

pgf-go



Manual for version 0.2026.10.30

Jonatán Perren Hug
jonatan.perren@gmail.com

October 30, 2026

Contents

I	Introduction	3
1	Overview	3
1.1	Motivation	3
1.2	First steps	3
1.3	How to read this manual	4
II	Tutorials	5
2	Kāru's boards	5
2.1	The first game	5
2.2	A position by hand	6
2.3	Moves	8
2.4	Partial board and marks	9
2.5	Explaining a fight: arrows, letters and views	11
2.6	A ko: captures and notes	13
2.7	A whole game	14
2.8	Variations: remembering a board	16
2.9	The end of the game: territory	17
2.10	A problem book	19
2.11	A wooden board: a profile with images	21
2.12	Slides and animations	24
2.13	A joseki dictionary	25
2.14	Other languages	26
2.15	Writing less: the coordinate syntax	27
III	Reference	28
3	Settings	28
3.1	The compute kernel	30
3.2	Languages	30
3.3	One board per page	31

4	Coordinate syntax	32
4.1	Groups of stones	33
4.2	Additive and subtractive groups	38
5	The board	40
5.1	Grid	42
5.2	Star points	47
5.3	Border	49
5.4	Partial board	56
5.5	Labels of the board	58
6	Stones, moves and prisoners	62
6.1	Stones	62
6.1.1	Images and the size of the PDF	73
6.2	Moves	73
6.3	Prisoners	77
7	Marks, arrows and lines	78
7.1	Arrows and lines	88
7.2	Drawing with TikZ	92
8	The game: captures, notes and territory	92
8.1	Captures	92
8.2	Moves on points already played	94
8.3	Territory	95
9	Remembering boards and views	101
9.1	Remembering a board	101
9.2	Views in the text	103
9.3	A stone in the text	106
9.4	Views before the board	106
9.5	Using a board in another document	107
10	Games in SGF	107
10.1	Names for the files	107
10.2	Sensei's Library diagrams	109
10.3	A game on a board	110
10.4	Information and comments	115
10.5	Animations	117
10.6	Diagrams as in books	120
10.7	Problems	127
11	Layout of the diagrams	131
12	Players and profiles	134
12.1	Players	134
12.2	Profiles	135
13	Catalog of profiles	138
13.1	default	138
13.2	pgfgo	138
13.3	pgfgosmooth	140
13.4	wain	141
13.5	simple	143
13.6	draft	144
13.7	shell	145
13.8	yunzi	146
13.9	baduktv	147
A	What's new	147

Part I

Introduction

1 Overview

1.1 Motivation

The following package offers a collection of macros designed to represent Go boards and positions. This project, although still under development¹, incorporates features aimed at simplifying and streamlining the creation of Go diagrams:

- Reading games in SGF: a game on one board, series of diagrams as in books, animations with buttons, and problems with their solutions.
- The rules of the game, with the Lua kernel: captures, “N at M” notes for kos, and the computation of territory and score.
- Visual profiles for boards and players, which can use images: wood in the background, slate and shell stones.
- Named boards, to follow a variation from a saved position or to show part of a board in the middle of a paragraph.
- Marks, arrows and lines, and marks of your own.
- A grid to lay out diagrams of the same size.
- The texts of the package translated into several languages, and border coordinates in the Western, Japanese or Chinese style.
- Loading multiple coordinates simultaneously using a syntax that some might find familiar.

The development of PGFGO stems from my frustration in 2021 with not finding packages that met my needs. The packages I explored allow for the representation of black-and-white boards, suitable for printing, but impractical when combined with technologies like **beamer** or in documents where a color presentation is desired.

Initially, PGFGO was intended to be limited to these tasks; however, it has evolved a lot since it was published almost two years ago. I used to see it more as an interface for managing styles and states, avoiding excessive use of direct formatting². Now, with the Lua kernel, I see it as a complete suite.

Regarding the name: the drawing is written on the basic layer of PGF, and the computations (coordinates, moves, captures, territory, reading SGF) are done by a *kernel* written in L3 or in Lua (p. 30). The **goban** environment is, all the same, a **tikzpicture**, so that users familiar with TikZ can use its macros on the board (p. 40).

In this version, I used LLMs to translate into English the documentation I wrote in Spanish. This let me focus more on the language I handle best. They were also used to support other languages in the texts of the package (such as the captions).

1.2 First steps

PGFGO is loaded like any other package. I recommend (strongly) compiling with Lua^{La}T_EX: with the other engines the package works, but it cannot read SGF files, remove captured stones or find the territory by itself (p. 30). Obviously, if your intention is to create boards and set up positions for teaching, these features may not be that necessary.

¹But, frankly, finished.

²This is reflected mainly in the structure of the *keys* tree.

```

\documentclass{article}
\usepackage{pgf-go}

\begin{document}
\begin{goban}
\stones[b]{C3, D4}
\stones[w]{E5}
\end{goban}
\end{document}

```

The package can be loaded with two options: `kernel`, which chooses the kernel (p. 30), and `lang`, the language of the texts it writes (p. 30). Everything else is set with `\pgfgoset` (p. 28).

1.3 How to read this manual

The manual has three parts: this introduction; the tutorials (part II), in which Kāru prepares an article, a problem book and some lessons, and learns the package along the way; and the reference (part III), with every command and every *key*, ordered by object. At the end, the catalog shows the profiles that come with the package.

My recommendation would be to go through the tutorials first, in one go, to get an overall picture of everything. Then it is highly recommended to read the reference to see exactly everything that can be configured.

For something specific:

- **Drawing a game saved in SGF:** the tutorial on p. 5 and, for the whole game in diagrams, the one on p. 14. The reference is on p. 107.
- **Drawing a position by hand:** the tutorial on p. 6, and the stones on p. 62. To write fewer coordinates (rows, rectangles, paths), the tutorial on p. 27 and the complete syntax on p. 32.
- **Numbering moves:** the tutorial on p. 8, and `\moves` on p. 73.
- **Showing only part of the board or marking points:** the tutorial on p. 9; `partial` on p. 56, and the marks on p. 78.
- **Arrows, lines and letters:** the tutorial on p. 11, and the reference on p. 88.
- **Showing a stone or part of the board inside the text:** the end of the tutorial on p. 11, and `\pgfgoview` on p. 103. A single stone, without a board: `\pgfgostone` (p. 106). For a view before its board, or of a board from another document, pp. 106 and 107.
- **Captures, kos and “N at M” notes:** the tutorial on p. 13, and the reference on p. 92.
- **Marking the territory and counting the points:** the tutorial on p. 17, and `\territory` and `\pgfgoscore` on p. 95.
- **Saving a board and going on from it, for example to show variations:** the tutorial on p. 16, and `name` and `resume` on p. 101.
- **Josekis from a dictionary in SGF:** the tutorial on p. 25, and `variation` and `variations` on p. 107.
- **Problems (tsumego) with their solutions:** the tutorial on p. 19, and `\sgfproblem` on p. 127.
- **Laying out several diagrams in a grid:** the tutorials on pp. 14 and 19, and `layout` and `pgfgolayout` on p. 131.
- **Changing the style of every board:** the end of the tutorial on p. 6 and the catalog (p. 138). To change a detail, `\pgfgoset` (p. 28) and the *keys* of each object; to save a style of your own with a name, the profiles (p. 135).
- **Using images on the board and on the stones:** the tutorial on p. 21; the background and the grid on p. 40, and the stones on p. 70.
- **Slides and animations:** the tutorial on p. 24, and `\sgfanimation` on p. 117.
- **Writing in another language:** the tutorial on p. 26, and the reference on p. 30.

A general rule: the *keys* that turn something on or off (the grid, the background, the shadows, ...) are turned on alone or with `true`, turned off with `false`, and giving only options does not turn them on (p. 29).

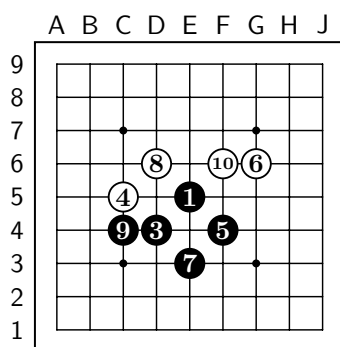
Part II

Tutorials

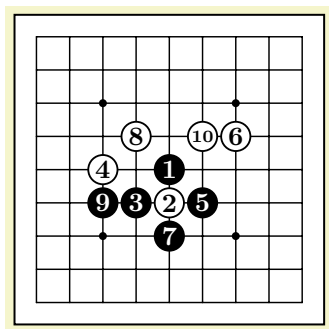
2 Kāru's boards

2.1 The first game

Kāru is writing an article to share with their students. They worry about the final size of the PDF and have noticed that diagramming *software* does not offer satisfactory formats or results. So they decide to try PGFGO. The first thing they want to show is a short game played in class. It is saved in an SGF file, `game.sgf`: the format in which almost every Go program and server saves games.



In PGFGO, every diagram is drawn inside the `goban` environment, and inside it, `\sgf` draws the game of the file. Reading SGF files needs Lua³_{TEX} (p. 30). Since the file will be used many times, Kāru first gives it a name with `\pgfgodefinesgf` (p. 107): from then on it is called `game`, and if the file moves, it is fixed in one place³.

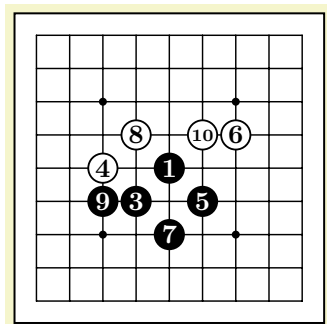


```
\pgfgodefinesgf{game}{game.sgf}

\begin{goban}
  \sgf{game}
\end{goban}
```

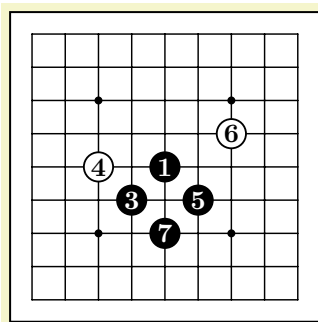
The moves are numbered, as in books. But with move 7 Black captured the white stone 2, which is still on the board: PGFGO draws what it is asked to, without applying the rules. With the *key* `captures`, the `goban` removes the captured stones (p. 92). The options of the board are written in brackets, after `\begin{goban}`.

³Unfortunately, I implemented it well into the documentation.



```
\begin{goban}[captures]
\sgf{game}
\end{goban}
```

To comment on the game in the text, Kāru stops it at move 7 with `to`. The file has a comment for that move, and `\sgfcomment` writes it.



Move 7: Tesuji. Black captures.

```
\begin{goban}[captures]
\sgf[to = 7]{game}
\end{goban}
```

Move 7: `\sgfcomment{game}{7}`

This game is 9×9 , the size of a `goban` by default. For a game of another size, `sgf size = game` takes the size from the file; Kāru will do so with a 19×19 game on p. 14. There they will also see how to draw a long game in a series of diagrams.

2.2 A position by hand

For the introduction of the article, Kāru wants a position of their own, which is not in any file: the one in Figure 1. It has to be drawn stone by stone.

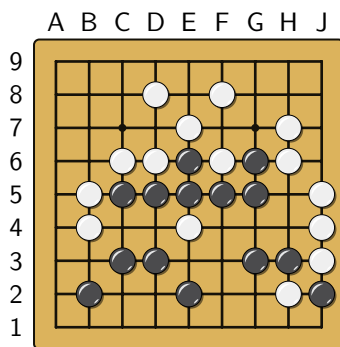
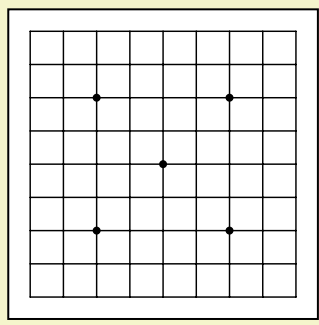


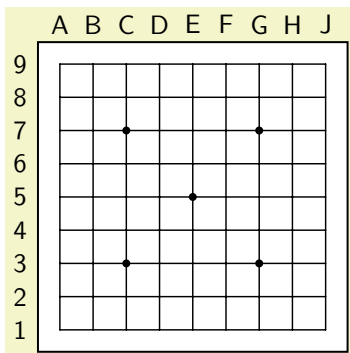
Figure 1

Kāru starts with the empty board. As in the previous tutorial, everything is drawn inside the `goban` environment, which takes care of the size of the board, its color, its coordinates and everything drawn on it.



```
\begin{goban}
\end{goban}
```

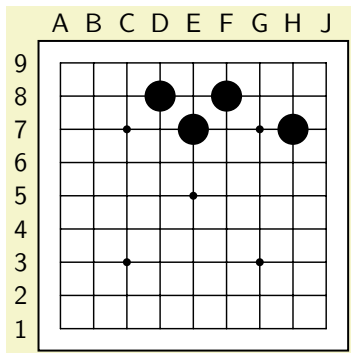
The board is 9×9 , just what is needed, but the coordinates on the borders are missing. The options of the `goban` go in brackets, and those of each part of the board are grouped in three *keys*: `grid`, for the grid; `background`, for the border and the fill, and `label`, for the coordinates (p. 40). Inside `label`, `at` says on which sides they go: `north`, `south`, `east` and `west` (or `above`, `below`, `right` and `left`), `all` or `none`. Kāru wants them above and on the left.



```
\begin{goban}[label = {at = {north, west}}]
\end{goban}
```

Kāru notices that the columns go from H to J: as is customary in Go, PGFGO skips the “I”, both in the coordinates and in the labels on the sides. If it is ever needed, the *key* `goban/use i` (p. 41) brings it back.

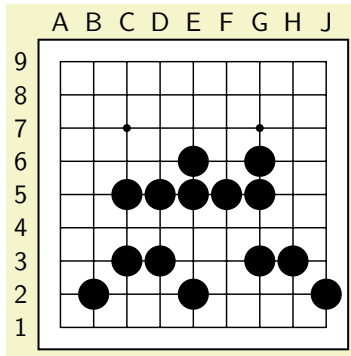
Now, the stones. The simplest is `\stone`, which takes the intersection, such as D8, and its options in brackets: the player, the style, a mark. Without a player, the stone is neutral; for Black, Kāru writes `black`, or `b`.



```
\begin{goban}[label = {at = {north, west}}]
\stone[black]{D8}
\stone[black]{F8}
\stone[b]{E7}
\stone[b]{H7}
\end{goban}
```

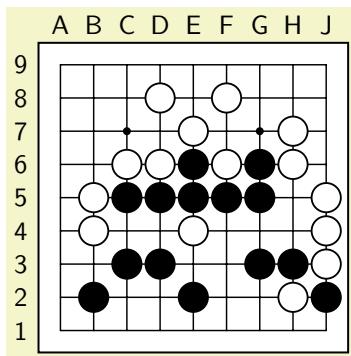
Kāru suspects this is not the best option for adding 200 stones or more. They find `\stones`, which takes a list of intersections separated by commas and accepts the same options as `\stone`.⁴

⁴Between `\stone{A1}` and `\stones{A1}`, the first is much faster. We will see later that many processes are involved in reading the mandatory argument of the plural macros.



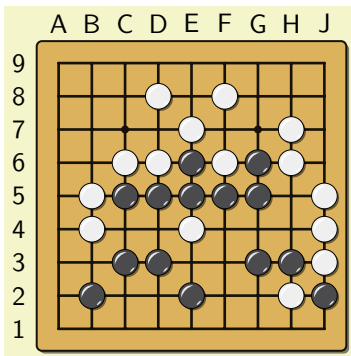
```
\begin{goban}[label = {at = {north, west}}]
\stones[b]{B2,E2,J2,C3,D3,G3,H3,C5,D5,E5,F5,G5,E6,G6}
\end{goban}
```

The white stones take `white`, or `w`. Strictly speaking, `black` and `white` are not colors but players, the first and the second: each stone takes the style of its player (p. 134).



```
\begin{goban}[label = {at = {north, west}}]
\stones[b]{B2,E2,J2,C3,D3,G3,H3,C5,D5,E5,F5,G5,E6,G6}
\stones[w]{D8,F8,E7,H7,C6,D6,F6,H6,B5,J5,B4,E4,J4,J3,H2}
\end{goban}
```

The style is missing. In the catalog (p. 138), Kāru finds one they like: the profile `pgfgo`. They apply it with `\pgfgoset{profile = pgfgo}` (p. 135), and from there on it holds for every board that follows.



```
\pgfgoset{profile = pgfgo}

\begin{goban}[label = {at = {north, west}}]
\stones[b]{B2,E2,J2,C3,D3,G3,H3,C5,D5,E5,F5,G5,E6,G6}
\stones[w]{D8,F8,E7,H7,C6,D6,F6,H6,B5,J5,B4,E4,J4,J3,H2}
\end{goban}
```

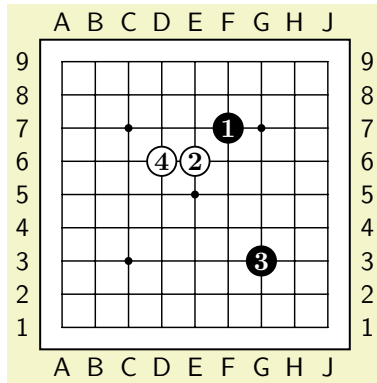
The lists of this tutorial are long, and Kāru's patience is short. In the last tutorial (p. 27), Kāru learns to write them shorter.

2.3 Moves

To explain how a fight began, Kāru wants to show the stones in the order in which they were played. With `\stone`, the player of each one would have to be given:

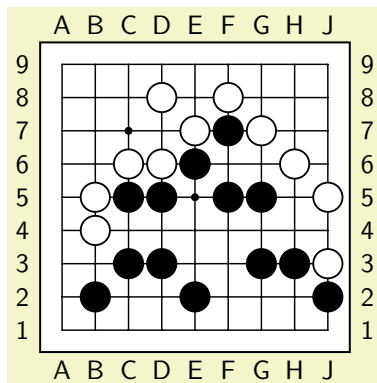
```
\begin{goban}
\stone[b]{F7}
\stone[w]{E6}
\stone[b]{G3}
\stone[w]{D6}
\end{goban}
```

That is what `\moves` is for (p. 73): it takes a list like `\stones`, but plays the intersections in the order in which they are written, alternates the colors and numbers the moves.



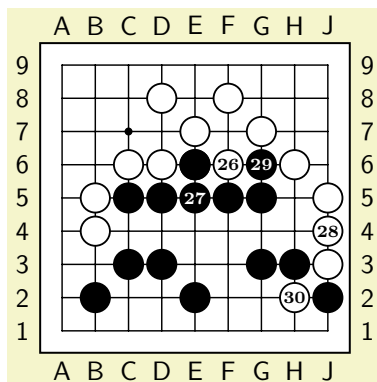
```
\begin{goban}[label = {at = all}]
\moves{F7, E6, G3, D6}
\end{goban}
```

The fight began with a black stone at F7, which Kāru puts with `\stone`. The moves that follow will not be commented on: with `numbers = false`, they are drawn without numbers. With `white`, the sequence starts with White: in `\moves`, black and white only say who plays first.



```
\begin{goban}[label = {at = all}]
\stone[b] {F7}
\moves[numbers = false, white] {E6, G3, D6, F5, E7, E2, F8, E6, D8, D3,
G7, C5, C6, D5, B5, G5, B4, B2, H6, C3,
J3, J2, J5, H3}
\end{goban}
```

The last moves are commented on, so they get numbers. The numbering and the turn go on from one `\moves` to the next, even if the previous moves have no numbers, and `from` chooses the number to go on from.



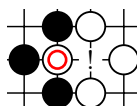
```
\begin{goban}[label = {at = all}]
\moves[numbers = false, white] {E6, G3, D6, F5, E7, E2, F8, E6, D8, D3,
G7, C5, C6, D5, B5, G5, B4, B2, H6, C3,
J3, J2, J5, H3}

\moves[from = 26] {F6, E5, J4, G6}
\moves{H2} % White and labeled 30, without saying so.
\end{goban}
```

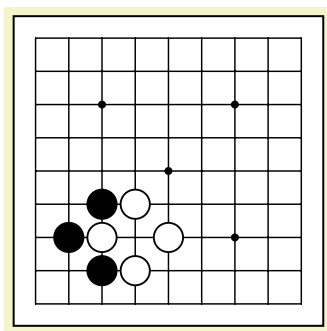
H2 is in another `\moves` and, without saying so, it is white and numbered 30: the previous move, G6, was 29, by Black.

2.4 Partial board and marks

For an exercise of the article, Kāru wants to show only part of the board, with two signs: a circle on the stone in atari and a “!” on the key point.

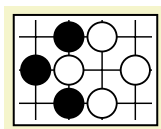


They start, as always, with the stones.



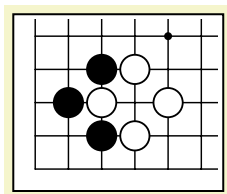
```
\begin{goban}
\stones[b]{C2,B3,C4}
\stones[w]{D4,E3,D2,C3}
\end{goban}
```

The *key partial* says which part of the board to show: two opposite corners of the rectangle, separated by a colon, such as B2:E4 (p. 56).



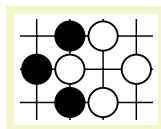
```
\begin{goban}[partial = B2:E4]
\stones[b]{C2,B3,C4}
\stones[w]{D4,E3,D2,C3}
\end{goban}
```

If Kāru does not want to look for the corners, with `partial = auto` the board chooses the part: the one used by what is drawn, with one more line around it. A side two lines or less from the edge reaches the edge, like the bottom and the left ones here.



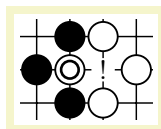
```
\begin{goban}[partial = auto]
\stones[b]{C2,B3,C4}
\stones[w]{D4,E3,D2,C3}
\end{goban}
```

Kāru prefers the tight crop, so they go on with B2:E4. On a partial board, the border of the board is too much: it is removed with `line = false` in the `background`.



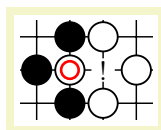
```
\begin{goban}[partial = B2:E4, background = {line = false}]
\stones[b]{C2,B3,C4}
\stones[w]{D4,E3,D2,C3}
\end{goban}
```

Marks are put with `\mark`, or with `\marks` for several intersections; there are many shapes (p. 78). Two are enough for Kāru: `circle`, the circle, and `custom`, which puts any text, such as “!”. A mark can also go in the options of the stone, with `mark`.



```
\begin{goban}[partial = B2:E4, background = {line = false}]
\stones[b]{C2,B3,C4}
\stones[w]{D4,E3,D2}
\mark[custom = {!}] {D3}
\stone[w, mark = circle] {C3}
\end{goban}
```

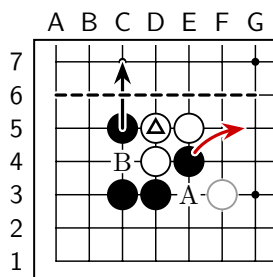
So that the circle stands out, Kāru paints it red, with `line/color` of the mark.



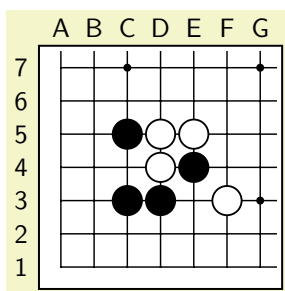
```
\begin{goban}[partial = B2:E4, background = {line = false}]
\stones[b]{C2,B3,C4}
\stones[w]{D4,E3,D2}
\mark[custom = {!}] {D3}
\stone[w, mark = {circle, line = {color = red}}] {C3}
\end{goban}
```

2.5 Explaining a fight: arrows, letters and views

In the second part of the article, Kāru wants to explain a fight. For that they need to point out the possible moves with letters, show where each group runs and, in the text, name the stones without making the reader look for them in the diagram.

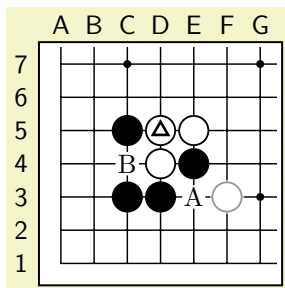


They start, as always, with the stones. To save space, only the corner is shown, with `partial`.



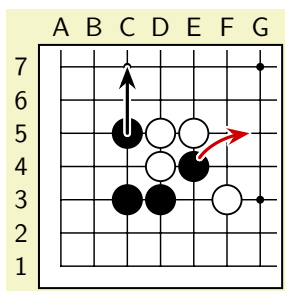
```
\begin{goban}[partial = A1:G7, label = {at = {north, west}}]
\stones[b]{C3, D3, E4, C5}
\stones[w]{D4, D5, E5, F3}
\end{goban}
```

The candidate moves are marked with letters, with `custom`, which accepts any text. The mark `triangle` highlights the stone the text talks about, and `dim` dims the white stone at F3, which no longer matters.



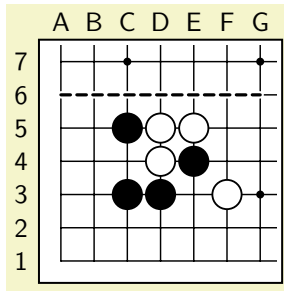
```
\begin{goban}[partial = A1:G7, label = {at = {north, west}}]
\stones[b]{C3, D3, E4, C5}
\stones[w]{D4, D5, E5, F3}
\marks[dim]{F3}
\marks[triangle]{D5}
\mark[custom = A]{E3}
\mark[custom = B]{C4}
\end{goban}
```

To show where the groups run, Kāru uses `\arrow`, which joins two intersections (p. 88). The arrow is drawn with a halo of the color of the board, so that it is not mistaken for the grid. With `bend`, the arrow is an arc; with `shorten`, it does not cover the stones at its ends.



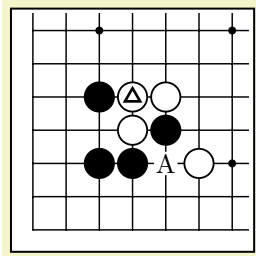
```
\begin{goban}[partial = A1:G7, label = {at = {north, west}}]
\stones[b]{C3, D3, E4, C5}
\stones[w]{D4, D5, E5, F3}
\arrow{C5}{C7}
\arrow[color = red!80!black, bend = 30, shorten = 4pt]{E4}{G5}
\end{goban}
```

`\line` is like `\arrow`, but without a tip, and accepts the same stroke *keys*, such as `dash`.



```
\begin{goban}[partial = A1:G7, label = {at = {north, west}}]
\stones[b]{C3, D3, E4, C5}
\stones[w]{D4, D5, E5, F3}
\line[dash = dashed]{A6}{G6}
\end{goban}
```

Now, the text. Kāru wants to write “the black stone at E4” and show it right there. For that, they give the board a name with `name` and, afterwards, `\pgfgoview` draws part of that board in the middle of the paragraph, as it was at the end (p. 103): a point or a rectangle.



```
\begin{goban}[partial = A1:G7, name = fight]
\stones[b]{C3, D3, E4, C5}
\stones[w]{D4, D5, E5, F3}
\marks[triangle]{D5}
\mark[custom = A]{E3}
\end{goban}
```

The black stone `\pgfgoview{fight}{E4}` is cut off from its companions.
 If Black plays at A, it connects: `\pgfgoview{fight}{C3:E5}`.
 Without marks: `\pgfgoview[marks = false]{fight}{C3:E5}`.

The black stone ● is cut off from its companions. If Black plays at A, it connects: . Without marks: .

The view has the style of the original board, but any option of `goban` can change it: Kāru prefers to remove the background and the coordinates in every view of the article.

```
\pgfgoset{view = {background = false, label = false}}

The black stone \pgfgoview{fight}{E4} is cut off from its companions.
If Black plays at A, it connects: \pgfgoview{fight}{C3:E5}.
```

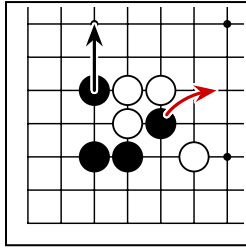
The black stone ● is cut off from its companions. If Black plays at A, it connects: .

Sometimes the text talks about a diagram that comes later, or about a figure that L^AT_EX moves to the next page. A view can also come before its `goban`: as with `\ref`, the `goban` is kept in the `.aux` file and the document has to be compiled twice. The first time, the view is ?? (p. 106).

```
The red arrow of the diagram below starts at the black stone \pgfgoview{arrows}{E4}.

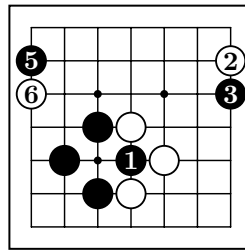
\begin{goban}[partial = A1:G7, name = arrows]
\stones[b]{C3, D3, E4, C5}
\stones[w]{D4, D5, E5, F3}
\arrow{C5}{C7}
\arrow[color = red!80!black, bend = 30, shorten = 4pt]{E4}{G5}
\end{goban}
```

The red arrow of the diagram below starts at the black stone ●.



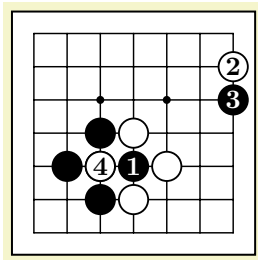
2.6 A ko: captures and notes

In the first tutorial, `captures` removed a captured stone from a game in SGF. To explain a ko, Kāru needs the same with moves written by hand: to put them in order and have PGFGO remove the captured stones.



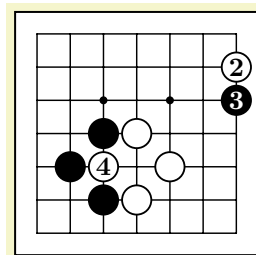
7 at 1

Black takes the ko at D3 (1), White threatens elsewhere (2), Black answers (3) and White takes back (4). With nothing else, the moves are drawn on top of one another and no stone leaves the board.



```
\begin{goban}[7]
\stones[b]{B3, C2, C4}
\stones[w]{C3, D2, D4, E3}
\moves{D3, G6, G5, C3}
\end{goban}
```

With the *key* `captures` (p. 92), the `goban` keeps count of the liberties and removes each group left without them. The white stone at C3 disappears with move 1, and black 1 with move 4. `\pgfgoprisoners` says how many stones each player captured.

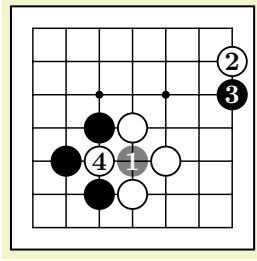


Black captured 1, White 1.

```
\begin{goban}[7, captures]
\stones[b]{B3, C2, C4}
\stones[w]{C3, D2, D4, E3}
\moves{D3, G6, G5, C3}
\end{goban}
```

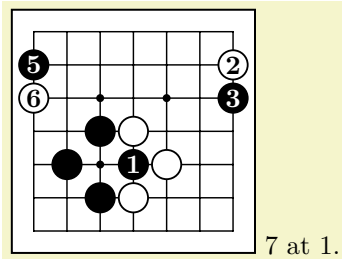
Black captured `\pgfgoprisoners{b}`,
White `\pgfgoprisoners{w}`.

If the captured stones must still be seen, `captures = fade` leaves them as prisoners.



```
\begin{goban}[7, captures = fade]
\stones[b]{B3, C2, C4}
\stones[w]{C3, D2, D4, E3}
\moves{D3, G6, G5, C3}
\end{goban}
```

The ko goes on: Black threatens (5), White answers (6) and Black takes back at D3 (7), where 1 had already been played. In books, that move is not drawn: “7 at 1” is written below the diagram. That is what the *key notes* does (p. 94), and `\pgfgomovenotes` writes the notes.



```
\begin{goban}[7, captures, notes]
\stones[b]{B3, C2, C4}
\stones[w]{C3, D2, D4, E3}
\moves{D3, G6, G5, C3, A6, A5, D3} % And so on until someone gets tired.
\end{goban}

\pgfgomovenotes.
```

2.7 A whole game

The article also comments on a long game: the first one between Lee Sedol and AlphaGo, from 2016, saved in `lee.sgf`, which Kāru calls `lee`. As in the first tutorial (p. 5), its 186 moves need not be written: PGFGO reads them from the file (p. 107).

First comes the information of the game, which `\sgfinfo` takes from the root of the file.

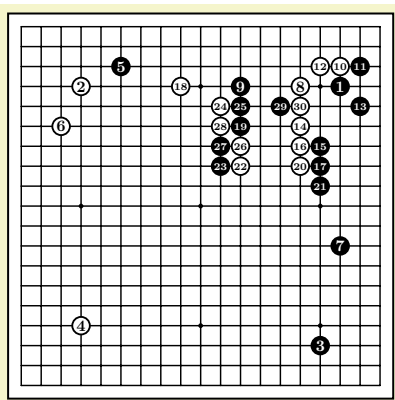
```
\pgfgodefinessgf{lee}{lee.sgf}

\sgfinfo{lee}{PB} (Black) against \sgfinfo{lee}{PW} (White),
\sgfinfo{lee}{EV}, \sgfinfo{lee}{RO}, \sgfinfo{lee}{DT}.
Result: \sgfinfo{lee}{RE}.
```

Lee Sedol (Black) against AlphaGo (White), Google DeepMind Challenge Match, Game 1, 2016-03-09. Result: W+Resign.

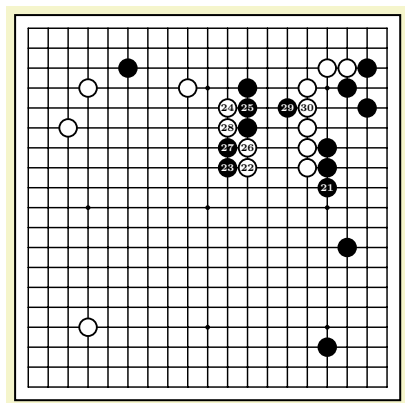
`\sgfinfo` writes the result as it is in the file; later, `\sgfdiagrams` writes it in words.

The game is 19×19 : `sgf size` gives the board the size that the file says. With `to`, Kāru shows it up to move 30, and with `scale`, makes the board smaller.



```
\begin{goban}[sgf size = lee, captures, scale = 0.6]
\sgf[to = 30]{lee}
\end{goban}
```

To comment on a stretch, Kāru numbers only the moves from 21 on with `from`: the previous ones are drawn without a number.



```
\begin{goban}[sgf size = lee, captures, scale = 0.6]
\sgf[from = 21, to = 30]{lee}
\end{goban}
```

Then Kāru wants the whole game, as in books: one diagram every 50 moves, each with its caption and its “N at M” notes, and the result at the end. That is what `\sgfdiagrams` does; it is used outside `goban`. With `layout`, the diagrams go in a grid of two columns, and with `fit` they are scaled to fill them (p. 131). Only the last two are shown here, with `from`.

```
\sgfdiagrams[step = 50, from = 101, layout = {columns = 2, fit}]{lee}
```

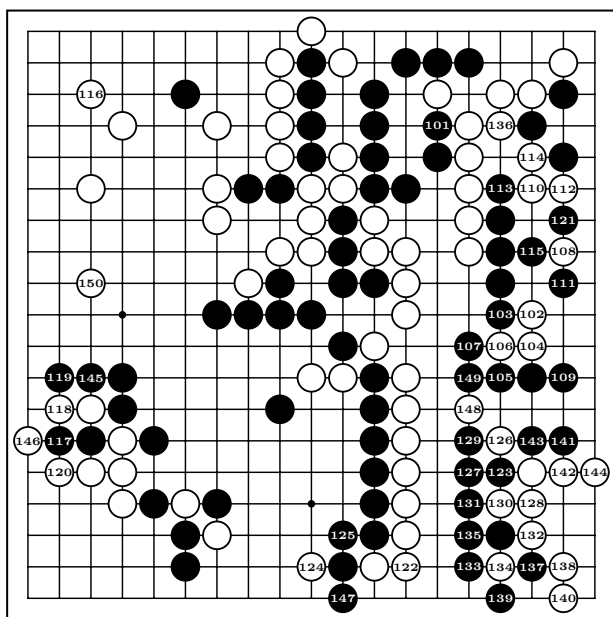


Diagram 3 (101–150)

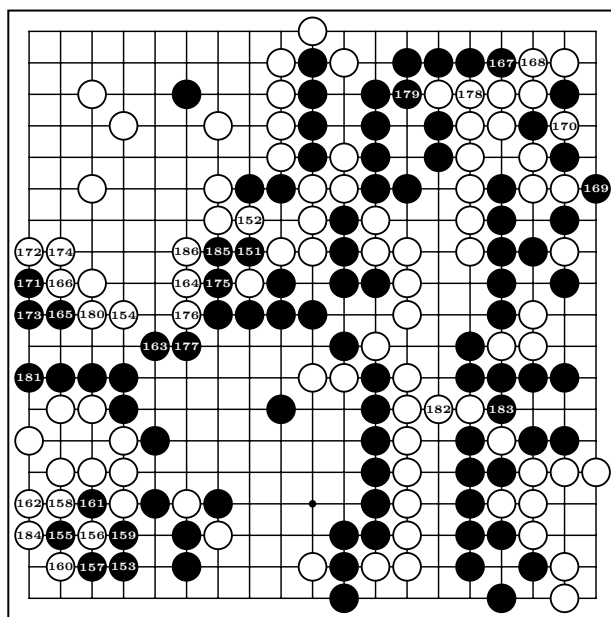


Diagram 4 (151–186)

Result: White wins by resignation

The caption and the result are in English because the document uses `babel` in English (p. 30). Some books restart the numbering in each diagram; with `restart`, each diagram, from the second one on, is numbered from 1.

```
\sgfdiagrams[step = 50, from = 51, to = 150, restart,
goban = {label = {at = {north, west}}}, layout = {columns = 2, fit}]{lee}
```

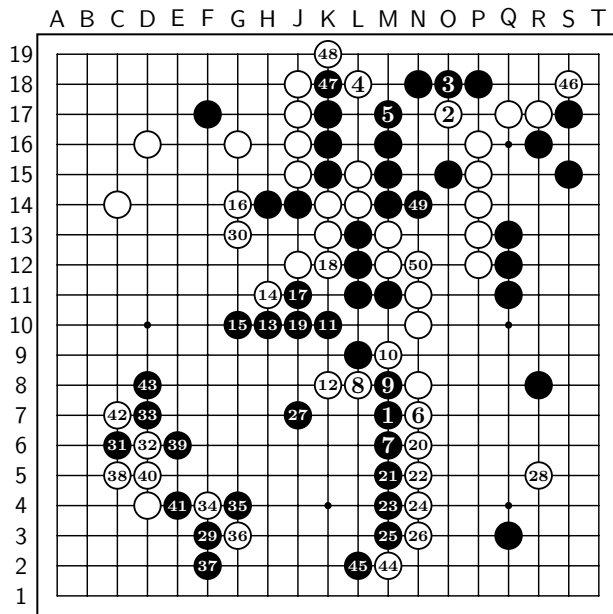


Diagram 2 (51-100)

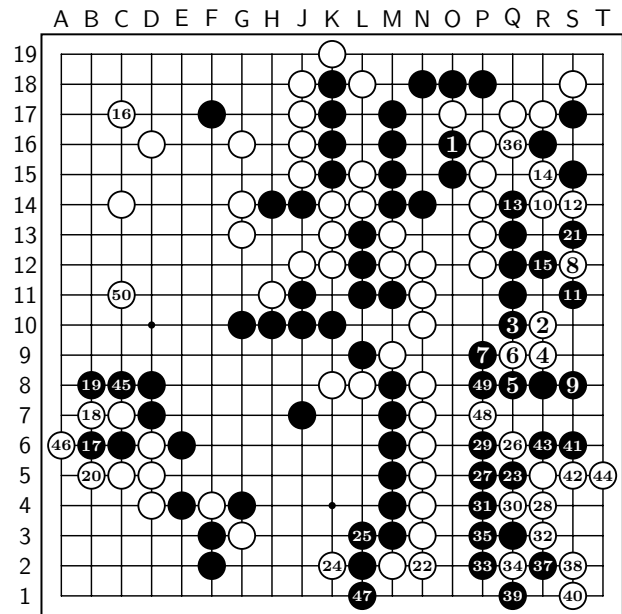
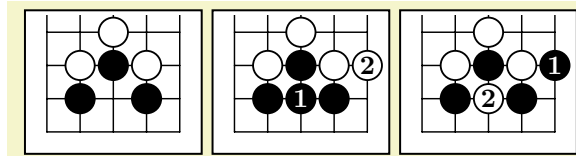


Diagram 3 (101-150)

2.8 Variations: remembering a board

In comments, Kāru often shows what would have happened with another move. Copying the position in each diagram is tedious and error-prone: if a stone changes, it has to be changed in all of them (I speak from experience). In the fight, Kāru gave a board a name with `name` to show it in the text. That board keeps everything it draws, and with `resume` another goban starts with that position (p. 101).



