \ifnum9<\numexpr#1\relax% 
    \ifLP@KKR@solution% 
      \setcell{\arabic{LP@counti}}{\arabic{LP@countii}}{#1}% 
    \else% 
      \fi% 
    \else% 
      #1% 
    \fi% 
}
\end{verbatim}
\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@counti}%;
% }
% \setcounter{LP@counti}{1}
% \setcounter{LP@countii}{#1}
% \foreach \LP@element in {#2}
% {\LP@KKR@print@element{\LP@element}
% \stepcounter{LP@countii}%;
% }

\KXR
\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.
\KXR{⟨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})
% \begin{pgfonlayer}{LPdump}\
% \node [shape=rectangle,inner sep=0pt] (A_{\theLP@counter@unique})

112
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}\stepcounter{LP@rows}\setcounter{LP@columns}{\LP@KKR@columns}\stepcounter{LP@columns}\begin{minipage}[t]{\LP@KKR@width}\ifthenelse{\equal{\LP@KKR@title}{}}{\par\enspace\par}% empty\else\enspace\par
\hspace{\LP@KKR@titleindent}\parbox{\LP@KKR@titlewidth}{\strut\LP@titleformat}\LP@KKR@title\vspace{3mm}\par\fi
\begin{tikzpicture}\LP@drawbackground{1}{1}{\LP@KKR@columns}{\LP@KKR@rows}{\LP@KKR@bgcolor}}{\end{tikzpicture}}
```

113
\begin{tikzpicture}
\end{tikzpicture}
\stepcounter{LP@puzzlecounter}
\end{minipage}
\end{align}

\begin{minipage}{5.1cm}
% killersudoku environment and options
% killersudoku
\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@package}{Large}\let\killersudokucell\setcell\newcommand*{\killersudokusetup}[1]{\setkeys{killersudoku.sty}{#1}}\setkeys{killersudoku.sty}{#1}

killersudoku
\newenvironment{killersudoku}[1][\setkeys{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}{}}{\enspace\par}% empty\enspace\par\noindent\hspace{\LP@KSDK@titleindent}\parbox{\LP@KSDK@titlewidth}{\strut\LP@titleformat{\LP@KSDK@title}\begin{tikzpicture}[LPpreset, scale=\LP@KSDK@scale]\LP@set@background{1}{1}{\LP@KSDK@rows}{\LP@KSDK@bgcolor}\LP@drawgrid{1}{1}{\LP@KSDK@columns}{\LP@KSDK@rows}{1cm}\end{minipage}}\end{minipage}
Place a cross in the bottom left corner of the grid cell.
\placecross{⟨column⟩}{⟨row⟩}

Place a mirror in the bottom left corner of the grid cell.
\placemirror

\placearrow{⟨column⟩}{⟨row⟩}{⟨direction⟩} may be: 
RightUp, LeftUp, LeftDown or RightDown

\LP@Arrow We define a Ti\kZ picture for an arrow in four directions: 
RightUp, LeftUp, LeftDown or RightDown

\LP@Cross We define a Ti\kZ picture of a cross.
We define a TikZ picture for a mirror.

\LP@Mirror{(direction)}

Based on \textit{(direction)}, we define the angle of the mirror.

\LP@set@LP@scale{\LP@env@prefix}
\begin{tikzpicture}
Mirrors should be on top of everything, therefore we draw them on the LPforegroundtwo layer and use the predefined color LP@c@mirror.

\begin{pgfonlayer}{LPforegroundtwo}
\pgfmathsetmacro{\LPlinewidth}{3pt*\LP@scale}
\draw[line width=\LPlinewidth,scale=\LP@scale,color=LP@c@mirror,rotate=\LP@rotate](.35,.5) -- (0.65,.5);\end{pgfonlayer}
\end{tikzpicture}

\newenvironment{laserbeam}{\setkeys{laserbeam}{#1}}{\LP@drawcounter{\LP@LB@counterstyle}}
For the \texttt{lpsudoku} and skyscrapers environments, we need to add the typical thick Sudoku lines to the standard grid.

\begin{verbatim}
\LP@drawsudokugrid
\end{verbatim}
For the magnets environment, we need to add some lines to the standard grid.

First, we copy the values of \texttt{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.

Now, we can draw the additional lines and the + and - signs.

```latex
\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\footnotesize] at (0.5,\arabic{LP@counti}.5) {$-$};
\node[font=\LP@fontsize\bfseries\footnotesize] at (-0.5,\arabic{LP@counti}.5) {$+$};
```

\lpsudoku
\newenvironment{magiclabyrinth}{\setkeys{magiclabyrinth}{#1}}{\LP@set@package{magiclabyrinth}\setcounter{LP@rows}{\LP@ML@rows}\setcounter{LP@columns}{\LP@ML@columns}\begin{minipage}[t]{\LP@ML@width}\ifthenelse{\equal{\LP@ML@title}{}}{}{\enspace\par\noindent\hspace{\LP@ML@titleindent}\parbox{\LP@ML@titlewidth}{\strut\LP@titleformat{\LP@ML@title}}\vspace{3mm}\par}\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}}%
\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@element};\stepcounter{LP@counti}};}
\newcommand*{\magnetssetup}[1]{\setkeys{magnets.sty}{#1}}
\PMH
\newcommand*{\PMH}[1]{\LP@set@LP@fontsize{\LP@env@prefix}\LP@set@LP@rows{\LP@env@prefix}\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}]{\pm};\stepcounter{LP@counti}\node[font=\LP@fontsize{\bfseries}]{\mp};}
\MPH
\newcommand*{\MPH}[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}]{\pm};\stepcounter{LP@counti}\node[font=\LP@fontsize{\bfseries}]{\mp};}
\MPH
\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@titleformat{\LP@MN@title}} \vspace{3mm} \par}% \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}} \newcommand*{\MasyuB}{\begin{tikzpicture}[scale=0.7*\LP@scale] \fill[color=black] (0.5,0.5) circle (0.5cm); \end{tikzpicture}} \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}} \newenvironment{masyu}[1][]{\setkeys{masyu}{#1}}{\setkeys{masyu}{#1}}
We define a TiKZ picture for a mine.