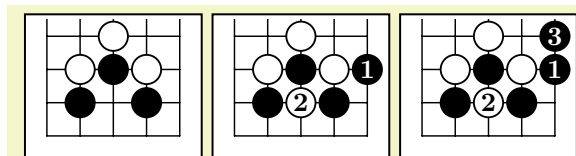
```
\pgfgobset{goban = {5, partial = A1:E4}}

\begin{goban}[name = base]
\stones[b]{B2, C3, D2}
\stones[w]{B3, C4, D3}
\end{goban}

\begin{goban}[resume = base]
\moves[b]{C2, E3}
\end{goban}

\begin{goban}[resume = base]
\moves[b]{E3, C2}
\end{goban}
```

Variations can have a name too, and be continued: with `name` and `resume` in the same `goban`, the new name keeps what it brought plus its own. By default, `resume` brings only the stones; `numbers` brings the numbers too.



```

\pgfgoset{goban = {5, partial = A1:E4}}

\begin{goban}[name = base]
  \stones[b]{B2, C3, D2}
  \stones[w]{B3, C4, D3}
\end{goban}

\begin{goban}[name = variation, resume = base]
  \moves[b]{E3, C2}
\end{goban}

\begin{goban}[resume = {variation, numbers}]
  \moves{E4}
\end{goban}

```

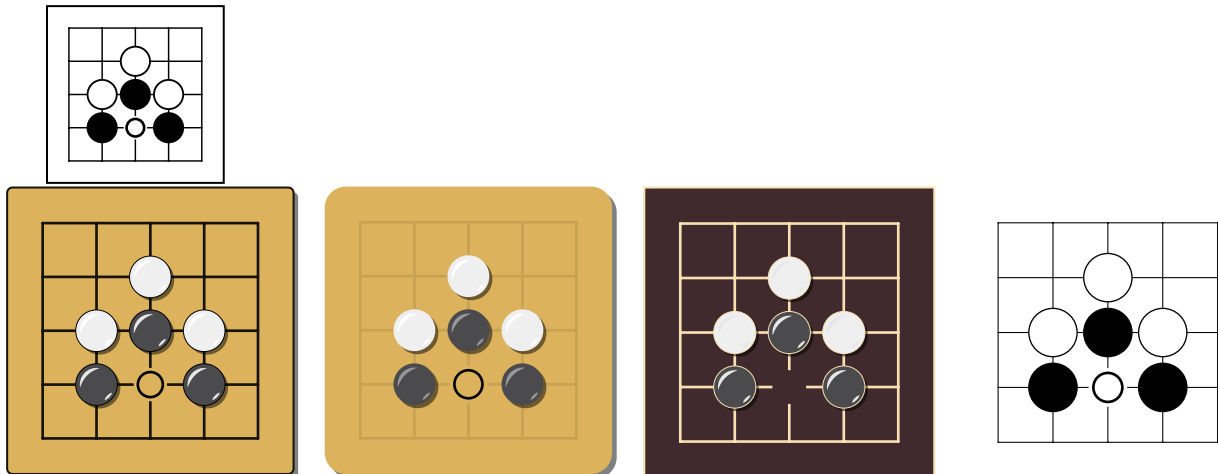
A named board also helps to compare styles. Before choosing the profile of the article, Kāru draws the same board with several profiles of the catalog (p. 138). To put them side by side, with the same width, they use the `pgfgolayout` environment (p. 131). With `marks`, `resume` brings the marks too.

```

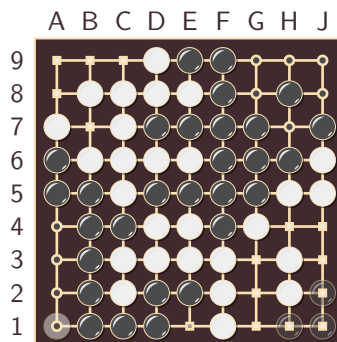
\begin{goban}[5, name = styles]
  \stones[b]{B2, C3, D2}
  \stones[w]{B3, C4, D3}
  \marks[circle]{C2}
\end{goban}

\begin{pgfgolayout}[columns = 4, fit]
  \begin{goban}[resume = {styles, marks}, profile = pgfgo] \end{goban}
  \begin{goban}[resume = {styles, marks}, profile = pgfgosmooth] \end{goban}
  \begin{goban}[resume = {styles, marks}, profile = wain] \end{goban}
  \begin{goban}[resume = {styles, marks}, profile = simple] \end{goban}
\end{pgfgolayout}

```

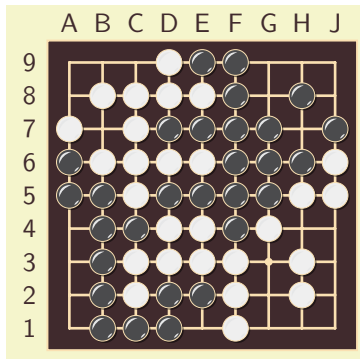


2.9 The end of the game: territory



To end the article, Kāru wants to show how a game is counted. They start from the final position. Since it has many stones and will be drawn several times, they keep it with `name`, like the variations of the previous

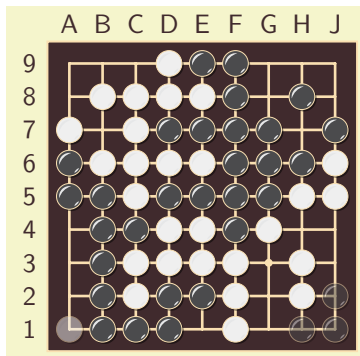
tutorial, and the following boards start with **resume**, without repeating the coordinates.



```
\pgfgoset{profile = wain}

\begin{goban}[label = {at = {above, left}}, name = final]
\stones[b] {A5,A6,B1,B2,B3,B4,B5,C1,C4,D1,D2,D5,D7,E2,E5,
E7,E9,F4,F5,F6,F7,F8,F9,G5,G6,G7,H6,H8,J7}
\stones[w] {A7,B6,B8,C2,C3,C5,C6,C7,C8,D3,D4,D6,D8,D9,E3,
E4,E6,E8,F1,F2,F3,G4,H2,H3,H5,J5,J6}
\end{goban}
```

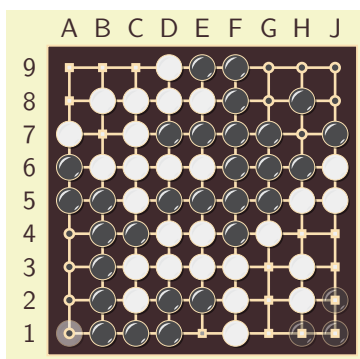
Next come the prisoners: the dead stones, which are removed when counting. They are drawn with `\prisoner` and `\prisoners`, or with the *key* prisoner of `\stone` and `\stones` (p. 62). Kāru keeps this board too, with another name: with **name** and **resume** in the same goban, as in the variations.



```
\pgfgoset{profile = wain}

\begin{goban}[label = {at = {above, left}},
resume = final, name = final2]
\stones[b, prisoner]{H1, J1, J2} % Same as \prisoners[b]{...}
\prisoner[w] {A1} % Same as \stone[w, prisoner]{...}
\end{goban}
```

Now, the territory. One way is to mark each point with **t-square** or **t-circle**. These marks always take as fill the fill color of the stone they belong to, and as border color and width those of the grid.



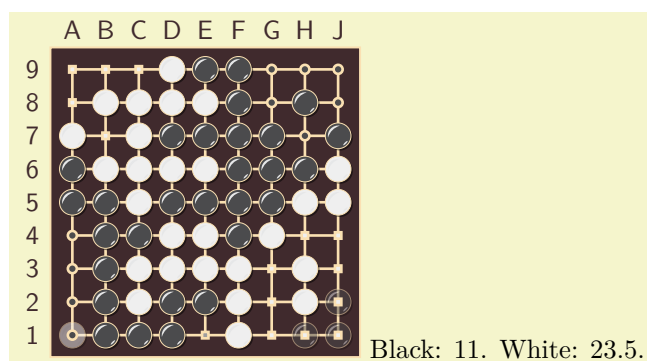
```

\pgfgoset{profile = wain}

\begin{goban}[label = {at = {above, left}}, resume = final2]
\marks[b, t-circle]{A1, A2, A3, A4, G8, G9, H9, J9, J8, H7}
\marks[w, t-square]{A9, B9, C9, A8, B7, H4, J4, J3, J2, J1,
H1, G1, G2, G3}
\marks[t-square, fill = {color = red}]{E1} % Not filled in red: it takes the color of neutral
\end{goban}

```

Another way is `\territory` (p. 95). With a polygon, it marks the points of that region that have no living stone; with `auto` and the Lua kernel, it finds the territory of each player by itself. `\pgfgoscore` writes the score.



```

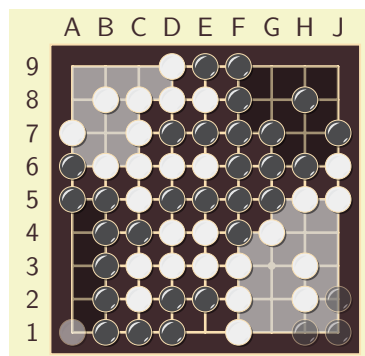
\pgfgoset{profile = wain}

\begin{goban}[label = {at = {above, left}}, resume = final2]
\territory[dame]{auto}
\end{goban}

Black: \pgfgoscore{black}. White: \pgfgoscore{komi = 6.5}{white}.

```

For the article, Kāru prefers the territory to look like a patch of color and not point by point. With `draw = region`, the territory of each color is a single polygon, whose sides run along the lines of the board.



```

\pgfgoset{profile = wain}

\begin{goban}[label = {at = {above, left}}, resume = final2]
\territory[draw = region]{auto}
\end{goban}

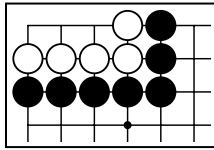
```

2.10 A problem book

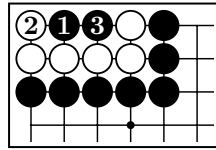
For their students, Kāru prepares a book of life-and-death problems. Problems are often saved in SGF too: the initial position, who plays and a tree with each variation of the solution, right or wrong, with its comments. `\sgfproblem` draws all of that: first the statement and then one diagram per variation. If the file does not say which part of the board to show, only the part the problem uses is shown, as with `partial = auto`.

```
\pgfdefinesgf{problem}{problem.sgf}

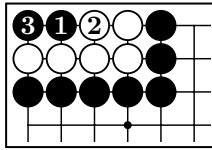
\sgfproblem[title = Problem 1]{problem}
```



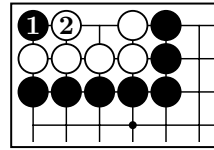
Problem 1: Black
to play
Black to kill.



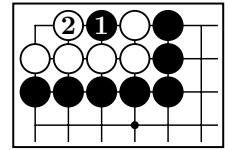
Variation 1
1: Tesuji. Correct:
the vital point.

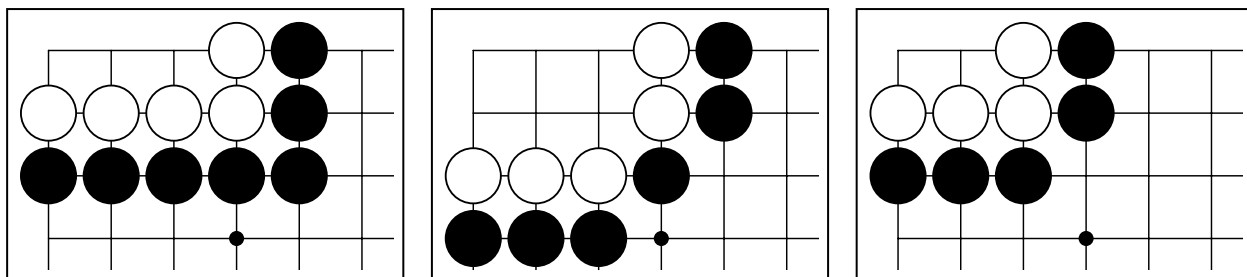


Variation 2
1: Tesuji. Correct:
the vital point.
3: White has only
one eye.



Variation 3
1: Bad move.
Wrong: White
lives.

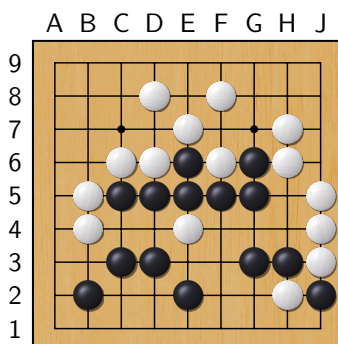




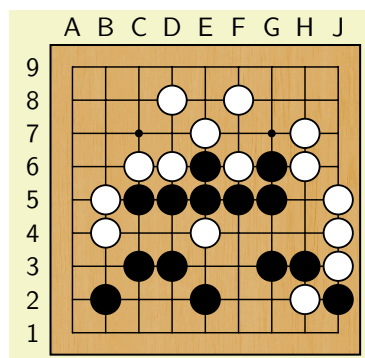
Problem 1: Black to play
Black to kill.

2.11 A wooden board: a profile with images

For the cover of the book, Kāru wants a board that looks like a real one, since the budget does not stretch to a kaya one: wood, stones with volume and white stones with stripes. The profiles of the catalog draw the stones as circles; this needs images (PNG, JPG or PDF). PGFGO comes with some, next to its .sty files, and Kāru starts with them: a photo of wood, `pgfgo-board.jpg`, a black slate stone, `pgfgo-shell-black.png`, and five white shell stones, `pgfgo-shell-white-1.png` to `pgfgo-shell-white-5.png`. They are used by name, from any document; images of your own are used the same way, as with `\includegraphics`.



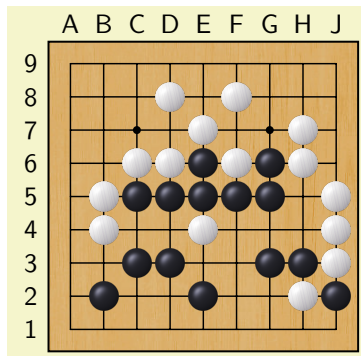
They start from the first board of the article. The wood goes in the background of the board, with `background/fill/image` (p. 40). By default, the image covers the whole board, cropped if needed; with `fit`, it can be stretched or repeated in tiles.



```
\pgfgoreset{goban = {background = {fill = {image = {file = pgfgo-
board.jpg}}}}}

\begin{goban}[label = {at = {north, west}}]
\stones[b]{B2,E2,J2,C3,D3,G3,H3,C5,D5,E5,F5,G5,E6,G6}
\stones[w]{D8,F8,E7,H7,C6,D6,F6,H6,B5,J5,B4,E4,J4,J3,H2}
\end{goban}
```

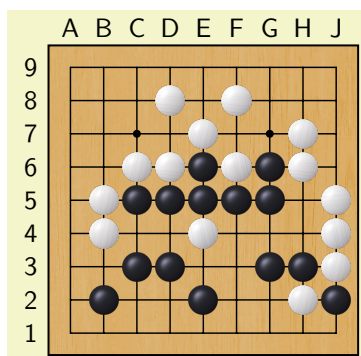
The stones of a player are drawn with an image with `stone/image` (p. 70), in the options of that player. The image takes the place of the circle: neither the fill nor the border is drawn.



```
\pgfset{
  goban = {background = {fill = {image = {file = pgfgo-board.jpg}}}},
  players = {
    black = {stone = {image = {file = pgfgo-shell-black.png}}},
    white = {stone = {image = {file = pgfgo-shell-white-1.png}}}}
}

\begin{goban}[label = {at = {north, west}}]
  \stones[b]{B2,E2,J2,C3,D3,G3,H3,C5,D5,E5,F5,G5,E6,G6}
  \stones[w]{D8,F8,E7,H7,C6,D6,F6,H6,B5,J5,B4,E4,J4,J3,H2}
\end{goban}
```

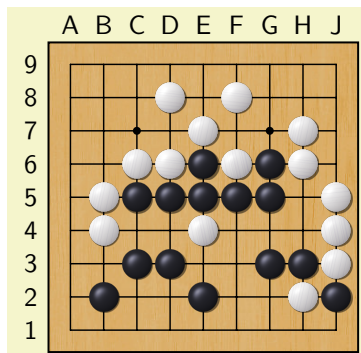
The black stones look good, but the white ones are all the same, and on a real board each shell stone has its own stripes. With several files in `file`, each stone uses one of them, chosen by its position: a stone on a point always shows the same image, in every diagram.



```
\pgfset{
  goban = {background = {fill = {image = {file = pgfgo-board.jpg}}}},
  players = {
    black = {stone = {image = {file = pgfgo-shell-black.png}}},
    white = {stone = {image = {file = {
      pgfgo-shell-white-1.png, pgfgo-shell-white-2.png, pgfgo-shell-white-
      3.png,
      pgfgo-shell-white-4.png, pgfgo-shell-white-5.png}}}}}}
}

\begin{goban}[label = {at = {north, west}}]
  \stones[b]{B2,E2,J2,C3,D3,G3,H3,C5,D5,E5,F5,G5,E6,G6}
  \stones[w]{D8,F8,E7,H7,C6,D6,F6,H6,B5,J5,B4,E4,J4,J3,H2}
\end{goban}
```

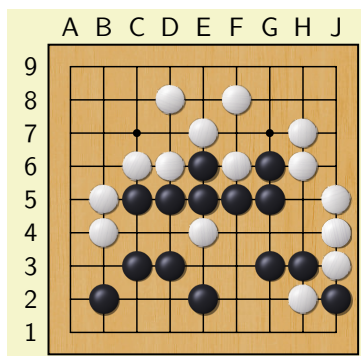
For the stones to look as if they rest on the wood, a shadow is missing. `shadow` draws it with the shape of the image (if the image has transparency, as these PNG do). Since it is the same for both players, it goes in `all`. Each file is kept only once in the PDF, however many stones use it.



```
\pgfset{
  goban = {background = {fill = {image = {file = pgfgo-board.jpg}}}},
  players = {
    all = {stone = {shadow = {true, offset = 0.75pt, angle = -45}}},
    black = {stone = {image = {file = pgfgo-shell-black.png}}},
    white = {stone = {image = {file = {
      pgfgo-shell-white-1.png, pgfgo-shell-white-2.png, pgfgo-shell-white-
      3.png,
      pgfgo-shell-white-4.png, pgfgo-shell-white-5.png}}}}}}
}

\begin{goban}[label = {at = {north, west}}]
  \stones[b]{B2,E2,J2,C3,D3,G3,H3,C5,D5,E5,F5,G5,E6,G6}
  \stones[w]{D8,F8,E7,H7,C6,D6,F6,H6,B5,J5,B4,E4,J4,J3,H2}
\end{goban}
```

With five images, a repeated one may still show. `rotate` turns each stone by a different angle, also chosen by its position. These stones have the light at the top left, so Kāru turns them only a little, between -15 and 15 degrees; the shadow turns with the image, but still falls to the same side.



```
\pgfset{
  goban = {background = {fill = {image = {file = pgfgo-board.jpg}}}},
  players = {
    all = {stone = {shadow = {true, offset = 0.75pt, angle = -45}}},
    black = {stone = {image = {file = pgfgo-shell-black.png}}},
    white = {stone = {image = {file = {
      pgfgo-shell-white-1.png, pgfgo-shell-white-2.png, pgfgo-shell-white-
      3.png,
      pgfgo-shell-white-4.png, pgfgo-shell-white-5.png}},
      rotate = -15:15}}}}
}

\begin{goban}[label = {at = {north, west}}]
  \stones[b]{B2,E2,J2,C3,D3,G3,H3,C5,D5,E5,F5,G5,E6,G6}
  \stones[w]{D8,F8,E7,H7,C6,D6,F6,H6,B5,J5,B4,E4,J4,J3,H2}
\end{goban}
```

Kāru will use this style in the whole book, and maybe in other documents. Instead of repeating the `\pgfset`, they save it as a profile with `\pgfdefineprofile` (p. 135): `goban` holds the options of the board; `all`, those of the three players, and `black` and `white`, those of each one. With `from = default`, each part starts with the profile `default`. Then the profile is used like any other, with the *key* profile.

```
\pgfdefineprofile[from = default]{wood}{
  goban = {background = {fill = {image = {file = pgfgo-board.jpg}}}},
  all = {stone = {shadow = {true, offset = 0.75pt, angle = -45}}},
  black = {stone = {image = {file = pgfgo-shell-black.png}}},
  white = {stone = {image = {file = {
    pgfgo-shell-white-1.png, pgfgo-shell-white-2.png, pgfgo-shell-white-3.png,
    pgfgo-shell-white-4.png, pgfgo-shell-white-5.png}}}}}}

\pgfset{profile = wood}
```

An almost identical profile, with a soft shadow and without the border of the board, comes with PGFGO: `shell` (p. 145); with `profile = shell` nothing has to be defined. There is also `yunzi`, with matte stones. Their definitions are in the catalog.

A profile holds for everything that PGFGO draws, also for the diagrams of an SGF. This is the game of Lee Sedol and AlphaGo on wood, with one diagram every 50 moves and the numbering restarted in each one.

```
\pgfset{profile = wood}

\sgfdiagrams[step = 50, restart, layout = {columns = 2, fit}]{lee}
```

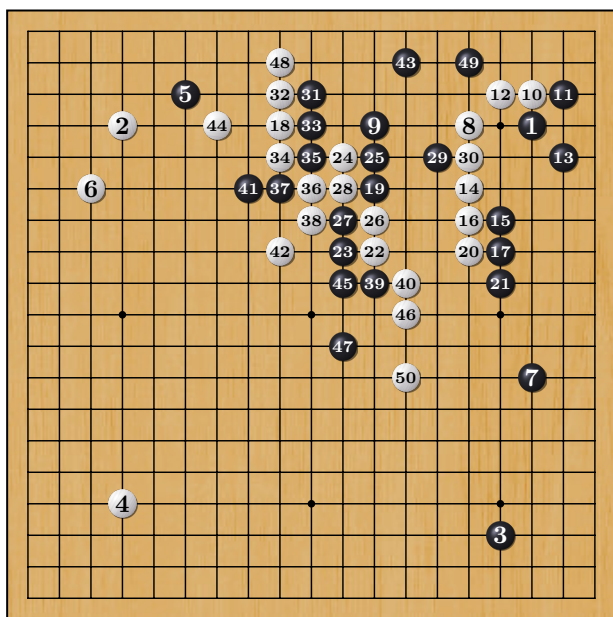


Diagram 1 (1–50)

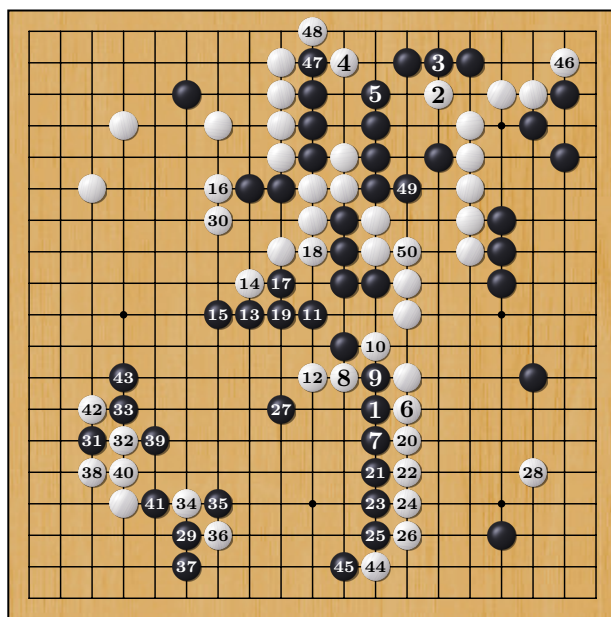


Diagram 2 (51–100)

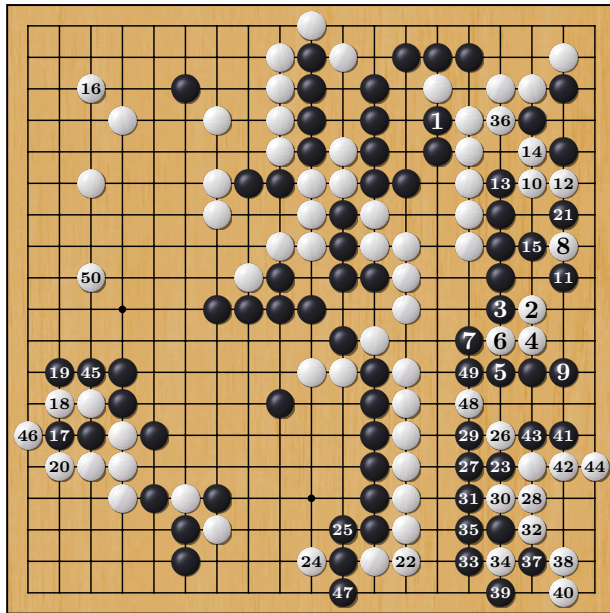


Diagram 3 (101-150)

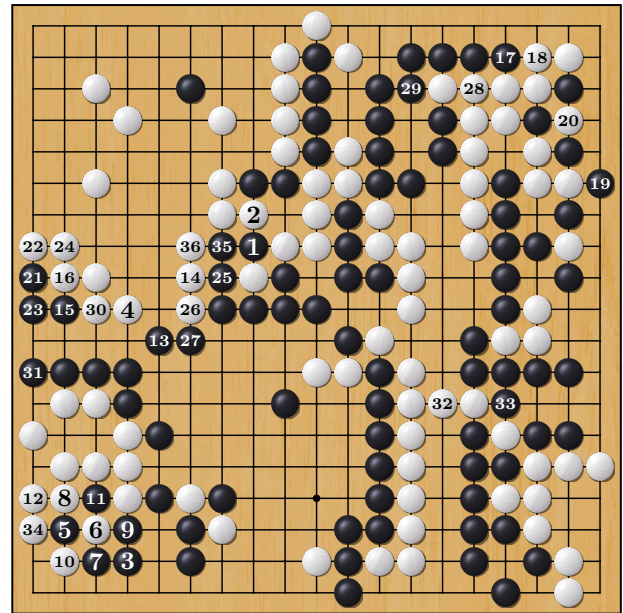


Diagram 4 (151-186)

Result: White wins by resignation

2.12 Slides and animations

Kāru also teaches with slides. A goban works inside a `beamer` frame as anywhere else, and `\sgf` with `to` shows the game move by move, one slide per move, with its comment beside it.

```
\documentclass{beamer}
\usepackage{pgf-go}

\begin{document}
\begin{frame}[t]{A short game}
\begin{columns}[T]
\column{0.5\textwidth}
\foreach \k in {1, ..., 10} {
\only<\k>{
\begin{goban}[sgf size = game, captures, scale = 1.5]
\sgf[to = \k]{game}
\end{goban}}
\column{0.45\textwidth}
\foreach \k in {1, ..., 10} {
\only<\k>{\small\sgfcomment{game}{\k}}}
\end{columns}
\end{frame}
\end{document}
```

To go over the whole game, on the other hand, the diagrams of Lee Sedol and AlphaGo from the article are enough (p. 14), one per slide. With `per frame`, `\sgfdiagrams` puts each diagram on its own slide of the frame, with its caption (p. 131). Outside `beamer`, `per page` does the same with pages.

```
\begin{frame}{Lee Sedol against AlphaGo}
\centering
\sgfdiagrams[step = 50, goban = {scale = 0.6}, layout = {per frame}]{lee}
\end{frame}
```

The slides are another document, but Kāru does not want to copy the boards of the article. In the article, they add `export` to the `goban` of the fight, which is written to the file `fight.pgfgo`. In the slides, they bring it in with `\pgfgoimport`, with its name and its style, and use it as if it had been drawn there: with `\pgfgoview` or with `resume` (p. 107).

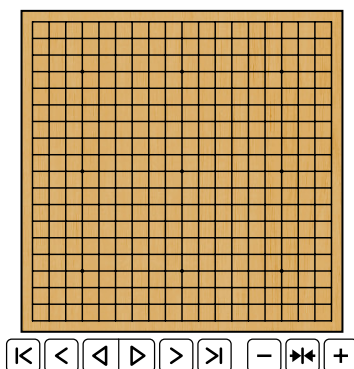
```
% In the article:
\begin{goban}[partial = A1:G7, name = fight, export = fight]
...
\end{goban}

% In the slides:
\pgfgoimport{fight}
...
\begin{frame}{If Black connects}
\begin{goban}[resume = {fight, marks}, partial = auto]
\moves{E3}
\end{goban}
\end{frame}
```

In a PDF to be read on screen, on the other hand, they prefer an animation: `\sgfanimation` draws one board per move with the `animate` package, and its buttons step forward, back or play the game (p. 107). The profile holds here too. The animation can be seen in Acrobat Reader and in some other viewers; in the rest, only the first frame.

```
\pgfgoreset{profile = wood}

\sgfanimation[to = 40, fps = 4, goban = {captures, scale = 0.5},
animate = {controls, loop}]{lee}
```

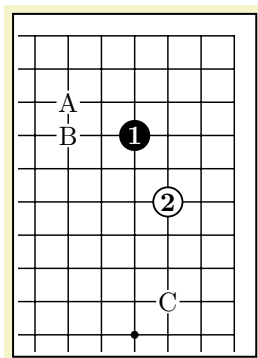


2.13 A joseki dictionary

For another lesson, Kāru wants to explain the answers to the low approach to the 4-4 point. The known sequences of the opening in a corner, the *josekis*⁵, are often saved in an SGF shaped like a dictionary: a tree with a branch for each possible move, comments on the ones that matter and letters on the moves that can follow. Kāru puts together a small one, `joseki.sgf`, and calls it `josekis`.

```
\pgfgodefinessgf{josekis}{joseki.sgf}
```

First, the position: the first two moves, with the letters of the answers. The board is 19×19 , but everything happens in a corner; with `partial = auto` (p. 9), the goban shows only the part used by what it draws, the letters too.



```
\begin{goban}[sgf size = josekis, partial = auto]
\sgf[to = 2]{josekis}
\end{goban}
```

⁵Learning them is easy; forgetting them, as the proverb asks, is harder.

For the answers, Kāru chooses the line with `variation`. In a dictionary there are so many branches that it is easier to give the moves of the line, from the first one, than to look for its number: `variation = {Q16, R14}` (p. 107). With `variations`, `\sgfdiagrams` also draws, each in its own diagram, the commented variations that leave that line, and with `from = 3`, only those that start at move 3 or later: the answers, and not the invasion at the 3-3 point, which is another move 2. With `partial = auto`, every diagram shows the same part of the board.

```
\sgfdiagrams[variation = {Q16, R14}, variations = {commented, from = 3},
partial = auto, comments, layout = {columns = 3, fit}]{josekis}
```

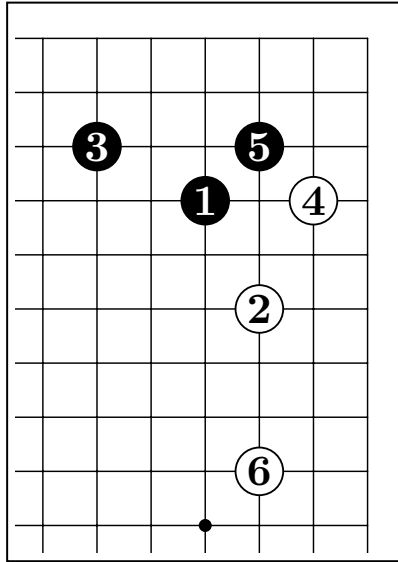


Diagram 1 (1–6)

2: The low approach.

3: A: the knight's move.

4: White slides into the corner.

6: And extends along the side.

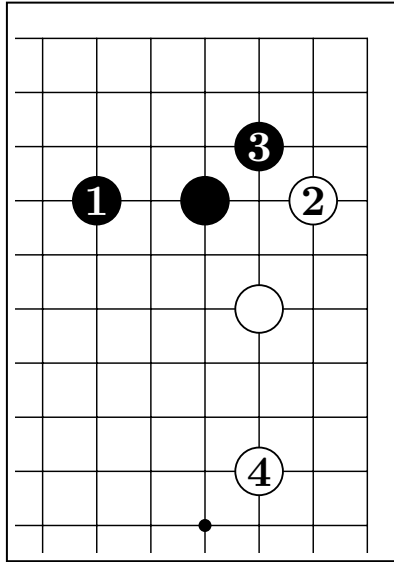


Diagram 1a (variation at 3)

1: B: the one-space jump.

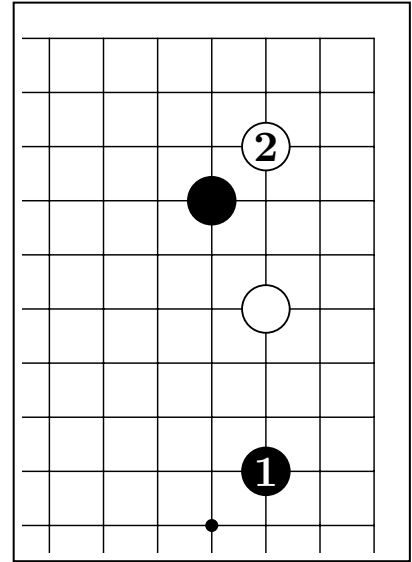


Diagram 1b (variation at 3)

1: C: the low pincer.

2: White invades at the 3-3 point.

The first diagram is the chosen line, to the end. The moves of each variation are numbered from 1, and the caption says at which move it leaves the line. In a large dictionary, each move comes with letters on the points where play can go on; with `marks = false`, the diagrams do not draw them.

2.14 Other languages

A colleague of Kāru wants to translate the book into French. The texts that PGFGO writes (the captions, the result, the “N at M” notes and the annotations of the comments) follow the language of `babel` or `polyglossia`, also when it changes inside the document: with `\selectlanguage{french}` they are translated by themselves. The language can also be chosen with the `key lang` (p. 30).

```
\pgfset{lang = fr}

\sgfdiagrams[step = 5, layout = {columns = 2}]{game}
\sgfproblem[solutions = false]{problem}
```

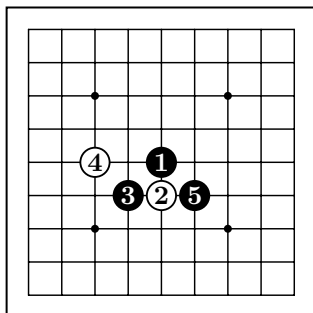


Diagramme 1 (1–5)

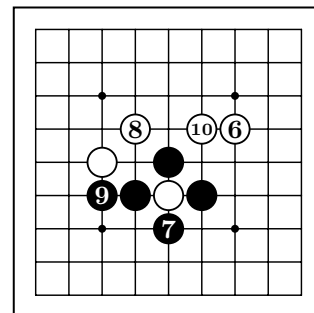
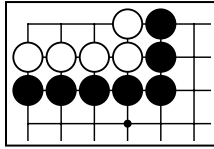


Diagramme 2 (6–10)



Noir joue

Black to kill.

The comment of the problem, “Black to kill.”, is not translated: it is written that way in the file. Each text of the package can also be changed with its own *key*, in any language: for example, the caption of the diagrams, with `title`.

```
\pgfgoset{sgf diagrams = {title = Figure}}
```

2.15 Writing less: the coordinate syntax

In all the tutorials, Kāru wrote the stones one by one. When there are many, PGFGO allows the lists of `\stones`, `\marks` and `\prisoners` to be shortened (p. 32). The most useful forms are the stretches of a row or of a column: `C:G5` is row 5, from C to G, and `J3:5` is column J, from 3 to 5. With them, the first board of the article becomes:

```
\pgfgoset{profile = pgfgo}

\begin{goban}[label = {at = {north, west}}]
\stones[b]{B2, E2, J2, C3, D3, G3, H3, C:G5, E6, G6}
\stones[w]{D8, F8, E7, H7, C6, D6, F6, H6, B5, B4, E4, H2, J3:5}
\end{goban}
```

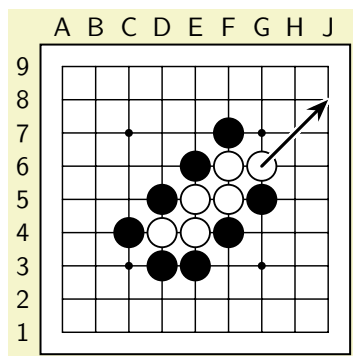
For larger groups, there are paths that turn at right angles. `A6 |- B5` goes from A6 down to row 5 and then right to B5: the stones A6, A5 and B5. `E9 -| F4` goes right first and then down. `D5 -- D7` is a straight line. Paths can be chained, and with them the final position of the territory tutorial (p. 17) fits in two lines:

```
\pgfgoset{profile = wain}

\begin{goban}[label = {at = {above, left}}]
\stones[b]{A6|-B5|-D1|-E2, C4, E9-|F4, D5-|G7--D7, H6, H8, J7}
\stones[w]{E8-|C5, B:E6, A7, B8, D9, J6|-H5, H2:3, F1|-C3, C2, D:E4, G4}
\prisoners[b]{H1-|J2}
\prisoner[w]{A1}
\territory[dame]{auto}
\end{goban}
```

Between two intersections that are not on the same row or the same column, `--` is a slanted line: `D5 -- F7` are D5, E6 and F7. Those stones touch only at the corners; for a connected chain, `-|-` fills each step, first

sideways and then up or down: D4 -|- F6 are D4, E4, E5, F5 and F6 (-|- starts with the column). With that, a ladder (*shicho*) is written with its ends:



```
\begin{goban}[label = {at = {north, west}}]
\stones[w]{D4 -|- F6 -- G6}
\stones[b]{C4, D3, D5 -- F7, E3 -- G5}
\arrow{G6}{J8} % White runs. Not for long.
\end{goban}
```

There is more: filled rectangles (A1:C3) or only their border (A1 r C3), whole columns and rows (D, 4), the whole board (*) and, after a ;, stones that are taken out of the list. Everything is in the reference, on p. 32.

Part III

Reference

3 Settings

`\pgfgoreset{...}` sets options for everything that follows: the look of the board, of the stones, of the moves, etc. It is equivalent to `\keys_set:nn{pgfgo}{...}`. It is also a door to direct formatting.⁶

`\pgfgoreset{<options>}`

`/pgfgo/goban = <goban options>` (no default)

Options of the board (p. 40): the size, the grid, the background, the coordinates, etc.

`/pgfgo/stone = <stone options>` (no default)

Options of the stones of the three players (p. 62), the same as `players = {all = {stone = {...}}}`.

`/pgfgo/move = <move options>` (no default)

Options of the moves, such as the format of their numbers. Except for `numbers` (p. 73), which holds for every move, it is the same as `players = {all = {move = {...}}}`.

`/pgfgo/mark = <mark options>` (no default)

Options of the marks: the default shape of `\marks` (`\pgfgoreset{mark = {triangle}}`) and the style of the marks of the three players (`line`, `fill`, `label`, `sequence` and `scale`): `mark = {scale = 2}` is the same as `players = {all = {mark = {scale = 2}}}`.

`/pgfgo/sgf animation = <sgf animation options>` (no default)

Options of every `\sgfanimation` of the document (p. 117).

`/pgfgo/sgf diagrams = <sgf diagrams options>` (no default)

Options of every `\sgfdiagrams` of the document (p. 120).

`/pgfgo/sgf problem = <sgf problem options>` (no default)

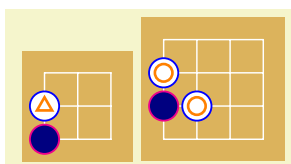
Options of every `\sgfproblem` of the document (p. 127).

`/pgfgo/sgf comment = <sgf comment options>` (no default)

Options of every `\sgfcomment` of the document (p. 115).

⁶Use with discretion... not like footnotes.

<code>/pgfgo/sgf info = <sgf info options></code>	(no default)
Options of every <code>\sgfinfo</code> of the document (p. 115).	
<code>/pgfgo/sgf annotations = <texts></code>	(no default)
The texts of the annotations of SGF files (p. 115).	
<code>/pgfgo/arrow = <arrow options></code>	(no default)
Options of every arrow of the document (p. 88).	
<code>/pgfgo/line = <line options></code>	(no default)
Options of every line of the document (p. 88).	
<code>/pgfgo/territory = <territory options></code>	(no default)
Options of every <code>\territory</code> of the document, such as <code>draw = fill</code> (p. 95).	
<code>/pgfgo/score = <score options></code>	(no default)
Options of every <code>\pgfgoscore</code> of the document, such as <code>komi = 6.5</code> (p. 95).	
<code>/pgfgo/players = <players options></code>	(no default)
Options of each player: black, white and neutral (p. 134).	



```

\pgfgoreset{
  goban = {
    background = {
      fill = {color = pgfgo},
      line = false},
    grid = {line = {color = white}},
    players = {
      black = {
        stone = {
          line = {color = magenta},
          fill = {color = blue!50!black}},
        white = {
          stone = {line = {color = blue}},
          mark = {line = {color = orange}}
        }
      }
    }
  }

\begin{goban}[3]
  \stone[black] {A1}
  \stone[white, mark = {triangle}] {A2}
\end{goban}

\begin{goban}[4]
  \stone[black] {A2}
  \stones[white, mark = {circle}] {B2,A3}
\end{goban}

```

The `goban` environments that follow inherit these options, and each one can change them. The options of a `goban` accept, besides those of the board, those of `\pgfgoreset`, which hold only for that board: `\begin{goban}[9, stone = {radius = 4pt}, mark = {line = {color = red}}]`.

Many *keys* turn something on or off: the grid, the background, the coordinates, the border and the fill of the stones, their shadows and their shines, etc. They all follow the same rule:

- the *key* alone, or with `true`, turns it on: `shine, shine = true;`
- with `false` it turns it off: `shine = false;`
- giving options does not turn it on: `shine = {angle = 45}` changes the angle, but the shine stays as it was. To turn it on and change it at once, `shine = {true, angle = 45}`.

The exceptions are the *keys* whose value is the content itself, such as `partial` (the part of the board) and `sequence` (where the numbering starts).

3.1 The compute kernel

PGFGO separates the computations from the drawing. The computations (reading the coordinates, knowing which part of the board is visible, numbering the moves and keeping the turn, remembering the boards, etc.) are done by a *kernel*. The drawing, the *keys* and the profiles are always the same. There are two kernels:

- **lua**: written in Lua. It needs Lua $\text{\LaTeX}$, and it is the default with Lua $\text{\LaTeX}$.
- **expl3**: written in L3. It works with any engine, and it is the default with the others.

Both produce exactly the same diagrams⁷. The Lua kernel also knows three things the other does not: captures (p. 92), finding the territory with `\territory{auto}` (p. 95) and reading games in SGF (p. 107). This manual is compiled with the Lua kernel.

`/pgfgo/package/kernel` = $\langle expl3 \text{ or } lua \rangle$ (no default, initially lua with Lua $\text{\LaTeX}$, expl3 with the other engines)

Package option that chooses the kernel. It is only needed to use **expl3** with Lua $\text{\LaTeX}$. It can only be given when PGFGO is loaded; it cannot be changed with `\pgfgoset`. If `kernel = lua` is asked for with pdf $\text{\LaTeX}$, the package warns and uses **expl3**. With **expl3**, `captures`, `\territory{auto}` and `\sgf` give an error.

```
\usepackage[kernel = expl3]{pgf-go}
```

3.2 Languages

Some texts are written by the package itself: the captions of `\sgfdiagrams` and `\sgfproblem`, the result of the game, the annotations of the comments (`/pgfgo/sgf annotations`) the word of the “N at M” notes and the “and” between several views. They are translated into English (**en**), Spanish (**es**), French (**fr**), German (**de**), Portuguese (**pt**), Italian (**it**), Esperanto (**eo**), Japanese (**ja**), Chinese (**zh**) and Korean (**ko**). Japanese, Chinese and Korean need a font with those characters.

By default, the texts are in the language set by babel or polyglossia, and they follow it if it changes inside the document (`\selectlanguage`, `otherlanguage`). Without babel or polyglossia, or in a language without a translation, they are in English. Each text can be changed with its *key* (for example, `/pgfgo/sgf diagrams/title`); what is set that way holds in any language.

`/pgfgo/lang` = $\langle language \rangle$ (no default, initially **auto**)

Language of the texts: **auto** (that of babel or polyglossia), the code of a language (**fr**) or the name babel or polyglossia give it (**french**). If the language is not known, the package warns and uses English.

```
\pgfgoset{lang = fr}
```

`/pgfgo/package/lang` = $\langle language \rangle$ (no default, initially **auto**)

The same, as a package option: `\usepackage[lang = fr]{pgf-go}`.

`\pgfgotext{\text}`

Writes one of the texts in the current language. The texts, with their value in English and in Spanish, are:

⁷Or so the tests say.

tesuji	Tesuji	Tesuji
very good tesuji	Very good tesuji	Muy buen tesuji
bad move	Bad move	Mala jugada
very bad move	Very bad move	Muy mala jugada
doubtful	Doubtful move	Jugada dudosa
interesting	Interesting move	Jugada interesante
good for black	Good for Black	Bueno para Negro
very good for black	Very good for Black	Muy bueno para Negro
good for white	Good for White	Bueno para Blanco
very good for white	Very good for White	Muy bueno para Blanco
even	Even position	Posición pareja
unclear	Unclear position	Posición incierta
diagram	Diagram	Diagrama
variation	Variation	Variante
black to play	Black to play	Juega Negro
white to play	White to play	Juega Blanco
at	at	en
and	and	y
result	Result	Resultado
black	Black	Negro
white	White	Blanco
win	<i>winner</i> wins	Gana <i>winner</i>
win by points	<i>winner</i> wins by <i>points</i> points	Gana <i>winner</i> por <i>points</i> puntos
win by resignation	<i>winner</i> wins by resignation	Gana <i>winner</i> por abandono
win on time	<i>winner</i> wins on time	Gana <i>winner</i> por tiempo
win by forfeit	<i>winner</i> wins by forfeit	Gana <i>winner</i> por incomparecencia
draw	Draw (jigo)	Empate (jigo)
void	No result	Sin resultado
unknown result	Unknown result	Resultado desconocido

Black to play. Tesuji. Noir joue.

```
\pgfgotext{black to play}. \pgfgotext{tesuji}.

\pgfgoset{lang = fr}
\pgfgotext{black to play}.
```

\pgfgodefinelanguage{*language*}{*texts*}

Adds a language, or changes texts of one that already exists. The missing texts are taken from English. In the texts of the result, `\pgfgowinner` is the name of the winner and `\pgfgopoints` the difference (p. 120).

```
\pgfgodefinelanguage{es}{black = Negro, white = Blanco,
win by points = Gana \pgfgowinner{} por \pgfgopoints{} puntos}
```

\pgfgolanguagealias{*name*}{*language*}

Another name for a language, for example the one babel or polyglossia give it, if PGFGO does not know it.

```
\pgfgolanguagealias{castellano}{es}
```

3.3 One board per page

With the option `pages`, the document comes out with one **goban** per page, each one the size of its board, and nothing else: the text, the captions and the notes do not come out. Neither do the views in the text (`\pgfgoview`) and the animations. So the boards of a document can be used in another program (a word processor, an ebook, a web page) without copying their code.