\LP@Mine

\begin{tikzpicture}[scale=\LP@scale]
\end{tikzpicture}
We simply draw a shaded ball with four spikes.

\begin{tikzpicture}
\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

\newenvironment{minesweeper}{\setkeys{minesweeper}{#1}}{\LP@drawcounter}\LP@drawcounter

\setkeys{minesweeper}{rows}{5}\setkeys{minesweeper}{columns}{5}\setkeys{minesweeper}{extracells}{5}\setkeys{minesweeper}{helplines}{5}
\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}}{\LP@fe@length}\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{\arabic{LP@countiii}}{\LP@fe@row}\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}{\fillcell{\arabic{\LP@line}}{\arabic{\LP@countiii}}\addtocounter{LP@countiii}{1}\addtocounter{LP@whiledo@i}{-1}));%}
\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}}{\LP@fe@length}\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{\arabic{LP@countiii}}{\LP@fe@row}\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}{\fillcell{\arabic{\LP@line}}{\arabic{\LP@countiii}}\addtocounter{LP@countiii}{1}\addtocounter{LP@whiledo@i}{-1}));%}
\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}[]%
\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}{}}{}{\enspace\par
\noindent\hspace{\LP@NG@titleindent}\parbox{\LP@NG@titlewidth}{\strut\LP@titleformat{\LP@NG@title}}}%
\begin{tikzpicture}[
\LP@set@library{tikzpicture}{\LP@NG@library}]
\LP@drawbackground\{1\}\{\LP@NG@columns\}\{\LP@NG@rows\}\{\LP@NG@bgcolor\}%
\LP@drawgrid\{1\}\{\LP@NG@columns\}\{\LP@NG@rows\}\{1cm\}%
\framepuzzle%
\ifthenelse{\equal{\LP@NG@solution}{false}}{\LP@nonogramgrid}{}%
\end{tikzpicture}%
\end{minipage} %
\end{tikzpicture}\LP@drawcounter{LP@NG@counterstyle}\stepcounter{LP@puzzlecounter}\end{minipage}
For the resuko environment, we need to draw different tiles of track segments. First, we start with a curve from the bottom to the left.
A generic command for drawing track curves with a rotation \([\langle\text{angle}\rangle]\).

\[
\text{\LP@G@trackC[\langle\text{angle}\rangle]}
\]

\newcommand{\LP@G@trackC}[1][0]{%
\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}%
}

We draw a TikZ 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.

135
We need horizontal and vertical straights, therefore we allow an rotate angle.

\newcommand*{\trackS}[1]{% 
\begin{tikzpicture}[scale=\scale] 
\clip (0,0) rectangle (1,1); 
\draw[color=LP@track, line width=.3cm*\scale] (.5,0)--(.5,1); 
\end{tikzpicture}}% 

\StraightV \newcommand*{\StraightV}{\trackS} 

\StraightH \newcommand*{\StraightH}{\trackS[90]} 

\Straight \let\Straight=\StraightV 

Crossings are needed as well!

\newcommand*{\trackCR}{% 
\begin{tikzpicture}[scale=\scale] 
\draw[color=LP@track, line width=.3cm*\scale] (0,0.5)--(.5,.5); 
\draw[color=LP@track, line width=.3cm*\scale] (0.5,0)--(.5,.2); 
\draw[color=LP@track, line width=.3cm*\scale] (0.5,1)--(.5,.8); 
\end{tikzpicture}}% 

136
Same game again, we need also rotated versions.

\newcommand*{\LP@G@trackCR}[1][0]\
\begin{tikzpicture}[scale=#1, rotate=#1]
\clip (0,0) rectangle (1,1);
\draw[color=LP@c@track, line width=.3cm*#1] (0,0.5)--(1,.5);
\draw[color=LP@c@track, line width=.3cm*#1] (0.5,0)--(.5,.2);
\draw[color=LP@c@track, line width=.3cm*#1] (0.5,1)--(.5,.8);
\end{tikzpicture}
\newcommand*{\CrossH}{\LP@G@trackCR}
\newcommand*{\CrossV}{\LP@G@trackCR[90]}
\let\Cross=\CrossH

We also need a gravel trap.

\newcommand*{\LP@graveltrap}{%\
\begin{tikzpicture}[scale=1]
\clip (0,0) rectangle (1,1);
\fill[color=LP@c@track] (.5,.5) circle (.1cm);
\end{tikzpicture}\nlet\Graveltrap=\LP@graveltrap%

And a parking lot!