At the end, the package also writes `<document>-pages.lua`, which turns each page into a file in the folder `<document>-pages`. The file is named like the goban (name; only letters, digits, ., _ and -), or `diagram-` and the number of the page if it has no name. It is run with `texlua`:

```
texlua book-pages.lua           % SVG, with pdftocairo or dvisugm
texlua book-pages.lua png 600   % PNG at 600 dpi (300 if not given), with pdftocairo or pdftoppm
texlua book-pages.lua pdf       % one PDF per board, with pdftocairo or pdfseparate
```

`/pgfgo/package/pages = <boolean>` (default true, initially false)

Package option: one goban per page. It needs Lua_{La}T_EX and `kernel = lua`; otherwise, the package warns and the document comes out as always.

```
\usepackage[pages]{pgf-go}
```

4 Coordinate syntax

Something that sets PGFGO apart from other packages is the possibility of entering coordinates with several syntaxes:

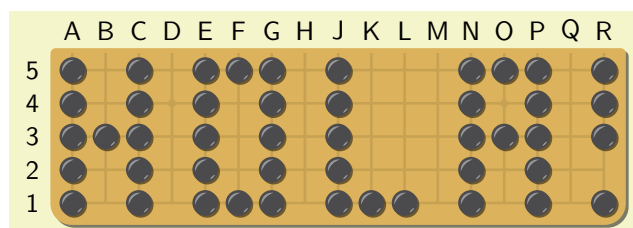
- Multiple entries, separated by commas.
- Removing stones by adding the subtractive group after a «;».

These forms only work in the plural commands (`\stones`, `\marks`, `\prisoners`); `\stone`, `\mark` and `\prisoner` take a single intersection. `\moves` accepts all of them except the subtractive groups, and warns if they have no fixed order (p. 38). The writing has some restrictions. For example, to refer to an intersection, the alphabetic coordinate must be written in capitals and without spaces. So the intersection B5 must be entered as B5, and not as B 5, B 5 or b5, since spaces between letter and number are not valid in this context. On boards of more than 25 columns, the columns after Z have two letters: AA, AB, ... (AA5, AA:AC 3, `partial = AZ5:BG8`). The restrictions will be explained in each case.

When PGFGO has to read the argument of the coordinates, it may find something like this

```
\stones{<item>, <item>, ...; <item>, <item>, ...; ...}
```

The first thing it will try to do is split the argument at the ; and build two groups: the additive group and the subtractive group. The additive group adds stones with our syntax. The subtractive group removes them. These groups alternate automatically with each ;. This is seen in more detail in the corresponding subsection. If no ; is given, PGFGO will add all the stones to the additive group. Several stones or several groups can be added, separated by ,. Altogether, this provides an interface to create complex groups with few *tokens*;



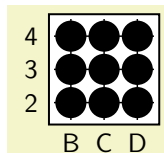
```
\pgfgoset{profile = pgfgosmooth}

\begin{goban}[17, partial = A1:R5, label = {at = {north, west}}]
  \stones[b]{*;B,D,H,M,R2,Q,F2:4,K2rL5,0[1:2,4];B3}
\end{goban}
```

4.1 Groups of stones

All the stones of the board (*Why not?*)

Adds to the corresponding group every **visible** intersection of the board. That is, on a partial board, only those of the visible region are added. This is worth making clear in case it is used together with **resume**. Its syntax is *****.

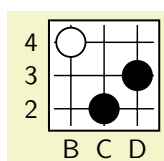


```
\pgfset{goban = {5, partial = B2:D4}}

\begin{goban}[label = {at = {south, west}}]
  \stones[b]{*}
\end{goban}
```

Intersection

Intersections are given by the letter followed by the number, without spaces between the alphabetic and the numeric component: LN



```
\pgfset{goban = {5, partial = B2:D4}}

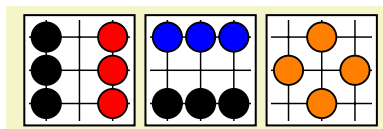
\begin{goban}[label = {at = {south, west}}]
  \stones[b]{C2, D3}
  \stone[w]{B4}
  \stone[w]{B 3} % Not recognized because it has a space.
\end{goban}
```

In the example, B 3 is not recognized because of the space, while B4 is drawn correctly as a white stone.

Vertical or horizontal lines

Lines are groups of vertical or horizontal stones implicitly bounded by the size of the board: a whole column or row.

- For a column (a vertical line), the syntax is L: D is column D.
- For a row (a horizontal line), the syntax is N: 4 is row 4.



```
\pgfset{goban = {5, partial = B2:D4}}

\begin{goban}
  \stones[b]{B2,B3,B4}
  \stones[fill = {color = red}]{D}
\end{goban}

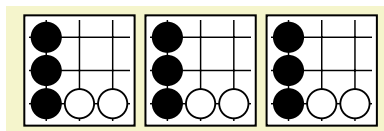
\begin{goban}
  \stones[b]{B2,C2,D2}
  \stones[fill = {color = blue}]{4}
\end{goban}

\begin{goban}
  %Add the lines C and 3. Then remove C3.
  \stones[fill = {color = orange}]{C, 3; C3} %CAREFUL: C,3 != C3.
\end{goban}
```

Segments

Segments are groups of vertical or horizontal stones explicitly bounded. Since they go in one direction (only vertical or only horizontal), one of their coordinates stays constant and need not be repeated; only the range of the other is given.

- For horizontal segments, the syntax is L:LN or L:L N.
- For vertical segments, the syntax is LN:N or L N:N.



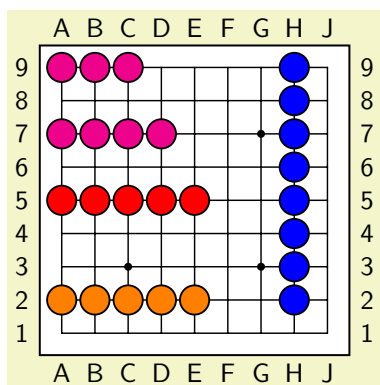
```
\pgfset{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{B2,B3,B4}
\stones[w]{C2,D2}
\end{goban}

\begin{goban}
\stones[b]{B2:4}
\stones[w]{C:D2}
\end{goban}

\begin{goban}
% Spaces work between different elements.
\stones[b]{B 2:4}
\stones[w]{C:D 2}
\end{goban}
```

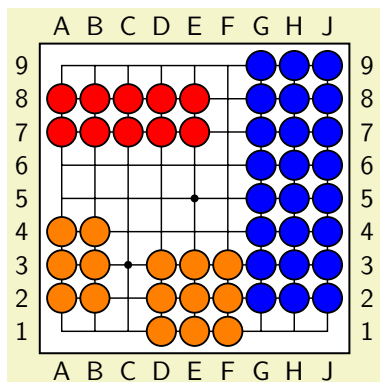
Here, B2:4 stands for a vertical segment from B2 to B4, while C:D2 is a horizontal segment from C2 to D2. Segments go from the first end to the second: H9:2 goes down from H9 to H2 and E:A2 goes from E2 to A2. They can also be chained with the operators we will see later.



```
\begin{goban}[label = {at = all}]
\stones[fill = {color = red}]{A:E5}
\stones[fill = {color = blue}]{H9:2}
\stones[fill = {color = orange}]{E:A2}
\stones[fill = {color = magenta}]{A:D7, A:C9} % Combined.
\end{goban}
```

Rectangle: an artificial consequence

The segment syntax creates ranges between coordinates to insert stones. I thought it useful to extend it to two ranges, as a first approach to rectangles (although it is not my favorite method, here it is). In this case, the syntax is L:L N:N or L:LN:N.



```
\begin{goban}[label = {at = all}]
% The space is not required.
\stones[fill = {color = red}]{A:E 7:8}
\stones[fill = {color = blue}]{I:G 9:2}
\stones[fill = {color = orange}]{A:B 2:4, F:D 1:3}
\end{goban}
```

For example, A:E 7:8 creates a rectangle that covers the columns from A to E in rows 7 and 8, while I:G 9:2 goes down from I9 to G2. It works, but the operators we will see later are clearer.

Cartesian products

In this context, the Cartesian product is a more powerful extension of the segments, but at the same time a generalization of all the notations presented so far. The idea behind it is the following: suppose we want to enter a group of stones on column A: A1:3, A5, A7:8. So far there was no simpler way to write this. But they all repeat A at the beginning: what if, instead of writing it three times, we write it only once? Something like

A[1:3, 5, 7:8]

In this case, the interpreter distributes the A on the left to each of the elements on the right.

The elements of the list on the right need not be made of numbers. Something like A:D 5:7, A1:2, A3, A:G9 can also be rewritten as

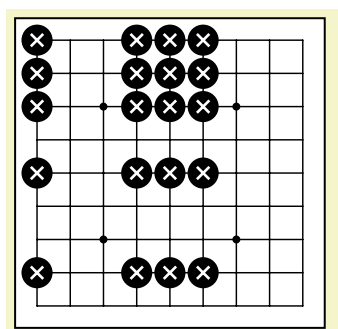
A[:D 5:7, 1:2, 3, :G9]

Of course, nobody abstracts that much when entering coordinates, so this is a mere whim. The only restriction on the content in brackets is that brackets cannot be nested.

Just as there is a syntax for letters on the left, there is one for numbers on the right, and it works the same way. The expression A12, C12, E:G12 can be written as [A,C,E:G] 12.

There is a third operator which, yes, is more of a whim than anything else: the unrestricted Cartesian product. I will not dwell on it, because it requires a higher degree of abstraction and really has no human use. This operator is a combination of the previous two: the expression [A,D:F] [2,5,7:9] is read as

A2, A5, A7:9, D:F2, D:F5, D:F 7:9.



```
\begin{goban}
\stones[b]{[A, D:F] [2, 5, 7:9]}
\marks[b]{A2, A5, A7:9, D:F2, D:F5, D:F 7:9} %<- what the interpreter reads.
\end{goban}
```

The Cartesian product syntax can then be used in three different ways:

- L[itm1, itm2,...].
- [itm1, itm2,...]N.
- [itm1,itm2,...][itm3,itm4,...].

where brackets cannot be nested.

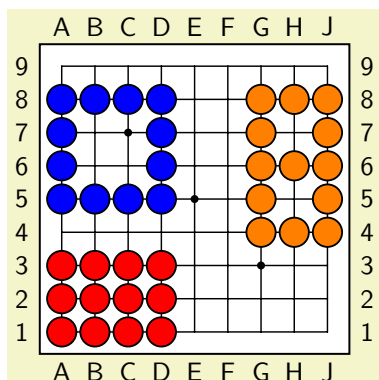
This can also be combined with some of the operators we will see next, to create nodes in the paths.

So far we have seen forms that cannot be combined with others in one element and that, therefore, **must** be separated by commas. Now we will see operators that join two intersections with a path and that can be chained.

Horizontal-vertical and vertical-horizontal

Given two intersections not aligned vertically or horizontally, these commands generate the intersections that connect the first with the second along a “horizontal-vertical” or “vertical-horizontal” path. This may remind you of the -| operator of TikZ, from which I borrowed the idea. The syntax is LN -| LN (for the first case) or LN |- LN (for the second), with or without spaces. These commands, together with the following ones, can be chained to make stairs or paths of the Manhattan metric.

The previous syntax for rectangles, L:L N:N, is not very intuitive and rather a legacy of the bounded segments. It is more direct to give two diametrically opposite vertices: LN:LN fills the rectangle (it is the same writing as that of `partial`) and LN r LN draws only its outline. Although the spaces around `r` are not required, I recommend them for readability.

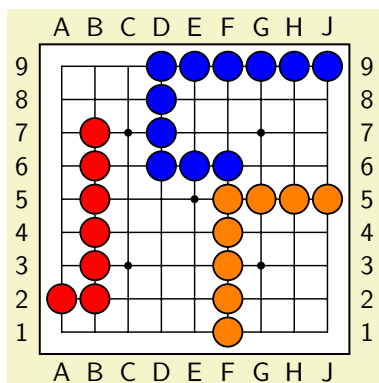


```
\begin{goban}[label = {at = all}]
\stones[fill = {color = red}]{A1:D3}
\stones[fill = {color = blue}]{A5 r D8}
\stones[fill = {color = orange}]{I8 r G4, H6} % It's alive!
\end{goban}
```

For example, A1:D3 fills a rectangle from A1 to D3, while A5 r D8 draws only the border of a larger rectangle. A rectangle of a single column or a single row is a segment: A5:A9 is A5:9.

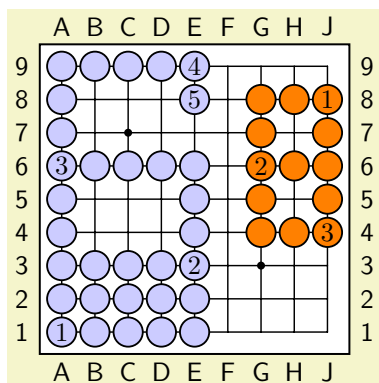
Chaining

The operators (`-|`, `|-`, `--`, `-|-`, `|-|` and `r`) can be chained: A1 -| C3 -- E3 is A1 -| C3 followed by C3 -- E3, and C3 appears only once. Between two operators there can be not only an intersection: also a segment (B5:7, A:B3) or a rectangle (A1:C3). Each operator goes from the last point of what comes before to the first of what comes after. So A2 -| B5:7 is A2 -| B5 followed by B5:7, and B7 -| A:B3 is B7 -| A3 -- B3.



```
\begin{goban}[label = {at = all}]
\stones[fill = {color = red}]{A2 -| B5:7}
\stones[fill = {color = blue}]{I9 -| D:F6}
\stones[fill = {color = orange}]{F1:3 |- I5}
\end{goban}
```

Rectangles can also be chained, to create more complex patterns.

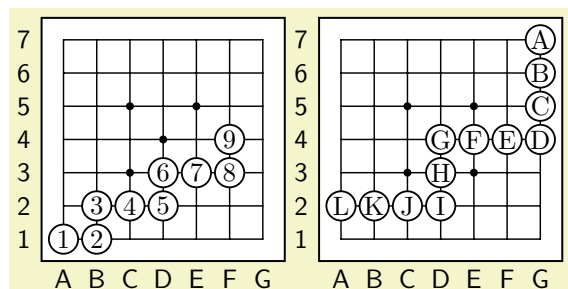


```
\begin{goban}[label = {at = all}]
\stones[fill = {color = orange}]{I8 r G6 r I4} % You guess.
\marks[sequence]{I8,G6,I4}
\stones[fill = {color = blue!20}]{A1:E3 r A6 |- E9 -- E8} % Wow!
\marks[sequence]{A1,E3,A6,E9,E8}
\end{goban}
```

In this case, I8 r G6 r I4 chains two outline rectangles, while A1:E3 r A6 |- E9 -- E8 combines a filled rectangle, an outline, a “vertical–horizontal” path and a line. The forms that go alone (`*`, `L`, `N`, `L:L`, `N:N` and `L:L N:N`) cannot be chained.

The order of the intersections

The intersections come out in the order in which they are written: segments go from the first end to the second (N10:7 goes down from N10 to N7), the operators follow their path and brackets are distributed in place (D4, B[3:1] is D4, B3:1). That is the order numbered by `\moves` and `sequence`.



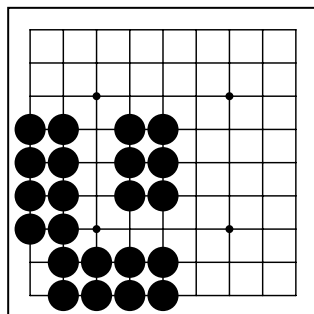
```
\begin{goban}[7, label = {at = {south, west}}]
\stones[w]{A1 -|- F4}
\marks[sequence]{A1 -|- F4}
\end{goban}

\begin{goban}[7, label = {at = {south, west}}]
\stones[w]{G7:4 -| D2 -- A2}
\marks[sequence = {format = Alph}]{G7:4 -| D2 -- A2}
\end{goban}
```

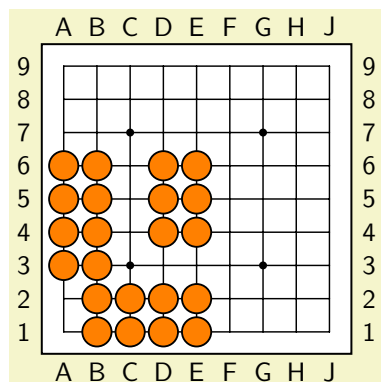
Some forms have no fixed path: rectangles (LN:LN, L:L N:N and LN r LN), whole columns and rows, and *. `\stones` and `\marks` accept them like the others, but `\marks[sequence]` and `\moves` warn (unordered-coordinate), because the order in which they go through them is not chosen by the writer.

4.2 Additive and subtractive groups

Suppose we want to draw the following diagram:

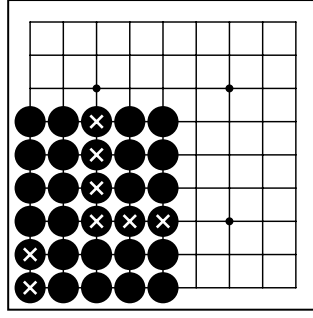


This can be done with the operators already seen, splitting the pattern into several parts:

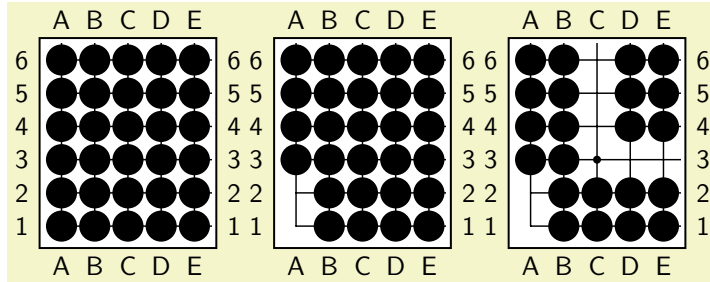


```
\begin{goban}[label = {at = all}]
\stones[fill = {color = orange}]{A6:B3, B2:E1, D4:E6}
\end{goban}
```

But we can also imagine that we start from a full rectangle from which the marked stones have been removed:



That is what additive and subtractive groups are about. An additive group adds intersections to the list, while a subtractive one removes them. By default, the commands always start by processing additive groups, which explains why they draw stones directly. In PGFGO, we can alternate between both kinds with ;.



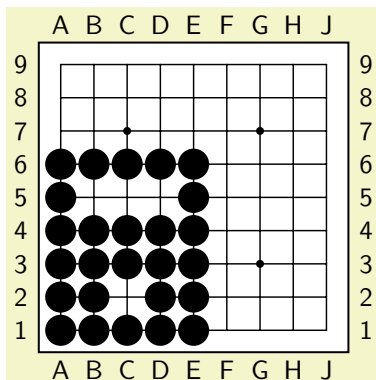
```
\begin{goban}[label = {at = all}, partial = A1:E6]
\stones[b]{A1:E6} % Additive group.
\end{goban}

\begin{goban}[label = {at = all}, partial = A1:E6]
\stones[b]{A1:E6; A 1:2} % Additive group; subtractive.
\end{goban}

\begin{goban}[label = {at = all}, partial = A1:E6]
\stones[b]{A1:E6; A 1:2, C6 |- E3} % Additive group; subtractive.
\end{goban}
```

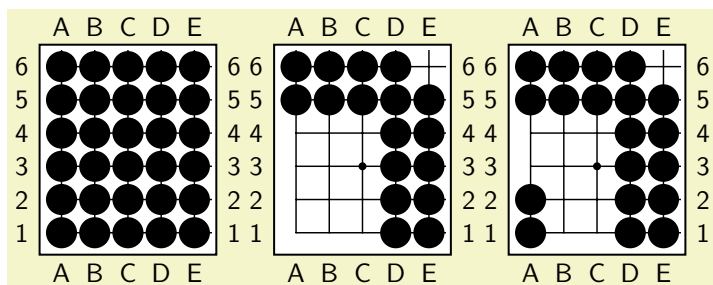
In this sequence, first a full rectangle is drawn (A1:E6), then the vertical segment A 1:2 is removed, and finally the path C6 |- E3 is taken out. The result is the diagram of the beginning.

Sometimes a pattern is ideal except for one or two stones: a subtractive group removes them without redoing everything.



```
\begin{goban}[label = {at = all}]
\stones[b]{A1:E6; B:D 5, C2}
\end{goban}
```

Here, B:D 5 removes a horizontal segment on row 5, and C2 removes one stone. Additive and subtractive groups alternate with each ;, as many times as needed.



```
\begin{goban}[label = {at = all}, partial = A1:E6]
\stones[b]{A1:E6} % AD
\end{goban}

\begin{goban}[label = {at = all}, partial = A1:E6]
\stones[b]{A1:E6; A1:C4, E6} % AD, SU
\end{goban}

\begin{goban}[label = {at = all}, partial = A1:E6]
\stones[b]{A1:E6; A1:C4, E6; A 1:2} % AD, SU, AD
\end{goban}
```

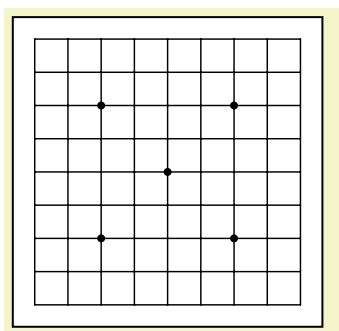
In this last example, a rectangle is created, two regions are removed (A1:C4 and E6), and then a segment is added (A 1:2).

5 The board

When we speak of the “board”, we mean the `goban` environment: the grid, the background, the coordinates and everything drawn on it.

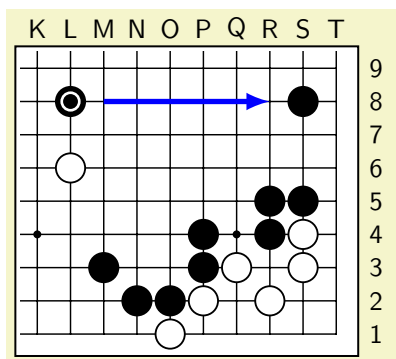
```
\begin{goban}[\langle options \rangle]
\langle environment contents \rangle
\end{goban}
```

Starts the `goban` environment. By default, it draws a square board of 9×9 intersections, with a visible border, a white background and no labels on the sides.



```
\begin{goban}
\end{goban}
```

`goban` is a `tikzpicture`, so TikZ can be used inside it. Each intersection is a point of its coordinate system, with A1 at (1,1), or it is written as in the commands, with (go cs:A1) (p. 92).



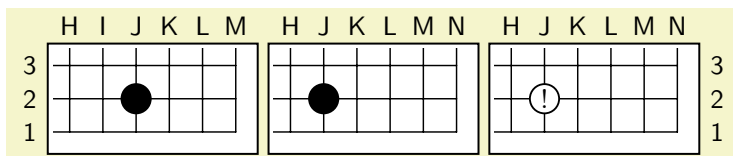
```
\begin{goban}[19, partial = K1:T9,
label = {at = {north, east}}]
\stones[b] {L8,S8,R5,S5,P4,R4,M3,P3,N2,O2}
\mark[black, circle] {L8}
\stones[white] {L6,S4,S3,Q3,O1,P2,R2}

\tikzset{myarrow/.style = {-{latex[]}, line width = 2pt, blue}}
\draw[myarrow] (12,8)--(17,8);
\end{goban}
```

`/pgfgo/goban/use i=<boolean>`

(default true, initially false)

Uses the letter “I” in the coordinates and in the labels of the board. By default it is not used, as is customary in Go: the columns go from “H” to “J”, and any reference to “I” is read as “J”. The “I” is skipped only between A and Z: among the columns of two letters, AI is a column (the one after AH).



```
\pgfgoreset{goban = {13, label = {at = north}}}}

\begin{goban}[use i, label = {at = west}, partial = H1:M3]
\stone[b]{J2}
\end{goban}

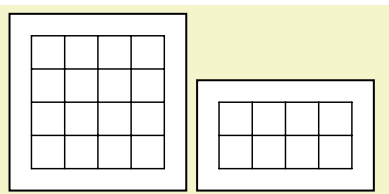
\begin{goban}[partial = H1:N3]
\stone[b]{J2}
\end{goban}

\begin{goban}[label = {at = east}, partial = H1:N3]
\stone[white, mark = {custom = !}] {I2} % If it is I, it becomes J.
\end{goban}
```

`/pgfgo/goban/size=<nlines or nlines x nlines>`

(no default, initially 9)

Sets the number of vertical and horizontal lines of the board. The word “size” can be left out and the value given directly. With a single integer (such as 9), the board is square (9×9). With an expression such as 5x3, the board is rectangular, with 5 columns and 3 rows.



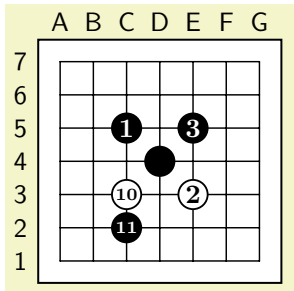
```
\begin{goban}[5]
\end{goban}

\begin{goban}[5x3]
\end{goban}
```

`/pgfgo/goban/sgf size=<file>`

(no default)

Takes the size of the board from an SGF game (its property SZ; with a name with **transpose**, turned, p. 107) and, if it is a Sensei’s Library diagram, its **blank** points (the `_`). Only with the Lua kernel (p. 30).



```
\begin{goban}[sgf size = variations, label = {at = {north, west}}]
\sgf{variations}
\end{goban}
```

`/pgfgo/goban/sgf view=<file>` (no default)

Like `sgf size` and, if the file has a view (VW), shows only that part of the board, like `partial`. The view can change during the game; the one of the move `sgf view move` is used.

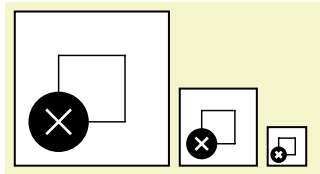
`/pgfgo/goban/sgf view move=<number>` (no default, initially 0)

The move from which `sgf view` takes the view. A VW holds from the move of its node (or from the next one, if the node has no move) to the next VW, and VW[] shows the whole board again. With 0, it is the view at the beginning. It can go before or after `sgf view`. `\sgfdiagrams`, `\sgfanimation` and `\sgfproblem` choose the view by themselves.

```
\begin{goban}[sgf view = game, sgf view move = 120]
\sgf[from = 101, to = 120]{game}
\end{goban}
```

`/pgfgo/goban/scale=<float>` (no default, initially 1)

Scale of the board: the sizes. The width of the lines does not change, except with `scale line width`.



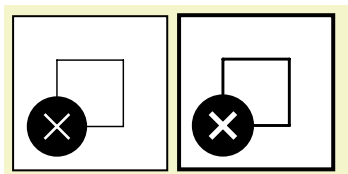
```
\begin{goban}[2, scale = 2]
\stone[b, mark]{A1}
\end{goban}

\begin{goban}[2, scale = 1]
\stone[b, mark]{A1}
\end{goban}

\begin{goban}[2, scale = 0.5]
\stone[b, mark]{A1}
\end{goban}
```

`/pgfgo/goban/scale line width=<boolean>` (default true, initially false)

With `true`, `scale` also multiplies the width of the lines: the grid, the border, the stones, the marks, the arrows, the lines and the territory. With `layout/fit`, which chooses the scale to fill the column, the width does not change (p. 131).



```
\begin{goban}[2, scale = 2]
\stone[b, mark]{A1}
\end{goban}

\begin{goban}[2, scale = 2, scale line width]
\stone[b, mark]{A1}
\end{goban}
```

5.1 Grid

The grid is the lines of the board.

`/pgfgo/goban/grid=<options>` (default `true`)

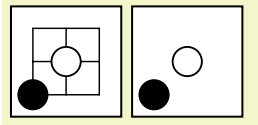
Options of the grid.

`/pgfgo/goban/grid/true` (style, no value)

Draws the grid again after a `grid = false` (for example, a global one set with `\pgfgoset`).

`/pgfgo/goban/grid/false` (style, no value)

Removes the grid and the star points.



```
\begin{goban}[3]
\stone[b]{A1}
\stone[w]{B2}
\end{goban}

\begin{goban}[3, grid = false]
\stone[b]{A1}
\stone[w]{B2}
\end{goban}
```

The first board shows the default grid.

`/pgfgo/goban/grid/line=<options>` (default `true`)

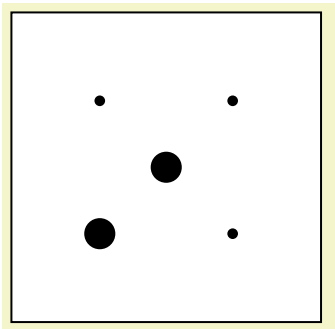
Options of the lines of the grid.

`/pgfgo/goban/grid/line/true` (style, no value)

Draws the lines. It is the initial value.

`/pgfgo/goban/grid/line/false` (style, no value)

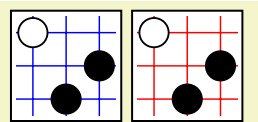
Does not draw the lines, but does draw the star points (unlike `grid = false`).



```
\begin{goban}[9, grid = {line = false}, hoshi = {radius = 2pt}]
\stones[b]{C3, E5}
\end{goban}
```

`/pgfgo/goban/grid/line/color = <color>` (no default, initially `black`)

Color of the lines of the grid.



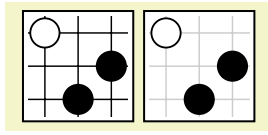
```
\pgfgoset{goban = {5, partial = B2:D4}}

\begin{goban}[grid = {line = {color = blue}}, name = a]
\stones[b]{C2, D3}
\stone[w]{B4}
\end{goban}

\begin{goban}[grid = {line = {color = red}}, resume = a]
\end{goban}
```

`/pgfgo/goban/grid/line/opacity` = $\langle float \rangle$ (no default, initially 1)

Opacity of the lines of the grid.



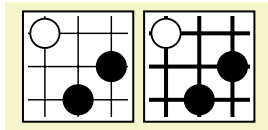
```
\pgfgo{goban} = {5, partial = B2:D4}}

\begin{goban}[name = a]
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}[grid = {line = {opacity = 0.2}}, resume = a]
\end{goban}
```

`/pgfgo/goban/grid/line/width` = $\langle dimension \rangle$ (no default, initially 0.55pt)

Width of the lines of the grid.



```
\pgfgo{goban} = {5, partial = B2:D4}}

\begin{goban}[name = a]
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}[grid = {line = {width = 1.5pt}}, resume = a]
\end{goban}
```

`/pgfgo/goban/grid/x sep` = $\langle dimension \rangle$ (no default, initially 12.5pt)

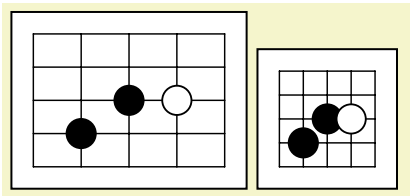
Distance between the vertical lines.

`/pgfgo/goban/grid/y sep` = $\langle dimension \rangle$ (no default, initially 12.5pt)

Distance between the horizontal lines.

`/pgfgo/goban/grid/sep` = $\langle dimension \rangle$ (no default, initially 12.5pt)

Both distances at once. The initial sizes of the board (distances, radii, sizes of the numbers) are in pt: the board does not change with the size of the font of the document. A size in em is computed with the font where the *key* is set.

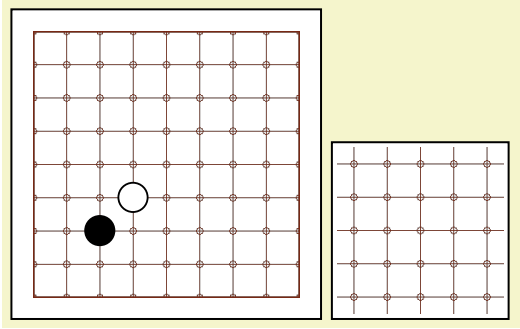


```
\begin{goban}[5, grid = {x sep = 1.8em}]
\stones[b]{B2, C3}
\stone[w]{D3}
\end{goban}

\begin{goban}[5, grid = {sep = 0.9em}]
\stones[b]{B2, C3}
\stone[w]{D3}
\end{goban}
```

`/pgfgo/goban/grid/image` = $\langle options \rangle$ (no default)

Draws an image instead of the lines of the grid, with the opacity of `grid/line/opacity`. With `partial` the corresponding part is seen. The star points are still drawn (see `hoshi`).

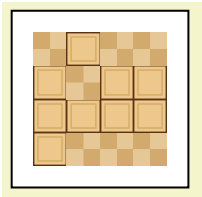


```
\begin{goban}[9, grid = {image = {file = pgfgo-grid.png}}, hoshi = false]
\stones[b]{C3}\stone[w]{D4}
\end{goban}

\begin{goban}[9, grid = {image = {file = pgfgo-grid.png}}, hoshi = false, partial = C3:G7]
\end{goban}
```

/pgfgo/goban/grid/image/file = $\langle file, file, \dots \rangle$ (no default)

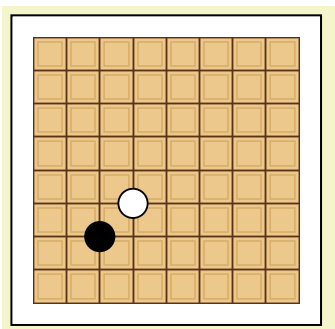
The file of the image. With several files and **fit** = **tile**, each tile uses one, chosen by its place in the mesh: always the same one in each place, in every diagram. With **fit** = **stretch**, the first one is used.



```
\begin{goban}[5, hoshi = false,
grid = {image = {file = {pgfgo-cell.png, pgfgo-tile.png}}, fit = tile}]
\end{goban}
```

/pgfgo/goban/grid/image/fit = $\langle stretch \text{ or } tile \rangle$ (no default, initially **stretch**)

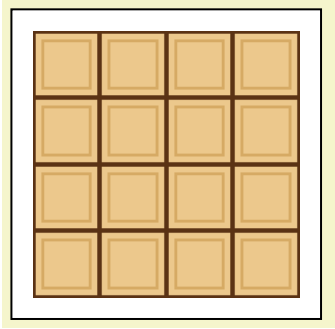
With **stretch**, the image is stretched from A1 to the opposite corner of the whole board, so the outer lines have to be on the edges of the image. With **tile**, it is drawn once in each square of the grid (between four intersections), stretched to the size of the square: with the image of one square the whole board is built.



```
\begin{goban}[9, grid = {image = {file = pgfgo-cell.png, fit = tile}}, hoshi = false]
\stones[b]{C3}\stone[w]{D4}
\end{goban}
```

/pgfgo/goban/grid/image/tiles = $\langle columns \rangle \times \langle rows \rangle$ (no default, initially one per square)

With **fit** = **tile**, the mesh in which the image is repeated, over the area that goes from A1 to the opposite corner of the whole board; each image is stretched to the size of its cell. A single number holds for both directions. Without this *key*, there is one image per square.



```
\begin{goban}[9, grid = {image = {file = pgfgo-cell.png, fit = tile, tiles = 4}}, hoshi = false]
\end{goban}
```

`/pgfgo/goban/grid/image/rotate` = $\langle angle \rangle$, random or $\langle minimum \rangle : \langle maximum \rangle$ (no default, initially 0)

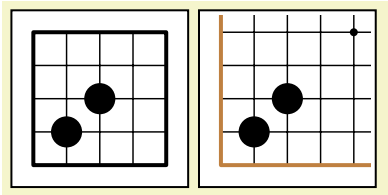
Turns the image, like `background/fill/image/rotate`: in quarter turns, and with `random` or a range, each tile with its own.

`/pgfgo/goban/grid/image/false` (style, no value)

Removes the image: the lines come back.

`/pgfgo/goban/grid/edge` = $\langle options \rangle$ (no default)

The edge of the grid: the four outer lines, with their width, color and opacity, over the lines or the image. What is not given is as in the rest of the grid. With `partial`, only the edges of the board that are in view are seen; with `blank`, the stretches of the first and last row and column.



```
\begin{goban}[5, grid = {edge = {width = 1.5pt}}]
\stones[b]{B2, C3}
\end{goban}

\begin{goban}[9, partial = A1:E5, grid = {edge = {width = 1.5pt, color = brown}}]
\stones[b]{B2, C3}
\end{goban}
```

`/pgfgo/goban/grid/edge/width` = $\langle dimension \rangle$ (no default, initially that of `grid/line/width`)

The width of the edge.

`/pgfgo/goban/grid/edge/color` = $\langle color \rangle$ (no default, initially that of `grid/line/color`)

The color of the edge.

`/pgfgo/goban/grid/edge/opacity` = $\langle float \rangle$ (no default, initially that of `grid/line/opacity`)

The opacity of the edge.

`/pgfgo/goban/grid/edge/false` (style, no value)

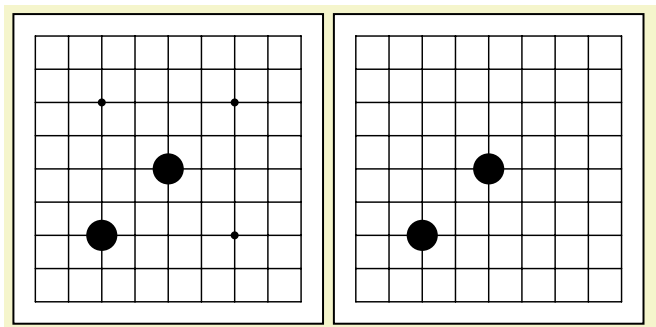
Back to an edge like the rest of the grid.

5.2 Star points

The star points (*hoshi*) are drawn on the grid, with its color. By default they are the usual ones: on each side, the fourth line from the edge (the third on boards of fewer than 13 lines); the center if the size is odd; and, from 15 lines on, the middle of the sides. A 19 board has nine, and a 13 or a 9 board, five. Boards of fewer than 7 lines have none. Stones cover them.

`/pgfgo/goban/hoshi=<options>` (default `true`)

Options of the star points.



```
\begin{goban}[9]
\stones[b]{C3, E5}
\end{goban}

\begin{goban}[9, hoshi = false]
\stones[b]{C3, E5}
\end{goban}
```

`/pgfgo/goban/hoshi/true` (style, no value)

Draws the star points. It is the initial value.

`/pgfgo/goban/hoshi/false` (style, no value)

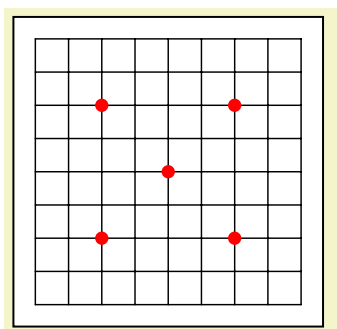
Does not draw the star points.

`/pgfgo/goban/hoshi/radius=<dimension>` (no default, initially 1.5pt)

Radius of the points.

`/pgfgo/goban/hoshi/color=<color>` (no default)

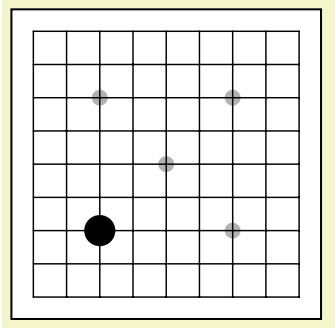
Color of the points. Without this *key*, they take the color of the grid.



```
\begin{goban}[9, hoshi = {radius = 2.5pt, color = red}]
\end{goban}
```

`/pgfgo/goban/hoshi/opacity=<float>` (no default)

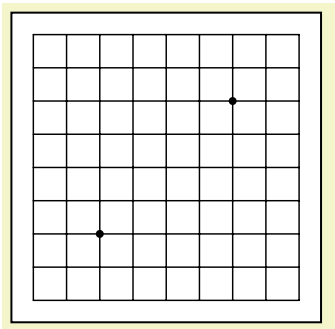
Opacity of the points (also of their image). Without this *key*, they take the opacity of the grid.



```
\begin{goban}[9, hoshi = {radius = 3pt, opacity = 0.3}]
\stones[b]{C3}
\end{goban}
```

`/pgfgo/goban/hoshi/at=<coordinates>` (no default)

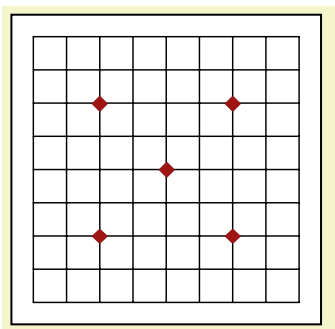
Chooses the points, with the same syntax as `\stones`. Empty, the usual ones are used.



```
\begin{goban}[9, hoshi = {at = {C3, G7}}]
\end{goban}
```

`/pgfgo/goban/hoshi/image=<options>` (no default)

Draws an image on each star point, instead of the circle.



```
\begin{goban}[9, hoshi = {image = {file = pgfgo-hoshi.png}, radius = 3pt}]
\end{goban}
```

`/pgfgo/goban/hoshi/image/file=<file>` (no default)

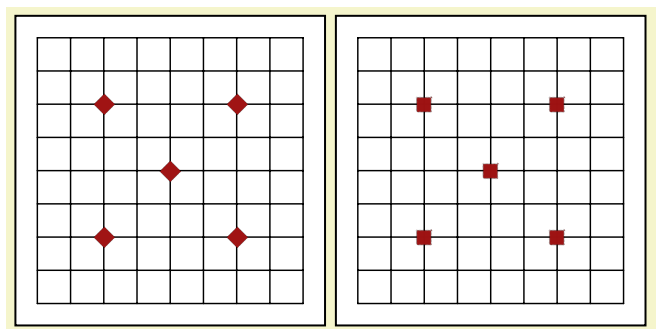
The file of the image.

`/pgfgo/goban/hoshi/image/width=<dimension>` (no default, initially $2 \times \text{radius}$)

The width of the image; the height keeps the proportion.

`/pgfgo/goban/hoshi/image/rotate=<angle>`, `random` or `<minimum>:<maximum>` (no default, initially 0)

Turns the image, like `stone/image/rotate` (p. 70): with `random` or a range, each point with a different angle, always the same.



```
\begin{goban}[9, hoshi = {image = {file = pgfgo-hoshi.png, width = 8pt}}]
\end{goban}

\begin{goban}[9, hoshi = {image = {file = pgfgo-hoshi.png, width = 8pt, rotate = 45}}]
\end{goban}
```

`/pgfgo/goban/hoshi/image/false` (style, no value)

Removes the image: the circles come back.

5.3 Border

The background of the board is the rectangle that goes under the grid, with its fill and its border. By default, the fill is white and the border, black.

`/pgfgo/goban/background=<options>` (default `true`)

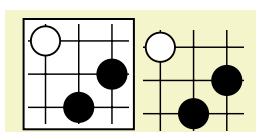
Options of the background and of the border.

`/pgfgo/goban/background/true` (style, no value)

Draws the background and the border again after a `background = false`.

`/pgfgo/goban/background/false` (style, no value)

Removes the background: the fill and the border.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}[name = a]
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

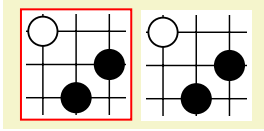
\begin{goban}[background = false, resume = a]
\end{goban}
```

`/pgfgo/goban/background/fill=<options>` (default `true`)

Options of the fill.

`/pgfgo/goban/background/fill=<boolean>` (no default, initially `true`)

With `false`, the background is transparent and only the border remains.