\newcommand*{\parkinglot}[2]\
\begin{tikzpicture}[scale=1,\LP@set@LP@color,\LP@set@LP@bgcolor,\LP@set@LP@fontsize]
\ifthenelse{\equal{\LP@color}{}}{\gdef\LP@c@parkinglot{black}}{\gdef\LP@c@parkinglot{\LP@color}}\n\ifthenelse{\equal{\LP@bgcolor}{}}{\gdef\LP@c@bg@parkinglot{white}}{\gdef\LP@c@bg@parkinglot{\LP@bgcolor}}
\stepcounter{LP@counti} \stepcounter{LP@countii}
\draw[color=\LP@c@parkinglot, line width=\LP@normallines, %

And finally, the pitlane!

This macro provides the "box" placed next to the grid, showing the straights, curves and crossings, which are needed in that line!

This macro provides the "box" placed next to the grid, showing the straights, curves and crossings, which are needed in that line!
\trackV \textbf{This macro places the vertical track lines and expects a csv list in the format straight/crossings!}
\trackV{\textit{csv list}}
\newcommand*{\trackV}[1]{%}
\LP@set@LP@columns{\LP@env@prefix}%
\setcounter{LP@counti}{\LP@columns}%
\setcounter{LP@countii}{1}%
\foreach \LP@c@straight/\LP@c@cross in {#1}%
\LP@c@straight \LP@c@cross\LP@env{\LP@trackline{\LP@c@straight}{\LP@c@cross}{\LP@c@cross}};%
}\stepcounter{LP@countii}%;%
}
\trackH \textbf{The same for the horizontal track lines!}
\trackH{\textit{csv list}}
\newcommand*{\trackH}[1]{%
\setcounter{LP@counti}{1}%
\setcounter{LP@countii}{1}%
\foreach \LP@c@straight/\LP@c@cross in {#1}%
\LP@c@straight \LP@c@cross\LP@env{\LP@trackline{\LP@c@straight}{\LP@c@cross}{\LP@c@cross}};%
\stepcounter{LP@counti}%;%
}
\track \textbf{We still need to draw the race track. The used decoration - which allows automatic over-/underbridges - was provided by Frédéric in this question!}
\track{\textit{Tikz path}}
\newcommand*{\track}[1]{%}
\LP@set@LP@scale{\LP@env@prefix}%
\LP@set@LP@bgcolor{\LP@env@prefix}%
\ifthenelse{\equal{\LP@bgcolor}{}}%
{\gdef\LP@c@bg@track{white}}%
{\gdef\LP@c@bg@track{\LP@bgcolor}}%
\bggroup%
For the `schatzsuche` environment, we need to draw diamonds.

\begin{tikzpicture}[scale=0.9]
\clip (0.05,0.905) rectangle (0.95,0.1);
\draw[line width=0.1mm,fill=yellow!20] (0.5,0.1) -- (0.7,0.9) -- (0.3,0.9) -- cycle;
\draw[line width=0.1mm,fill=orange!20] (0.5,0.1) -- (0.7,0.9) -- (0.9,0.8) -- cycle;
\draw[line width=0.1mm,fill=orange!20] (0.5,0.1) -- (0.3,0.9) -- (0.1,0.8) -- cycle;
\draw (0.1,0.8) -- (0.9,0.8);
\end{tikzpicture}

A user command for drawing diamonds.

\let\Diamond\LP@Diamond

\begin{minipage}{0.5\textwidth}
\ifthenelse{\equal{\LP@SS@title}{}{\enspace\par
\noindent\hspace{\LP@SS@titleindent}\parbox{\LP@SS@titlewidth}{\strut\LP@titleformat{\LP@SS@title}}\vspace{3mm}\par}}{}\enspace\par
\end{minipage}
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.

\LP@drawgriddots

The dots are drawn immediately after the grid, so we can recycle LP@counti and LP@countii.

\newcommand*{\LP@drawgriddots}{% 
\pgfmathsetmacro{\START}{1}% 
\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][1]{% 
\setkeys{slitherlink}{#1}[1]%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
(144)
For several environments, we need to draw stars.

\LP@Star For several environments, we need to draw stars.

\LP@Star% %
\tikz\node[shape=star,fill=yellow,draw,\scale,\]{\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}
The we define some arrows with standard angles.

\newcommand\Right{\LP@c@stararrow(0)}
\newcommand\RightUp{\LP@c@stararrow(45)}
\newcommand\Up{\LP@c@stararrow(90)}
\newcommand\LeftUp{\LP@c@stararrow(135)}
\newcommand\Left{\LP@c@stararrow(180)}
\newcommand\LeftDown{\LP@c@stararrow(225)}
\newcommand\Down{\LP@c@stararrow(270)}
\newcommand\RightDown{\LP@c@stararrow(315)}
\LP@DarkCloud For the sunandmoon environment, we need to draw a dark cloud and differently lighted moons.

\LP@DarkCloud
\newcommand*{\LP@Moon}{\begin{tikzpicture}[scale=0.7*\LP@scale]\fill[color=black!60](0,0) circle (0.5cm);\end{tikzpicture}}
\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}}
\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}}
\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}}
\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}}
\newcommand*{\LP@MoonTL}{\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}}
For the tentsandtree environments, we need to draw trees.

```
\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 (0,8) .. (0,10);%}
\end{tikzpicture}
```
\Tree \ A user command for drawing a tree.
\let \Tree \LP@Tree\\n
\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}}\\n
\Tent \ A user command for drawing tents.