```
\pgfset{goban = {5, partial = B2:D4}}

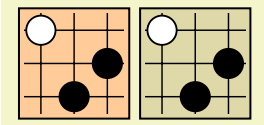
\begin{goban}[background = {line = {color = red}}, name = a]
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}[background = {line = false}, resume = a]
\end{goban}
```

`/pgfgo/goban/background/fill/color = <color>`

(no default, initially white)

Color of the fill.



```
\pgfset{goban = {5, partial = B2:D4}}

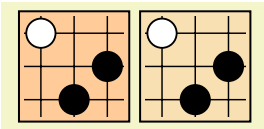
\begin{goban}[background = {fill = {color = orange!40}}, name = a]
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}[background = {fill = {color = olive!30}}, resume = a]
\end{goban}
```

`/pgfgo/goban/background/fill/opacity = <float>`

(no default, initially 1)

Opacity of the fill.



```
\pgfset{goban = {5, partial = B2:D4}}

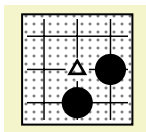
\begin{goban}[background = {fill = {color = orange!40}}, name = a]
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}[resume = a,
background = {fill = {color = orange!40}, fill = {opacity = 0.5}}]
\end{goban}
```

`/pgfgo/goban/background/fill/pattern = <options>`

(no default)

A pattern over the color and the image of the background, as `stone/fill/pattern` (p. 62). The marks on empty points and the mark `dim`, which cover the grid with the color of the background, repeat the pattern. Here `fill/color` cannot be `none`.



```
\pgfset{goban = {5, partial = B2:D4}}

\begin{goban}[background = {fill = {
pattern = {name = dots, color = gray}}}]]
\stones[b]{C2,D3}
\mark[triangle]{C3}
\end{goban}
```

`/pgfgo/goban/background/fill/pattern/name = <pattern>`

(no default)

The pattern: any of the TikZ libraries `patterns` and `patterns.meta`, which PGFGO loads. One with options goes in braces: `name = {Lines[angle = 45, distance = 2pt]}`. Without `name` there is no pattern.

`/pgfgo/goban/background/fill/pattern/color = <color>`

(no default, initially black)

Color of the pattern.

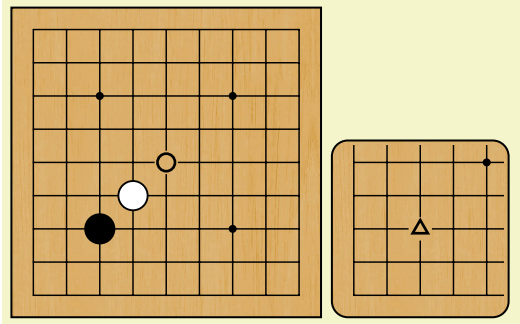
`/pgfgo/goban/background/fill/pattern/false`

(no value)

No pattern.

/pgfgo/goban/background/fill/image = $\langle options \rangle$ (no default)

A background image (for example, the wood of a board), over the fill color and with its same opacity. The image is placed according to the whole board: with **partial** the corresponding part is seen, cut by the border and its rounded corners. The marks on empty points, which cover the grid with the color of the background, show the part of the image underneath, and so does the mark **dim**. The halo of arrows and lines is still the color of the background.

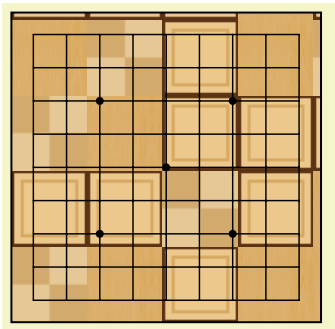


```
\begin{goban}[9,
  background = {fill = {image = {file = pgfgo-board.jpg}}}
  \stones[b]{C3}\stone[w]{D4}
  \marks[circle]{E5, C3}
\end{goban}

\begin{goban}[9, partial = A1:E5,
  background = {fill = {image = {file = pgfgo-board.jpg}},
    rounded corners = 6pt}]
  \marks[triangle]{C3}
\end{goban}
```

/pgfgo/goban/background/fill/image/file = $\langle file, file, ... \rangle$ (no default)

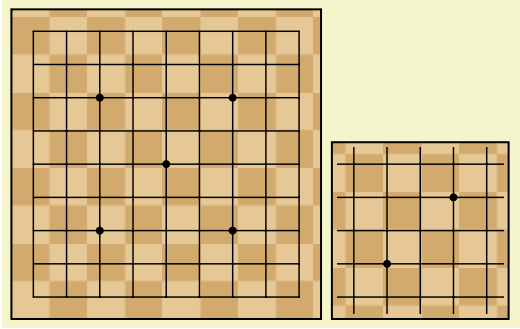
The file of the background image. With several files and **fit = tile**, each tile uses one, chosen by its place: always the same one in each place, also with **partial**. That way the board does not look repeated. The tiles have the size of the first image (or that of their cell, with **tiles**). With **stretch** and **cover**, the first one is used.



```
\begin{goban}[9, background = {fill = {image = {
  file = {pgfgo-tile.png, pgfgo-cell.png, pgfgo-board.jpg},
  fit = tile, width = 1cm}}}]
\end{goban}
```

/pgfgo/goban/background/fill/image/fit = $\langle stretch, cover or tile \rangle$ (no default, initially **cover**)

How the image fills the background of the whole board. **stretch** stretches it to fill the whole background, without keeping the proportion. **cover** enlarges or reduces it, keeping the proportion, until it covers the background, centered: what is left over is not seen. **tile** repeats it as tiles, from a corner, an edge or the center of the whole board (see **anchor**).

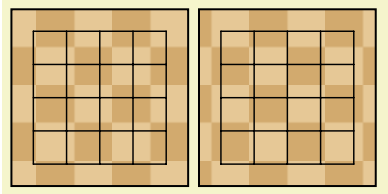


```
\begin{goban}[9, background = {fill = {image = {
  file = pgfgo-tile.png, fit = tile, width = 1cm}}}]
\end{goban}

\begin{goban}[9, partial = D4:H8, background = {fill = {image = {
  file = pgfgo-tile.png, fit = tile, width = 1cm}}}]
\end{goban}
```

`/pgfgo/goban/background/fill/image/anchor` = $\langle position \rangle$ (no default, initially **south west**)

With `fit = tile`, where the tiles start, on the background of the whole board. With a corner (south west, south east, north west or north east), a whole tile is in that corner. With an edge (north, south, east or west), the first tile is against that edge, centered along it. With **center**, a tile is in the center of the board.



```
\begin{goban}[5, background = {fill = {image = {
  file = pgfgo-tile.png, fit = tile, width = 1cm, anchor = north east}}}]
\end{goban}

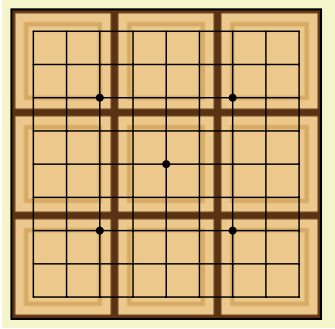
\begin{goban}[5, background = {fill = {image = {
  file = pgfgo-tile.png, fit = tile, width = 1cm, anchor = north}}}]
\end{goban}
```

`/pgfgo/goban/background/fill/image/width` = $\langle dimension \rangle$ (no default, initially **the natural width**)

With `fit = tile`, the width of each tile; the height keeps the proportion.

`/pgfgo/goban/background/fill/image/tiles` = $\langle columns \rangle \times \langle rows \rangle$ (no default)

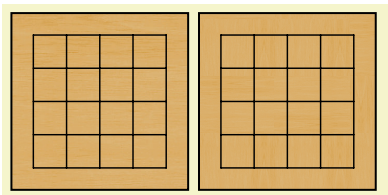
With `fit = tile`, instead of tiles of width `width` from **anchor**, divides the background of the whole board into that mesh and puts the image in each cell, stretched to its size. A single number holds for both directions.



```
\begin{goban}[9, background = {fill = {image = {
file = pgfgo-cell.png, fit = tile, tiles = 3x3}}}]
\end{goban}
```

`/pgfgo/goban/background/fill/image/rotate` = $\langle angle \rangle$, `random` or $\langle minimum \rangle : \langle maximum \rangle$ (no default, initially 0)

Turns the image in quarter turns: the angle is rounded to 0, 90, 180 or 270 degrees, so that the turned image still fills its place without leaving gaps. With 90 or 270, the width and the height are swapped: with `cover` the image keeps the proportion, but a tile of width `width` that is not square is stretched. With `random` or a range, as in `stone/image/rotate` (p. 70), each tile has its own turn, always the same.



```
\begin{goban}[5, background = {fill = {image = {
file = pgfgo-board.jpg, rotate = 90}}}]
\end{goban}

\begin{goban}[5, background = {fill = {image = {
file = pgfgo-board.jpg, fit = tile, width = 1cm, rotate = random}}}]
\end{goban}
```

`/pgfgo/goban/background/fill/image/false` (style, no value)

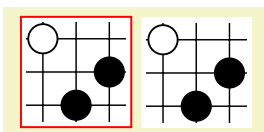
Removes the background image.

`/pgfgo/goban/background/line` = $\langle options \rangle$ (default `true`)

Options of the border.

`/pgfgo/goban/background/line` = $\langle boolean \rangle$ (no default, initially `true`)

Draws (`true`) or removes (`false`) the border.

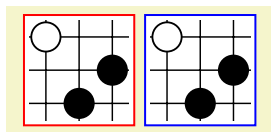


```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}[background = {line = {color = red}}, name = a]
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}[background = {line = false}, resume = a]
\end{goban}
```

`/pgfgo/goban/background/line/color` = $\langle color \rangle$ (no default, initially black)
 Color of the border.

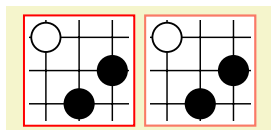


```
\pgfgoset{goban = {5, partial = B2:D4}}

\begin{goban}[background = {line = {color = red}}, name = a]
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}[background = {line = {color = blue}}, resume = a]
\end{goban}
```

`/pgfgo/goban/background/line/opacity` = $\langle float \rangle$ (no default, initially 1)
 Opacity of the border.

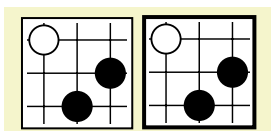


```
\pgfgoset{goban = {5, partial = B2:D4}}

\begin{goban}[background = {line = {color = red}}, name = a]
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}[resume = a,
background = {line = {color = red}, line = {opacity = 0.5}}]
\end{goban}
```

`/pgfgo/goban/background/line/width` = $\langle dimension \rangle$ (no default, initially 0.75pt)
 Width of the border.

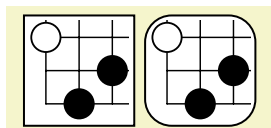


```
\pgfgoset{goban = {5, partial = B2:D4}}

\begin{goban}[name = a]
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}[background = {line = {width = 1.5pt}}, resume = a]
\end{goban}
```

`/pgfgo/goban/background/rounded corners` = $\langle dimension \rangle$ (no default, initially 0pt)
 Radius of the corners of the background. With 0pt, they are square.

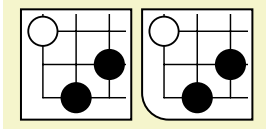


```
\pgfgoset{goban = {5, partial = A1:C3}}

\begin{goban}[name = a]
\stones[b]{B1,C2}
\stone[w]{A3}
\end{goban}

\begin{goban}[background = {rounded corners = 10pt}, resume = a]
\end{goban}
```

`/pgfgo/goban/background/rounded corners/outer` = $\langle dimension \rangle$ (no default, initially 0pt)
 Only that of the outer corners.

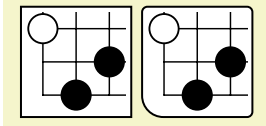


```
\pgfgoset{goban = {5, partial = A1:C3}}

\begin{goban}[background = {rounded corners = Opt}, name = a]
\stones[b]{B1,C2}
\stone[w]{A3}
\end{goban}

\begin{goban}[background = {rounded corners = {outer = 10pt}}, resume = a]
\end{goban}
```

`/pgfgo/goban/background/rounded corners/inner` = $\langle dimension \rangle$ (no default, initially 0pt)
Only that of the inner corners (they are seen on partial boards).

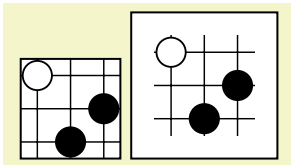


```
\pgfgoset{goban = {5, partial = A1:C3}}

\begin{goban}[background = {rounded corners = Opt}, name = a]
\stones[b]{B1,C2}
\stone[w]{A3}
\end{goban}

\begin{goban}[resume = a,
background = {rounded corners = {outer = 8pt, inner = 2pt}}]
\end{goban}
```

`/pgfgo/goban/background/sep` = $\langle dimension \rangle$ (no default, initially 8.25pt)
Distance between the grid and the border. By default, 1.5 times the radius of the stones.

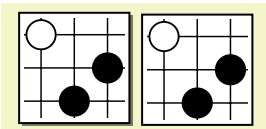


```
\pgfgoset{goban = {5, partial = B2:D4}}

\begin{goban}[background = {sep = 0.625em}, name = a]
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}[background = {sep = 1.5em}, resume = a]
\end{goban}
```

`/pgfgo/goban/background/shadow` = $\langle options \rangle$ (default true)
Draws a shadow under the board. Without a value, turns it on.



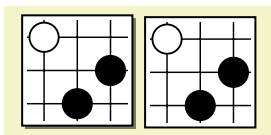
```
\pgfgoset{goban = {5, partial = B2:D4}}

\begin{goban}[background = {shadow}, name = a]
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}[background = {shadow = {color = blue, opacity = 0.3}}, resume = a]
\end{goban}
```

`/pgfgo/goban/background/shadow` = $\langle boolean \rangle$ (no default, initially false)
Turns the shadow on (true) or off (false).

`/pgfgo/goban/background/shadow/angle` = $\langle float \rangle$ (no default, initially -45)
Direction of the shadow, in degrees.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

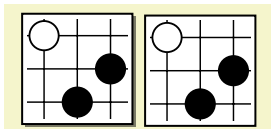
\begin{goban}[background = {shadow}, name = a]
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}[background = {shadow = {angle = 45}}, resume = a]
\end{goban}
```

/pgfgo/goban/background/shadow/offset = $\langle dimension \rangle$

(no default, initially 1.5pt)

Distance between the board and its shadow.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}[background = {shadow}, name = a]
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}[background = {shadow = {offset = 4pt}}, resume = a]
\end{goban}
```

/pgfgo/goban/background/shadow/opacity = $\langle float \rangle$

(no default, initially 0.5)

Opacity of the shadow.

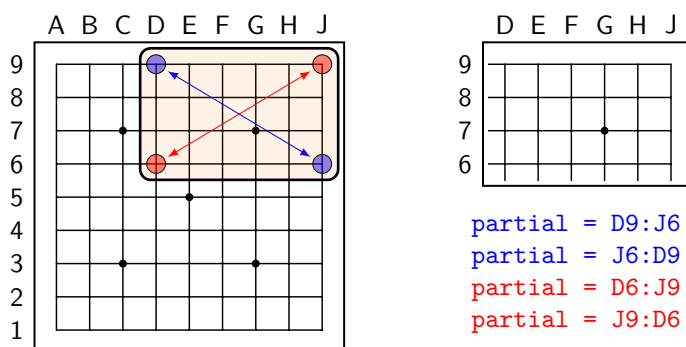
/pgfgo/goban/background/shadow/color = $\langle color \rangle$

(no default, initially black)

Color of the shadow.

5.4 Partial board

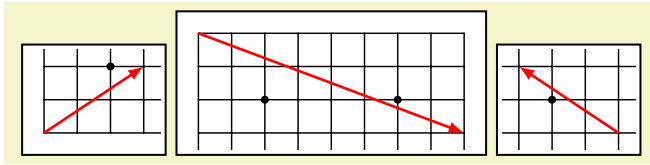
Often the whole board is not needed, only a part. The *key* **partial** sets it with two opposite corners of a rectangle, in any order. For example, **partial** = A1:D3 is equivalent to **partial** = D3:A1.



/pgfgo/goban/partial = $\langle intersection \rangle$: $\langle intersection \rangle$, **auto** or **false**

(no default)

Shows only part of the board: the rectangle that has those two intersections as opposite corners, in any order. It can also be written **partial** = {B2, D4} or **partial** = {B2 to D4}. With **auto**, the board chooses the part: the one used by the stones, moves, marks, arrows, lines and territories that are drawn, with **partial margin** more lines. What is drawn directly with *TikZ* does not count. With **false**, the whole board is shown again (for example, after a global **partial** set with **\pgfgoreset**).



```
\tikzset{myarrow/.style = {-{latex[]}, line width = 1pt, red}}

\begin{goban}[partial = A1:D3]
\draw[myarrow] (1,1)--(4,3); % From A1 to D3
\end{goban}

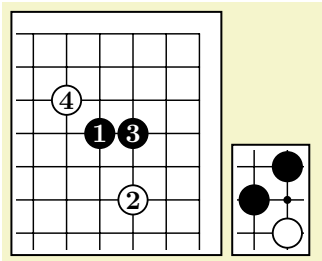
\begin{goban}[partial = A9:J6]
\draw[myarrow] (1,9)--(9,6); % From A9 to J6
\end{goban}

\begin{goban}[partial = E2:B4]
\draw[myarrow] (5,2)--(2,4); % From E2 to B4
\end{goban}
```

/pgfgo/goban/partial margin=*<number>*

(no default, initially 1)

Lines that **partial = auto** adds around what is drawn. A side two lines or less from the edge reaches the edge. With nothing drawn, the whole board is seen.



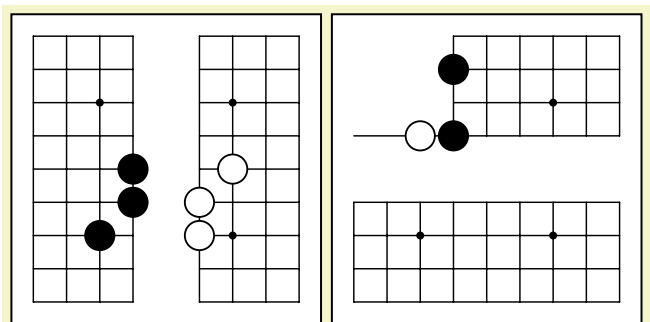
```
\begin{goban}[19, partial = auto]
\moves{Q16, R14, R16, P17}
\end{goban}

\begin{goban}[19, partial = auto, partial margin = 0]
\stones[b]{J10, K11}\stones[w]{K9}
\end{goban}
```

/pgfgo/goban/blank=*<coordinates>*

(no default)

The intersections that are not part of the board: they have no lines and no star point. A line joins only two neighbouring intersections of the board, so next to a **blank** point it ends as at the edge; an intersection without neighbours has no lines. The stones, the marks and the background do not change. With **grid/image**, the image is seen only in the cells of the intersections of the board. The **_** of Sensei's Library diagrams are **blank** points (p. 109).



```

\begin{goban}[9, blank = E]
\stones[b]{C3, D4, D5}
\stones[w]{F3, F4, G5}
\end{goban}

\begin{goban}[9, blank = {A9:C7, 5}]
\stones[b]{D8, D6}
\stones[w]{C6}
\end{goban}

```

5.5 Labels of the board

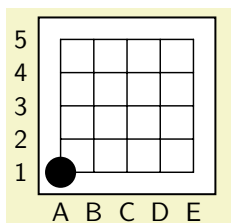
The labels are the coordinates written on the sides of the board.

`/pgfgo/goban/label=<options>` (default **true**)

Options of the labels.

`/pgfgo/goban/label/true` (no value)

Puts the letters below and the numbers on the left: **A1** is read from the lower left corner. It is the same as `label = {at = {south, west}}`; for the four sides, `label = {at = all}`.



```

\begin{goban}[5, label]
\stones[b]{A1}
\end{goban}

```

`/pgfgo/goban/label/false` (no value)

Turns off all the labels.

`/pgfgo/goban/label/at =<options>` (no default)

On which sides of the board the labels go: one or several cardinal points, **all** or **none**.

`/pgfgo/goban/label/at/all` (style, no value)

On the four sides.

`/pgfgo/goban/label/at/none` (style, no value)

The same as `label = false`.

`/pgfgo/goban/label/at/east` (style, no value)

On the right (the numbers).

`/pgfgo/goban/label/at/right` (style, no value)

Equivalent to **east**.

`/pgfgo/goban/label/at/north` (style, no value)

Above (the letters).

`/pgfgo/goban/label/at/above` (style, no value)

Equivalent to **north**.

`/pgfgo/goban/label/at/south` (style, no value)

Below (the letters).

`/pgfgo/goban/label/at/below` (style, no value)

Equivalent to **south**.

`/pgfgo/goban/label/at/west`

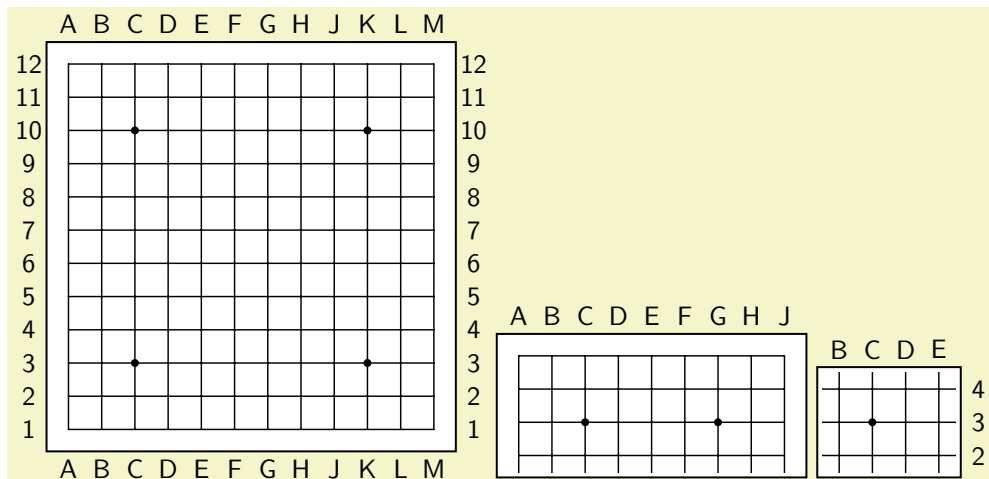
(style, no value)

On the left (the numbers).

`/pgfgo/goban/label/at/left`

(style, no value)

Equivalent to `west`.



```
\begin{goban}[12, label = {at = {all}}]
\end{goban}

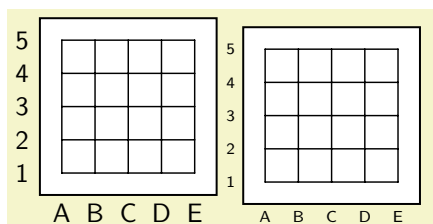
\begin{goban}[partial = A6:I9, label = {at = north}]
\end{goban}

\begin{goban}[partial = B2:E4, label = {at = {north, east}}]
\end{goban}
```

`/pgfgo/goban/label/font size =  $\langle dimension \rangle$`

(no default, initially 10pt)

Size of the font of the labels. The labels move away from the board half of this size, besides `label/sep`. Empty, it is that of the current font.



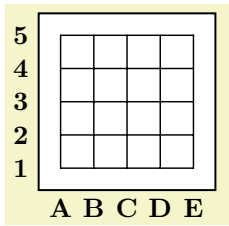
```
\begin{goban}[5, label = {at = {south, west}}]
\end{goban}

\begin{goban}[5, label = {at = {south, west}, font size = 6pt}]
\end{goban}
```

`/pgfgo/goban/label/font =  $\langle font commands \rangle$`

(no default, initially `\sffamily`)

Font of the labels.



```
\begin{goban}[5, label = {at = {south, west}, font = \bfseries}]
\end{goban}
```

`/pgfgo/goban/label/sep =  $\langle dimension \rangle$`

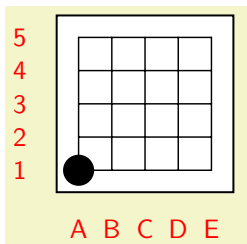
(no default, initially 0.35cm)

Distance between the labels and the border of the board.

`/pgfgo/goban/label/color =  $\langle color \rangle$`

(no default, initially black)

Color of the labels.



```
\begin{goban}[5, label = {at = {south, west}, color = red, sep = 0.6cm}]
\stones[b]{A1}
\end{goban}
```

`/pgfgo/goban/label/columns =  $\langle letters, numbers or kanji \rangle$`

(no default, initially letters)

How the coordinates of the columns (above and below) are written: with letters (as in the commands, without the I except with `use i`), with numbers or with numbers in kanji (一, 二, ..., 十, 十一, ...).

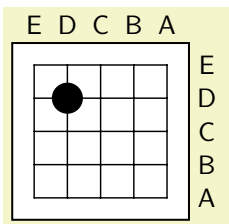
`/pgfgo/goban/label/rows =  $\langle letters, numbers or kanji \rangle$`

(no default, initially numbers)

The same for the rows (on the sides). The kanji of the rows are aligned on the side of the board, so that those of two characters do not run into it.

`/pgfgo/goban/label/columns direction =  $\langle left to right or right to left \rangle$` (no default, initially left to right)

In which direction the columns are counted.

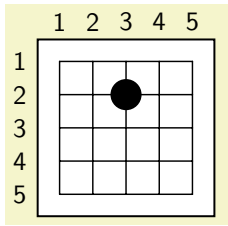


```
\begin{goban}[5, label = {at = {north, east},
columns direction = right to left, rows = letters}]
\stones[b]{B4}
\end{goban}
```

`/pgfgo/goban/label/rows direction =  $\langle bottom to top or top to bottom \rangle$`

(no default, initially bottom to top)

In which direction the rows are counted. These four options change only what is written on the border: the coordinates of the commands (C3) stay the same. On a partial board the count starts from the edge of the whole board.



```
\begin{goban}[5, label = {at = {north, west}, columns = numbers,
rows direction = top to bottom}]
\stones[b]{C4}
\end{goban}
```

`/pgfgo/goban/label/style = <western, japanese or chinese>` (no default, initially **western**)

Sets the four previous options according to the custom of each place. The coordinates of the commands do not change: A1 is still at the bottom left.

`/pgfgo/goban/label/style/western` (style, no value)

Columns with letters from the left and rows with numbers from below.

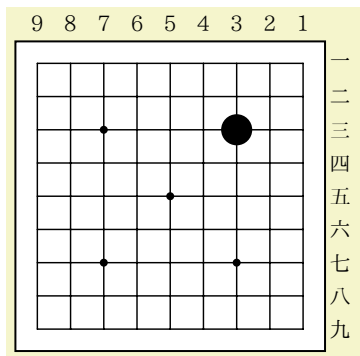
`/pgfgo/goban/label/style/japanese` (style, no value)

Columns with numbers from the right and rows with kanji from above, as in Japanese books: the upper right corner is 1-一.

`/pgfgo/goban/label/style/chinese` (style, no value)

Columns with numbers from the left and rows with kanji from above.

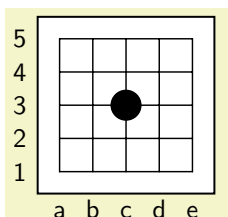
Kanji need a font that has them. With Lua^AT_EX, for example, IPAex (included in T_EX Live and in MiK_TE_X):



```
\font\kanji = {file:ipaexm.ttf} at 8pt
\begin{goban}[9, label = {at = {north, east}, style = japanese, font = \kanji}]
\stones[b]{G7}
\end{goban}
```

`/pgfgo/goban/label/letter case = <upper or lower>` (no default, initially **upper**)

The letters of the coordinates, in upper case or in lower case. Like the directions, it changes only what is written on the border: the commands still use C3.



```
\begin{goban}[5, label = {at = {south, west}, letter case = lower}]
\stones[b]{C3}
\end{goban}
```

6 Stones, moves and prisoners

Stones are drawn with `\stone` and `\stones`; moves, with `\moves`, and prisoners, with `\prisoner` and `\prisoners`.

6.1 Stones

`\stone[⟨options⟩]{⟨intersection⟩}`

Draws a stone on the intersection (for example, C2). The options change its style: the color, the size, a mark, etc. If the player is not given (b, w or n), the stone is *neutral*.

`\stones[⟨options⟩]{⟨intersection , intersection, ...⟩}`

Like `\stone`, but with several intersections, written with the coordinate syntax (p. 32). If the player is not given, the stones are *neutral* too: for black stones, write `\stones[b]{...}`.

Stones, marks and labels that fall outside the visible region of a partial board are not drawn. A coordinate that does not exist on the board (for example, E9 on a 5×5 board) is ignored and gives a warning.

An option can be changed for every player with `\pgfgoreset{stone = {...}}`. This is equivalent to `\pgfgoreset{players = {all = {stone = {...}}}}`.

`/pgfgo/stone/fill =⟨options⟩` (default true)

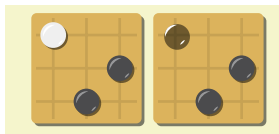
Options of the fill of the stone.

`/pgfgo/stone/fill/true` (no value)

Draws the fill.

`/pgfgo/stone/fill/false` (no value)

Does not draw the fill.



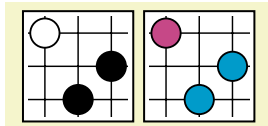
```
\pgfgoreset{profile = pgfgosmooth}
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

% What is seen in the figure is what is left of the stone: shadows and shines.
\begin{goban}
\stones[b]{C2,D3}
\stone[w, fill = false]{B4}
\end{goban}
```

`/pgfgo/stone/fill/color = ⟨color⟩` (no default)

Color of the fill. With `none`, no color: the pattern is left, if there is one.



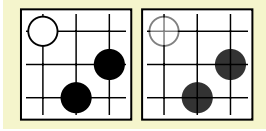
```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[fill = {color = cyan!80!black}]{C2,D3}
\stone[fill = {color = magenta!80!black}]{B4}
\end{goban}
```

`/pgfgo/stone/fill/opacity = ⟨float⟩` (no default, initially 1)

Opacity of the fill.



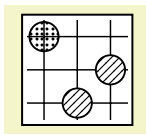
```
\pgfgo\set{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b, fill = {opacity = 0.8}]{C2,D3}
\stone[w, fill = {opacity = 0.5}]{B4}
\end{goban}
```

/pgfgo/stone/fill/pattern = *<options>* (no default)

A TikZ pattern over the fill color, with its same opacity. With **fill/color** = none, the stone has only the pattern. The marks **t-circle** and **t-square** of the player use the same fill. The pattern does not change size with **goban/scale**.



```
\pgfgo\set{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b, fill = {color = white,
pattern = {name = north east lines}}]{C2,D3}
\stone[w, fill = {color = none,
pattern = {name = {Dots[distance = 2pt]}}}]{B4}
\end{goban}
```

/pgfgo/stone/fill/pattern/name = *<pattern>* (no default)

The pattern: any of the TikZ libraries **patterns** and **patterns.meta**, which PGFGO loads. One with options goes in braces: **name** = {Lines[angle = 45, distance = 2pt]}. Without **name** there is no pattern.

/pgfgo/stone/fill/pattern/color = *<color>* (no default, initially black)

Color of the pattern.

/pgfgo/stone/fill/pattern/false (no value)

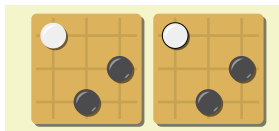
No pattern.

/pgfgo/stone/line = *<option>* (default true, initially true)

Options of the border of the stone.

/pgfgo/stone/line/true (no value)

Draws the border.



```
\pgfgo\set{profile = pgfgosmooth}
\pgfgo\set{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w, line = true]{B4}
\end{goban}
```

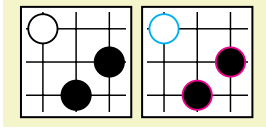
/pgfgo/stone/line/false (no value)

Does not draw the border.

`/pgfgo/stone/line/color` = $\langle color \rangle$

(no default, initially black)

Color of the border.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

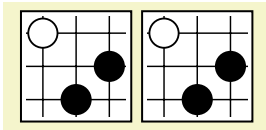
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b, line = {color = magenta}]{C2,D3}
\stone[w, line = {color = cyan}]{B4}
\end{goban}
```

`/pgfgo/stone/line/opacity` = $\langle float \rangle$

(no default, initially 1)

Opacity of the border.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

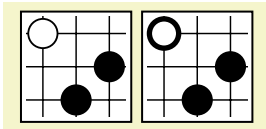
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b, line = {opacity = 0.5}]{C2,D3}
\stone[w, line = {opacity = 0.8}]{B4}
\end{goban}
```

`/pgfgo/stone/line/width` = $\langle dimension \rangle$

(no default, initially 0.7pt)

Width of the border.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

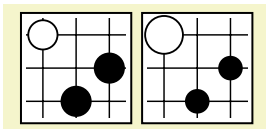
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w, line = {width = 2pt}]{B4}
\end{goban}
```

`/pgfgo/stone/radius` = $\langle dimension \rangle$

(no default, initially 5.5pt)

Radius of the stone.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

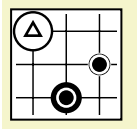
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b, radius = 0.15cm]{C2,D3}
\stone[w, radius = 0.25cm]{B4}
\end{goban}
```

`/pgfgo/stone/scale` = $\langle float \rangle$

(no default, initially 1)

Scales the whole stone: the radius, the border and the marks.

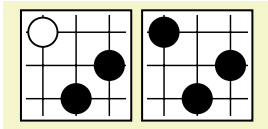


```
\begin{goban}[5, partial = B2:D4]
\stones[b, mark = circle]{C2}
\stones[b, scale = 0.7, mark = circle]{D3}
\stone[w, scale = 1.3, mark = triangle]{B4}
\end{goban}
```

/pgfgo/stone/black

(no value)

The stone belongs to the first player, with its style. Without black (or b), \stone and \stones draw neutral stones.



```
\pgfgo{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[black]{B4}
\end{goban}
```

/pgfgo/stone/b

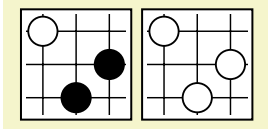
(no value)

Short for black.

/pgfgo/stone/white

(no value)

The stone belongs to the second player, with its style.



```
\pgfgo{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[white]{B4}
\end{goban}

\begin{goban}
\stones[white]{C2,D3}
\stone[white]{B4}
\end{goban}
```

/pgfgo/stone/w

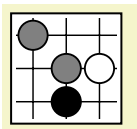
(no value)

Short for white.

/pgfgo/stone/neutral

(no value)

The stone belongs to the neutral player. It is the default player of \stone and \stones.



```
\begin{goban}[5, partial = B2:D4]
\stones[b]{C2}
\stones[w]{D3}
\stone{B4}
\stone[n]{C3}
\end{goban}
```

/pgfgo/stone/n

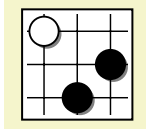
(no value)

Short for “neutral”.

/pgfgo/stone/shadow =*<options>*

(default true)

Shadow of the stone, drawn underneath. By default it has the silhouette of the stone: the circle, or the shape of the image if the stone is an image (image). With shadow/image, the shadow is an image of its own. In both cases it is moved offset towards angle and enlarged scale times.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}
  \stones[b, shadow = {true, angle = -45, offset = 1.5pt}]{C2,D3}
  \stone[w, shadow = {true, angle = -45, offset = 1.5pt}]{B4}
\end{goban}
```

/pgfgo/stone/shadow/true

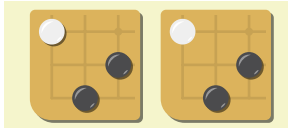
(no value)

Draws the shadow.

/pgfgo/stone/shadow/false

(no value)

Does not draw the shadow.



```
\pgfgoreset{profile = pgfgosmooth}
\pgfgoreset{goban = {partial = A1:C3}}

\begin{goban}
  \stones[b]{B1,C2}
  \stone[w]{A3}
\end{goban}

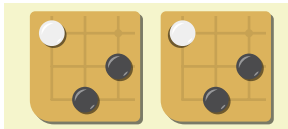
\pgfgoreset{players = {white = {stone = {shadow = false}}}}

\begin{goban}
  \stones[b]{B1,C2}
  \stone[w]{A3}
\end{goban}
```

/pgfgo/stone/shadow/angle = $\langle float \rangle$

(no default, initially 0)

Where the shadow falls, in degrees: with 0, to the right; with -45, down to the right. It does not turn the shape of the shadow.



```
\pgfgoreset{profile = pgfgosmooth}
\pgfgoreset{goban = {partial = A1:C3}}

\begin{goban}
  \stones[b]{B1,C2}
  \stone[w]{A3}
\end{goban}

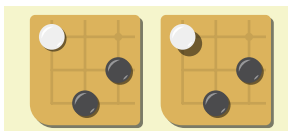
\pgfgoreset{players = {white = {stone = {shadow = {angle = 225}}}}}}

\begin{goban}
  \stones[b]{B1,C2}
  \stone[w]{A3}
\end{goban}
```

/pgfgo/stone/shadow/offset = $\langle dimension \rangle$

(no default, initially 1pt)

How far the shadow is moved from the center of the stone, in the direction **angle**.



```
\pgfgoreset{profile = pgfgosmooth}
\pgfgoreset{goban = {partial = A1:C3}}

\begin{goban}
  \stones[b]{B1,C2}
  \stone[w]{A3}
\end{goban}

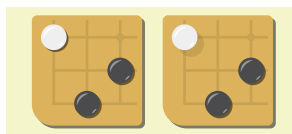
\pgfgoreset{players = {white = {stone = {shadow = {offset = 0.125cm}}}}}}

\begin{goban}
  \stones[b]{B1,C2}
  \stone[w]{A3}
\end{goban}
```

`/pgfgo/stone/shadow/opacity = <float>`

(no default, initially 0.5)

Opacity of the shadow (also that of the shadow with an image).



```
\pgfgo\set{profile = pgfgosmooth}
\pgfgo\set{goban = {partial = A1:C3}}

\begin{goban}
  \stones[b]{B1,C2}
  \stone[w]{A3}
\end{goban}

\pgfgo\set{players = {white = {stone = {
  shadow = {offset = 0.125cm, opacity = 0.1}}}}}

\begin{goban}
  \stones[b]{B1,C2}
  \stone[w]{A3}
\end{goban}
```

`/pgfgo/stone/shadow/color = <color>`

(no default, initially black)

The color of the shadow. A shadow with an image has its own colors, except with `image/silhouette`.



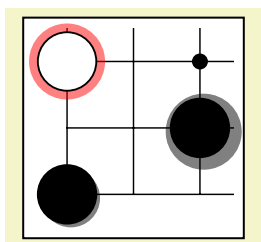
```
\pgfgo\set{profile = pgfgosmooth}
\pgfgo\set{goban = {partial = A1:C3}}
\pgfgo\set{players = {white = {stone = {
  shadow = {color = blue, offset = 0.1cm}}}}}

\begin{goban}
  \stones[b]{B1,C2}
  \stone[w]{A3}
\end{goban}
```

`/pgfgo/stone/shadow/scale = <float>`

(no default, initially 1)

Enlarges (or reduces) the shadow with respect to the stone. A shadow larger than the stone makes it look taller. It also enlarges the shadow with an image.



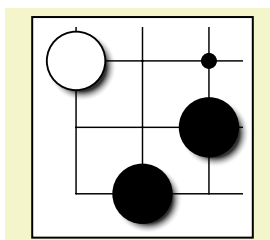
```
\pgfgo\set{goban = {partial = A1:C3, scale = 2}}

\begin{goban}
  \stone[b, shadow = {true, angle = -45}]{A1}
  \stone[b, shadow = {true, angle = -45,
    scale = 1.3}]{C2}
  \stone[w, shadow = {true, offset = 0pt,
    scale = 1.3, color = red}]{A3}
\end{goban}
```

`/pgfgo/stone/shadow/image = <options>`

(no default)

A shadow of your own: an image with transparency (PNG or PDF), centered on the stone and drawn as it is instead of the silhouette, for example a soft shadow. It is moved and enlarged like the silhouette (`angle`, `offset`, `scale`) and uses `opacity`, but not `color`. It does not turn with the image of the stone (`stone/image/rotate`), so it always falls to the same side. Each file is kept only once in the PDF.



```
\pgfgo\set{goban = {partial = A1:C3, scale = 2}}
\pgfgo\set{stone = {shadow = {true, angle = -45,
  offset = 1.5pt, opacity = 0.8,
  image = {file = pgfgo-shadow.png, width = 13pt}}}}

\begin{goban}
  \stones[b]{B1,C2}
  \stone[w]{A3}
\end{goban}
```

`/pgfgo/stone/shadow/image/file =  $\langle file \rangle$` (no default)

The file of the image. It is looked for as with `\includegraphics`.

`/pgfgo/stone/shadow/image/width =  $\langle dimension \rangle$` (no default, initially the diameter of the stone)

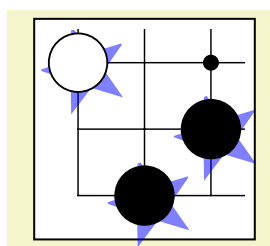
The width of the image; the height keeps the proportion.

`/pgfgo/stone/shadow/image/rotate =  $\langle angle \rangle$` (no default, initially 0)

Turns the image of the shadow, the same on every stone.

`/pgfgo/stone/shadow/image/silhouette =  $\langle boolean \rangle$` (default `true`, initially `false`)

Uses only the shape of the image (its transparency), painted with `shadow/color`, instead of the image as it is. It needs pdfL^AT_EX or LuaL^AT_EX with PDF output; with other engines the shadow is not drawn and a warning appears.



```
\pgfgoreset{goban = {partial = A1:C3, scale = 2}}
\pgfgoreset{stone = {shadow = {true, angle = -45,
offset = 2pt, color = blue,
image = {file = pgfgo-star.pdf, width = 16pt,
rotate = 18, silhouette}}}}

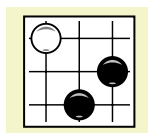
\begin{goban}
\stones[b]{B1,C2}
\stone[w]{A3}
\end{goban}
```

`/pgfgo/stone/shadow/image/false` (style, no value)

Removes the image: the silhouette of the stone comes back.

`/pgfgo/stone/shine =  $\langle options \rangle$` (default `true`)

Shine of the stone, drawn on top of it, be it a circle or an image.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

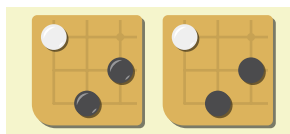
\begin{goban}
\stones[b, shine]{C2,D3}
\stone[w, shine = {true, color = gray}]{B4}
\end{goban}
```

`/pgfgo/stone/shine/true` (no value)

Draws the shine.

`/pgfgo/stone/shine/false` (no value)

Does not draw the shine.



```
\pgfgoreset{profile = pgfgosmooth}
\pgfgoreset{goban = {partial = A1:C3}}

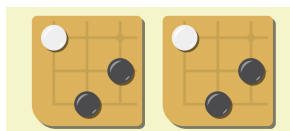
\begin{goban}
\stones[b]{B1,C2}
\stone[w]{A3}
\end{goban}

\pgfgoreset{players = {black = {stone = {shine = false}}}}

\begin{goban}
\stones[b]{B1,C2}
\stone[w]{A3}
\end{goban}
```

/pgfgo/stone/shine/angle = $\langle float \rangle$ (no default, initially 0)

Turns the drawn shine. With 0, the light comes from the top left.



```
\pgfgo\set{profile = pgfgosmooth}
\pgfgo\set{goban = {partial = A1:C3}}

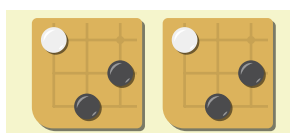
\begin{goban}
  \stones[b]{B1,C2}
  \stone[w]{A3}
\end{goban}

\pgfgo\set{players = {black = {stone = {shine = {angle = 45}}}}}

\begin{goban}
  \stones[b]{B1,C2}
  \stone[w]{A3}
\end{goban}
```

/pgfgo/stone/shine/opacity = $\langle float \rangle$ (no default, initially 1)

Opacity of the shine (also that of the shine with an image).



```
\pgfgo\set{profile = pgfgosmooth}
\pgfgo\set{goban = {partial = A1:C3}}

\begin{goban}
  \stones[b]{B1,C2}
  \stone[w]{A3}
\end{goban}

\pgfgo\set{players = {black = {stone = {shine = {opacity = 0.5}}}}}

\begin{goban}
  \stones[b]{B1,C2}
  \stone[w]{A3}
\end{goban}
```

/pgfgo/stone/shine/color = $\langle color \rangle$ (no default, initially gray!50!white)

Color of the drawn shine.

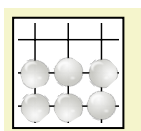


```
\pgfgo\set{profile = pgfgosmooth}
\pgfgo\set{goban = {partial = A1:C3}}
\pgfgo\set{players = {black = {stone = {shine = {color = red}}}}}

\begin{goban}
  \stones[b]{B1,C2}
  \stone[w]{A3}
\end{goban}
```

/pgfgo/stone/shine/image = $\langle options \rangle$ (no default)

A shine of your own: an image with transparency (PNG or PDF), centered on the stone, instead of the drawn shine. If the stone is an image too, the shine is drawn on top and does not turn with it (**stone/image/rotate**): the stones can have stripes turned at random and the light always on the same side. Each file is kept only once in the PDF.



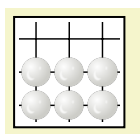
```
\begin{goban}[5, partial = B2:D4]
  \stones[w, image = {file = pgfgo-yunzi-white-1.png, rotate = random},
    shine = {true, image = {file = pgfgo-shine.png}}]{B2, C2, D2, B3, C3, D3}
\end{goban}
```

/pgfgo/stone/shine/image/file = $\langle file \rangle$ (no default)

The file of the image. It is looked for as with `\includegraphics`.

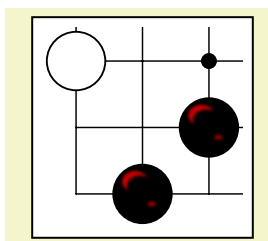
`/pgfgo/stone/shine/image/width =  $\langle dimension \rangle$` (no default, initially the diameter of the stone)
 The width of the image; the height keeps the proportion.

`/pgfgo/stone/shine/image/rotate =  $\langle angle \rangle$` (no default, initially 0)
 Turns the image of the shine, the same on every stone.



```
\begin{goban}[5, partial = B2:D4]
\stones[w, image = {file = pgfgo-yunzi-white-1.png},
shine = {true, image = {file = pgfgo-shine.png, width = 0.3cm,
rotate = 90}}]{B2:D3}
\end{goban}
```

`/pgfgo/stone/shine/image/silhouette =  $\langle boolean \rangle$` (default true, initially false)
 Uses only the shape of the image (its transparency), painted with `shine/color`, instead of the image as it is. It needs pdfL^AT_EX or LuaL^AT_EX with PDF output; with other engines the shine is not drawn and a warning appears.



```
\pgfgoset{goban = {partial = A1:C3, scale = 2}}
\pgfgoset{players = {black = {stone = {shine = {true,
color = red, image = {file = pgfgo-shine.png, silhouette}}}}}}
\begin{goban}
\stones[b]{B1,C2}
\stone[w]{A3}
\end{goban}
```

`/pgfgo/stone/shine/image/false` (style, no value)
 Removes the image: the drawn shine comes back.

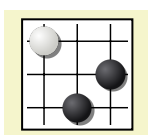
`/pgfgo/stone/image =  $\langle options \rangle$` (no default)
 Draws the stone with an image (PNG, JPG or PDF) instead of the circle: neither the fill nor the border is drawn, and the image need not be round. If the shine (**shine**) is on, it is drawn on top of the image. The shadow (**shadow**) is drawn with the shape of the image; for that the image has to have transparency (a PNG with an alpha channel or a PDF): the shadow of a JPG is a rectangle. A prisoner with an image looks translucent, like any prisoner.

The files are looked for as with `\includegraphics` (for example, in the folders of `\graphicspath`) and, if they are not there, among the images that come with PGFGO, next to `pgf-go.sty` (those of the profiles `shell`, `yunzi` and `baduktv`, p. 145). Each file, at each size, is kept only once in the PDF, however many stones use it (p. 73). The shadow with the shape of the image needs pdfL^AT_EX or LuaL^AT_EX (with PDF output); with other engines the stone is drawn without a shadow and with a warning.

A file name with a comma goes in double braces, `file = {{a,b.png}}`; with single braces, the comma separates two files. As with `\includegraphics`, the name cannot have `&`, `#`, `%`, `$`, `{` or `}`, nor two spaces in a row.

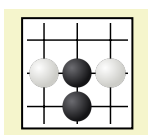
SVG images cannot be used directly: they have to be converted to PDF first, for example with Inkscape or with `rsvg-convert`. The same holds for the background, the grid, the star points and the marks.

```
inkscape --export-type=pdf stone.svg
rsvg-convert -f pdf -o stone.pdf stone.svg
```



```
\pgfgoset{goban = {5, partial = B2:D4}}
\begin{goban}
\stones[b, image = {file = pgfgo-yunzi-black.png},
shadow = {true, angle = -45, offset = 1.5pt}]{C2,D3}
\stone[w, image = {file = pgfgo-yunzi-white-1.png},
shadow = {true, angle = -45, offset = 1.5pt}]{B4}
\end{goban}
```

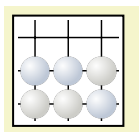
For all the stones of a player, the image is given in its options:



```
\pgfgoreset{players = {
  black = {stone = {image = {file = pgfgo-yunzi-black.png}}},
  white = {stone = {image = {file = pgfgo-yunzi-white-1.png}}}}
\begin{goban}[5, partial = B2:D4]
\moves[numbers = false]{C3, D3, C2, B3}
\end{goban}
```

/pgfgo/stone/image/file = $\langle file, file, \dots \rangle$ (no default)

The file of the image. With several files, each stone uses one, chosen by its position: a stone on a point always shows the same image, in every diagram.



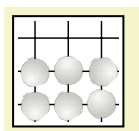
```
\begin{goban}[5, partial = B2:D4]
\stones[w, image = {file = {pgfgo-yunzi-white-1.png, pgfgo-yunzi-white-
2.png}}]{B2, C2, D2, B3, C3, D3}
\end{goban}
```

/pgfgo/stone/image/width = $\langle dimension \rangle$ (no default, initially the diameter of the stone)

The width of the image; the height keeps the proportion. By default it is the diameter of the stone: $2 \times \text{radius} \times \text{scale}$.

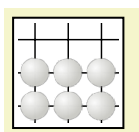
/pgfgo/stone/image/rotate = $\langle angle \rangle$, **random** or $\langle minimum \rangle : \langle maximum \rangle$ (no default, initially 0)

Turns the image $\langle angle \rangle$ degrees, counterclockwise. With **random**, each stone is turned by a different angle, chosen by its position: a stone on a point always looks the same, in every diagram and in every frame of an animation. The shadow turns with the image, but still falls to the same side.



```
\begin{goban}[5, partial = B2:D4]
\stones[w, image = {file = pgfgo-yunzi-white-1.png,
  rotate = random}]{B2, C2, D2, B3, C3, D3}
\end{goban}
```

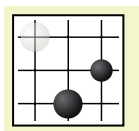
With a range $\langle minimum \rangle : \langle maximum \rangle$, the angle is chosen as with **random**, but between $\langle minimum \rangle$ and $\langle maximum \rangle$ degrees (**random** is the same as 0:360).



```
\begin{goban}[5, partial = B2:D4]
\stones[w, image = {file = pgfgo-yunzi-white-1.png,
  rotate = -20:20}]{B2, C2, D2, B3, C3, D3}
\end{goban}
```

/pgfgo/stone/image/opacity = $\langle float \rangle$ (no default, initially 1)

The opacity of the image.



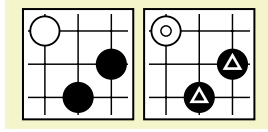
```
\begin{goban}[5, partial = B2:D4]
\stones[b, image = {file = pgfgo-yunzi-black.png}]{C2}
\stones[b, image = {file = pgfgo-yunzi-black.png, width = 8pt}]{D3}
\stone[w, image = {file = pgfgo-yunzi-white-1.png, opacity = 0.5}]{B4}
\end{goban}
```

/pgfgo/stone/image/false (style, no value)

Removes the image: the stone is a circle again.

/pgfgo/stone/mark = $\langle mark\ keys \rangle$ (no default)

A mark on the stone, with the options of `\mark` (p. 78). For example, `mark = triangle` or `mark = {circle, line = {color = red}}`.



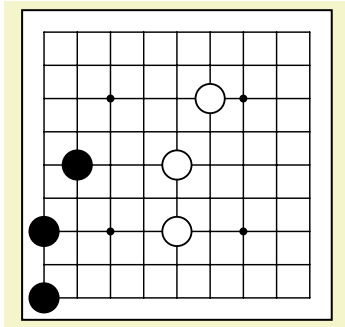
```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b, mark = triangle]{C2,D3}
\stone[w, mark = t-circle]{B4}
\end{goban}
```

`/pgfgo/stone/shift` = $\langle columns \rangle, \langle rows \rangle$ (no default, initially 0, 0)

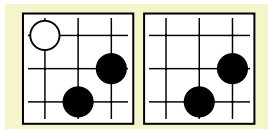
Moves the coordinates of the command: `shift = {2, -1}` is the same as writing them two columns to the right and one row lower. With a single number, it moves only the columns. It is always to the right and up, even when the labels of the border go the other way. The stones are in their new place for everything (captures, territory, `resume`). Points that end up off the board are not drawn, with a warning. It applies only to that command. It also works in `\stone`.



```
\def\corner{A1--A3, B5; A2}
\begin{goban}[9]
\stones[b]{\corner}
\stones[w, shift = {4, 2}]{\corner}
\end{goban}
```

`/pgfgo/stone/forget` (no value)

`resume` does not bring this stone (p. 101). It stays in the memory of the named goban: `\pgfgoview` shows it with `forgotten`.



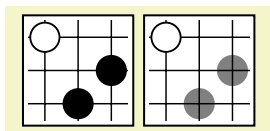
```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}[name = a]
\stones[b]{C2,D3}
\stone[w, forget]{B4}
\end{goban}

\begin{goban}[resume = a]
\end{goban}
```

`/pgfgo/stone/prisoner` = $\langle prisoner options \rangle$ (default fill opacity = 0.5)

The stone is a prisoner: by default, with the fill at half opacity. The options added are applied after the style of the prisoners (`/pgfgo/prisoner` and that of the player).

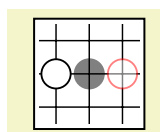


```
\pgfgoset{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b, prisoner]{C2,D3}
\stone[white]{B4}
\end{goban}
```

In the options of a player (`\pgfgoset{players = {white = {stone = {prisoner = ...}}}}`), it is the style of its prisoners: those of `\prisoner`, `\prisoners` and the *key* prisoner of `\stone` and `\stones`, and the stones captured with `captures = fade`. It is applied after the fill opacity 0.5 and `/pgfgo/prisoner`, and before the options of the stone. The other stones of the player do not change.



```
\pgfgoset{goban = {5, partial = B2:D4}}
\pgfgoset{players = {white = {stone = {prisoner = {line = {color = red}}}}}}

\begin{goban}
\stones[w]{B3}
\prisoner[b]{C3}
\prisoner[w]{D3}
\end{goban}
```

6.1.1 Images and the size of the PDF

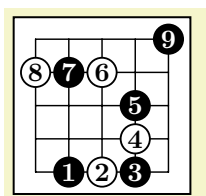
With pdfL^AT_EX or LuaL^AT_EX (with PDF output), each image (of the stones, the background, the grid, the hoshi, the marks, the shine or the shadow) is kept in the PDF only once, the first time it is used; afterwards, in the whole document, each stone or tile that uses it only names it. A game drawn with `profile = shell` keeps the wood, the six stones and the shadow once, even if it has hundreds of stones spread over many diagrams. Turning an image (`rotate`) does not add another copy, and the shadow with the shape of the image (also with `silhouette`) uses the same copy. What does add a copy is another size: the same image with another `width`, or on boards with another `scale` (for example, with `fit`), is kept once for each size. With other engines (XeL^AT_EX, or DVI output), the image is included each time it is used, and the PDF grows with each stone.

Drawn stones, without an image, are not kept that way: each one is a circle and a few drawing instructions, almost the same as naming a kept copy. A more elaborate vector stone (gradients, drawn stripes) can be made in a PDF and used with `stone/image`: it looks sharp at any scale and is kept only once, like the other images.

6.2 Moves

`\moves[options]{intersection , intersection , ...}`

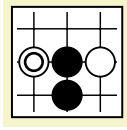
Plays the intersections in the order in which they are written, alternating Black and White, and numbers them (**numbers**). An intersection can be repeated (for example, to take back a ko with `captures`). It accepts the syntax of `\stones` (p. 32) and plays each element in its order (p. 38): a path (A1 -- A3 -- C3) goes from its first point to the last, and a segment, from the first end to the second (C:A 4 goes from C4 to A4; C5:1, from C5 to C1). What has no fixed path (rectangles, `r`, whole columns and rows, `*`) is played all the same, with a warning. Subtractive groups (`:`) give an error: `\moves` only adds moves.



```
\begin{goban}[5]
\moves{B1 -| D3, C:A 4, E5}
\end{goban}
```

Inside one `goban`, the numbering and the turn go on from one `\moves` to the next. Each new `goban` starts at move 1 with Black; to continue a game in another diagram, use `from` and `b` or `w`, or `resume`.

The options of `\moves` also accept those of the style of the stone (`fill`, `line`, `radius`, `shine`, `image`, etc.) and `mark`, like `\stones`. For example, to mark the last move:



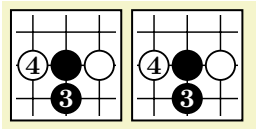
```
\begin{goban}[5, partial = B2:D4]
\moves[numbers = false]{C3, D3, C2}
\moves[numbers = false, mark = circle]{B3}
\end{goban}
```

The options of every player can be changed with `\pgfgoset{move = {...}}`, which is equivalent to `\pgfgoset{players = {all = {move = {...}}}}`, except `numbers`, which does not belong to a player: it holds for every move.

`/pgfgo/move/numbers=<boolean>`

(default `true`, initially `true`)

Whether the moves carry their number. Without numbers, the moves still count: the next move goes on with the numbering and the turn. As an option of `\moves` it holds for those moves; with `\pgfgoset{move = {numbers = false}}` or `\begin{goban}[move = {numbers = false}]`, for all the ones that follow.



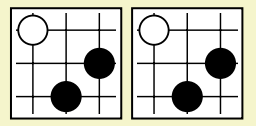
```
\begin{goban}[5, partial = B2:D4]
\moves[numbers = false]{C3, D3}
\moves{C2, B3}
\end{goban}

\begin{goban}[5, partial = B2:D4, move = {numbers = false}]
\moves{C3, D3}
\moves[numbers]{C2, B3}
\end{goban}
```

`/pgfgo/move/black`

(no value)

The sequence starts with Black.



```
\pgfgoset{goban = {5, partial = B2:D4}}

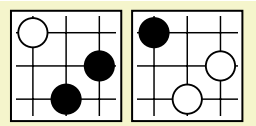
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\moves[numbers = false, b]{C2,B4}
\moves[numbers = false]{D3}
\end{goban}
```

`/pgfgo/move/white`

(no value)

The sequence starts with White.



```
\pgfgoset{goban = {5, partial = B2:D4}}

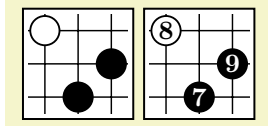
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\moves[numbers = false, white]{C2,B4}
\moves[numbers = false]{D3}
\end{goban}
```

`/pgfgo/move/b` (no value)
Short for black.

`/pgfgo/move/w` (no value)
Short for white.

`/pgfgo/move/from` = $\langle number \rangle$ (no default)
Number of the first move of the sequence.

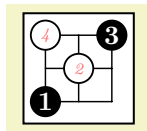


```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\moves[from = 7]{C2,B4}
\moves{D3}
\end{goban}
```

`/pgfgo/move/label` = $\langle options \rangle$ (no default)
Options of the numbers of the moves.

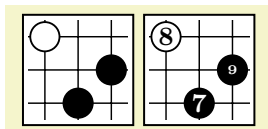


```
\pgfgoreset{players = {white = {move = {label = {
color = red, font = \itshape, font size = 6pt, opacity = 0.6}}}}}

\begin{goban}[3]
\moves{A1, B2, C3, A3}
\end{goban}
```

`/pgfgo/move/label/font` = $\langle font commands \rangle$ (no default, initially `\bfseries`)
Font of the numbers.

`/pgfgo/move/label/font size` = $\langle dimension \rangle$ (no default, initially automatic)
Size of the numbers. If it is not given, it adjusts by itself according to the number of digits: 10pt with one, 6.5pt with two and 4.5pt with more.



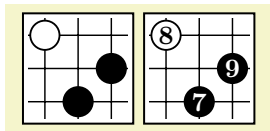
```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\moves[from = 7]{C2,B4}
\moves[label = {font size = 0.5em}]{D3}
\end{goban}
```

`/pgfgo/move/label/format` = $\langle options \rangle$ (no default, initially `arabic`)
Format of the numbers: `arabic`, `alph`, `Alph`, `roman`, `Roman` or `kanji`.

`/pgfgo/move/label/format/arabic` (no value)
Arabic numbers. It is the initial value.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

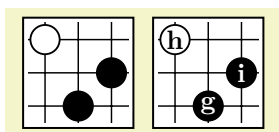
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\moves[from = 7]{C2,B4}
\moves{D3}
\end{goban}
```

`/pgfgo/move/label/format/alph`

(no value)

Lowercase letters.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

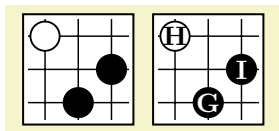
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\pgfgoreset{move={label = {format=alph}}}}
\begin{goban}
\moves[from = 7]{C2,B4}
\moves{D3}
\end{goban}
```

`/pgfgo/move/label/format/Alph`

(no value)

Uppercase letters.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

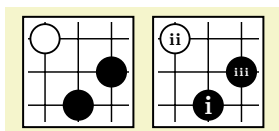
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\pgfgoreset{move = {label = {format = Alph}}}}
\begin{goban}
\moves[from = 7]{C2,B4}
\moves{D3}
\end{goban}
```

`/pgfgo/move/label/format/roman`

(no value)

Lowercase Roman numbers.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

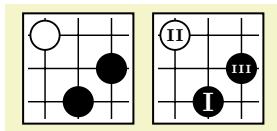
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\pgfgoreset{move = {label = {format = roman}}}}
\begin{goban}
\moves{C2,B4}
\moves{D3}
\end{goban}
```

`/pgfgo/move/label/format/Roman`

(no value)

Uppercase Roman numbers.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\pgfgoreset{move = {label = {format = Roman}}}

\begin{goban}
\moves{C2,B4}
\moves{D3}
\end{goban}
```

/pgfgo/move/label/format/kanji

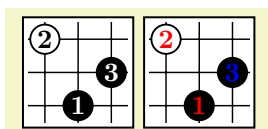
(no value)

Numbers in kanji (一, 二, …, 十, 十一, …, 百), like those of `goban/label/columns = kanji`. They need a font that has them, chosen with `label/font` (see `/pgfgo/mark/label/format`, p. 86).

/pgfgo/move/label/color = *⟨color⟩*

(no default)

Color of the numbers.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}
\moves{C2,B4}
\moves{D3}
\end{goban}

\begin{goban}
\moves[label = {color = red}]{C2,B4}
\moves[label = {color = blue}]{D3}
\end{goban}
```

/pgfgo/move/label/opacity = *⟨float⟩*

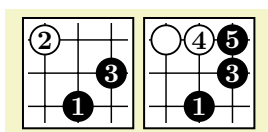
(no default, initially 1)

Opacity of the numbers.

/pgfgo/move/label/forget

(no value)

With `resume = {..., numbers}`, this move comes without its number (p. 101). With `\pgfgoview`, the number is only seen with `forgotten`.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}[name = a]
\moves{C2}
\moves[label = forget]{B4}
\moves{D3}
\end{goban}

\begin{goban}[resume = {a, numbers}]
\moves{C:D4}
\end{goban}
```

/pgfgo/move/shift = *⟨columns⟩, ⟨rows⟩*

(no default, initially 0, 0)

Moves the moves, like `stone/shift`.

/pgfgo/move/forget

(no value)

`resume` does not bring these moves, neither the stone nor the number. The numbering goes on from the last move it brings.

6.3 Prisoners

Prisoners are captured stones that remain on the board; by default they are drawn translucent.

`\prisoner[<options>]{<intersection>}`

Draws a prisoner on the intersection. It accepts the options of `\stone`.

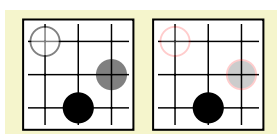
`\prisoners[<options>]{<intersection , intersection, ...>}`

Like `\prisoner`, with several intersections.

`/pgfgo/prisoner=<stone options>`

(default fill opacity = 0.5)

Style of the prisoners: by default, the fill with opacity 0.5. It accepts the options of `\stone`. It holds for the prisoners of the three players; the style of each one (`stone/prisoner` in the options of the player) comes after.



```
\pgfgo\set{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2}
\prisoners[b]{D3}
\prisoner[w]{B4}
\end{goban}

\pgfgo\set{prisoner = {fill = {opacity = 0.2}, line = {color = red}}}

\begin{goban}
\stones[b]{C2}
\prisoners[b]{D3}
\prisoner[w]{B4}
\end{goban}
```

7 Marks, arrows and lines

Marks go on an intersection, with or without a stone: shapes (circle, triangle, cross, ...), numbers in sequence or any text.

`\mark[<options>]{<intersection>}`

Draws a mark on the intersection. The options choose its shape and its style.

`\marks[<options>]{<intersection , intersection, ...>}`

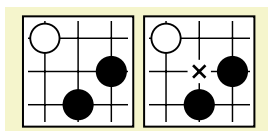
Like `\mark`, with several intersections (p. 32).

`\mark` and `\marks` are TeX primitives. So as not to override them, the `\mark` and `\marks` macros of PGFGO only exist inside the `goban` environment. Outside it they are still the primitives, so page marks and headers are not affected.

`/pgfgo/mark =<options>`

(default cross)

The shape of the mark. Without a shape, it is a cross (`cross`).



```
\pgfgo\set{goban = {5, partial = B2:D4}}

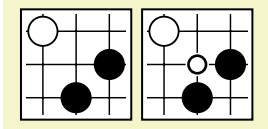
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark{C3}
\end{goban}
```

`/pgfgo/mark/circle`

(no value)

A circle.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

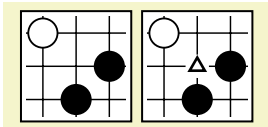
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[circle]{C3}
\end{goban}
```

/pgfgo/mark/triangle

(no value)

A triangle.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

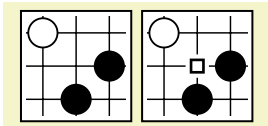
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[triangle]{C3}
\end{goban}
```

/pgfgo/mark/square

(no value)

A square.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

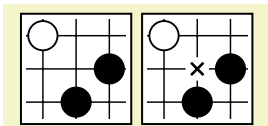
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[square]{C3}
\end{goban}
```

/pgfgo/mark/cross

(no value)

A cross.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

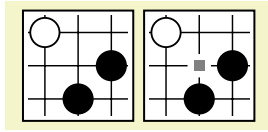
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[cross]{C3}
\end{goban}
```

/pgfgo/mark/filled square

(no value)

A filled square.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

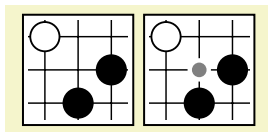
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[filled square]{C3}
\end{goban}
```

`/pgfgo/mark/filled circle`

(no value)

A filled circle.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

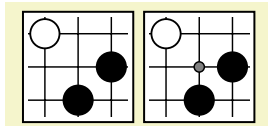
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[filled circle]{C3}
\end{goban}
```

`/pgfgo/mark/t-circle`

(no value)

A small circle, with the fill of the stones of the player and the border of the grid. It is the mark of Black's territory.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

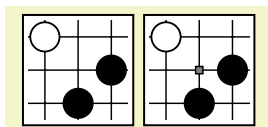
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[t-circle]{C3}
\end{goban}
```

`/pgfgo/mark/t-square`

(no value)

Like t-circle, but square. It is the mark of White's territory and of the neutral points.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

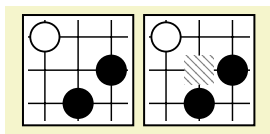
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[t-square]{C3}
\end{goban}
```

/pgfgo/mark/hatched lines

(no value)

A hatching of crossed lines.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

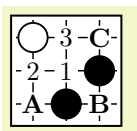
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[hatched lines]{C3}
\end{goban}
```

/pgfgo/mark/sequence=<options>

(default from = 1)

Numbers a series of marks in order. The font, the size and the color of the numbers are chosen with `label` (`label = {font = \bfseries, color = red}`), as the text of custom. The marks are numbered in the order of the coordinates (p. 38); if they have no fixed order, like a rectangle, `sequence` warns.



```
\begin{goban}[5, partial = B2:D4]
\stones[b]{C2,D3}
\stone[w]{B4}
\marks[sequence]{C3,B3,C4}
\marks[sequence = {format = Alph}, label = {font = \bfseries}]{B2,D2,D4}
\end{goban}
```

/pgfgo/mark/sequence/true

(no value)

Turns the sequence on. It is the same as `sequence` alone.

/pgfgo/mark/sequence/false

(no value)

Turns the sequence off (for example, one turned on with `\pgfgoreset`).

/pgfgo/mark/sequence/format = <format>

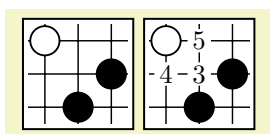
(no default)

Turns the sequence on and chooses the format of the numbers. It is the same as `sequence` with `label = {format = ...}` (see `label/format`).

/pgfgo/mark/sequence/from=<number>

(no default)

Number of the first mark.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

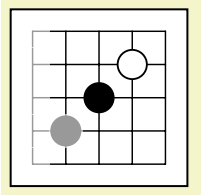
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\marks[sequence = {from = 3}]{C3,B3,C4}
\end{goban}
```

/pgfgo/mark/dim

(no value)

Dims the point: covers it with the color of the background of the board, translucent, over the stone and the grid. It is what `\sgf` draws for the property `DD`.

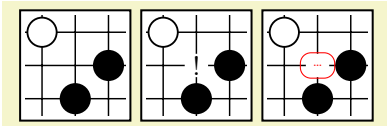


```
\begin{goban}[5]
\stones[b]{B2, C3}\stone[w]{D4}
\marks[dim]{A1:5, B2}
\end{goban}
```

`/pgfgo/mark/custom=<token list>`

(no default)

A mark with any content: a text, a symbol (“!”, “?”) or even a drawing.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[custom = {!}]{C3}
\end{goban}

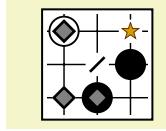
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[custom = {\tikz{\node[rounded corners, draw, inner sep = 1em, red]{\dots}};}] {C3}
\end{goban}
```

`\pgfdefinemark[<options>]{<name>}{<pgf code>}`

Defines a new mark, called `<name>`, which is used like the others: in `\marks`, in the option `mark` of the stones and of the players, and in `\territory`. It is drawn in the same layers as the other marks; on an empty point, it covers the grid with a circle of the color of the background. A mark with the name of another one replaces it.

The `<pgf code>` is drawn with the origin at the point. Coordinates without a unit (those of `\pgfpointxy` and those of TikZ) are measured in stone radii: (0,0) `circle` (1) has the size of the stone. The color of the lines is `mark/line/color` of the player; the fill color, `mark/fill/color`; and the width of the lines, `mark/line/width`. Those two colors are also called `pgfgomarkline` and `pgfgomarkfill`. With `mark/scale`, the mark changes size like the others.

With `\pgfdefinemark[image = {file = <file list>, width = <dimension>}]{<name>}`, without `<pgf code>`, the mark is an image (or one of several): see `image` below.



```
\pgfdefinemark{diamond}{
\pgfpathmoveto{\pgfpointxy{0}{0.7}}
\pgfpathlineto{\pgfpointxy{0.7}{0}}
\pgfpathlineto{\pgfpointxy{0}{-0.7}}
\pgfpathlineto{\pgfpointxy{-0.7}{0}}
\pgfpathclose
\pgfusepath{fill, stroke}}
\pgfdefinemark{bar}{\draw (-0.5,-0.5) -- (0.5,0.5);}
\pgfdefinemark[image = {file = pgfgo-star.pdf}]{star}

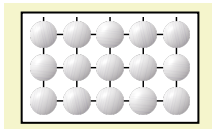
\begin{goban}[5, partial = B2:D4]
\stones[b]{C2,D3}
\stone[w]{B4}
\marks[diamond]{C2, B4, B2}
\marks[bar]{D3, C3}
\marks[star]{D4}
\end{goban}
```

/pgfgo/define mark/image = *<options>* (no default)

The image (PNG, JPG or PDF) of a mark defined with `\pgfdefinemark`. It is drawn centered on the point; the colors of the marks do not apply.

/pgfgo/define mark/image/file = *<file list>* (no default)

The file of the image, or several separated by commas. With several, each point uses one of them, chosen at random as in `stone/image/file`: the same point always gets the same file.

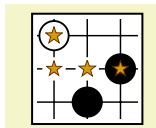


```
\pgfdefinemark[image = {file = {pgfgo-shell-white-1.png, pgfgo-shell-white-
3.png, pgfgo-shell-white-5.png}, width = 1em}]{shells}

\begin{goban}[5, partial = A1:E3]
\marks[shells]{A:E 1:3}
\end{goban}
```

/pgfgo/define mark/image/width = *<dimension>* (no default, initially 1.2 stone radii)

The width of the image; the height keeps the proportion. Without this *key*, the mark has the size of the circle (`circle`), and changes with the radius of the stones.

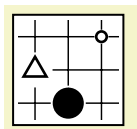


```
\pgfdefinemark[image = {file = pgfgo-star.pdf}]{star}
\pgfdefinemark[image = {file = pgfgo-star.pdf, width = 0.6em}]{small star}

\begin{goban}[5, partial = B2:D4]
\stones[b]{C2,D3}
\stone[w]{B4}
\marks[star]{C3, B4}
\marks[small star]{B3, D3}
\end{goban}
```

/pgfgo/mark/scale = *<float>* (no default, initially 1)

The size of the mark with respect to the usual one: the shapes, the images and the text.

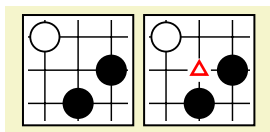


```
\begin{goban}[5, partial = B2:D4]
\stones[b]{C2}
\marks[triangle, scale = 1.5]{C2, B3}
\marks[circle, scale = 0.6]{D4}
\end{goban}
```

/pgfgo/mark/line = *<options>* (no default)

Options of the stroke of the mark.

`/pgfgo/mark/line/color =  $\langle color \rangle$` (no default)
 Color of the stroke.

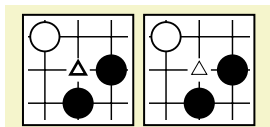


```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[triangle, line = {color = red}]{C3}
\end{goban}
```

`/pgfgo/mark/line/width =  $\langle dimension \rangle$` (no default)
 Width of the stroke.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[triangle]{C3}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[triangle, line = {width = 0.1pt}]{C3}
\end{goban}
```

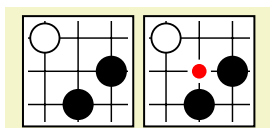
`/pgfgo/mark/line/opacity =  $\langle float \rangle$` (no default, initially 1)
 Opacity of the stroke.

`/pgfgo/mark/fill =  $\langle options \rangle$` (default true)
 Fill of the filled marks (filled square, filled circle and hatched lines) and color `pgfgomarkfill` of your own marks (p. 82).

`/pgfgo/mark/fill/true` (no value)
 The filled marks are filled with `fill/color`. It is the initial value.

`/pgfgo/mark/fill/false` (no value)
 The filled marks are not filled, so they are not seen.

`/pgfgo/mark/fill/color =  $\langle color \rangle$` (no default)
 Color of the fill (initially, gray).



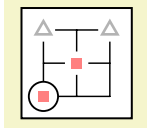
```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[filled circle, fill = {true, color = red}]{C3}
\end{goban}
```

`/pgfgo/mark/fill/opacity =  $\langle float \rangle$` (no default, initially 1)

Opacity of the fill.

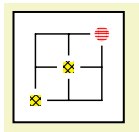


```
\pgfgo\set{players = {white = {mark = {
fill = {color = red, opacity = 0.5},
line = {opacity = 0.3}}}}}

\begin{goban}[3]
\stones[w]{A1}
\marks[w, filled square]{A1, B2}
\marks[w, triangle]{C3, A3}
\end{goban}
```

`/pgfgo/mark/fill/pattern =  $\langle options \rangle$` (no default)

A pattern over the fill color of filled square and filled circle, as stone/fill/pattern (p. 62). hatched lines and your own marks use only the color.



```
\begin{goban}[3]
\marks[filled square, fill = {color = yellow,
pattern = {name = crosshatch}}]{A1, B2}
\marks[filled circle, fill = {color = none,
pattern = {name = {Lines[distance = 1pt]}}, color = red}]{C3}
\end{goban}
```

`/pgfgo/mark/fill/pattern/name =  $\langle pattern \rangle$` (no default)

The pattern: any of the TikZ libraries patterns and patterns.meta, which PGFGO loads. One with options goes in braces: name = {Lines[angle = 45, distance = 2pt]}. Without name there is no pattern.

`/pgfgo/mark/fill/pattern/color =  $\langle color \rangle$` (no default, initially black)

Color of the pattern.

`/pgfgo/mark/fill/pattern/false` (no value)

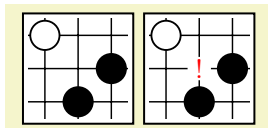
No pattern.

`/pgfgo/mark/label =  $\langle option \rangle$` (no default)

Options of the text of the marks that have one (sequence and custom).

`/pgfgo/mark/label/color =  $\langle color \rangle$` (no default)

Color of the text.



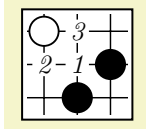
```
\pgfgo\set{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[custom = {!}, label = {color = red}]{C3}
\end{goban}
```

`/pgfgo/mark/label/font =  $\langle font commands \rangle$` (no default)

Font of the labels of the marks (sequence and custom).



```
\pgfset{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\marks[sequence, label = {font = \itshape}]{C3,B3,C4}
\end{goban}
```

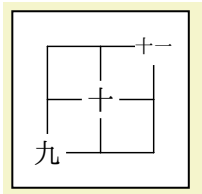
/pgfgo/mark/label/font size = $\langle dimension \rangle$ (no default)

Size of the text. If it is not set, it adjusts according to the number of characters, the same in **sequence** and in **custom**: 10pt with one, 6.5pt with two and 4.5pt with more.

/pgfgo/mark/label/format= $\langle options \rangle$ (no default, initially **arabic**)

Format of the numbers of **sequence**: **arabic**, **alph**, **Alph**, **roman**, **Roman** or **kanji**.

Kanji need a font that has them, chosen with **label/font** (of the marks or of the moves). It has to be a $\text{\LaTeX}$ family: with **fontspec**, **\newfontfamily\kanji{ipaexm.ttf}** and **font = \kanji**; without **fontspec**, with **Lua $\text{\LaTeX}$** , it can be declared as in this example (in the preamble). A $\text{\TeX}$ **\font** does not work, because it does not change size.



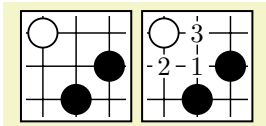
```
% In the preamble:
% \DeclareFontFamily{TU}{ipaex}{}
% \DeclareFontShape{TU}{ipaex}{m}{n}{<->file:ipaexm.ttf}{}
% \DeclareFontShape{TU}{ipaex}{bx}{n}{<->ssub*ipaex/m/n}{}

\begin{goban}[3, scale = 1.6]
\marks[sequence = {format = kanji, from = 9},
label = {font = \fontfamily{ipaex}\selectfont}]{A1, B2, C3}
\end{goban}
```

/pgfgo/mark/label/format/arabic

(no value)

Arabic numbers. It is the initial value.



```
\pgfset{goban = {5, partial = B2:D4}}

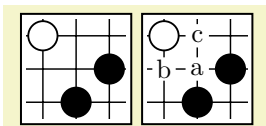
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\marks[label = {format = arabic}, sequence]{C3,B3,C4}
\end{goban}
```

/pgfgo/mark/label/format/alph

(no value)

Lowercase letters.



```
\pgfset{goban = {5, partial = B2:D4}}

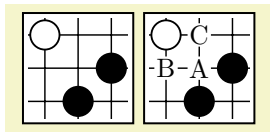
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\marks[label = {format = alph}, sequence]{C3,B3,C4}
\end{goban}
```

/pgfgo/mark/label/format/Alph

(no value)

Uppercase letters.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

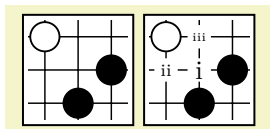
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\marks[label = {format = Alph}, sequence]{C3,B3,C4}
\end{goban}
```

/pgfgo/mark/label/format/roman

(no value)

Lowercase Roman numbers.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

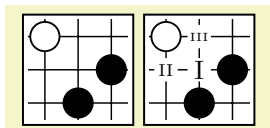
\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\marks[label = {format = roman}, sequence]{C3,B3,C4}
\end{goban}
```

/pgfgo/mark/label/format/Roman

(no value)

Uppercase Roman numbers.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\marks[label = {format = Roman}, sequence]{C3,B3,C4}
\end{goban}
```

/pgfgo/mark/label/format/kanji

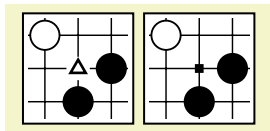
(no value)

Numbers in kanji (一, 二, …, 十, 十一, …, 百), like those of goban/label/columns = kanji.

/pgfgo/mark/black

(no value)

The mark belongs to the first player, with its style.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[triangle]{C3}
\end{goban}

\begin{goban}
\stones[b]{C2,D3}
\stone[w]{B4}
\mark[black, t-square]{C3}
\end{goban}
```

`/pgfgo/mark/white`

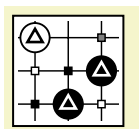
(no value)

The mark belongs to the second player, with its style.

`/pgfgo/mark/neutral`

(no value)

The mark belongs to the neutral player.



```
\begin{goban}[5, partial = B2:D4]
\stones[b]{C2,D3}
\stone[w]{B4}
\marks[b, t-square]{B2, C3}
\marks[w, t-square]{D2, B3}
\marks[n, t-square]{D4}
\marks[b, triangle]{C2, D3}
\marks[w, triangle]{B4}
\end{goban}
```

`/pgfgo/mark/b`

(no value)

Short for black.

`/pgfgo/mark/w`

(no value)

Short for white.

`/pgfgo/mark/n`

(no value)

Short for neutral.

`/pgfgo/mark/shift` = $\langle columns \rangle, \langle rows \rangle$

(no default, initially 0, 0)

Moves the marks, like `stone/shift`.

`/pgfgo/mark/forget`

(no value)

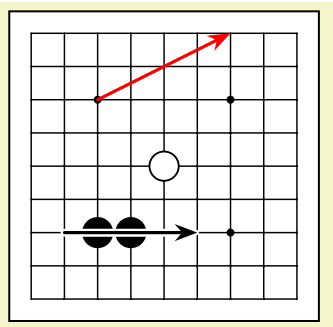
`resume` does not bring this mark, even if `marks` is asked for (p. 101).

7.1 Arrows and lines

Arrows and lines join two intersections. They are drawn on top of the stones, with a halo of the color of the background of the board that separates them from the stones and the grid. Like `\mark`, `\arrow` and `\line` only exist inside `goban`: outside, `\line` is still the L^AT_EX command. `\sgf` draws the arrows (AR) and the lines (LN) of the file.

`\arrow[ $\langle options \rangle$ ]{ $\langle from \rangle$ }{ $\langle to \rangle$ }`

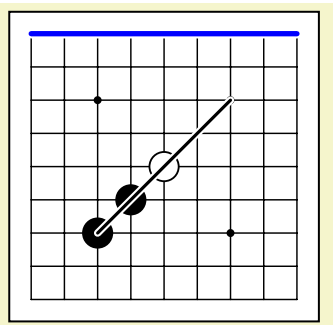
Draws an arrow from one intersection to another.



```
\begin{goban}[9]
\stones[b]{C3, D3}\stone[w]{E5}
\arrow{B3}{F3}
\arrow[color = red, halo = false]{C7}{G9}
\end{goban}
```

`\line[options]{from}{to}`

Draws a line from one intersection to another.

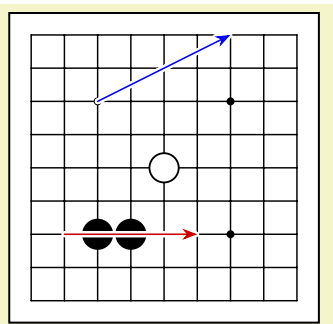


```
\begin{goban}[9]
\stones[b]{C3, D4}\stone[w]{E5}
\line{C3}{G7}
\line[color = blue, width = 2pt]{A9}{J9}
\end{goban}
```

`/pgfgo/goban/arrow=<arrow options>`

(no default)

The style of the arrows of the board, with the same keys as `\arrow` except `forget`. It is what a profile defines (p. 135): using a profile sets it back to the one of `default` before setting its own. The options of each `\arrow` go on top.



```

\begin{goban}[9, arrow = {color = red!80!black, width = 0.6pt}]
\stones[b]{C3, D3}\stone[w]{E5}
\arrow[B3]{F3}
\arrow[color = blue]{C7}{G9}
\end{goban}

```

/pgfgo/goban/line=*<line options>*

(no default)

The same for the lines.

/pgfgo/arrow/color=*<color>*

(no default, initially black)

Color of the arrow.

/pgfgo/arrow/width=*<dimension>*

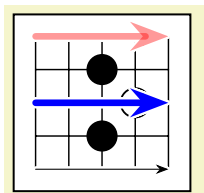
(no default, initially 1.2pt)

Width of the arrow. The tip grows with it.

/pgfgo/arrow/opacity=*<number>*

(no default, initially 1)

Opacity of the arrow.



```

\begin{goban}[5]
\stones[b]{C2, C4}\stone[w]{D3}
\arrow[width = 0.4pt]{A1}{E1}
\arrow[color = blue, width = 2.5pt]{A3}{E3}
\arrow[color = red, opacity = 0.4, width = 2.5pt]{A5}{E5}
\end{goban}

```

/pgfgo/arrow/dash=*<solid, dashed or dotted>*

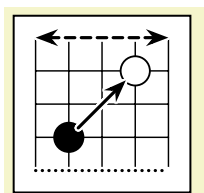
(no default, initially solid)

The stroke: solid, with dashes or with dots. The halo and the tip are always solid.