\let \Tent \LP@Tent\\n
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
\hspace{\LP@TAT@titleindent}\parbox{\LP@TAT@titlewidth}{\strut\LP@titleformat}\vspace{3mm}\par}\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}}\\n
% tunnel environment and options
\newcommand*{\LP@TN@init@prefix}{LP@TN}\\n\newcommand*{\LP@TN@init@package}{tunnel}\\n
% 153
We define a Ti\textit{k}Z picture that symbolizes a tunnel portal.
\begin{verbatim}
\newcommand*{\LP@Portal}{\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}}
\end{verbatim}

We define a user command to set a tunnel portal into a grid cell.
\begin{verbatim}
\newcommand*{\portal}[2]{\begin{puzzlebackground}
\LP@G@setcellcontent[hcenter,vcenter]{#1}{#2}{\LP@Portal}
\end{puzzlebackground}}
\end{verbatim}

To avoid interference with the grid lines, we use the \texttt{LPbackground} layer.
\begin{verbatim}
\begin{puzzlebackground}
  \LP@G@setcellcontent[hcenter,vcenter]{#1}{#2}{\LP@Portal}
\end{puzzlebackground}
\end{verbatim}

For the tunnel environments, we need to draw tubes.

\texttt{\LP@Portal}

\texttt{\portal}

\texttt{\tube}
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.

\begin{macrocode}
\def\LP@rel@tikzpath{.5}
% We draw on the \layer{LPbackgroundtwo} layer to get a segmented look behind the grid lines.
\begin{pgfonlayer}{LPbackgroundtwo}
\pgfsetcornersarced{\pgfpoint{3mm}{3mm}}
We draw the tube several times with slightly different colors to get a 3D effect.
\begin{macrocode}
\draw[color=LPC@tunnel!80!black, line width=.4cm*\LP@scale] #1;
\draw[color=LPC@tunnel, line width=.38cm*\LP@scale] #1;
\draw[color=LPC@tunnel!85, line width=.35cm*\LP@scale] #1;
\draw[color=LPC@tunnel!70, line width=.32cm*\LP@scale] #1;
\draw[color=LPC@tunnel!55, line width=.29cm*\LP@scale] #1;
\draw[color=LPC@tunnel!45, line width=.25cm*\LP@scale] #1;
\draw[color=LPC@tunnel!35, line width=.2cm*\LP@scale] #1;
\end{pgfonlayer}
\end{macrocode}
\end{macrocode}
\LP@drawcounter{\LP@TN@counterstyle}\
\stepcounter{LP@puzzlecounter}\
\end{minipage}\
\}

\end{lpenv}
5 References


6 Change History

v1.0
General: first version of battleship.sty on CTAN ............ 72

v1.1
General: added \placeisland .......... 99
added \placerwater ............ 99
added \battleshipssetup for re-
setting global options .......... 100
implemented \placeship; place-
ment of a complete ship ........ 99
implemented \BS@ingrid checks
if coordinate is in the grid →
PackageError .................. 77
renamed \ship → \placesegment
................................ 98

v1.2
General: added \classicgame for
drawing game sheets for clas-
ic Battleship ...................... 100
added option bgcolor .......... 97
added option counterstyle .. 97
added option title ............ 97
moved code base for logic puz-
zles into logicpuzzle.sty ...... 72

v1.3
General: added option cvoffset .. 97
added support for bokkusu puz-
ze ..................... 72
added support for bokkusu puz-
ze ..................... 72
\fillcell: added \LP@ingrid to
\fillcell .................. 84
\LP@setcolumncontents: Bugfix: mix up column and row .... 80
\LP@setrowcontents: Bugfix: mix up column and row .... 79

v1.4
General: added support for
skyline puzzle ............... 72
\fillarea: added \fillarea .... 85
\framearea: added \framearea .. 85
\setcell: added \setcell ..... 81
\setcolumn: added \setcolumn . 81
\setrow: added \setrow ....... 81
puzzlebackground: added
puzzlebackground ............ 89
puzzleforeground: added
puzzleforeground ............ 89

v1.5
General: added support for hakyuu
puzzle .................... 72
added support for skyline and
chaossudoku puzzles ........... 72

v1.6
General: added support for
lpsudoku puzzle ............. 72

v1.7
General: added support for
ddsudoku puzzle ........... 72
added support for hitori puzzle
72

v1.8