/pgfgo/arrow/shorten=*<dimension>*

(no default, initially 0pt)

Shortens both ends.



```

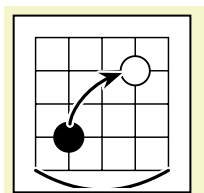
\begin{goban}[5]
\stones[b]{B2}\stones[w]{D4}
\arrow[shorten = 5pt]{B2}{D4}
\arrow[dash = dashed, tips = both]{A5}{E5}
\line[dash = dotted]{A1}{E1}
\end{goban}

```

/pgfgo/arrow/bend=*<angle>*

(no default, initially 0)

Draws an arc instead of a straight line: the arrow leaves turned that angle to the left and arrives turned the same, like `bend left` of TikZ. With a negative angle, it bends to the right.



```

\begin{goban}[5]
\stones[b]{B2}\stones[w]{D4}
\arrow[bend = 40, shorten = 5pt]{B2}{D4}
\line[bend = -30]{A1}{E1}
\end{goban}

```

/pgfgo/arrow/tip=*<arrow tip>*

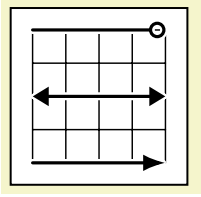
(no default, initially Stealth)

The tip of the arrow, one of those of the TikZ library `arrows.meta`, such as `Latex` or `Triangle`.

`/pgfgo/arrow/tips=<end or both>`

(no default, initially end)

Where the tip goes: only at the end, or at both ends.



```
\begin{goban}[5]
\arrow[tip = LaTeX]{A1}{E1}
\arrow[tip = Triangle, tips = both]{A3}{E3}
\arrow[tip = {Circle[open]}]{A5}{E5}
\end{goban}
```

`/pgfgo/arrow/halo=<halo options>`

(default true)

The halo around the arrow.

`/pgfgo/arrow/halo/true`

(style, no value)

Draws the halo. It is the initial value.

`/pgfgo/arrow/halo/false`

(style, no value)

Does not draw the halo.

`/pgfgo/arrow/halo/color=<color>`

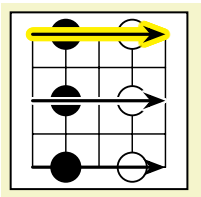
(no default)

Color of the halo. Without this *key*, it is that of the background of the board.

`/pgfgo/arrow/halo/width=<dimension>`

(no default, initially 0.8pt)

Width of the halo on each side of the arrow.



```
\begin{goban}[5]
\stones[b]{B1, B3, B5}\stones[w]{D1, D3, D5}
\arrow[halo = false]{A1}{E1}
\arrow{A3}{E3}
\arrow[halo = {color = yellow, width = 2pt}]{A5}{E5}
\end{goban}
```

`/pgfgo/arrow/forget`

(no value)

`resume` does not bring this arrow, even if `marks` is asked for.

Lines have the same options, except `tip`:

`/pgfgo/line/color=<color>`

(no default, initially black)

Like `/pgfgo/arrow/color`.

`/pgfgo/line/width=<dimension>`

(no default, initially 1.2pt)

Like `/pgfgo/arrow/width`.

`/pgfgo/line/opacity=<number>`

(no default, initially 1)

Like `/pgfgo/arrow/opacity`.

`/pgfgo/line/dash=<solid, dashed or dotted>`

(no default, initially solid)

Like `/pgfgo/arrow/dash`.

`/pgfgo/line/shorten=<dimension>`

(no default, initially 0pt)

Like `/pgfgo/arrow/shorten`.

`/pgfgo/line/bend=<angle>`

(no default, initially 0)

Like `/pgfgo/arrow/bend`.

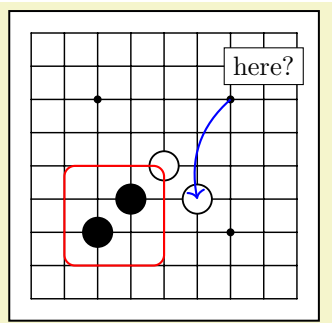
<code>/pgfgo/line/halo=<halo options></code>	(default true)
Like <code>/pgfgo/arrow/halo</code> .	
<code>/pgfgo/line/halo/true</code>	(style, no value)
Like <code>/pgfgo/arrow/halo/true</code> .	
<code>/pgfgo/line/halo/false</code>	(style, no value)
Like <code>/pgfgo/arrow/halo/false</code> .	
<code>/pgfgo/line/halo/color=<color></code>	(no default)
Like <code>/pgfgo/arrow/halo/color</code> .	
<code>/pgfgo/line/halo/width=<dimension></code>	(no default, initially 0.8pt)
Like <code>/pgfgo/arrow/halo/width</code> .	
<code>/pgfgo/line/forget</code>	(no value)
Like <code>/pgfgo/arrow/forget</code> .	

7.2 Drawing with TikZ

A `goban` is a TikZ picture, so `\draw`, `\node` and the other TikZ commands can be used inside it. The unit is the distance between two lines of the board, and (1,1) is the lower left corner. What is drawn this way goes on top of everything, `partial = auto` does not take it into account and it is not remembered: `resume` and the views do not bring it.

(go cs:<intersection>)

The coordinate system `go` gives the intersection of the board written as in `\arrow` and `\line`: (go cs:C3) is C3. It can be outside the board, like (go cs:C0), below column C. If the coordinate is not understood, it warns (go-coordinate) and uses the origin, (0,0).



```
\begin{goban}[9]
\stones[b]{C3, D4}\stones[w]{E5, F4}
\draw[red, thick, rounded corners] (go cs:B2) rectangle (go cs:E5);
\draw[blue, ->, thick] (go cs:G7) to[bend right] (go cs:F4);
\node[fill = white, draw] at (go cs:H8) {here?};
\end{goban}
```

8 The game: captures, notes and territory

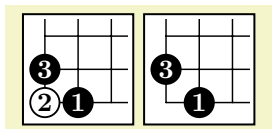
With the Lua kernel, a `goban` can follow the rules of the game: remove the captured stones, note the moves that fall on points already played and count the territory.

8.1 Captures

With the Lua kernel (p. 30), PGFGO can keep track of the stones on the board and remove those left without liberties, as in a game.

/pgfgo/goban/captures=(*remove, fade or false*) (default **remove**, initially **false**)

Turns captures on. When a move leaves an opposing group without liberties, that group is removed from the board. If it leaves its own group without liberties (a suicide), that group is removed. Stones put with `\stone` and `\stones` also count; neutral stones and prisoners do not. A move outside a partial board also captures. A move on an occupied point gives a warning, because it is not possible in a game. With `kernel = expl3`, this *key* gives an error.



```
\pgfgo{goban = {5, partial = A1:C3}}

\begin{goban}
\moves{B1,A1,A2}
\end{goban}

\begin{goban}[captures]
\moves{B1,A1,A2} % 3 captures 2.
\end{goban}
```

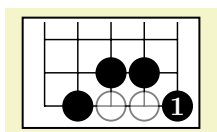
Since PGF cannot erase what it has already drawn, with `captures` the stones are drawn when the `goban` ends. So, if the style is changed with `\pgfgo` inside a `goban` with `captures`, the change does not reach the stones; the options of each command do.

/pgfgo/goban/captures/remove (style, no value)

The captured stones disappear. It is the same as `captures` or `captures = true`.

/pgfgo/goban/captures/fade (style, no value)

The captured stones stay on the board, drawn as prisoners (with the style of `\prisoner` and of `/pgfgo/prisoner`).



```
\pgfgo{goban = {5, partial = A1:E3}}

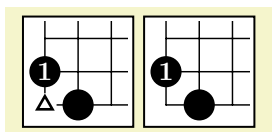
\begin{goban}[captures = fade]
\stones[w]{C1,D1}
\stones[b]{B1,C2,D2}
\moves{E1}
\end{goban}
```

/pgfgo/goban/captures/false (style, no value)

Turns captures off. It is the initial value.

/pgfgo/goban/captured marks=(*keep or remove*) (no default, initially **keep**)

Decides what happens to the marks on a stone when it is captured. It only has an effect with `captures`. Marks on empty points do not change.



```
\pgfgo{goban = {5, partial = A1:C3, captures}}

\begin{goban}
\stones[w]{A1}\stones[b]{B1}
\marks[triangle]{A1}
\moves{A2}
\end{goban}

\begin{goban}[captured marks = remove]
\stones[w]{A1}\stones[b]{B1}
\marks[triangle]{A1}
\moves{A2} % the triangle goes with the white stone.
\end{goban}
```

/pgfgo/goban/captured marks/keep (style, no value)

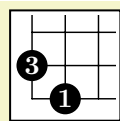
The marks stay on the empty point. It is the initial value.

/pgfgo/goban/captured marks/remove (style, no value)

The marks go with the stone, also with `captures = fade`. Sequences keep their numbering.

`\pgfgoprisoners{⟨player⟩}`

Writes how many prisoners the player (black or b, white or w) has in the last goban with captures: the opposing stones captured by its moves. With another player, it gives an error and writes 0.



Black: 1. White: 0.

```
\pgfgoset{goban = {5, partial = A1:C3}}
```

```
\begin{goban}[captures]
```

```
\moves{B1,A1,A2}
```

```
\end{goban}
```

```
Black: \pgfgoprisoners{black}. White: \pgfgoprisoners{white}.
```

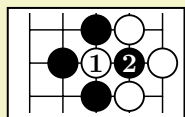
8.2 Moves on points already played

In numbered diagrams, when a move falls on a point where there was already a move (for example, when taking back a ko), books do not draw it: the point still shows the first stone and below the diagram it is noted, for example, “3 at 1”.

`/pgfgo/goban/notes=⟨boolean⟩`

(default true, initially false)

A numbered move on a point where there was already a numbered move of the same goban is not drawn: the point still shows the first one, even if it was captured, and “N at M” is noted. Only the numbered moves (`\moves` without `numbers = false`) that are seen on the board count. It works with both kernels.



3 at 1

```
\pgfgoset{goban = {5, partial = A2:E4}}
```

```
\begin{goban}[notes]
```

```
\stones[b]{B3,C2,C4}
```

```
\stones[w]{D2,D4,E3}
```

```
\moves[w]{C3}
```

```
\moves{D3}
```

```
\moves[from = 3, w]{C3}
```

```
\end{goban}
```

```
\pgfgomovenotes
```

`\pgfgomovenotes[⟨options⟩]`

Writes the notes of the last goban, separated by commas. If there are no notes, it writes nothing. The options are those of `/pgfgo/notes`.

`/pgfgo/notes = ⟨options⟩`

(no default)

Options of `\pgfgomovenotes` for the whole document, for example `\pgfgoset{notes = {at = en}}`.

`/pgfgo/notes/at = ⟨text⟩`

(no default, initially at or its translation)

Text between the two numbers.

`/pgfgo/notes/pass = ⟨text⟩`

(no default, initially pass or its translation)

A pass of an SGF file (p. 107) counts, but it is not drawn. With `notes`, if it would be numbered, “N: pass” is noted, in order with the other notes. The passes that end the game are not noted. This key is the text after the number.

`/pgfgo/notes/font = ⟨font commands⟩`

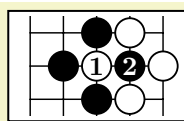
(no default)

The font of the notes. Without it, that of the text; in `\sgfdiagrams` and `\sgfproblem`, that of the caption.

`/pgfgo/notes/separator = <text>`

(no default, initially a comma and a space)

Text between the notes.



```
\begin{goban}[5, partial = A2:E4, notes]
\stones[b]{B3,C2,C4}
\stones[w]{D2,D4,E3}
\moves[w]{C3}
\moves{D3}
\moves[from = 3, w]{C3}
\moves{D3}
\end{goban}

\pgfgomovenotes[separator = {; }]
```

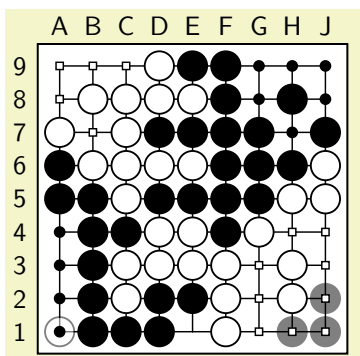
8.3 Territory

At the end of the game, the territory can be marked by hand with `\marks` (p. 78) or with `\territory`, which also counts it.

`\territory[<options>]{<polygon>}`

Marks as territory the points of the polygon that has those vertices (in order), including its border. It does not mark the points that have a living stone: prisoners count as territory. The vertices are written as in `\stones`; a single one marks a point, and two, a segment. It works with both kernels.

The territory is drawn when the goban ends. Each point keeps the last `\territory` that marked it, with the player and the options of that `\territory` (so `\territory[n]{A1}` after another `\territory` makes A1 neutral), and it is neither drawn nor counted if at the end it has a living stone.



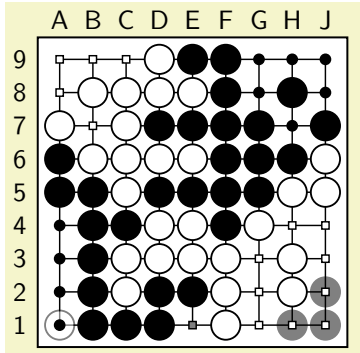
```
\begin{goban}[label = {at = {above, left}}]
\stones[b]{A6|B5|D1|E2, C4, E9|F4, D5|G7--D7, H6, H8, I7}
\stones[w]{E8|C5, B:E6, A7, B8, D9, I6|H5, H2:3, F1|C3, C2, D:E4, G4}
\prisoners[b]{H1|I2}
\prisoner[w]{A1}

\territory[b]{A1,A5,B5,B1}
\territory[b]{F9,F7,G7,G6,I6,I9}
\territory[w]{F1,F4,G4,G5,I5,I1}
\territory[w]{A9,A6,C6,C8,D8,D9}
\end{goban}
```

`\territory[<options>]{<auto>}`

With `auto` instead of a polygon, it finds the territory by itself: each region of points without living stones belongs to the color that surrounds it. If both colors touch it (dame, seki) or neither does, it is

neutral. Prisoners count as dead stones. Without a player, it marks the territory of both; with **b** or **w**, only that of that player. Only with the Lua kernel (p. 30).



```
\begin{goban}[label = {at = {above, left}}]
\stones[b] {A6|-B5|-D1|-E2, C4, E9-|F4, D5-|G7--D7, H6, H8, I7}
\stones[w] {E8-|C5, B:E6, A7, B8, D9, I6|- H5, H2:3, F1|-C3, C2, D:E4, G4}
\prisoners[b]{H1-|I2}
\prisoner[w] {A1}

\territory[dame]{auto}
\end{goban}
```

`auto` only looks at which color each region touches; it cannot read the position. So:

- It is for finished games. In an open position, almost the whole board touches both colors and is neutral; with stones of a single color, the whole board is theirs. If a border has a gap, the territory inside joins the dame and is neutral too.
- It does not know which stones are dead (sometimes the players do not either): they have to be put with `\prisoners`. A dead stone put with `\stones` counts as alive, and the region it touches is neutral.
- The eyes of a group in seki count as territory (under Japanese rules they do not). They can be made neutral afterwards, with a `\territory[n]` for each one.
- Neutral stones (without a player) occupy their point, but belong to nobody: a region that only touches neutral stones is neutral.
- It finds the regions with the stones there are when `\territory` appears, so it is best put at the end of the `goban`.
- With `partial = auto`, the board grows to show the whole territory (and the dame, with `dame`).

`/pgfgo/territory/black` (no value)

The territory is Black's. It can also be written `b`.

`/pgfgo/territory/white` (no value)

The territory is White's. It can also be written `w`.

`/pgfgo/territory/neutral` (no value)

Neutral points. It is what a polygon without a player marks. It can also be written `n`.

`/pgfgo/territory/b` (no value)

Short for black.

`/pgfgo/territory/w` (no value)

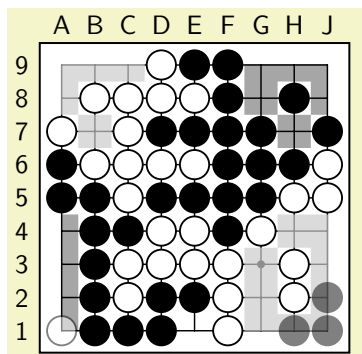
Short for white.

`/pgfgo/territory/n` (no value)

Short for neutral.

`/pgfgo/territory/draw=(marks, fill or region)` (no default, initially `marks`)

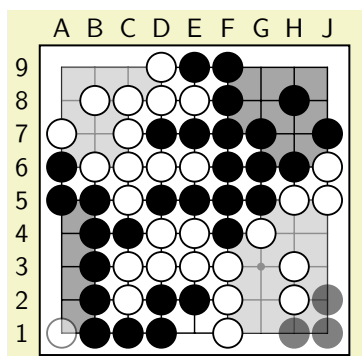
With `marks`, each point gets the territory mark of the player. With `fill`, the region is filled under the stones.



```
\begin{goban}[label = {at = {above, left}}]
\stones[b] {A6|-B5|-D1|-E2, C4, E9-|F4, D5-|G7--D7, H6, H8, I7}
\stones[w] {E8-|C5, B:E6, A7, B8, D9, I6|- H5, H2:3, F1|-C3, C2, D:E4, G4}
\prisoners[b]{H1-|I2}
\prisoner[w] {A1}

\territory[draw = fill]{auto}
\end{goban}
```

With `region`, the territory of each color is a single polygon: each square of the grid that has a corner in the territory is painted (dead stones count as territory). Its sides run along the lines of the board, its vertices are always on intersections, and it reaches the edge of the board. Two zones that touch at a single corner stay separate. All the `\territory` of a color in the same `goban` (also the territory `TB` and `TW` of `\sgf`) form a single region, which is drawn when the `goban` ends with the style of the last one. Neutral points are filled one by one, as with `fill`. The outline and its vertices can be drawn on top, with `line` and `vertices`, and `\pgfgoregion` gives its properties.



```
\begin{goban}[label = {at = {above, left}}]
\stones[b] {A6|-B5|-D1|-E2, C4, E9-|F4, D5-|G7--D7, H6, H8, I7}
\stones[w] {E8-|C5, B:E6, A7, B8, D9, I6|- H5, H2:3, F1|-C3, C2, D:E4, G4}
\prisoners[b]{H1-|I2}
\prisoner[w] {A1}

\territory[draw = region]{auto}
\end{goban}
```

`/pgfgo/territory/dame=(boolean)` (default `true`, initially `false`)

With `auto`, also marks the neutral points.

`/pgfgo/territory/forget` (no value)

`resume` does not bring this territory, even if `territory` is asked for (p. 101).

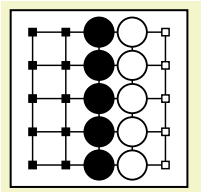
The other options of `\territory` change the style of the player only for that command, like `mark` and `fill` in `\territory[w, mark = {cross}]{auto}`. Without a player, they change that of the three. The style of each player is set with `/pgfgo/players`:

`/pgfgo/players/black/territory=<territory options>` (no default)

Style of the territory of each player (black, white or neutral).

`/pgfgo/players/black/territory/mark=<mark options>` (no default, initially `t-circle`)

Mark of each point, with `draw = marks`. It is `t-circle` for Black and `t-square` for White and the neutral points.



```
\begin{goban}[5]
\stones[b]{C1:5}
\stones[w]{D1:5}
\territory[b, mark = {t-square}]{A1, A5, B5, B1}
\territory[w]{E1, E5}
\end{goban}
```

`/pgfgo/players/black/territory/fill=<fill options>` (default `true`)

Fill, with `draw = fill` and `draw = region`.

`/pgfgo/players/black/territory/fill/true` (no value)

Draws the fill. It is the initial value.

`/pgfgo/players/black/territory/fill/false` (no value)

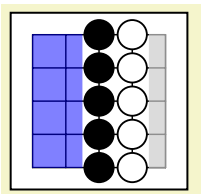
Does not draw the fill: with `draw = region` only the outline or the vertices remain.

`/pgfgo/players/black/territory/fill/color=<color>` (no default, initially `black`)

Color of the fill: `black` for Black, `white!80!black` for White and `gray` for the neutral points. With `none`, no color.

`/pgfgo/players/black/territory/fill/opacity=<number>` (no default, initially 0.35)

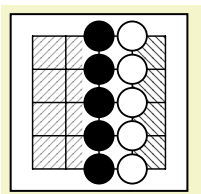
Opacity of the fill: 0.35 for Black and the neutral points, 0.7 for White.



```
\begin{goban}[5]
\stones[b]{C1:5}
\stones[w]{D1:5}
\territory[b, draw = fill,
fill = {color = blue, opacity = 0.5}]{A1, A5, B5, B1}
\territory[w, draw = fill]{E1, E5}
\end{goban}
```

`/pgfgo/players/black/territory/fill/pattern=<options>` (no default)

A pattern over the fill color, as `stone/fill/pattern` (p. 62). With `fill/color = none`, only the pattern is left.



```
\begin{goban}[5]
\stones[b]{C1:5}
\stones[w]{D1:5}
\territory[b, draw = fill, fill = {color = none,
pattern = {name = north east lines}}]{A1, A5, B5, B1}
\territory[w, draw = region, fill = {color = none,
pattern = {name = north west lines}}]{E1:5}
\end{goban}
```

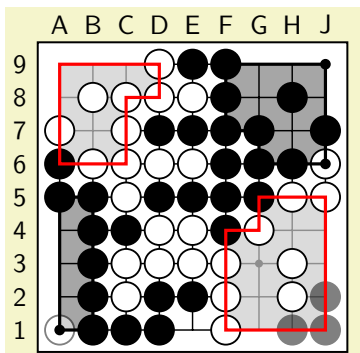
`/pgfgo/players/black/territory/fill/pattern/name = <pattern>` (no default)

The pattern: any of the TikZ libraries `patterns` and `patterns.meta`, which PGFGO loads. One with options goes in braces: `name = {Lines[angle = 45, distance = 2pt]}`. Without `name` there is no pattern.

`/pgfgo/players/black/territory/fill/pattern/color =  $\langle color \rangle$` (no default, initially **black**)
 Color of the pattern.

`/pgfgo/players/black/territory/fill/pattern/false` (no value)
 No pattern.

`/pgfgo/players/black/territory/line= $\langle line options \rangle$` (default **true**)
 Outline of the region, with `draw = region`. Alone or with **true**, it draws the outline; options alone do not turn it on.



```
\begin{goban}[label = {at = {above, left}}]
\stones[b] {A6|-B5|-D1|-E2, C4, E9-|F4, D5-|G7--D7, H6, H8, I7}
\stones[w] {E8-|C5, B:E6, A7, B8, D9, I6|- H5, H2:3, F1|-C3, C2, D:E4, G4}
\prisoners[b] {H1-|I2}
\prisoner[w] {A1}

\territory[b, draw = region, line,
vertices]{auto}
\territory[w, draw = region,
line = {true, color = red}]{auto}
\end{goban}
```

`/pgfgo/players/black/territory/line/true` (no value)
 Draws the outline.

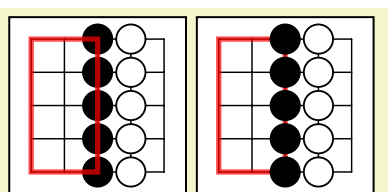
`/pgfgo/players/black/territory/line/false` (no value)
 Does not draw the outline. It is the initial value.

`/pgfgo/players/black/territory/line/color= $\langle color \rangle$` (no default, initially **black**)
 Color of the outline: **black** for Black and **white** for White.

`/pgfgo/players/black/territory/line/width= $\langle dimension \rangle$` (no default, initially 1.2pt)
 Width of the outline.

`/pgfgo/players/black/territory/line/opacity= $\langle number \rangle$` (no default, initially 1)
 Opacity of the outline and of the vertices.

`/pgfgo/players/black/territory/line/above= $\langle boolean \rangle$` (default **true**, initially **true**)
 With **true**, the outline and the vertices are drawn on top of the stones and the marks. With **false**, under the stones, together with the fill.



```

\begin{goban}[5]
\stones[b]{C1:5}
\stones[w]{D1:5}
\territory[b, draw = region, fill = false,
  line = {true, color = red, width = 2pt, opacity = 0.7}]{A1, A5, B5, B1}
\end{goban}

\begin{goban}[5]
\stones[b]{C1:5}
\stones[w]{D1:5}
\territory[b, draw = region, fill = false,
  line = {true, color = red, width = 2pt, opacity = 0.7, above = false}]{A1, A5, B5, B1}
\end{goban}

```

`/pgfgo/players/black/territory/vertices=<options>` (default **true**)

Points on the vertices of the outline, with `draw = region`, like the star points. They are drawn in the same layer as the outline. Alone or with **true**, it draws them; options alone do not turn them on.

`/pgfgo/players/black/territory/vertices/true` (no value)

Draws the vertices.

`/pgfgo/players/black/territory/vertices/false` (no value)

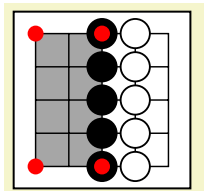
Does not draw the vertices. It is the initial value.

`/pgfgo/players/black/territory/vertices/radius=<dimension>` (no default, initially 2pt)

Radius of the points.

`/pgfgo/players/black/territory/vertices/color=<color>` (no default)

Color of the points. Without this *key*, they take the color of the outline.



```

\begin{goban}[5]
\stones[b]{C1:5}
\stones[w]{D1:5}
\territory[b, draw = region,
  vertices = {true, radius = 3pt, color = red}]{A1, A5, B5, B1}
\end{goban}

```

`\pgfgoscore[<options>]{<player>}`

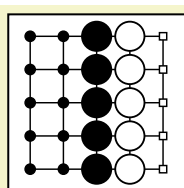
Writes the score of the player (black or b, white or w) in the last goban, under Japanese rules: the points of its territory, plus the dead opposing stones in it, plus the stones it captured (with `captures`), plus the komi for White. A point marked twice counts only once. With `neutral` or n, it writes the number of neutral points marked. With another player, it gives an error and writes 0.

Black: `\pgfgoscore{black}`. White: `\pgfgoscore[komi = 6.5]{white}`.

With the board above, it writes “Black: 11. White: 23.5”.

`/pgfgo/score/komi=<number>` (no default, initially 0)

Komi added to White’s score. For the whole document: `\pgfgoset{score = {komi = 6.5}}`.



Black: 10. White: 11.5.

```

\begin{goban}[5]
\stones[b]{C1:5}
\stones[w]{D1:5}
\territory{auto}
\end{goban}

Black: \pgfgoscore{b}. White: \pgfgoscore[komi = 6.5]{w}.

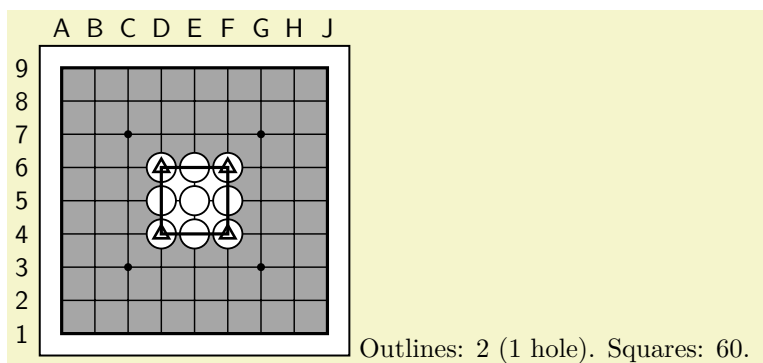
```

`\pgfgoregion{<player>}{<property>}`

Writes a property of the region (`draw = region`) of the player (black or `b`, white or `w`) in the current goban or in the last one. It is expandable, so it can be used in a computation or inside the coordinates of `\marks`. The properties are those of the whole outline, even if the board is partial. Without a region, the numbers are 0. The properties are:

- **polygons**: the number of outlines, outer ones and holes.
- **outlines**: the number of outer outlines, one for each separate zone.
- **holes**: the number of holes.
- **vertices**: the number of vertices of all the outlines.
- **area**: the number of painted squares.
- **perimeter**: the length of all the outlines, in squares.
- **polygon <i>**: the vertices of outline <i>, as coordinates separated by commas (for example, A1,D1,D4,A4). Empty if there are not that many outlines.

Each outline starts at its lowest vertex and, among those, at the leftmost one. Outer outlines go counterclockwise and holes, clockwise. The outlines are sorted by that first vertex, from bottom to top.



```

\begin{goban}[label = {at = {above, left}}]
\stones[w]{D4:F6}
\territory[b, draw = region, line]{B2,B8,H8,H2}
\marks[triangle]{\pgfgoregion{b}{polygon 2}}
\end{goban}

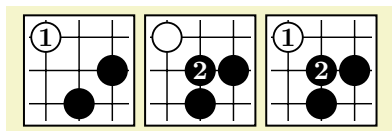
Outlines: \pgfgoregion{b}{polygons}
(\pgfgoregion{b}{holes} hole).
Squares: \pgfgoregion{b}{area}.

```

9 Remembering boards and views

9.1 Remembering a board

A named `goban` keeps everything it draws (stones, moves, marks, arrows, lines and territory, also what comes from `\sgf`), together with its options. Another `goban` can start with that board as it was left, with `resume`, and `\pgfgoview` shows it in the text (p. 103).



```
\pgfgo goban = {5, partial = B2:D4}

\begin{goban}[name = board1]
\stones[b]{C2,D3}
\moves[w]{B4}
\end{goban}

\begin{goban}[resume = board1]
\moves{C3} % Goes on with 2 (Black).
\end{goban}

\begin{goban}[resume = {board1, numbers}]
\moves{C3}
\end{goban}
```

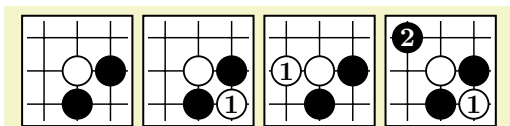
/pgfgo/goban/name=*<name>* (no default)

Keeps with that name everything drawn in this **goban**, together with its options. The name is free (for example, **gob:board1**). Another **goban** with the same name replaces it.

/pgfgo/goban/resume=*<name, options>* (no default)

Starts the **goban** with the board that has that name, as it was at the end (with the captures resolved: if that **goban** used **captures**, this one does too). The numbering and the turn go on from there. By default it brings only the stones, without numbers, marks or territory; the options ask for the rest. The size is that of that board, unless this **goban** gives its own (**size**, a number, **sgf size** or **sgf view**; one set before with **\pgfgo set** does not count); the other options of the board (**partial**, coordinates) are given in each **goban**. If there is no **goban** with that name, an error is given.

With **name** and **resume** in the same **goban** (in any order), the new name keeps what it brought, as it was drawn, plus its own, and the board of **resume** does not change: that is how variations are built. With the same name in both, the **goban** keeps adding to that board.



```
\pgfgo goban = {5, partial = B2:D4}

\begin{goban}[name = base]
\stones[b]{C2,D3}
\stones[w]{C3}
\end{goban}

\begin{goban}[name = variation1, resume = base]
\moves[w]{D2}
\end{goban}

\begin{goban}[name = variation2, resume = base]
\moves[w]{B3}
\end{goban}

\begin{goban}[resume = {variation1, numbers}]
\moves{B4}
\end{goban}
```

/pgfgo/goban/resume/numbers=*<boolean>* (default true, initially false)

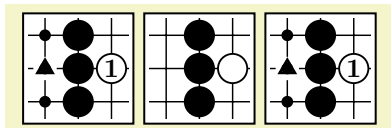
Also brings the numbers of the moves.

`/pgfgo/goban/resume/marks=<boolean>` (default true, initially false)

Also brings the marks, the arrows and the lines.

`/pgfgo/goban/resume/territory=<boolean>` (default true, initially false)

Also brings the territory.



```
\pgfgoreset{goban = {5, partial = B2:D4}}

\begin{goban}[name = saved]
\stones[b]{C2:4}
\moves[w]{D3}
\marks[triangle]{B3}
\territory[b]{B2, B4}
\end{goban}

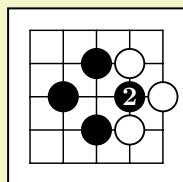
\begin{goban}[resume = saved]
\end{goban}

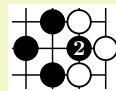
\begin{goban}[resume = {saved, numbers, marks, territory}]
\end{goban}
```

So that `resume` does not bring a stone, a move, a mark, an arrow or a territory, it is drawn with the option `forget` (`\stone[forget]{C3}`, `\marks[forget, circle]{A1}`, ...); with `label = forget`, a move comes without its number even if `numbers` is asked for. What is forgotten stays kept: `\pgfgoview` shows it with `forgotten`.

9.2 Views in the text

A named `goban` keeps everything it draws. Afterwards, `\pgfgoview` shows part of that board, as it was at the end (with the captures resolved), in the middle of a paragraph: a single stone, a box or several.



Black at D3 ② captures White at C3 +. The ko: .

```
\begin{goban}[5, name = board1, captures]
\stones[b]{B3, C2, C4}
\stones[w]{D2, D4, E3}
\moves[w]{C3, D3}
\end{goban}

Black at D3 \pgfgoview{board1}{D3} captures White at C3 \pgfgoview{board1}{C3}.
The ko: \pgfgoview{board1}{B2:E4}.
```

`\pgfgoview[<options>]{<name>}{<parts>}`

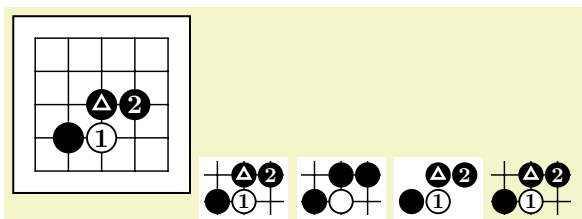
Draws the intersection or the rectangle (written as in `partial`) of the `goban` with that name, as it was at the end. With several parts separated by commas (`{F15, G15, C3:E5}`), it draws one view per part, joined with `view/separator` and `view/last`. By default everything the `goban` drew in that part is seen (stones, numbers of the moves, marks, arrows and territory) with the background and the coordinates of that `goban`, at the size of the text and centered at the height of the lowercase letters. The options are those of `/pgfgo/view`, which remove the marks, the numbers or the territory, or add what was

forgotten, and also any option of `goban`, which is applied after those of the named `goban`: that way the background, the coordinates, the grid or the star points are removed, or the style of the stones, the marks and the board is changed (`players`, `stone`, `mark`, `grid`, `background`, ...).

The view uses the style that `goban` had when it was drawn (the profile, what was set with `\pgfgoset` and its options), even if it changes later. On top go the profile of the views (`view/profile`), `\pgfgoset{view = ...}` and the options of the view.

The `goban` can come after the view (p. 106) or from another document (p. 107).

A view changes nothing of what the last `goban` keeps: `\pgfgoscore`, `\pgfgoprisoners` and `\pgfgomovenotes` still speak of it, and what each named `goban` keeps stays the same.



```
\begin{goban}[5, name = board2]
  \stones[b]{B2, C3}
  \moves[w]{C2, D3}
  \marks[b, triangle]{C3}
\end{goban}

\pgfgoview{board2}{B2:D3}
\pgfgoview[marks = false, numbers = false]{board2}{B2:D3}
\pgfgoview[label = false, grid = false]{board2}{B2:D3}
\pgfgoview[background = false]{board2}{B2:D3}
```

`/pgfgo/view=<options>` (no default)

Options for every view, with `\pgfgoset`. The options of `goban` that are not of `/pgfgo/view` are added to those the views bring: `scale = 0.8`, `background = {line = false}` (the background without border, even if the original has one).

Without background and larger: .

```
\pgfgoset{view = {background = false, scale = 1}}
Without background and larger: \pgfgoview{board2}{C3}.
```

`/pgfgo/view/marks=<boolean>` (default true, initially true)

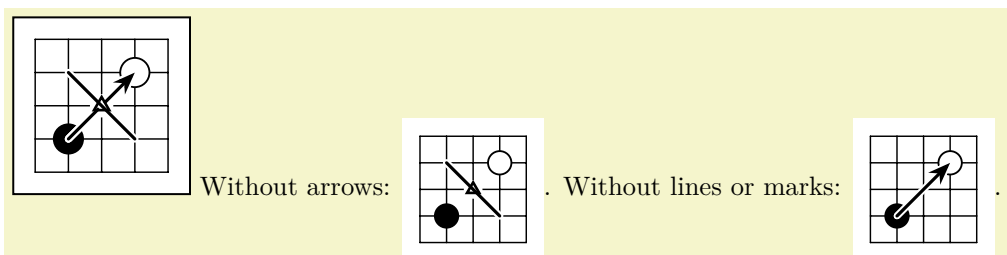
With false, does not draw the marks.

`/pgfgo/view/arrows=<boolean>` (default true, initially true)

With false, does not draw the arrows (`\arrow` and `AR` of `\sgf`).

`/pgfgo/view/lines=<boolean>` (default true, initially true)

With false, does not draw the lines (`\line` and `LN` of `\sgf`).



```

\begin{goban}[5, name = board-arrows]
\stones[b]{B2}\stones[w]{D4}
\arrow{B2}{D4}
\line{B4}{D2}
\marks[triangle]{C3}
\end{goban}

```

Without arrows: `\pgfgoview[arrows = false]{board-arrows}{A1:E5}`.

Without lines or marks: `\pgfgoview[lines = false, marks = false]{board-arrows}{A1:E5}`.

`/pgfgo/view/numbers=<boolean>`

(default true, initially true)

With false, does not draw the numbers of the moves.

`/pgfgo/view/territory=<boolean>`

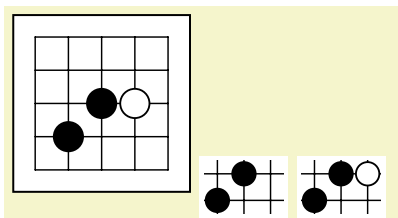
(default true, initially true)

With false, does not draw the territory.

`/pgfgo/view/forgotten=<boolean>`

(default true, initially false)

Also draws what was drawn with `forget` (p. 101).



```

\begin{goban}[5, name = board3]
\stones[b]{B2, C3}
\stone[w, forget]{D3}
\end{goban}

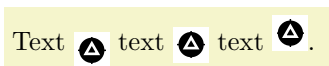
\pgfgoview{board3}{B2:D3}
\pgfgoview[forbidden]{board3}{B2:D3}

```

`/pgfgo/view/raise=<dimension>`

(no default, initially 0.5ex)

At what height above the baseline of the text the center of the view is.



```

Text \pgfgoview[raise = Opt]{board2}{C3}
text \pgfgoview{board2}{C3}
text \pgfgoview[raise = 1.5ex]{board2}{C3}.

```

`/pgfgo/view/separator=<text>`

(no default, initially a comma and a space)

What goes between the views of several parts.

`/pgfgo/view/last=<text>`

(no default, initially and between spaces, or its translation)

What goes before the last view, instead of `separator`. With two parts, only this goes. The spaces are part of the text.



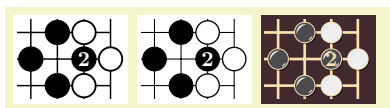
```

The stones \pgfgoview{board1}{B3, C2, C4} surround C3.
\pgfgoview[separator = {, }, last = { or }]{board1}{B3, C2, C4}.

```

`/pgfgo/view/profile=<name>` (no default, initially false)

A profile for the views (p. 135), for example `simple`. It goes after the style of the named `goban` and before the other options of the view. With `false`, the view uses the style of the `goban` as it is.



```
\pgfgoview{board1}{B2:E4}
\pgfgoview[profile = simple]{board1}{B2:E4}
\pgfgoview[profile = wain]{board1}{B2:E4}
```

9.3 A stone in the text

`\pgfgostone[<options>]`

Draws a single stone in the text, without a board. It uses the style of that moment and, on top, the one of the views: `view/profile`, `view/raise` and the `goban` options of `\pgfgoset{view = ...}`, such as `scale`. So it has the same size and height as the view of one intersection. The options are those of `\stone` (`b`, `w`, `n`, `prisoner`, `mark` and the style of the stone) and `number`. Without `b` or `w`, the stone is neutral.

It changes nothing of what the last `goban` keeps.

The black stone  cuts and the white stone  answers.     

```
The black stone \pgfgostone[b, mark = triangle] cuts and
the white stone \pgfgostone[w, mark = {custom = A}] answers.
```

```
\pgfgostone[b] \pgfgostone[w] \pgfgostone[n]
\pgfgostone[b, prisoner] \pgfgostone[w, mark = cross]
```

`/pgfgo/pgfgostone/number=<number>` (no default)

The stone has the number, with the style of the moves of its color (`move/label`). A neutral stone has no number: it is drawn without it, with a warning.

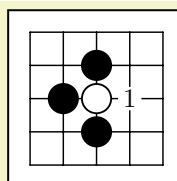
Black  cuts, white  ataris.

```
Black \pgfgostone[b, number = 5] cuts,
white \pgfgostone[w, number = 6] ataris.
```

9.4 Views before the board

A view can come before the `goban` it shows, like a reference with `\ref`: the `goban` is kept in the `.aux` file and the view uses it in the next compilation. The first time the view is ?? and there is a warning; if the `goban` changes afterwards, another warning asks to compile again. If two `goban` environments have the same name, a view that comes before both shows the first one.

Further down, the white stone at C3  is in atari.



```
Further down, the white stone at C3 \pgfgoview{board4}{C3} is in atari.
\begin{goban}[5, name = board4]
\stones[b]{B3, C2, C4}
\stones[w]{C3}
\marks[sequence]{D3}
\end{goban}
```

9.5 Using a board in another document

A named goban can be saved in a file and used in another document, with its stones, numbers, marks, territory and style.

/pgfgo/goban/export=*<file>* (no default)

When it ends, the goban is written to that file (if it has no extension, .pgfgo), for \pgfgoimport. It needs name. For security, T_EX only writes inside the folder of the document: the file goes with a relative name, such as boards/ko.

```
\begin{goban}[19, name = ko, export = ko]
...
\end{goban}
```

\pgfgoimport[*<options>*]{*<file>*}

Brings the goban of an export file (without an extension, .pgfgo), as if it had been drawn in this document: it can be used with \pgfgoview and with resume. It can be used in the preamble.

```
\pgfgoimport{ko}
\pgfgoimport[name = other ko]{ko}
...
The ko of \pgfgoview{ko}{Q16:S18}.
\begin{goban}[resume = {other ko, marks}]
\moves[numbers = false]{R17}
\end{goban}
```

A mark defined with \pgfgodefinemark has to be defined in this document too; otherwise, there is a warning.

/pgfgo/import/name=*<name>* (no default)

The name of the imported goban. Without name, the one it had.

10 Games in SGF

SGF is the format almost every Go program uses to save games. With the Lua kernel (p. 30), PGFGO can read them.

In every command that reads SGF files, the name can have spaces, accents, parentheses and other scripts. The special characters of T_EX are written with a backslash: \sgf{a\&b.sgf} reads a&b.sgf, and the same holds for \#, \%, \\$, \{ and \}. Two spaces in a row are read as one, because T_EX joins them before PGFGO sees the name.

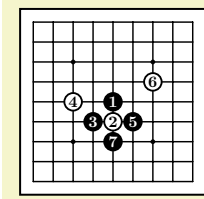
Each file is read only once in the whole document, however many commands use it (if it changes during the compilation, it is read again).

10.1 Names for the files

\pgfgodefinesgf[*<options>*]{*<name>*}{*<file>*}

Gives a name to a game: from then on, *<name>* works in every place where an SGF file goes (\sgf, \sgfinfo, \sgfcomment, \sgfdiagrams, \sgfproblem, \sgfanimation, sgf size and sgf view). So the path of the file is written only once, and so is what says which game it is (game in a collection, the charset, the syntax, the orientation). A name that is not defined is a file, as always; if a

name matches that of a file, the name wins. The options of each command win over those of the name: `\sgfinfo[game = 1]{<name>}{PB}` reads game 1 even if the name says another one. With `kernel = expl3`, the name is defined but cannot be used, like the SGF commands.