General: added support for
killersudoku and kendokut
puzzles .................... 72
\colorarea: added \colorarea
(\framearea without frame) . 86

v1.9
General: added support for
laserbeam puzzle ............ 72
added support for slitherlink
puzzle ..................... 72

v2.0
General: added support for
minesweeper puzzle ........... 72
added support for tunnel puzzle
72
moved code from packages to
logicpuzzle.sty ............. 72

v2.1
General: added support for
kakuro puzzle ............... 72
added support for schatzsuche
puzzle .................... 72
added support for tentsandtrees
puzzle ..................... 72

v2.2
General: added support for
bridges puzzle .............. 72
added support for starbattle
puzzle .................... 72
added support for sunandmoon
and starsandarrows puzzles 72
reduced counters to \LP@rows
and \LP@columns instead of
\LP@XX@... .............. 73
v2.3
General: added support for
fourwinds puzzle ............ 72
added support for numberlink
puzzle ........................ 72
added support for resuko puzzle 72
\framepuzzle: Bugfix: \framepuzzle
only worked correctly for
quadratic puzzles ............ 88
v2.4
General: added support for
magiclabyrinth snd masyu
puzzles ....................... 72
added support for magnets puz-
kle ............................ 72
\setcells: added \setcells ... 82
v2.5
General: added LPpreset TikZ
style to all puzzle environments 72
added support for nonogram
puzzle. Feature request by:
Theresa Spannbauer ........... 72
\LP@drawgriddots: removed un-
necessary scale update .... 144
### Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>@Ship</td>
<td>852, 861</td>
</tr>
<tr>
<td>@ShipB</td>
<td>855</td>
</tr>
<tr>
<td>@ShipC</td>
<td>853, 863, 864, 865, 866, 867, 868, 869, 870</td>
</tr>
<tr>
<td>@ShipL</td>
<td>856, 862, 863, 864, 865, 866, 867, 868, 869, 870</td>
</tr>
<tr>
<td>@ShipR</td>
<td>857, 862, 863, 864, 865, 866, 867, 868, 869, 870</td>
</tr>
<tr>
<td>@ShipT</td>
<td>854</td>
</tr>
</tbody>
</table>

### B

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>battleship (environment)</td>
<td>1001</td>
</tr>
<tr>
<td>\battleshipsetup</td>
<td>13, 968</td>
</tr>
<tr>
<td>bgcolor (Puzzle option)</td>
<td>9, 24, 25, 27, 28, 32, 34, 35, 37, 39, 41, 43, 45, 46, 48, 49, 51, 53, 54, 56, 60, 62, 63, 65, 66, 69, 70, 72</td>
</tr>
<tr>
<td>\Black</td>
<td>15, 1544</td>
</tr>
<tr>
<td>bokkusu (environment)</td>
<td>1054</td>
</tr>
<tr>
<td>bokkususetup</td>
<td>6</td>
</tr>
<tr>
<td>\bridge</td>
<td>14, 1142</td>
</tr>
<tr>
<td>bridges (environment)</td>
<td>1146</td>
</tr>
<tr>
<td>\bridgescell</td>
<td>1097</td>
</tr>
<tr>
<td>\bridgescolumn</td>
<td>14, 1122</td>
</tr>
<tr>
<td>\bridgesrow</td>
<td>14, 1103</td>
</tr>
<tr>
<td>\bridgesetup</td>
<td>14, 1099</td>
</tr>
</tbody>
</table>

### C

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>chaossudoku (environment)</td>
<td>1204</td>
</tr>
<tr>
<td>\chaossudoku</td>
<td>14, 1198</td>
</tr>
<tr>
<td>\chaossudokusetup</td>
<td>14, 1200</td>
</tr>
<tr>
<td>\classicgame</td>
<td>13, 973</td>
</tr>
<tr>
<td>\Cloud</td>
<td>21, 3120</td>
</tr>
<tr>
<td>color (Puzzle option)</td>
<td>9, 10, 27, 28, 32, 37, 48, 51, 53, 54</td>
</tr>
<tr>
<td>\colorarea</td>
<td>10, 449</td>
</tr>
<tr>
<td>columns (Puzzle option)</td>
<td>7–10, 23, 25, 27, 28, 30, 32, 34, 35, 37, 39, 41, 43, 44, 46, 47, 49, 50, 52, 54, 56, 60, 61, 63, 64, 69, 70, 72</td>
</tr>
<tr>
<td>counterstyle (Puzzle option)</td>
<td>9, 24, 25, 27, 29, 30, 32, 34, 36, 37, 40, 42, 43, 45, 46, 48, 49, 51, 53, 54, 56, 60, 62, 63, 65, 66, 69, 70, 72</td>
</tr>
</tbody>
</table>

---

162
\LP@BG@rows . 1151, 1160, 1167, 1170
\LP@BG@scale ............ 1159
\LP@BG@title ......... 1156, 1158
\LP@BG@titleindent .... 1158
\LP@BG@titlewidth ...... 1158
\LP@BG@width .......... 1155
\LP@bgcolor ............. 41, 206, 208, 341, 343, 696, 1299, 1301, 1521, 1523, 2585, 2587, 2667, 2669
\LP@BK@bgcolor ........ 1068
\LP@BK@columns 1060, 1068, 1069
\LP@BK@counterstyle .... 1073
\LP@BK@init@package 1030, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043
\LP@BK@init@prefix 1029, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043
\LP@BK@rows ..... 1059, 1068, 1069
\LP@BK@title ........... 1067
\LP@BK@titleindent .... 1066
\LP@BK@titlewidth ..... 1066
\LP@BK@width ........... 1063
\LP@Block ... 414, 532, 534, 639
\LP@bottomrow . 580, 1045, 2813
\LP@bottomrow@edge . 584, 1689
\LP@BS@bgcolor ... 1016
\LP@BS@columns .. 893, 900, 905, 1008, 1016, 1017
\LP@BS@counterstyle ... 1021
\LP@BS@init@package . 822, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838
\LP@BS@init@prefix .. 821, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838
\LP@BS@printship ... 859, 878
\LP@BS@printshipbox 873, 1022
\LP@BS@rows 893, 900, 905, 1007, 1016, 1017
\LP@BS@shindent ........ 1022
\LP@BS@sbshipscale .. 861, 862, 863, 864, 865, 866, 867, 868, 869, 870
\LP@BS@shbbox . 843, 844, 845, 846, 847, 848, 849, 850, 1015
\LP@BS@shipbox . 840, 912, 1022
\LP@BS@shipcolor 843, 844, 845, 846, 847, 848, 852, 853, 854, 855, 856, 857
\LP@BS@title ........... 1012, 1014
\LP@BS@titleindent ..... 1014
\LP@BS@titlewidth ..... 1014
\LP@BS@width .......... 1011
\LP@c@bg@fw ........ 1300, 1301, 1311
\LP@c@bg@parkinglot 2586, 2587, 2593
\LP@c@bg@track 2668, 2669, 2676
\LP@c@cross . 2644, 2647, 2655, 2659
\LP@c@curve . 2644, 2647, 2655, 2659
\LP@c@parkinglot .. 2583, 2584, 2592, 2594
\LP@c@pitlane 2602, 2603, 2605, 2607
\LP@c@stararrow ... 2996, 2997, 2998, 2999, 3000, 3001, 3002, 3003, 3004
\LP@c@straight .... 2644, 2647, 2655, 2659
\LP@color ............... 40, 526, 641, 680, 1114, 1133, 1306, 1532, 1823, 1830, 1836, 2304, 2641
\LP@counterstyle ........ 45
\LP@Cross ............ 1705, 1727
\LP@CS@bgcolor ........ 1218
\LP@CS@columns 1210, 1218, 1219
\LP@CS@counterstyle ... 1223
\LP@CS@init@package 1184, 1186, 1187, 1188, 1189, 1190, 1191, 1192, 1193, 1194,
<table>
<thead>
<tr>
<th>Command</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>\LP@KD@scale</td>
<td>1612</td>
</tr>
<tr>
<td>\LP@KD@title</td>
<td>1609, 1611</td>
</tr>
<tr>
<td>\LP@KD@titleindent</td>
<td>1611</td>
</tr>
<tr>
<td>\LP@KD@titlewidth</td>
<td>1611</td>
</tr>
<tr>
<td>\LP@KD@width</td>
<td>1608</td>
</tr>
<tr>
<td>\LP@KKR@bgcolor</td>
<td>1566</td>
</tr>
<tr>
<td>\LP@KKR@columns</td>
<td>1558, 1566, 1567</td>
</tr>
<tr>
<td>\LP@KKR@counterstyle</td>
<td>1571</td>
</tr>
<tr>
<td>\LP@KKR@init@package</td>
<td>1440, 1442, 1444, 1445, 1446, 1447, 1448, 1449, 1450, 1451, 1452, 1453</td>
</tr>
<tr>
<td>\LP@KKR@init@prefix</td>
<td>1439, 1442, 1444, 1445, 1446, 1447, 1448, 1449, 1450, 1451, 1452, 1453</td>
</tr>
<tr>
<td>\LP@KKR@print@element</td>
<td>1485, 1502, 1512</td>
</tr>
<tr>
<td>\LP@KKR@rows</td>
<td>1557, 1566, 1567</td>
</tr>
<tr>
<td>\LP@KKR@scale</td>
<td>1565</td>
</tr>
<tr>
<td>\LP@KKR@solution</td>
<td>1457, 1460, 1463, 1468, 1471, 1474</td>
</tr>
<tr>
<td>\LP@KKR@solutionfalse</td>
<td>1455, 1464, 1475</td>
</tr>
<tr>
<td>\LP@KKR@solutiontrue</td>
<td>1461, 1472</td>
</tr>
<tr>
<td>\LP@KKR@title</td>
<td>1562, 1564</td>
</tr>
<tr>
<td>\LP@KKR@titleindent</td>
<td>1564</td>
</tr>
<tr>
<td>\LP@KKR@titlewidth</td>
<td>1564</td>
</tr>
<tr>
<td>\LP@KKR@width</td>
<td>1561</td>
</tr>
<tr>
<td>\LP@KSDK@bgcolor</td>
<td>1660</td>
</tr>
<tr>
<td>\LP@KSDK@columns</td>
<td>1652, 1660, 1661</td>
</tr>
<tr>
<td>\LP@KSDK@counterstyle</td>
<td>1665</td>
</tr>
<tr>
<td>\LP@KSDK@init@package</td>
<td>1626, 1628, 1629, 1630, 1631, 1632, 1633, 1634, 1635, 1636, 1637, 1638</td>
</tr>
<tr>
<td>\LP@KSDK@init@prefix</td>
<td>1625, 1628, 1629, 1630, 1631, 1632, 1633, 1634, 1635, 1636, 1637, 1638</td>