Move 7: Tesuji. Black captures.

```
\pgfdefinesgf{game}{game.sgf}

\begin{goban}[sgf size = game, scale = 0.6]
  \sgf[to = 7]{game}
\end{goban}

Move 7: \sgfcomment{game}{7}
```

`/pgfgo/define sgf/game = <number>` (no default, initially that of each command)

In a collection, the game used by the commands with this name, like `/pgfgo/sgf/game`.

```
\pgfdefinesgf[game = 3]{third}{collection.sgf}
```

`/pgfgo/define sgf/charset = <charset>` (no default, initially that of sgf charset)

The charset with which the file is read if it is not UTF-8, like `/pgfgo/sgf charset`, only for this name.

```
\pgfdefinesgf[charset = GBK]{qipu}{games/qipu-without-ca.sgf}
```

`/pgfgo/define sgf/syntax = <syntax>` (no default, initially that of sgf syntax)

The syntax with which the file is read, like `/pgfgo/sgf syntax`, only for this name.

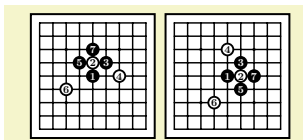
```
\pgfdefinesgf[syntax = sensei]{josekis}{diagrams/josekis.txt}
```

`/pgfgo/define sgf/origin = <corner>` (no default, initially top left)

The corner where the point aa of the file goes: top left, top right, bottom left or bottom right. With top left it is read as SGF says. The orientation belongs only to the name: no option of the commands changes it, and the same file used by its path is read as SGF says. It holds for everything read from the file (stones, moves, marks, arrows, views) and for `partial = auto`. The coordinates written in the commands, like those of `variation = {R16, P17}`, are those of the board, not those of the file.

`/pgfgo/define sgf/transpose = <boolean>` (default true, initially false)

The first letter of each point is the row and the second the column; the size is turned too (a 13×9 game becomes 9×13). Together with `origin`, it gives the eight symmetries of the board.



```
\pgfdefinesgf[origin = bottom right]{mirror}{game.sgf}
\pgfdefinesgf[transpose]{transposed}{game.sgf}

\begin{goban}[sgf size = mirror, scale = 0.4]
  \sgf[to = 7]{mirror}
\end{goban}

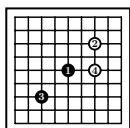
\begin{goban}[sgf size = transposed, scale = 0.4]
  \sgf[to = 7]{transposed}
\end{goban}
```

```
\begin{pgfgosgf}[\langle options \rangle]{\langle name \rangle}
  \langle environment contents \rangle
\end{pgfgosgf}
```

A file written in the document, which every SGF command reads as the file $\langle name \rangle$, in any syntax (sgf syntax). The $\langle options \rangle$ are those of `\pgfgdefinesgf`; `charset` does nothing, because the text is already in the encoding of the document. The lines are read as they are: `\begin{pgfgosgf}[\langle options \rangle]{\langle name \rangle}` ends its line and `\end{pgfgosgf}` goes on a line of its own. It does not work inside the argument of another command or inside `pgfgolayout`. If $\langle name \rangle$ is also the name of a file, the environment wins. With `kernel = expl3`, the content is skipped and there is an error.

```
\begin{pgfgosgf}{short}
  (;SZ[9];B[ee];W[gc];B[cg];W[ge])
\end{pgfgosgf}

\begin{goban}[sgf size = short, scale = 0.4]
  \sgf{short}
\end{goban}
```

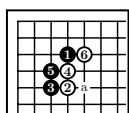


10.2 Sensei's Library diagrams

Sensei's Library and forums such as L19 write diagrams as text, with lines that start with `$$`. A file with `sgf syntax = sensei`, or with `auto` if it has no SGF nodes and has those lines, is read as a collection: each block of consecutive `$$` lines is a game (`game = 2` is the second diagram), and what is between the blocks is not read. The `[go]` and `[/go]` tags of L19 can be left in. The file works everywhere an SGF file goes, and also with `\pgfgdefinesgf` and in the environment `pgfgosgf`.

```
\begin{pgfgosgf}{nadare}
  $$$Bc Nadare
  $$ -----
  $$ | . . . . .
  $$ | . . . . .
  $$ | . . . 1 6 .
  $$ | . . 5 4 . .
  $$ | . . 3 2 a .
  $$ | . . . . .
\end{pgfgosgf}

\sgfdiagrams[goban = {scale = 0.5}]{nadare}
```



Nadare (1–6)

The first line is the header: `$$`, B or W (who plays move 1; black if absent), a board size (`$$$B13`; 19 if absent), m and the number of move 1 (`$$$Wm67`) and the title. The c for coordinates is accepted but not used: the coordinates are those of `goban/label`. Each diagram is read like this:

- The title is the name of the game (GN) and of its figure (FG): `\sgfdiagrams` puts it in the caption.
- X and O are setup stones (AB and AW). B, #, Y and Z are black stones with a circle, square, triangle and cross; W, @, Q and P, white ones. C, S, T and M are those marks on empty points, and lowercase letters are letters.
- The numbers (1 to 9, and 0 is 10) are moves, in order: the odd ones belong to whoever plays 1. If a number is missing ("5 at 1"), the following moves keep theirs (MN).

A line that is not understood is skipped with a warning. A diagram bigger than the size of the header makes the board bigger, also with a warning.

10.3 A game on a board

$$\backslash\text{sgf}[\langle options \rangle]\{\langle file \rangle\}$$

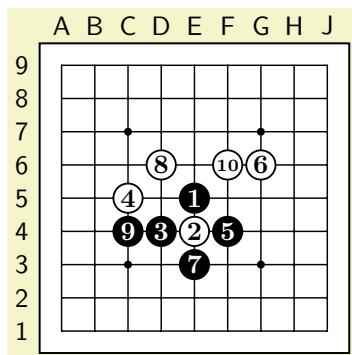
Draws on the `goban` the main line of the first game of the file: the initial stones (`AB` and `AW`) as with `\stones`, and the moves as with `\moves`. Variations are not drawn. Passes count for the numbering, but are not drawn (with `goban/notes` they are noted, “N: pass”). It only exists inside `goban`.

The stones that the nodes after the first move add (**AB**, **AW**) or remove (**AE**) are applied together with the previous move. Removing stones needs `captures`; without `captures`, `\sgf` warns once and does not remove them.

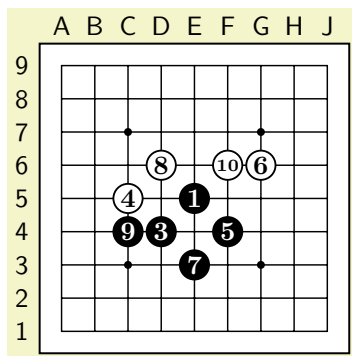
At the end, the numbering and the turn are ready to continue the game with `\moves`. `\sgf` works with `partial`, `captures`, `name` and `resume`. If the game is for a board of another size, a warning is given. With `kernel = expl3`, `\sgf` gives an error.

In the examples, the files are used by their name (p. 107). `game` is `game.sgf`, a short 9×9 game:

```
(;GM[1]FF[4]SZ[9]PB[Black]PW[White]C[A short game.]
;B[ee];W[ef];B[df];W[ce];B[ff];W[gd];B[eg]C[Black captures.];W[dd];B[cf];W[fd])
```



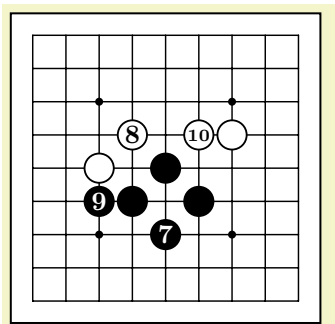
```
\begin{goban}[label = {at = {north, west}}]
  \sgf{game} % Without captures, 2 is not removed.
\end{goban}
```



```
\begin{goban}[label = {at = {north, west}}, captures]
\sgf{game}
\end{goban}
```

`/pgfgo/sgf/from` = $\langle number \rangle$ (no default, initially 1)

Number of the first numbered move. The previous moves are drawn without a number.

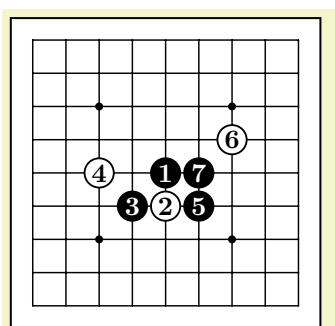


```
\begin{goban}[captures]
\sgf[from = 7]{game}
\end{goban}
```

`/pgfgo/sgf/to =  $\langle number \rangle$  or last`

(no default, initially last)

Number of the last move played. The following ones are not drawn. With **last** (or a negative number), up to the last one.



```
\begin{goban}[captures]
\sgf[to = 6]{game}
\moves{F5} % Goes on with 7 (Black), elsewhere.
\end{goban}
```

from and **to** count the moves of the file. The number shown can be another: the property **MN** of the file changes it (the following moves go on from there), and so does **restart**. In the examples that follow, **variations** is **variations.sgf**, a 7×7 game with two branchings, marks and **MN[10]** on the fourth move of the main line.

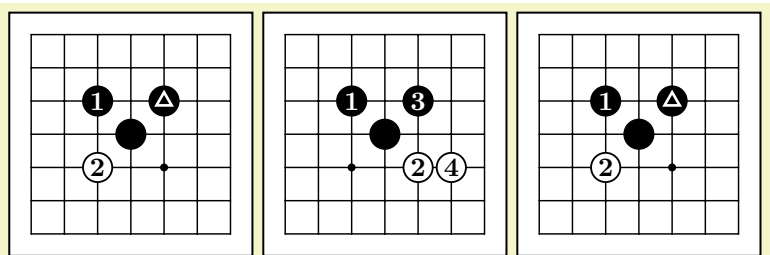
`/pgfgo/sgf/variation =  $\langle numbers or moves \rangle$`

(no default, initially the main line)

Chooses a variation: at each branching of the game, the given variation is taken, in order (1 is the main one). For example, **variation = 2.1** takes the second variation at the first branching and the first one at the second.

In a file with many branchings, such as a joseki dictionary, it is easier to give the moves of the line from the first one, separated by commas or spaces: **variation = {R16, P17, Q15}**. At each branching, the variation that plays the next move is taken; a pass is written **pass**. Each intersection is a column (one or two capital letters) and a number, as in **\stone**.

In both cases, when the list ends, the first variation is followed. If a variation (or a move) does not exist, a warning is given and the first one is followed from there.



```

\begin{goban}[sgf size = variations]
\sgf[variation = 2]{variations}
\end{goban}

\begin{goban}[sgf size = variations]
\sgf[variation = 1.2]{variations}
\end{goban}

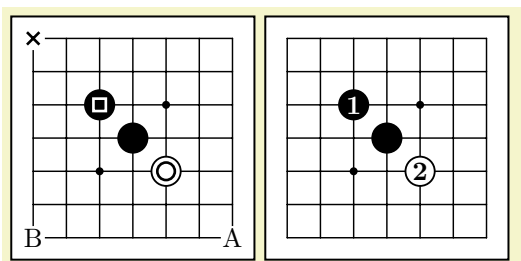
\begin{goban}[sgf size = variations]
\sgf[variation = {C5, C3}]{variations}
\end{goban}

```

/pgfgo/sgf/marks = *<boolean>*

(default **true**, initially **true**)

Draws the marks of the file (TR, SQ, CR, MA, SL and the labels of LB), the arrows (AR) and the lines (LN) of the last node shown, and dims the points of DD (which stay dimmed until the next DD; p. 88). Each mark takes the color of the stone on that point, so that it is seen, and replaces the number of the move. Labels are the exception: they do not cover a number, so on a numbered stone they are not drawn. The territory of the file (TB and TW) is drawn like that of `\territory` (p. 95) and counts for `\pgfgoscore`.



```

\begin{goban}[sgf size = variations]
\sgf[to = 2]{variations}
\end{goban}

\begin{goban}[sgf size = variations]
\sgf[to = 2, marks = false]{variations}
\end{goban}

```

/pgfgo/sgf/step = *<number>* or *<list of numbers>*

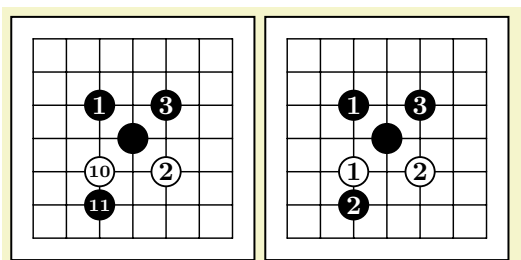
(no default, initially 100)

The size of the stretches in which **restart** restarts the numbering. With a list, each number is the size of a stretch, and the last one is repeated: with `step = {50, 100}`, the first stretch has 50 moves and the others 100. It is the same *key* as `/pgfgo/sgf diagrams/step`, where each stretch is a diagram.

/pgfgo/sgf/restart = *<restart options>*

(default **true**)

From the second stretch of `step` moves on, each one is numbered from 1 (or from `restart/from`), as in books: with `step = 100`, `restart`, move 101 is shown as 1. The first stretch keeps the numbers of the game. `\sgfanimation` and `\sgfdiagrams` have the same options.



```

\begin{goban}[sgf size = variations]
\sgf[]{variations}
\end{goban}

\begin{goban}[sgf size = variations]
\sgf[step = 3, restart]{variations}
\end{goban}

```

/pgfgo/sgf/restart/true (no value)

Restarts the numbering in each stretch.

/pgfgo/sgf/restart/false (no value)

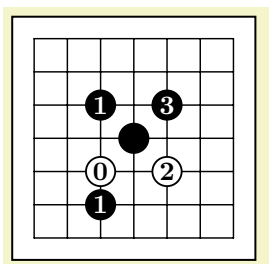
Does not restart it: the moves carry the numbers of the game (with MN).

/pgfgo/sgf/restart/auto (no value)

Does what the file asks for with PM: with PM[2], the numbering restarts every 100 moves, regardless of step (101 is shown as 1); otherwise, it does not restart. It is the initial value.

/pgfgo/sgf/restart/from = $\langle number \rangle$ (no default, initially 1)

The number of the first move of each stretch, from the second on. It also turns **restart** on. With **step = 99** and **restart = {from = 0}**, move 100 is shown as 0, 101 as 1, and so on.



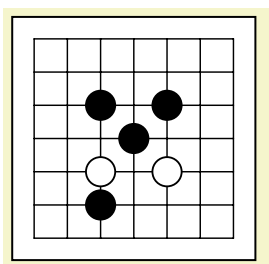
```

\begin{goban}[sgf size = variations]
\sgf[step = 3, restart = {from = 0}]{variations}
\end{goban}

```

/pgfgo/sgf/numbers = $\langle true, false \text{ or } auto \rangle$ (default **true**, initially **auto**)

With **false**, the moves are drawn without a number. With **auto**, it does what the file asks for with PM: with PM[0], they are not numbered. **\sgfdiagrams** has its own *key numbers*, and **\sgfproblem** always numbers.



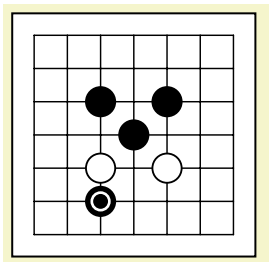
```

\begin{goban}[sgf size = variations]
\sgf[numbers = false]{variations}
\end{goban}

```

/pgfgo/sgf/last mark = $\langle mark options \rangle$ or false (no default, initially **false**)

Marks the last move shown, with the options of a mark (**circle**, **triangle**, **custom = ...**, etc.). A pass is not marked, nor is a point the file already marks or a move with a number (the number already says which is the last): in practice, it marks something with **numbers = false**.



```
\begin{goban}[sgf size = variations]
\sgf[numbers = false, last mark = circle]{variations}
\end{goban}
```

/pgfgo/sgf/game = *<number>*

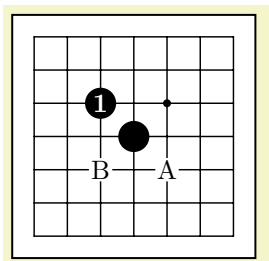
(no default, initially 1)

An SGF file can have several games (a collection). Chooses which one is read; if it does not exist, the first one is read with a warning. It also holds for `\sgfanimation`, `\sgfdiagrams`, `\sgfcomment`, `\sgfinfo` and `sgf size`, which have their own *key game*; for the whole document, `\pgfgoset{sgf = {game = 2}}`.

/pgfgo/sgf/variations = *<none, children, siblings or auto>*

(default auto, initially none)

Marks with letters the variations of the last move shown: with `children`, the moves that can follow; with `siblings`, the alternatives to that move (the letter replaces its number). `auto` does what the file asks for with ST. They are only marked if there is more than one.



```
\begin{goban}[sgf size = variations]
\sgf[to = 1, marks = false, variations = children]{variations}
\end{goban}
```

`\sgf` also reads PL (who plays next: `\moves` goes on with that color) and CA (the charset of the file; see `sgf charset`).

/pgfgo/sgf charset = *<charset>*

(no default, initially auto)

Charset of the SGF files that are not in UTF-8, for every command that reads files. With `auto`, it is the one the file gives with CA or, if it gives none, Latin-1 (the SGF default). `Latin-1`, `GBK` (and `GB2312`), `Big5`, `Shift_JIS`, `EUC-JP` and `EUC-KR` are read; the names ignore case and punctuation (`shift-jis` and `EUC_KR` also work). A file that is valid UTF-8 is always read as UTF-8, even if CA says otherwise; one in UTF-16 (with the byte order mark at the beginning) is read too.

Many files from Asian servers do not give their charset; to read them, for example, `\pgfgoset{sgf charset = GBK}`. If a file without CA seems to be in one of those charsets, a warning reminds of it. With another charset a warning is given and the file is read as Latin-1: it has to be converted to UTF-8 first, for example with `iconv -f KOI8-R -t UTF-8 game.sgf > game-utf8.sgf`.

To write the comments in Chinese, Japanese or Korean, the package of that language is also needed: `ctex`, `luatexja` or `luatexko`. Examples 14, 18 and 19 of `doc/examples` show a commented game in each one. Without it, LaTeX only notes the missing characters in the `.log`; so, if the font with which a comment or an `\sgfinfo` is written lacks some Chinese, Japanese or Korean character, a warning is given, once per font. With those packages (or `xeCJK`) this is not checked: they write those characters with another font.

/pgfgo/sgf syntax = *<syntax>*

(no default, initially auto)

Syntax of the files, for every command that reads files: `sgf`, `sensei` (Sensei's Library and L19 diagrams, p. 109) or `auto`, which is `sensei` if the file has no SGF nodes (`(;`) and has lines that start with `$$`, and `sgf` otherwise. Every syntax is read as an SGF game, so every command works the same with any of them. With another syntax a warning is given and `auto` is used.

/pgfgo/sgf = *<sgf options>*

(no default)

Options of every `\sgf` of the document, for example `\pgfgoset{sgf = {variations}}`.

10.4 Information and comments

\sgfinfo[*<options>*]{*<file>*}{*<property>*}

Writes a property of the root of the game, for example PB and PW (the players), RE (the result), KM (the komi) or DT (the date), or with **move** that of a move. If it is not there, it writes nothing. It can be used outside **goban**.

Ann (Black) against Bob (White), komi 6.5. Result: B+R.

```
\sgfinfo{variations}{PB} (Black) against
\sgfinfo{variations}{PW} (White),
komi \sgfinfo{variations}{KM}.
Result: \sgfinfo{variations}{RE}.
```

The value is written as it is in the file (with the escaped characters already resolved and in the charset of CA): RE stays as B+R, not in words. The properties with information about the game, according to the SGF format, are:

<i>Property</i>	<i>What it is</i>	<i>Example</i>
PB, PW	The name of Black and that of White.	Lee Sedol
BR, WR	The rank of Black and that of White.	9p
BT, WT	The team of Black and that of White.	
RE	The result: B+ or W+ and the points, R (resignation), T (time) or F (forfeit); 0 or Draw, a draw; Void, no result; ?, unknown.	W+3.5
KM	The komi.	6.5
HA	The handicap: how many stones.	2
DT	The date, YYYY-MM-DD (or several, separated by commas).	2016-03-09
EV	The tournament or event.	Meijin
RO	The round.	3 (final)
PC	The place.	Tokyo
GN	The name of the game.	
GC	A comment about the game: the background, the players.	
ON	The opening (fuseki).	chinese
RU	The rules.	Japanese
TM	The time of each player, in seconds.	7200
OT	The overtime (byo-yomi).	5x30 byo-yomi
SO	The source: a book, a magazine.	
AN	Who commented on the game.	
US	Who wrote the file.	
CP	The copyright.	
AP	The program that wrote the file and its version.	CGoban:3
SZ	The size of the board (19, or 19:13 if it is not square).	19
CA	The charset of the file.	UTF-8
FF, GM	The version of the format and the game (1: Go).	4

Any other property of the root (or of the move, with **move**) works. If it has several values, they are written separated by commas: **\sgfinfo{game}{AB}** writes the black stones put before the start in SGF coordinates, dd, pp. The long names of the old versions of the format (PlayerBlack) are read as their abbreviation: they are asked for with PB.

\sgfcomment[*<options>*]{*<file>*}{*<move>*}

Writes the comment of a move of the file (0: the one before the first move): the name of the node (N), in bold; the annotations of the move (TE, BM, DO, IT) and of the position (GB, GW, DM, UC); and the comment (C). The annotations are written with the texts of /pgfgo/sgf annotations. The values (V) and the hotspots (HO), which are for the game tree, are not written. It can be used outside **goban**. **\sgfanimation** and **\sgfdiagrams** write the comments the same way.

Move 7: Tesuji. Black captures.

```
Move 7: \sgfcomment{game}{7}
```

`/pgfgo/sgf info/game = <number>` (no default, initially 1)

Like `/pgfgo/sgf/game`, for `\sgfinfo`.

`/pgfgo/sgf info/move = <number>` (no default, initially 0)

The property of that move: that of its node and that of the nodes without a move that follow it. With 0, that of the root (and of the nodes before the first move). For example, the time White has left at move 120 of a KGS or OGS game (WL, in seconds): `\sgfinfo[move = 120]{game}{WL}`.

`/pgfgo/sgf info/variation = <numbers>` (no default, initially the main line)

With move, the line of the move, like `/pgfgo/sgf/variation`.

`/pgfgo/sgf comment/game = <number>` (no default, initially 1)

Like `/pgfgo/sgf/game`.

`/pgfgo/sgf comment/variation = <numbers>` (no default, initially the main line)

Like `/pgfgo/sgf/variation`.

Bad move. Wrong: White lives.

```
\sgfcomment[variation = 2]{problem}{1}
```

`/pgfgo/sgf comment/annotations = <boolean>` (default true, initially true)

Writes the name and the annotations. With false, only the comment.

With: Tesuji. Black captures. Without: Black captures.

```
With: \sgfcomment{game}{7}
\quad Without: \sgfcomment[annotations = false]{game}{7}
```

`/pgfgo/sgf comment/chat = <boolean>` (default true, initially true)

With false, removes the chat of the servers from the comment: on OGS, everything after the line “-- chat --” (what comes before is the comment of the reviewer); on KGS, the lines “name [rank]: ...” and, if there is any, also the lines “name: ...” of the same comment (those of the players without a rank). If nothing is left, the move has no comment.

The texts of the annotations are in the language of the document (p. 30), and they can be changed for the whole document with `\pgfgoset{sgf annotations = {...}}`, for example `bad move = Mistake`. The emphasized annotations of DM and UC use the same text.

`/pgfgo/sgf annotations/tesuji = <text>` (no default, initially Tesuji or its translation)

Text of TE.

Good move. Black captures.

```
\pgfgoset{sgf annotations = {tesuji = Good move}}
\sgfcomment{game}{7}
```

`/pgfgo/sgf annotations/very good tesuji = <text>` (no default, initially Very good tesuji or its translation)
Text of TE[2].

`/pgfgo/sgf annotations/bad move = <text>` (no default, initially Bad move or its translation)
Text of BM.

`/pgfgo/sgf annotations/very bad move = <text>` (no default, initially Very bad move or its translation)
Text of BM[2].

`/pgfgo/sgf annotations/doubtful = <text>` (no default, initially Doubtful move or its translation)
Text of DO.

`/pgfgo/sgf annotations/interesting = <text>` (no default, initially Interesting move or its translation)
Text of IT.

`/pgfgo/sgf annotations/good for black = <text>` (no default, initially Good for Black or its translation)
Text of GB.

`/pgfgo/sgf annotations/very good for black = <text>` (no default, initially Very good for Black or its translation)
Text of GB[2].

`/pgfgo/sgf annotations/good for white = <text>` (no default, initially Good for White or its translation)
Text of GW.

`/pgfgo/sgf annotations/very good for white = <text>` (no default, initially Very good for White or its translation)
Text of GW[2].

`/pgfgo/sgf annotations/even = <text>` (no default, initially Even position or its translation)
Text of DM.

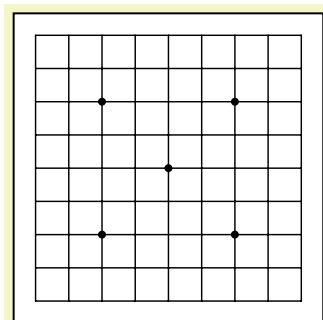
`/pgfgo/sgf annotations/unclear = <text>` (no default, initially Unclear position or its translation)
Text of UC.

10.5 Animations

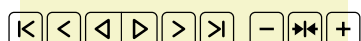
`\sgfanimation[<options>]{<file>}`

Steps through the game with the `animate` package: it draws one `goban` per move, and the buttons of `animate` step forward, back or play it. Frame k shows the position after move k ; frame 0, the position before the first move. It is used outside `goban`. The boards have the size of the game (`SZ`), unless `goban` gives another one. It needs `\usepackage{animate}` and two compilations.

The animations of `animate` work in Acrobat Reader and in some other viewers (Foxit, PDF-XChange). In the rest only the first frame is seen. These options can be set for the whole document with `\pgfgoreset{sgf animation = {...}}`.



A short game.



```
\sgfanimation[comments, goban = {captures}]{game}
```

`/pgfgo/sgf animation/from =  $\langle number \rangle$` (no default, initially 0)
First frame.

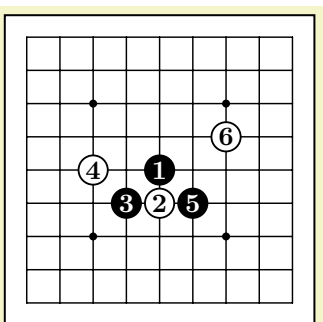
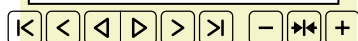
`/pgfgo/sgf animation/to =  $\langle number \rangle$  or last` (no default, initially last)
Last frame. With `last` (or `-1`), that of the last move.

`/pgfgo/sgf animation/fps =  $\langle number \rangle$` (no default, initially 1)
Frames per second when playing.

`/pgfgo/sgf animation/goban =  $\langle goban options \rangle$` (no default)
Options of each board, as in `\begin{goban}[...]`. As in `\sgfdiagrams`, the view of the file is applied after them: to fix the visible part there is `partial`.

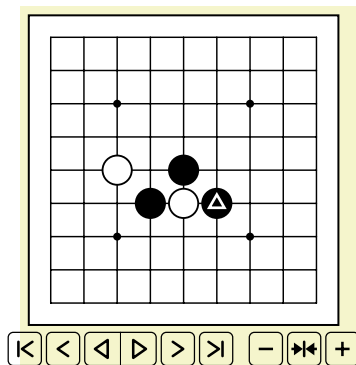
`/pgfgo/sgf animation/animate =  $\langle animate options \rangle$` (no default, initially controls)
Options of the `animateinline` environment of `animate`, for example `controls`, `loop`.

`/pgfgo/sgf animation/numbers =  $\langle boolean \rangle$` (default true, initially false)
Numbers every move, like `\sgf`. Without `numbers`, the moves have no number and the last one is marked.



```
\sgfanimation[from = 6, to = 8, numbers, goban = {captures}]{game}
```

`/pgfgo/sgf animation/last mark =  $\langle mark options or false \rangle$` (no default, initially circle)
Mark of the last move, with the color of whoever played it. With `false`, the last move is not marked. It is not used with `numbers`, nor when the file already marks that point.



```
\sgfanimation[from = 5, to = 7, last mark = triangle, fps = 2,
animate = {controls, loop}, goban = {captures}]{game}
```

`/pgfgo/sgf animation/comments =  $\langle$ comments options $\rangle$` (default true)

Shows the comment (C) of the move below the board. That of frame 0 is that of the nodes before the first move. It is turned on with `comments` or `comments = true` and off with `false` (the initial value); `height` and `font` do not turn it on: `comments = {true, height = 6em}`.

`/pgfgo/sgf animation/comments/true` (style, no value)

Shows the comments.

`/pgfgo/sgf animation/comments/false` (style, no value)

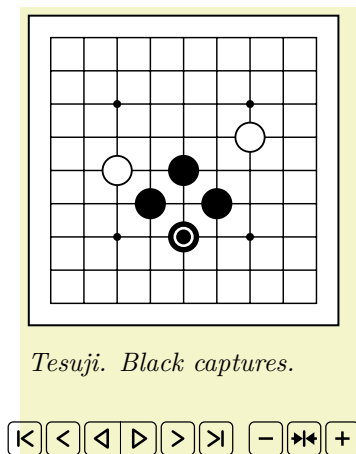
Does not show the comments. It is the initial value.

`/pgfgo/sgf animation/comments/height =  $\langle$ dimension $\rangle$` (no default, initially 4em)

Height of the space for the comments. It is the same in every frame, so that they do not change size.

`/pgfgo/sgf animation/comments/font =  $\langle$ font commands $\rangle$` (no default, initially \small)

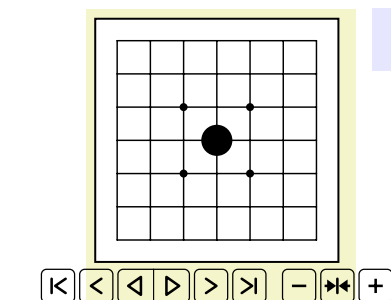
Font of the comments.



```
\sgfanimation[from = 7, to = 9, goban = {captures},
comments = {true, height = 2.5em, font = \itshape}]{game}
```

`/pgfgo/sgf animation/variation =  $\langle$ numbers $\rangle$` (no default, initially the main line)

Like `/pgfgo/sgf/variation`: the animation goes through the line that leads to that variation.



```
\sgfanimation[variation = 2, numbers]{variations}
```

`/pgfgo/sgf animation/marks =<boolean>` (default `true`, initially `true`)
 Like `/pgfgo/sgf/marks`: each frame shows the marks of its move.

`/pgfgo/sgf animation/step = <number> or <list of numbers>` (no default, initially 100)
 Like `/pgfgo/sgf/step`.

`/pgfgo/sgf animation/restart =<restart options>` (default `true`)
 Like `/pgfgo/sgf/restart`.

`/pgfgo/sgf animation/restart/true` (no value)
 Like `/pgfgo/sgf/restart/true`.

`/pgfgo/sgf animation/restart/false` (no value)
 Like `/pgfgo/sgf/restart/false`.

`/pgfgo/sgf animation/restart/auto` (no value)
 Like `/pgfgo/sgf/restart/auto`. It is the initial value.

`/pgfgo/sgf animation/restart/from = <number>` (no default, initially 1)
 Like `/pgfgo/sgf/restart/from`.

`/pgfgo/sgf animation/annotations =<boolean>` (default `true`, initially `true`)
 With `comments`, also writes the name and the annotations of each move, like `\sgfcomment`.

`/pgfgo/sgf animation/chat =<boolean>` (default `true`, initially `true`)
 With `comments`, like `/pgfgo/sgf comment/chat`: with `false`, without the chat of KGS and OGS.

`/pgfgo/sgf animation/game = <number>` (no default, initially 1)
 Like `/pgfgo/sgf/game`.

`/pgfgo/sgf animation/view =<boolean>` (default `true`, initially `true`)
 If the file has a view (VW), the boards show only that part, like `sgf view`. Every frame has the same board: if the view changes during the game, the board covers the views of every frame (and if in any of them the whole board is seen, the whole board is seen).

`/pgfgo/sgf animation/partial = <auto, false or corners>` (no default, initially `false`)
 With `auto`, where the file has no view, the boards show only the part used by the line that is animated (that of `variation`), as in `\sgfdiagrams`. With two corners (`partial = M12:T19`), every board shows that part, even if the file has a view. Every frame has the same board.

`/pgfgo/sgf animation/layers =<boolean>` (default `true`, initially `true`)
 Builds the animation in layers: the board is drawn only once, and so is each stone, on a transparency that `animate` shows while the stone is still on the board (with the `timeline` option of `animate`). The only thing drawn in each frame is what changes: the mark of the last move, the marks of the file and the comment. So the PDF grows with the number of moves and not with its square: a game of 200 moves takes less than 1 MB, instead of more than 10 MB, and it also compiles faster. With `layers = false`, each frame is a whole goban. The `timeline` is kept in an auxiliary file, `<jobname>-pgfgo-<n>.tln`.

10.6 Diagrams as in books

`\sgfdiagrams[<options>]{<file>}`

Draws the game as in books: one diagram every `step` moves (1–50, 51–100, etc.). Each diagram starts from the real position, with the previous moves without a number and their captures; the stones captured inside the diagram are not removed (they are removed in the next one). Below each one go its caption and its “N at M” notes (p. 94). The caption gives the numbers of the first and the last move, the ones on the stones if the file changes the numbering (MN). A game without moves is a single diagram, of its position, with no numbers in the caption. It is used outside `goban`. The options can be set for the whole document with `\pgfgoset{sgf diagrams = {...}}`.

```
\sgfdiagrams[step = 5, title = Diagram]{game}
```

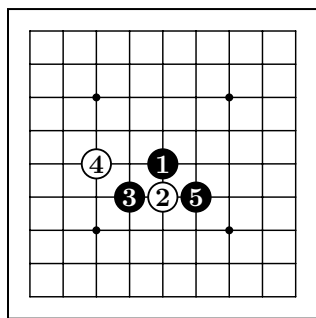


Diagram 1 (1–5)

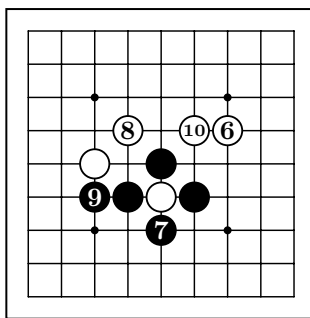


Diagram 2 (6–10)

`/pgfgo/sgf diagrams/step` = $\langle number \rangle$ or $\langle list\ of\ numbers \rangle$ (no default, initially 50)

Number of moves of each diagram. With a list, the first diagram has as many moves as the first number says, the second as many as the second says, and so on; the last number is repeated: with `step = {5, 10, 3}`, the diagrams have 5, 10, 3, 3, 3... moves.

`/pgfgo/sgf diagrams/from` = $\langle number \rangle$ (no default, initially 1)

First move. The diagrams are the same as without `from`, with the same number: only those with moves from `from` on are drawn, and the first one starts at `from`. With `step = 50` and `from = 120`, the first one is “Diagram 3 (120–150)”.

`/pgfgo/sgf diagrams/to` = $\langle number \rangle$ or `last` (no default, initially `last`)

Last move. With `last` (or `-1`), the last one of the game.

`/pgfgo/sgf diagrams/numbers` = $\langle boolean \rangle$ (default `true`, initially `true`)

With `false`, the moves of each diagram are drawn without a number (the “N at M” notes still use the numbers of the game).

`/pgfgo/sgf diagrams/last mark` = $\langle mark\ options \rangle$ or `false` (no default, initially `false`)

Like `/pgfgo/sgf/last mark`, in each diagram.

```
\sgfdiagrams[step = 5, to = 10, numbers = false, last mark = circle]{game}
```

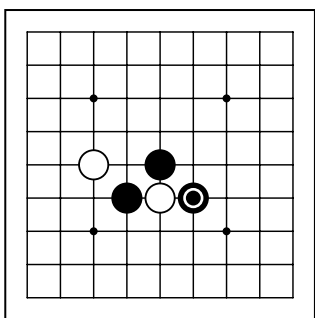


Diagram 1 (1–5)

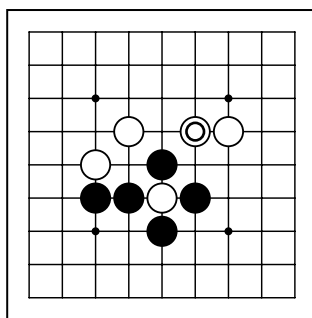


Diagram 2 (6–10)

`/pgfgo/sgf diagrams/restart` = $\langle restart\ options \rangle$ (default `true`)

Like `/pgfgo/sgf/restart`, with one diagram per stretch: from the second diagram on, each one is numbered from 1 (or from `restart/from`), as some books do. The first diagram keeps the numbers of the game.

```
\sgfdiagrams[step = {2, 3}, restart = {from = 0}]{game}
```

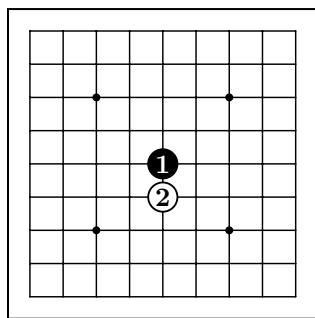


Diagram 1 (1-2)

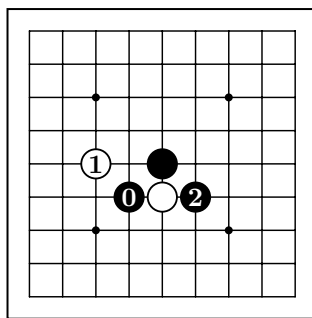


Diagram 2 (3-5)

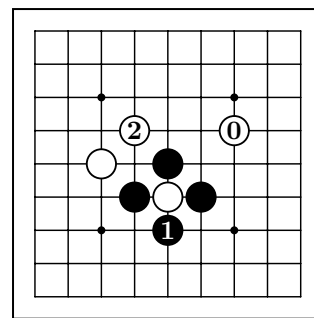


Diagram 3 (6-8)

`/pgfgo/sgf diagrams/restart/true`

(no value)

Restarts the numbering in each diagram.

`/pgfgo/sgf diagrams/restart/false`

(no value)

Does not restart it: the moves carry the numbers of the game.

`/pgfgo/sgf diagrams/restart/auto`

(no value)

Like `/pgfgo/sgf/restart/auto`: with `PM[2]`, every 100 moves, regardless of the diagrams. It is the initial value.

`/pgfgo/sgf diagrams/restart/from = <number>`

(no default, initially 1)

The number of the first move of each diagram, from the second on. It also turns `restart` on. With `step = 99` and `restart = {from = 0}`, the first diagram goes from 1 to 99, and in the second one move 100 is shown as 0, 101 as 1, and so on.

`/pgfgo/sgf diagrams/marks = <boolean>`

(default `true`, initially `true`)

Like `/pgfgo/sgf/marks`: each diagram shows the marks, arrows and lines of its last move. With `marks = false` they are not drawn.

`/pgfgo/sgf diagrams/notes = <boolean>`

(default `true`, initially `true`)

Writes its notes below each diagram: “N at M” and the passes, “N: pass”.

`/pgfgo/sgf diagrams/title = <text>`

(no default, initially `Diagram or its translation`)

Word of the caption: “Diagram 2 (51-100)”.

`/pgfgo/sgf diagrams/caption = <caption options>`

(no default)

Options of the caption and of the notes.

`/pgfgo/sgf diagrams/caption/font = <font commands>`

(no default, initially `\small`)

Font of the caption and of the notes: `caption = {font = \itshape}`.

```
\sgfdiagrams[from = 6, title = Figure, caption = {font = \itshape},
goban = {label = {at = {north, west}}}] {game}
```

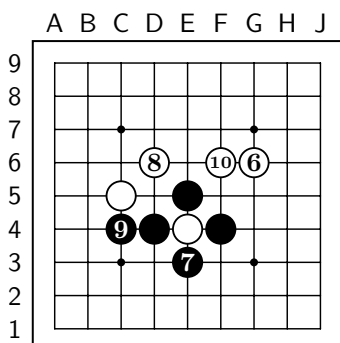


Figure 1 (6-10)

`/pgfgo/sgf diagrams/caption/min width =  $\langle dimension \rangle$` (no default, initially 8em)

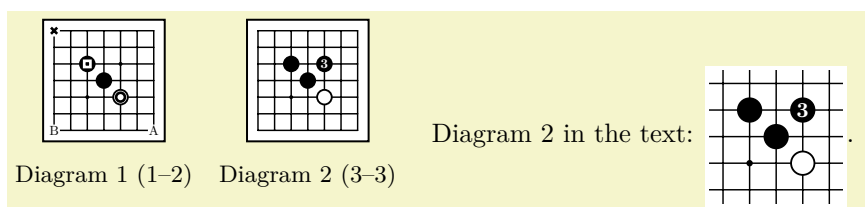
With `layout = {columns = auto}`, a caption wider than its board is broken into lines at the width of the board. If the board is narrower than this measure (in the font of the caption), it is broken at this measure; it is never broken narrower than its longest word, nor widened beyond its width on one line. In the grid, the caption is broken at the width of the cell.

`/pgfgo/sgf diagrams/goban =  $\langle goban options \rangle$` (no default)

Options of each board. By default, the boards have the size of the game and use `captures` and `notes`. The view of the file (`view`) is applied after these options: if the file has a view, `goban = {partial = ...}` does not change the visible part; for that there is `partial`.

`/pgfgo/sgf diagrams/name =  $\langle name \rangle$` (no default)

Each diagram is a named goban (`/pgfgo/goban/name`): this name followed by the number of its caption. With `name = var`, the diagrams are called `var1`, `var2...` and the variations `var2a`, `var2b...`. They can be used with `resume` and in the views in the text (p. 103). Each one keeps the whole position up to its last move, with the captures resolved: a stone of the numbered stretch captured in the same diagram stays in the drawing, but not in the memory.



```
\sgfdiagrams[step = 2, to = 3, name = var, goban = {scale = 0.5}]{variations}
Diagram 2 in the text: \pgfgoview{var2}{B2:F6}.
```

`/pgfgo/sgf diagrams/layout =  $\langle layout options \rangle$` (no default)

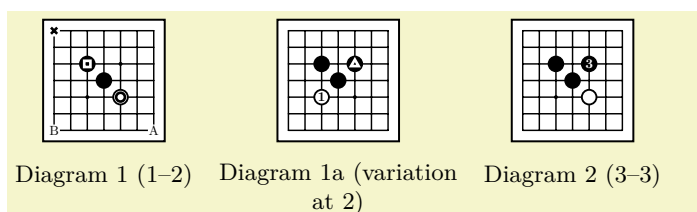
How the diagrams are laid out: options of `/pgfgo/layout` (p. 131) only for `\sgfdiagrams`. For example, `layout = {columns = 3}`.

`/pgfgo/sgf diagrams/variation =  $\langle numbers \rangle$` (no default, initially the main line)

Like `/pgfgo/sgf/variation`.

`/pgfgo/sgf diagrams/variations =  $\langle variations options \rangle$` (default commented)

Also draws the variations of the game, each one in its own diagram, after the diagram where its first move starts: “Diagram 3a”, “Diagram 3b”... The moves of each variation are numbered from 1, and the caption says which move the first one replaces (`variation title`). The previous ones are seen without a number, as they were. The options say which ones are drawn, and they add up: `variations = {commented, 5}`. Besides the ones below, anything else is the path of a variation, as in `variation` (2, 2.3.1, or its moves in braces and separated by spaces: `{R16 P17 Q15}`): that one is always drawn, at any depth. The 1’s at the end do not change the line: 2.3.1 is the same as 2.3. If the file does not have it, a warning is given. Its comments and notes are written like those of the other diagrams. With `variation`, the variations are those that leave that line (the main one too).



```
\sgfdiagrams[step = 2, to = 3, variations = all, goban = {scale = 0.5}]{variations}
```

`/pgfgo/sgf diagrams/variations/none` (style, no value)
 None. It is the initial value, and clears the previous ones.