</tr>
<tr>
<td>\LP@KSDK@rows</td>
<td>1651, 1660, 1661</td>
</tr>
<tr>
<td>\LP@KSDK@scale</td>
<td>1659</td>
</tr>
<tr>
<td>\LP@KSDK@title</td>
<td>1656, 1658</td>
</tr>
<tr>
<td>\LP@KSDK@titleindent</td>
<td>1658</td>
</tr>
<tr>
<td>\LP@KSDK@titlewidth</td>
<td>1658</td>
</tr>
<tr>
<td>\LP@KSDK@width</td>
<td>1655</td>
</tr>
<tr>
<td>\LP@LB@bgcolor</td>
<td>1766</td>
</tr>
<tr>
<td>\LP@LB@columns</td>
<td>1758, 1766, 1767</td>
</tr>
<tr>
<td>\LP@LB@counterstyle</td>
<td>1771</td>
</tr>
<tr>
<td>\LP@LB@init@package</td>
<td>1673, 1675, 1676, 1677, 1678, 1679, 1680, 1681, 1682, 1683, 1684, 1685</td>
</tr>
<tr>
<td>\LP@LB@init@prefix</td>
<td>1672, 1675, 1676, 1677, 1678, 1679, 1680, 1681, 1682, 1683, 1684, 1685</td>
</tr>
<tr>
<td>\LP@LB@rows</td>
<td>1757, 1766, 1767</td>
</tr>
<tr>
<td>\LP@LB@scale</td>
<td>1765</td>
</tr>
<tr>
<td>\LP@LB@title</td>
<td>1762, 1764</td>
</tr>
<tr>
<td>\LP@LB@titleindent</td>
<td>1764</td>
</tr>
<tr>
<td>\LP@LB@titlewidth</td>
<td>1764</td>
</tr>
<tr>
<td>\LP@LB@width</td>
<td>1761</td>
</tr>
<tr>
<td>\LP@leftcolumn</td>
<td>588, 1046, 1688, 1947, 2815, 2976, 3172, 3253</td>
</tr>
<tr>
<td>\LP@length</td>
<td>36, 2613, 2616, 2629</td>
</tr>
<tr>
<td>\LP@Line</td>
<td>643, 1144, 2442</td>
</tr>
<tr>
<td>\LP@line</td>
<td>2225, 2226, 2287, 2292</td>
</tr>
<tr>
<td>\LP@Line@double</td>
<td>645, 660, 666</td>
</tr>
<tr>
<td>\LP@Line@linelinecolor</td>
<td>650, 661, 668, 670</td>
</tr>
<tr>
<td>\LP@Line@linewidth</td>
<td>655, 662, 667, 669, 671</td>
</tr>
<tr>
<td>\LP@LP@bgcolor</td>
<td>805</td>
</tr>
<tr>
<td>\LP@LP@columns</td>
<td>794, 804, 806</td>
</tr>
<tr>
<td>\LP@LP@counterstyle</td>
<td>810</td>
</tr>
<tr>
<td>\LP@LP@init@package</td>
<td>749, 751, 753, 755, 757, 759, 761, 763, 765, 767, 769, 771, 774</td>
</tr>
<tr>
<td>\LP@LP@init@prefix</td>
<td>748, 751, 753, 755, 757, 759, 761, 763, 765, 767, 769, 771, 773</td>
</tr>
<tr>
<td>\LP@LP@rows</td>
<td>793, 804, 806</td>
</tr>
<tr>
<td>\LP@LP@scale</td>
<td>803</td>
</tr>
<tr>
<td>\LP@LP@title</td>
<td>798, 801</td>
</tr>
<tr>
<td>\LP@LP@titleindent</td>
<td>800</td>
</tr>
<tr>
<td>\LP@LP@titlewidth</td>
<td>801</td>
</tr>
<tr>
<td>\LP@LP@width</td>
<td>797</td>
</tr>
</tbody>
</table>
\LP@magnetsgrid ... 1818, 2074
\LP@Mine .......... 2171, 2181
\LP@Mirror ....... 1709, 1738
\LP@ML@bgcolor .... 1917
\LP@ML@columns 1909, 1917, 1918
\LP@ML@counterstyle .... 1922
\LP@ML@init@package 1873, 1875, 1876, 1877, 1878, 1879, 1880, 1881, 1882, 1883, 1884, 1885
\LP@ML@rows ..... 1908, 1917, 1918
\LP@ML@scale ......... 1916
\LP@ML@title ...... 1913, 1915
\LP@ML@titleindent .... 1915
\LP@ML@titlewidth .... 1915
\LP@ML@width ....... 1912
\LP@MN@bgcolor .... 2072
\LP@MN@columns 2064, 2072, 2073
\LP@MN@counterstyle .... 2079
\LP@MN@init@package 1930, 1932, 1933, 1934, 1935, 1936, 1937, 1938, 1939, 1940, 1941, 1942
\LP@MN@init@prefix 1929, 1932, 1933, 1934, 1935, 1936, 1937, 1938, 1939, 1940, 1941, 1942
\LP@MN@rows .... 2063, 2072, 2073
\LP@MN@scale ........ 2071
\LP@MN@title ..... 2068, 2070
\LP@MN@titleindent .... 2070
\LP@MN@titlewidth .... 2070
\LP@MN@width .... 2067
\LP@Moon .......... 3056, 3121
\LP@MoonB .... 3081, 3125
\LP@MoonBL .... 3096, 3127
\LP@MoonBR .... 3104, 3128
\LP@MoonL .... 3067, 3123
\LP@MoonR .... 3060, 3122
\LP@MoonT .... 3074, 3124
\LP@MoonTL .... 3088, 3126
\LP@MoonTR .... 3112, 3129
\LP@MS@bgcolor .... 2138
\LP@MS@columns 2188, 2196, 2197
\LP@MS@counterstyle .... 2201
\LP@MS@init@package 2151, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163
\LP@MS@init@prefix 2150, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163
\LP@MS@rows .. 2187, 2196, 2197
\LP@MS@scale ........ 2195
\LP@MS@title ...... 2192, 2194
\LP@MS@titleindent .... 2194
\LP@MS@titlewidth .... 2194
\LP@MS@width .... 2191
\LP@MY@bgcolor .... 2138
\LP@MY@columns 2130, 2138, 2139
\LP@MY@counterstyle .... 2143
\LP@MY@init@package 2087, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100
\LP@MY@init@prefix 2086, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100
\LP@MY@rows .... 2129, 2138, 2139
\LP@MY@scale ........ 2137
\LP@MY@title ...... 2134, 2136
\LP@MY@titleindent .... 2136
\LP@MY@titlewidth .... 2136
\LP@MY@width .... 2133
\LP@NG@bgcolor .... 2393
\LP@NG@columns 2385, 2393, 2394
\LP@NG@counterstyle .... 2404
\LP@NG@extracells ........ . 735, 2306, 2311, 2314, 2325, 2341, 2344, 2358
\LP@NG@helplines .. 2336, 2340, 2350, 2357, 2365
\LP@NG@init@package 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224
\LP@Star \hfill 2982
\LP@TAT@bgcolor \hfill 3225
\LP@TAT@columns \hfill 3217, 3225, 3226
\LP@TAT@counterstyle \hfill 3230
\LP@TAT@init@package \hfill 3157, 3159, 3160, 3161, 3162, 3163, 3164, 3165, 3166, 3167, 3168, 3169
\LP@TAT@init@prefix \hfill 3156, 3159, 3160, 3161, 3162, 3163, 3164, 3165, 3166, 3167, 3168, 3169
\LP@TAT@rows \hfill 3216, 3225, 3226
\LP@TAT@scale \hfill 3224
\LP@TAT@title \hfill 3221, 3223
\LP@TAT@titleindent \hfill 3223
\LP@TAT@titlewidth \hfill 3223
\LP@Tent \hfill 3200, 3210
\LP@thicklines \hfill 50, 443, 447, 555, 557, 560, 563, 725, 736, 1801, 1803, 1805, 1807, 1809, 1811, 1813, 1815, 1833, 1835, 1837, 1895, 1985, 2001, 2017, 2347, 2352, 2353, 2362, 2367, 2369
\LP@titleformat \hfill 46, 606, 801, 1014, 1066, 1158, 1216, 1263, 1331, 1378, 1425, 1564, 1611, 1658, 1764, 1857, 1915, 2070, 2136, 2194, 2391, 2457, 2699, 2760, 2835, 2898, 2946, 3017, 3142, 3223, 3312
\LP@TN@bgcolor \hfill 3314
\LP@TN@columns \hfill 3306, 3314, 3315
\LP@TN@counterstyle \hfill 3319
\LP@TN@init@package \hfill 3238, 3240, 3241, 3242, 3243, 3244, 3245, 3246, 3247, 3248, 3249, 3250
\LP@TN@init@prefix \hfill 3237, 3240, 3241, 3242, 3243, 3244, 3245, 3246, 3247, 3248, 3249, 3250
\LP@TN@rows \hfill 3305, 3314, 3315
\LP@TN@scale \hfill 3305, 3306, 3314, 3315
\LP@TN@title \hfill 3310, 3312
\LP@TN@titleindent \hfill 3312
\LP@TN@titlewidth \hfill 3312
\LP@TN@width \hfill 3309, 2814, 2975, 3171, 3252
\LP@toprow \hfill 592, 1047, 1687, 1946, 2120, 2347, 2352, 2353, 2362, 2367, 2369
\LP@trackC \hfill 2493, 2626
\LP@trackCR \hfill 2537, 2633
\LP@trackline \hfill 2610, 2647, 2659
\LP@tracks@scale \hfill 54, 2614
\LP@Tree \hfill 3178, 3199
\LPbackground (PGF layer) \hfill 7–9, 83, 88, 89, 154
\LPbackgroundtwo (PGF layer) \hfill 7, 8
\LPbgcolor (PGF layer) \hfill 7, 8, 76, 78
\LPdump (PGF layer) \hfill 7, 8, 78, 112
\LPenv.sty (Package) \hfill 97
\LPforeground (PGF layer) \hfill 7–9, 83, 89, 144
\LPforegroundtwo (PGF layer) \hfill 7, 8, 83, 118
\LPlinewidth \hfill 1722, 1723, 1732, 1733, 1745, 1746, 2991, 2992
\lpsudoku (environment) \hfill 1845
\lpsudokucell \hfill 21, 1793
\lpsudoku@setup \hfill 21, 1795

M

magiclabyrinth (environment) \hfill 1903
\magiclabyrinthcell \hfill 16, 1887
\magiclabyrinthsetup \hfill 17, 1889
magnets (environment) \hfill 2058
\magnetscell \hfill 1944
\magnetsH \hfill 17, 2042
\magnetssetup \hfill 17, 1974
\magnetsV \hfill 17, 2050
\main (PGF layer) \hfill 7–9, 78, 117
\masyu (environment) \hfill 2124
\MasyuB \hfill 18, 2108
\masyucell \hfill 17, 2102
\masyuline \hfill 18, 2114
\masyusetup \hfill 18, 2104

173
51, 52, 54, 56, 60, 61, 63, 65, 66, 69, 70, 72

titlewidth (Puzzle option) 8, 24, 25, 27, 28, 30, 32, 34, 35, 37, 39, 41, 43, 45, 46, 48, 49, 51, 53, 54, 56, 60, 62, 63, 65, 66, 69, 70, 72
\track 19, 2663
\trackH 19, 2651
\trackV 19, 2638
\Tree 22, 3199
\tube 22, 3278
\tunnel (environment) 3300
\tunnelH 22, 3252
\tunnelsetup 22, 3255
\tunnelV 22, 3253

U
\Up 21, 2999

V
\valueH 13, 1045
\valueV 13, 1046

W
\Water 850, 906
width (Puzzle option) 8, 23, 25, 27, 28, 30, 32, 34, 35, 37, 39, 41, 43, 44, 46, 47, 49, 50, 52, 54, 56, 60, 61, 63, 64, 66, 69, 70, 72

X
\xtikzpath 11, 484, 1307