`/pgfgo/sgf diagrams/variations/commented` (style, no value)
 Those that have some comment (C) in their own moves. It is what plain `variations` does.

`/pgfgo/sgf diagrams/variations/all` (style, no value)
 All of them.

`/pgfgo/sgf diagrams/variations/depth = <number>` (no default, initially 1)
 To what depth `commented` and `all` draw: 1, only those that leave the line that is drawn (the main one, or that of `variation`); 2, also the variations of those, and so on. Those of a variation go after it, and their caption says which of its moves they replace.

`/pgfgo/sgf diagrams/variations/from = <number>` (no default, initially 1)
 From which move `commented` and `all` draw: only the variations that start at that move or later.

```
\sgfdiagrams[variation = {Q16, R14}, variations = {commented, from = 3}]{josekis}
```

`/pgfgo/sgf diagrams/variation title = <text>` (no default, initially ``variation at'' and the number, or its translation)
 What goes in parentheses in the caption of a variation: “Diagram 3a (variation at 26)”. `\pgfgovariationmove` is the number of the move it replaces.

`/pgfgo/sgf diagrams/comments =<comments options>` (default true)
 Writes below each diagram the comments of its moves, like `\sgfcomment`, each one with the number of the move. It is turned on with `comments` or `comments = true` and off with `false` (the initial value); the other options do not turn it on: `comments = {true, align = justify}`. So `\pgfgoset{sgf diagrams = {comments = {font = \itshape}}}` sets the style of the whole document without turning the comments on.

`/pgfgo/sgf diagrams/comments/true` (style, no value)
 Writes the comments.

`/pgfgo/sgf diagrams/comments/false` (style, no value)
 Does not write the comments. It is the initial value.

`/pgfgo/sgf diagrams/comments/font = <font commands>` (no default, initially that of the caption)
 Font of the comments. Without this option, that of `caption/font`.

`/pgfgo/sgf diagrams/comments/align = <left, right, center or justify>` (no default, initially left)
 Alignment of the comments in the width of the diagram. With `justify`, a narrow diagram gets large spaces between the words.

`/pgfgo/sgf diagrams/comments/root =<boolean>` (default true, initially false)
 Also writes the comment of the root (that of the nodes before the first move), without a number, before those of the diagram that starts at move 1.

```
\sgfdiagrams[step = 10, comments = {true, root, font = \itshape}]{game}
```

`/pgfgo/sgf diagrams/comments/number = <text>` (no default, initially `\textbf{\#1}:\sim`)
 What goes before each comment; `\#1` is the number of the move, the one seen on the board (with `restart`, the restarted one). For example, `number = {Move~\#1.~}`.

`/pgfgo/sgf diagrams/comments/separator =  $\langle text \rangle$` (no default, initially one paragraph per comment)

What goes between the comments. With the initial value, each one goes in its own paragraph; with `separator = {\quad}`, they run on.

```
\sgfdiagrams[step = 10, comments = {true, number = {\textbf{\#1}.~}, separator = {~}}]{game}
```

`/pgfgo/sgf diagrams/comments/position =  $\langle below, right, left or after \rangle$` (no default, initially below)

Where the comments go: below the caption, beside the diagram, as in books of commented games, or with **after** all together after the diagrams, across the page, each group with the caption of its diagram as a heading (they can break across pages). Beside, they start at the height of the top border of the board; the caption, the notes and the result go on below the board.

```
\sgfdiagrams[step = 10, comments = {true, position = right}]{game}
```

`/pgfgo/sgf diagrams/comments/width =  $\langle dimension \rangle$  or auto` (no default, initially auto)

Width of the comments beside. With **auto**, what is left: outside the grid, the rest of the line (each diagram goes on its own line); in the grid (`layout = {columns = ...}`), the rest of the cell, and with **fit** the board takes half of the cell and the comments the other half. With a dimension, in the grid the board takes what is left of the cell.

`/pgfgo/sgf diagrams/comments/sep =  $\langle dimension \rangle$` (no default, initially 1em)

Space between the diagram and the comments beside it.

```
\sgfdiagrams[step = 10, layout = {columns = 2, fit},
comments = {true, position = left, sep = 2em}]{game}
```

`/pgfgo/sgf diagrams/comments/height =  $\langle dimension \rangle$  or none` (no default, initially none)

Maximum height of the comments of a diagram, below or beside. What does not fit goes after the diagrams, as with **after**, with the caption of the diagram as a heading.

```
\sgfdiagrams[step = 30, layout = {columns = 2},
comments = {true, height = 6\baselineskip}]{game}
```

`/pgfgo/sgf diagrams/annotations =  $\langle boolean \rangle$` (default true, initially true)

With comments, also writes the name and the annotations of each move.

```
\sgfdiagrams[step = 10, comments]{game}
\sgfdiagrams[step = 10, comments, annotations = false]{game}
```

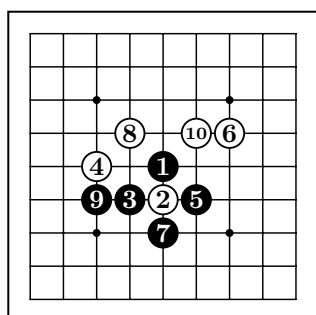


Diagram 1 (1-10)

7: Tesuji. Black captures.

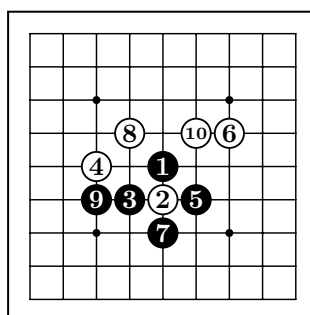


Diagram 1 (1-10)

7: Black captures.

`/pgfgo/sgf diagrams/chat =  $\langle boolean \rangle$` (default true, initially true)

With comments, like `/pgfgo/sgf comment/chat`: with **false**, without the chat of KGS and OGS. A move that only had chat left does not appear.

`/pgfgo/sgf diagrams/figures =<boolean>` (default `true`, initially `true`)

If the file marks figures (FG), each figure is a diagram, with its name as the caption, instead of one every step moves.

`/pgfgo/sgf diagrams/view =<boolean>` (default `true`, initially `true`)

If the file has a view (VW), each diagram shows only that part, like `sgf view`. If the view changes during the game, each diagram covers the views of its moves.

`/pgfgo/sgf diagrams/partial = <auto, false or corners>` (no default, initially `false`)

With `auto`, where the file has no view, each diagram shows only the part of the board used by the lines that are drawn (that of `variation` and those of `variations`), like `/pgfgo/goban/partial = auto`. It is the same in every diagram, also in those of the variations. With two corners, as in `/pgfgo/goban/partial (K10:T19, {K10, T19} or {K10 to T19})`, every diagram shows that part, even if the file has a view.

```
\sgfdiagrams[partial = auto, variations]{josekis}
\sgfdiagrams[partial = K10:T19, variations]{josekis}
\sgfdiagrams[partial = auto, variation = {Q16, R14},
  variations = {commented, from = 3}, marks = false]{josekis}
```

`/pgfgo/sgf diagrams/game = <number>` (no default, initially 1)

Like `/pgfgo/sgf/game`.

`/pgfgo/sgf diagrams/result =<result options>` (default `true`)

Writes below the last diagram, if it reaches the end of the game, the result (RE) and, if the file marks the territory (TB, TW), the score of each player: “Result: Black wins by 0.5 points (Black 6, White 5.5)”. The score is counted as with `\pgfgoscore` (p. 95), with the komi of the file (KM). It is turned off with `result = false`; `title` and `format` do not turn it on.

`/pgfgo/sgf diagrams/result/true` (style, no value)

Writes the result. It is the initial value.

`/pgfgo/sgf diagrams/result/false` (style, no value)

Does not write the result.

`/pgfgo/sgf diagrams/result/title = <text>` (no default, initially `Result` or its translation)

Word before the result.

`/pgfgo/sgf diagrams/result/format = <text or raw>` (no default, initially `text`)

How the result is written. `text`: in words, in the language of the document (p. 30), “Black wins by resignation”; `raw`: as it is in the file, “B+R”. With `text`, 0 or `Draw` (a draw), `Void` (no result), ? (unknown) and B+ or W+ followed by nothing, by the difference, by R or `Resign`, by T or `Time`, or by F or `Forfeit` are understood. Any other result is written as it is.

`/pgfgo/sgf diagrams/black name = <text>` (no default, initially `Black` or its translation)

Name of Black in the score and in the result.

`/pgfgo/sgf diagrams/white name = <text>` (no default, initially `White` or its translation)

Name of White in the score and in the result.

```
\sgfdiagrams[result = {title = Final, format = raw}]{variations}
\sgfdiagrams[black name = Ann, white name = Bob]{variations}
```

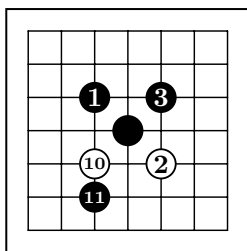


Diagram 1 (1-11)
Final: B+R

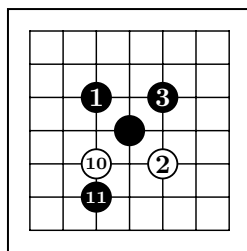


Diagram 1 (1-11)
Result: Ann wins by
resignation

`\pgfgowinner`

In the texts of the result (`\pgfgodefinelanguage`), the name of the winner: `black name` or `white name`.

`\pgfgovariationmove`

In the caption of a variation (`variation title`) and in its text in each language (`variation at`), the number of the move it replaces.

`\pgfgopoints`

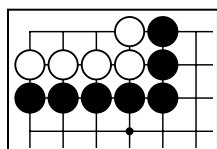
In the texts of the result, the difference in points, as it is in the file.

10.7 Problems

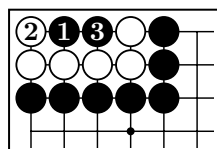
`\sgfproblem`[*options*]{*file*}

Draws a problem (tsumego). First, the statement: the initial position with its marks, who plays (PL, or the color of the first move) and the comment at the beginning. Then, one diagram for each line of the tree of the file (each variation of the solution, right or wrong), numbered from 1, with its “N at M” notes and the comments and annotations of its moves, which usually say whether the variation is right. The diagrams are drawn like those of `\sgfdiagrams`. If the file has no view (VW), only the part of the board the problem uses is shown. It is used outside `goban`. The options can be set for the whole document with `\pgfgoset{sgf problem = {...}}`. In the examples, `problem` is `problem.sgf`, a life-and-death problem: the right move and two wrong ones, with their comments.

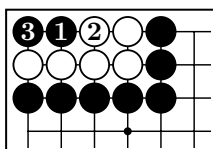
```
\sgfproblem[title = Problem 1, variation title = Line,
black to play = Black plays]{problem}
```



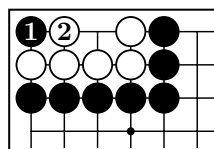
Problem 1: Black
plays
Black to kill.



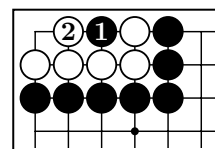
Line 1
1: Tesuji. Correct:
the vital point.



Line 2
1: Tesuji. Correct:
the vital point.
3: White has only
one eye.



Line 3
1: Bad move.
Wrong: White
lives.



Line 4
1: Wrong.

`/pgfgo/sgf problem/statement` = *(true, false or alone)*

(default `true`, initially `true`)

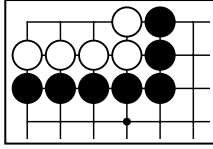
Draws the statement. With `alone`, the statement goes alone on its row, centered, and the variations below.

`/pgfgo/sgf problem/solutions` = *(boolean)*

(default `true`, initially `true`)

Draws the variations. With `solutions = false`, only the statement remains; the solutions go elsewhere, with `statement = false`.

```
\sgfproblem[solutions = false, title = Problem 2]{problem}
```



Problem 2: Black
to play
Black to kill.

`/pgfgo/sgf problem/title =  $\langle text \rangle$` (no default, initially `empty`)

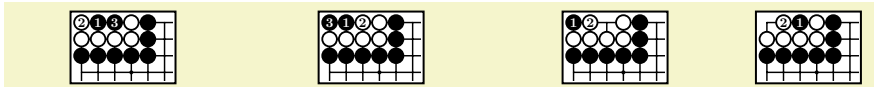
Beginning of the caption of the statement: “Problem 1: Black to play”. With several games (`game = 1:20`), it is followed by the number of each one; empty, it is `Problem` or its translation: “Problem 7: Black to play”. The solutions also say which problem they belong to: “Problem 7: Variation 1”.

`/pgfgo/sgf problem/to play =  $\langle auto, black, white \text{ or } none \rangle$` (no default, initially `auto`)

Who plays, for the caption of the statement. With `auto`, what PL says or, if it is not there, the color of the first move; if there is neither, it is not said. Many collections do not say it in the file: `to play = black`. With `none`, the caption does not say it.

`/pgfgo/sgf problem/verdict =  $\langle boolean \rangle$` (default `true`, initially `false`)

The caption of each solution says whether it is right or wrong (“Variation 2 (wrong)”), with the texts `correct` and `wrong` of the language. It is decided with the annotations of the line: the last GB or GW (the position is good for Black or for White), according to who plays; if there is none, the first move with TE or BM (the odd moves belong to whoever plays: a BM of theirs is a mistake, one of the opponent is not); otherwise, the word `RIGHT` in a comment, as in the files of `goproblems.com`. If there is none of that, it says nothing.



Variation 1 (correct) Variation 2 (correct) Variation 3 (wrong) Variation 4

```
\sgfproblem[verdict, statement = false, comments = false,
goban = {scale = 0.5}]{problem}
```

`/pgfgo/sgf problem/variation title =  $\langle text \rangle$` (no default, initially `Variation` or its translation)

Word of the caption of each variation: “Variation 2”.

`/pgfgo/sgf problem/black to play =  $\langle text \rangle$` (no default, initially `Black to play` or its translation)

Text of the statement when Black plays.

`/pgfgo/sgf problem/white to play =  $\langle text \rangle$` (no default, initially `White to play` or its translation)

Text of the statement when White plays.

`/pgfgo/sgf problem/comments =  $\langle comments \text{ options} \rangle$` (default `true`)

Writes the comment at the beginning below the statement and those of each variation below its diagram, like `/pgfgo/sgf diagrams/comments`. Here they are on at the start; they are turned off with `comments = false`. The options are `true`, `false`, `font`, `align`, `number`, `separator`, `position`, `width`, `sep` and `height`, as in `\sgfdiagrams`; `font` and `align` also hold for the comment of the statement.

`/pgfgo/sgf problem/comments/true` (style, no value)

Writes the comments. It is the initial value.

`/pgfgo/sgf problem/comments/false` (style, no value)

Does not write the comments.

`/pgfgo/sgf problem/comments/font =  $\langle$ font commands $\rangle$` (no default, initially that of the caption)
Like `/pgfgo/sgf diagrams/comments/font`.

`/pgfgo/sgf problem/comments/align =  $\langle$ left, right, center or justify $\rangle$` (no default, initially left)
Like `/pgfgo/sgf diagrams/comments/align`.

`/pgfgo/sgf problem/comments/number =  $\langle$ text $\rangle$` (no default, initially `\textbf{\#1}::~`)
Like `/pgfgo/sgf diagrams/comments/number`.

`/pgfgo/sgf problem/comments/separator =  $\langle$ text $\rangle$` (no default, initially one paragraph per comment)
Like `/pgfgo/sgf diagrams/comments/separator`.

`/pgfgo/sgf problem/comments/position =  $\langle$ below, right, left or after $\rangle$` (no default, initially below)
Like `/pgfgo/sgf diagrams/comments/position`.

`/pgfgo/sgf problem/comments/width =  $\langle$ dimension $\rangle$  or auto` (no default, initially auto)
Like `/pgfgo/sgf diagrams/comments/width`.

`/pgfgo/sgf problem/comments/sep =  $\langle$ dimension $\rangle$` (no default, initially 1em)
Like `/pgfgo/sgf diagrams/comments/sep`.

`/pgfgo/sgf problem/comments/height =  $\langle$ dimension $\rangle$  or none` (no default, initially none)
Like `/pgfgo/sgf diagrams/comments/height`.

`/pgfgo/sgf problem/annotations =  $\langle$ boolean $\rangle$` (default true, initially true)
With comments, also writes the name and the annotations of each move (TE, BM, etc.).

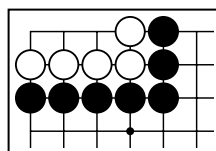
`/pgfgo/sgf problem/chat =  $\langle$ boolean $\rangle$` (default true, initially true)
With comments, like `/pgfgo/sgf comment/chat`: with false, without the chat of KGS and OGS.

`/pgfgo/sgf problem/notes =  $\langle$ boolean $\rangle$` (default true, initially true)
Writes the “N at M” notes of each variation.

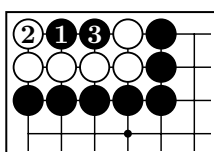
`/pgfgo/sgf problem/caption =  $\langle$ caption options $\rangle$` (no default)
Options of the captions, the notes and the comments.

`/pgfgo/sgf problem/caption/font =  $\langle$ font commands $\rangle$` (no default, initially `\small`)
Font of the captions, the notes and the comments.

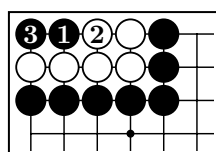
```
\sgfproblem[statement = alone, comments = false, notes = false,
caption = {font = \itshape}]{problem}
```



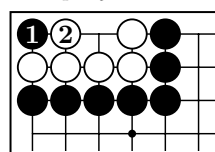
Black to play



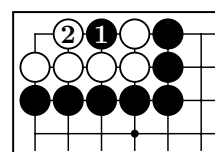
Variation 1



Variation 2



Variation 3



Variation 4

`/pgfgo/sgf problem/caption/min width =  $\langle$ dimension $\rangle$` (no default, initially 8em)
As in `\sgfdiagrams`: with `columns = auto`, the minimum width at which a caption wider than its board is broken (“Problem 3: White to play” below a small corner).

`/pgfgo/sgf problem/goban =  $\langle$ goban options $\rangle$` (no default)

Options of each board, like `/pgfgo/sgf diagrams/goban`.

`/pgfgo/sgf problem/name =  $\langle$ name $\rangle$` (no default)

Like `/pgfgo/sgf diagrams/name`: the statement is called `name`, and each variation `name` followed by its number. With several games (`game`), the game is added: the statement of problem 3 is `name3`, and its variations `name3-1`, `name3-2`...

`/pgfgo/sgf problem/layout =  $\langle$ layout options $\rangle$` (no default)

How the statement and the variations are laid out: options of `/pgfgo/layout` (p. 131) only for `\sgfproblem`.

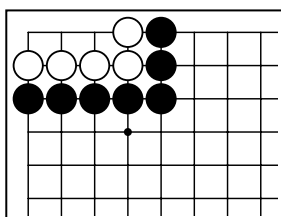
`/pgfgo/sgf problem/view =  $\langle$ boolean $\rangle$` (default `true`, initially `true`)

If the file has a view (VW) at the beginning, every diagram shows that part.

`/pgfgo/sgf problem/partial =  $\langle$ auto, false or corners $\rangle$` (no default, initially `auto`)

With `auto`, if the view of the file is not used, it shows only the part of the board where there are stones, moves, marks, arrows or lines in some line of the problem, like `/pgfgo/goban/partial = auto`. It is the same in every diagram. The lines added around are changed with `goban = {partial margin = ...}`. With `false`, the whole board is seen. With two corners (`partial = A14:G19`), every diagram shows that part, even if the file has a view.

```
\sgfproblem[solutions = false, goban = {partial margin = 3}]{problem}
```



Black to play

Black to kill.

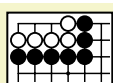
`/pgfgo/sgf problem/game =  $\langle$ numbers $\rangle$` (no default, initially 1)

Like `/pgfgo/sgf/game`: in a collection, the problem that is drawn. Also several, with numbers and ranges like those of the coordinates: `game = 1:20` or `game = {1:4, 9}`. Then all the statements go first, numbered (`title`), and then the solutions of those that have them. The games that the file does not have are left out, with a warning; if none is left, the first one is drawn. That is how a problem sheet is built:

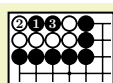
```
\sgfproblem[game = 1:24, to play = black, comments = false,
  layout = {columns = 6, fit}]{collection.sgf}
```

`/pgfgo/sgf problem/links =  $\langle$ boolean $\rangle$` (default `true`, initially `false`)

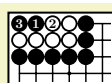
With `hyperref`, the caption of the statement is a link to its first solution, and that of each solution, a link to the statement, also if they are in another `\sgfproblem` (with `solutions = false` and `statement = false`). Two compilations are needed; a link whose target is not there is not set. Without `hyperref`, the package warns.



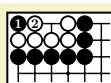
Black to play



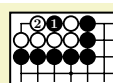
Variation 1



Variation 2



Variation 3



Variation 4

```
\sgfproblem[links, comments = false, notes = false,
  goban = {scale = 0.5}]{problem}
```

11 Layout of the diagrams

Several boards can go in a grid: the goban environments and the diagrams of `\sgfdiagrams` and `\sgfproblem`, inside the `pgfgolayout` environment, or those of a single `\sgfdiagrams` or `\sgfproblem`, with its *key* layout. Without a grid, the diagrams of those commands go in the paragraph, side by side, and move to the next line when they do not fit. With `columns`, they go in a grid of columns of the same width: each row is a separate line (the page can break between rows), the boards of a row are aligned at the top and an incomplete row leaves the last columns empty. With `fit`, the boards are also scaled to fill the width of their column.

```
\begin{pgfgolayout}[\langle layout options \rangle]
  \langle environment contents \rangle
\end{pgfgolayout}
```

Lays out in a grid the goban environments it contains, and the diagrams of the `\sgfdiagrams` and `\sgfproblem` it contains, in the order in which they appear. The options are those of `/pgfgo/layout`; the layout options of the commands inside are not used. There must be no text inside, only the boards and the commands. With `fit`, each goban is drawn several times to measure it: its content must have no effects outside the board (for example, counters that increase).

```
\begin{pgfgolayout}[columns = 4, fit]
  \begin{goban}[5]\stones[b]{C3}\end{goban}
  \begin{goban}[5]\stones[w]{C3}\end{goban}
  \sgfdiagrams[step = 5]{game}
\end{pgfgolayout}
```

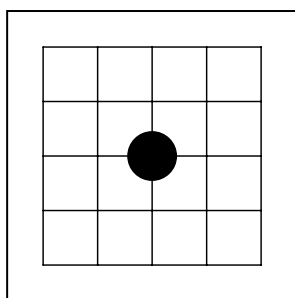
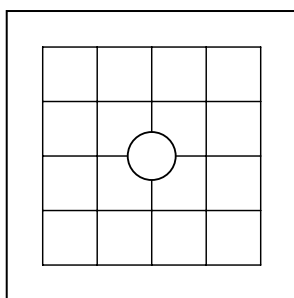
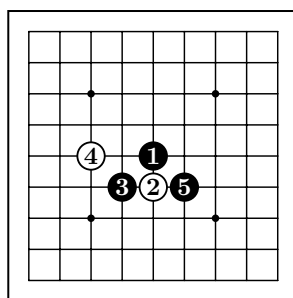
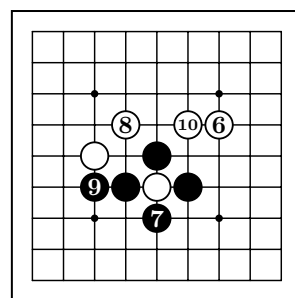


Diagram 1 (1–5)

Diagram 2 (6–10)

```
\sgfdiagrams[step = 2, to = 6, layout = {columns = 3, fit}]{game}
```

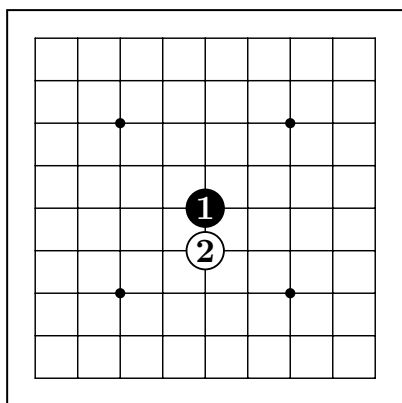
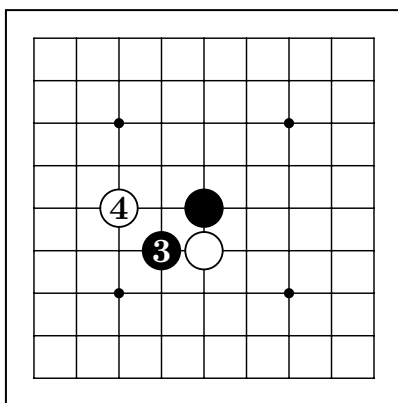
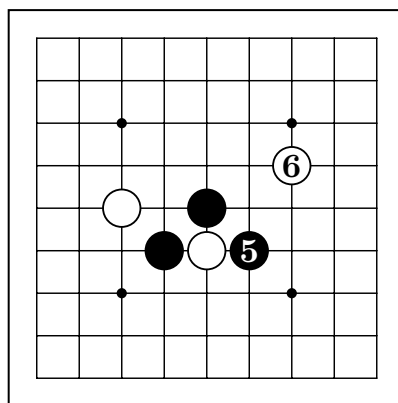


Diagram 1 (1–2)

Diagram 2 (3–4)

Diagram 3 (5–6)

`/pgfgo/layout = \langle layout options \rangle` (style, no default)

Options of the layout for the whole document (for example, `columns = 3, fit`). For a single command, `/pgfgo/sgf diagrams/layout` and `/pgfgo/sgf problem/layout`; also for the whole document, `\pgfgoset{sgf diagrams = {layout = {...}}}`.

`/pgfgo/layout/columns =  $\langle$ number or auto $\rangle$` (no default, initially auto)

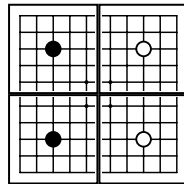
Number of columns of the grid. With `auto`, the diagrams go in the paragraph, without a grid.

`/pgfgo/layout/width =  $\langle$ dimension $\rangle$  or natural` (no default, initially `\linewidth`)

Width of the grid. If it is less than that of the line, the grid is centered. Each column measures what is left after subtracting the `column sep` from `width`, divided among the columns; a narrower board sits in its column according to `halign`, so between two boards there is more space than `column sep`.

With `natural`, the grid fits the diagrams: the columns have the width of the widest diagram (with its caption, which is broken as without a grid, see `caption/min width`) and between two boards of the same width there is exactly `column sep`. The grid measures that times the number of columns (or of diagrams, if there are fewer) and is centered on the line; if it is wider than the line, it goes into the margin. The diagrams are all drawn together at the end (or at the end of each group of `per page` or `per frame`). `fit` does not work with `natural`: the package warns and the boards keep their scale.

```
\begin{pgfgolayout}[columns = 2, width = natural, column sep = 0em, row sep = 0em]
\begin{goban}[partial = A9:E5, scale = 0.5]\stone[b]{C7}\end{goban}
\begin{goban}[partial = E9:I5, scale = 0.5]\stone[w]{G7}\end{goban}
\begin{goban}[partial = A1:E5, scale = 0.5]\stone[b]{C3}\end{goban}
\begin{goban}[partial = E1:I5, scale = 0.5]\stone[w]{G3}\end{goban}
\end{pgfgolayout}
```



`/pgfgo/layout/column sep =  $\langle$ dimension $\rangle$` (no default, initially 1em)

Space between columns.

`/pgfgo/layout/row sep =  $\langle$ skip $\rangle$` (no default, initially `\medskipamount`)

Space between rows.

`/pgfgo/layout/fit =  $\langle$ boolean $\rangle$` (default `true`, initially `false`)

Scales each board so that it fills the width of its column; it replaces the `scale` of the board. The lines do not change width. In `\sgfdiagrams` and `\sgfproblem`, every board uses the scale that fills the column with the first one, so the stones have the same size in all of them. If with that scale a board does not fit in the column (a problem wider than the first one, for example), only that board is reduced until it fits it. Only with `columns`, and not with `width = natural`.

`/pgfgo/layout/per page =  $\langle$ number $\rangle$  or false` (no default, initially 0)

Every $\langle$ number $\rangle$ diagrams, starts a new page (with `\newpage`): with `per page = 1`, one diagram per page. In the grid, a half row is drawn before the break; $\langle$ number $\rangle$ is best a multiple of `columns`. With 0 or `false`, there are no breaks. In `\sgfproblem` with `statement = alone`, the statement counts as a diagram.

```
\sgfdiagrams[step = 50, layout = {columns = 2, fit, per page = 4}]{lee}
```

`/pgfgo/layout/per frame =  $\langle$ number $\rangle$  or false` (default 1, initially 0)

Inside a `beamer frame`, every $\langle$ number $\rangle$ diagrams move to the next slide: with `per frame` alone, one diagram per slide, each with its caption. The diagrams use `\only<+>`, so they add up to the other overlays of the `frame`: after a `\pause`, the first diagram goes on the next slide. Outside `beamer`, the package warns and the `key` is not used. In `beamer`, if there is also `per page`, `per frame` wins.

```
\begin{frame}{The game}
\sgfdiagrams[step = 50, goban = {scale = 0.6}, layout = {per frame}]{lee}
\end{frame}
```

`/pgfgo/layout/halign` = $\langle center, left \text{ or } right \rangle$ (no default, initially `center`)

Where each diagram goes inside its column, with its caption and its notes. The comments are always aligned to the left.

```
\sgfdiagrams[step = 2, to = 6, layout = {columns = 2, width = 0.6\linewidth,
column sep = 3em, row sep = 2em, halign = left}]{game}
```

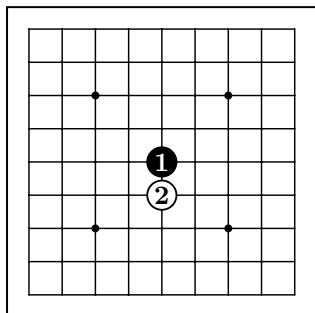


Diagram 1 (1–2)

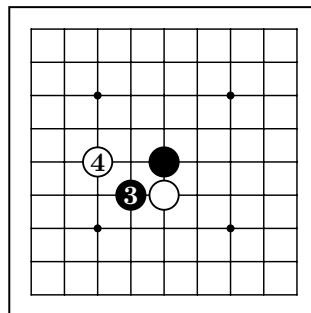


Diagram 2 (3–4)

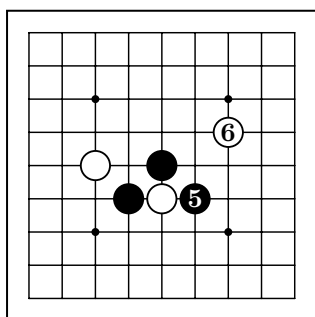
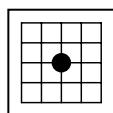
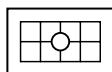
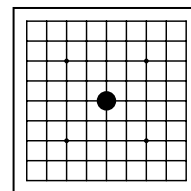


Diagram 3 (5–6)

`/pgfgo/layout/valign` = $\langle top \text{ or } bottom \rangle$ (no default, initially `top`)

How the diagrams of a row of the grid are aligned when their boards have different heights. With `top`, at the top. With `bottom`, the boards end at the same height, and their captions and notes all start on the same line, below. With `columns = auto`, the diagrams go on the baseline of the paragraph, as with `bottom`.

```
\begin{pgfgolayout}[columns = 3, valign = bottom]
\begin{goban}[5, scale = 0.6]\stones[b]{C3}\end{goban}
\begin{goban}[5x3, scale = 0.6]\stones[w]{C2}\end{goban}
\begin{goban}[9, scale = 0.6]\stones[b]{E5}\end{goban}
\end{pgfgolayout}
```



`/pgfgo/layout/align captions` = $\langle boolean \rangle$ (default `true`, initially `false`)

With `valign = top`, the captions, notes and comments of the diagrams of a row start on the same line, below the tallest board; the boards stay aligned at the top. With `valign = bottom` they are already on the same line. The `goban` environments of `pgfgolayout` have no caption: they do not change.

```
\sgfproblem[game = 1:3, solutions = false, partial = auto,
layout = {columns = 3, align captions}]{problems}
```

12 Players and profiles

Each stone, move and mark belongs to one of three players: the first (**black**), the second (**white**) or the neutral one (**neutral**). Their look is not in each command but in the player: the *keys* tree of each one gathers the options of its stones, its moves, its marks and its territory, which in the previous sections appear under **stone**, **move** and **mark**. Here are the players and, afterwards, the profiles.

12.1 Players

The options of the board are changed with `\pgfgoset{goban = {...}}`. Those of the players, with these *keys*:

`/pgfgo/players = <players options>` (no default)

Options of the players: **neutral**, **black** and **white**. Players are not to be confused with profiles: a profile is loaded into the players, and then these *keys* change them.

The options of the players can be set all together with the *key* **all**, or separately. The changes made here do not affect the definition of the profiles. That is, if we load the profile **pgfgo** and change the black player, this does not affect the definition of **pgfgo-black**, but the options of the first player itself.

`/pgfgo/players/all = <all player options>` (no default)

Applies the options to the three players.

`/pgfgo/players/neutral = <neutral player options>` (no default)

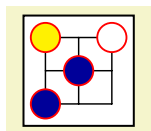
Options of the neutral player (the stones without a player).

`/pgfgo/players/black = <first player options>` (no default)

Options of the first player.

`/pgfgo/players/white = <second player options>` (no default)

Options of the second player.



```
\pgfgoset{players = {
  all = {stone = {line = {color = red}}},
  neutral = {stone = {fill = {color = yellow}}},
  black = {stone = {fill = {color = blue!60!black}}}}

\begin{goban}[3]
\stones[b]{A1, B2}
\stones[w]{C3}
\stone{A3}
\end{goban}
```

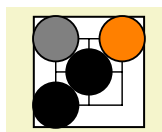
To avoid writing the whole path, there are four commands:

`\pgfgosetplayer{<player>}{<player options>}`

It is the same as `\pgfgoset{players = {<player> = {<player options>}}}`.

`\pgfgosetallplayers{<player options>}`

Applies the options to the three players. It is the same as `\pgfgoset{players = {all = {...}}}`.

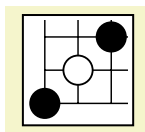


```
\pgfgosetallplayers{stone = {radius = 0.3cm}}
\pgfgosetplayer{white}{stone = {fill = {color = orange}}}

\begin{goban}[3]
\stones[b]{A1, B2}
\stones[w]{C3}
\stone{A3}
\end{goban}
```

\pgfgocopyplayer{*<target player>*}{*<source player>*}

Copies the current options of **source player** into **target player**.

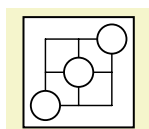


```
\pgfgocopyplayer{neutral}{white}

\begin{goban}[partial = A1:C3]
\stone{B2} % Neutral, but with the look of White.
\stones[b]{A1,C3}
\end{goban}
```

\pgfgocopyplayers{*<neutral source>*}{*<black source>*}{*<white source>*}

Copies, in this order, **neutral source** into the neutral player, **black source** into the black one and **white source** into the white one (with **\pgfgocopyplayer**). Since the copies are made one after the other, it cannot swap players.



```
\pgfgocopyplayers{neutral}{white}{black}
% It does not swap: both are white.

\begin{goban}[3]
\stones[b]{A1, B2}
\stones[w]{C3}
\end{goban}
```

Each player has the same *keys* tree. Each branch is documented with the object it applies to, without the player in the path: for the white player, `/pgfgo/stone/radius` is `/pgfgo/players/white/stone/radius`. The three branches below, although they mention **white**, hold for the three players.

/pgfgo/players/white/stone = *<stone options>* (no default)

Options of its stones and of its prisoners (p. 62).

/pgfgo/players/white/move = *<move options>* (no default)

Options of its moves (p. 73).

/pgfgo/players/white/mark = *<mark options>* (no default)

The style of its marks (p. 78). The shape (**triangle**, **custom**, ...) is not part of the style: it goes in the options of each mark (as in `\marks[w, custom = {?}] {A3}`) or in `/pgfgo/mark`. Inside `\pgfgoset{players = ...}` it gives an error.

The fourth branch, **territory**, is the style of its territory (p. 95).

12.2 Profiles

Profiles are equivalent to TikZ styles. In short, they save a configuration of board and stones with a name and call it whenever wanted.

\pgfgodefinegoban[*<options>*]{*<goban name>*}{*<goban options>*}

Gives a name to a group of **goban** options. It has two options.

/pgfgo/profiles/goban/from = *<goban name>* (no default)

Defines the goban from a copy of another one.

```
\pgfgodefinegoban[from = gobanA]{gobanB}{...}
```

A goban can copy itself to add options, although it is not the best alternative.

```
\pgfgodefinegoban[from = gobanB]{gobanB}{...}
```

/pgfgo/profiles/goban/append

(no value)

Adds options to a goban already defined.

```
\pgfgodefinegoban[append]{gobanB}{...}
```

It is equivalent to:

```
\pgfgodefinegoban[from = gobanB]{gobanB}{...}
```

\pgfgodefineplayer[*<options>*]{*<player name>*}{*<player options>*}

The same, for a player. Its options are those of `\pgfgodefinegoban`.

/pgfgo/profiles/player/from = *<player name>*

(no default)

Defines the player from a copy of another one.

/pgfgo/profiles/player/append

(no value)

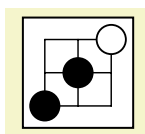
Extends the options of a player already defined. It is the same as `from = <player name>` with its own name.

/pgfgo/profile = *<name>*

(no default)

Applies the profile **name**: the goban **name** and the players **name-neutral**, **name-black** and **name-white**. It is used in `\pgfgoset`, in the options of a **goban** and in those of a **view** (p. 103).

Before applying it, the board and the players go back to the initial style (that of the profile **default**): so a profile does not inherit what it does not define, and a shadow or an image set before is lost. What is not style stays as it was: the size, **partial**, **scale**, where the coordinates go and which they are, **hoshi/at**, the captures and the notes. What comes after the profile is added to it.



```
\pgfgoset{stone = {shadow}}
\begin{goban}[3, profile = default]
\stones[b]{A1, B2}
\stones[w]{C3}
\end{goban}
```

The goban **name** and the players **name-neutral**, **name-black** and **name-white** must exist. For example, `profile = pgfgo` works because there is a goban called **pgfgo**, and players called **pgfgo-neutral**, **pgfgo-black** and **pgfgo-white**.

/pgfgo/goban profile = *<goban name>*

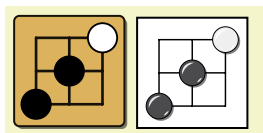
(no default)

Applies only the goban **goban name**, after taking the board back to the initial style.

/pgfgo/players profile = *<name>*

(no default)

Applies only the players **name-neutral**, **name-black** and **name-white**, each one after going back to the initial style.



```
\begin{goban}[3, goban profile = pgfgo]
\stones[b]{A1, B2}
\stones[w]{C3}
\end{goban}

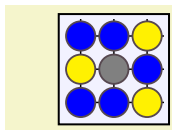
\begin{goban}[3, players profile = pgfgo]
\stones[b]{A1, B2}
\stones[w]{C3}
\end{goban}
```

/pgfgo/players/black/profile = *<player name>*

(no default)

Applies the player **player name** to one of the players (**neutral**, **black** or **white**), after taking it back to the initial style. With **all**, the same one to the three.

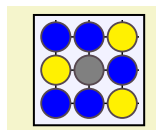
```
\pgfgoset{goban profile = pgfgo, players = {black = {profile = wain-black}}}
```



```
\pgfgdefinegoban{example}{background = {fill = {color = blue!05}}}  
\pgfgdefineplayer[from = default-neutral]{example-neutral}  
{stone = {  
  fill = {color = gray},  
  line = {color = magenta!20!black}}  
}  
  
\pgfgdefineplayer[from = example-neutral]{example-white}  
{stone = {fill = {color = yellow}}}  
  
\pgfgdefineplayer[from = example-neutral]{example-black}  
{stone = {fill = {color = blue}}}  
  
\pgfgoset{profile = example}  
  
\begin{goban}[partial = A1:C3]  
  \moves[numbers = false]{A1:3}  
  \stones[b]{B1,B3,C2}  
  \stones[w]{C1,C3}  
  \stone{B2}  
\end{goban}
```

\pgfgdefineprofile[*<options>*]{*<name>*}{*<profile options>*}

Defines at once a whole profile for profile: the goban name and the players name-neutral, name-black and name-white. It is the same as the example above, in a single command.



```
\pgfgdefineprofile[from = default]{example2}{  
  goban = {background = {fill = {color = blue!05}}},  
  all = {stone = {line = {color = magenta!20!black}}},  
  white = {stone = {fill = {color = yellow}}},  
  black = {stone = {fill = {color = blue}}}  
}  
  
\pgfgoset{profile = example2}  
  
\begin{goban}[partial = A1:C3]  
  \moves[numbers = false]{A1:3}  
  \stones[b]{B1,B3,C2}  
  \stones[w]{C1,C3}  
  \stone{B2}  
\end{goban}
```

/pgfgo/profiles/profile/goban = *<goban options>* (no default)

The options of the goban.

/pgfgo/profiles/profile/all = *<player options>* (no default)

Options of the three players. They go before those of each one.

/pgfgo/profiles/profile/neutral = *<player options>* (no default)

Options of the neutral player.

/pgfgo/profiles/profile/black = *<player options>* (no default)

Options of the black player.

/pgfgo/profiles/profile/white = *<player options>* (no default)

Options of the white player.

/pgfgo/profiles/profile/from = *<profile name>* (no default)

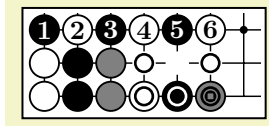
Option of the command: each part starts with that of the profile *<profile name>* (the goban *<profile name>*, the player *<profile name>-neutral*, etc.).

13 Catalog of profiles

The profiles that come with PGFGO, each one with a sample and with its definition, as it is in `pgf-go-profiles.sty`. They are used with `profile` (p. 135).

13.1 default

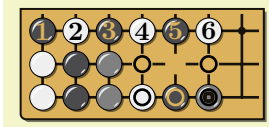
The initial style of the package: the one the board and the players have before a profile is used, and the one they go back to before any other is applied.



```
\pgfgoreset{profile = default}

\begin{goban}[partial = A1:G3]
\moves[from = 1]{A:F 3}
\stones[w]{A1:2}
\stones[b]{B1:2}
\stones[n]{C1:2}
\stone[w, mark = circle]{D1}
\mark[w, circle]{D2}
\stone[b, mark = circle]{E1}
\mark[b, circle]{E2}
\stone[n, mark = circle]{F1}
\mark[n, circle]{F2}
\mark[w, t-square]{F1}
\mark[b, t-circle]{F1}
\mark[n, t-square]{F1}
\end{goban}
```

13.2 pgfgo



```
\pgfgoreset{profile = pgfgo}

\begin{goban}[partial = A1:G3]
\moves[from = 1]{A:F 3}
\stones[w]{A1:2}
\stones[b]{B1:2}
\stones[n]{C1:2}
\stone[w, mark = circle]{D1}
\mark[w, circle]{D2}
\stone[b, mark = circle]{E1}
\mark[b, circle]{E2}
\stone[n, mark = circle]{F1}
\mark[n, circle]{F2}
\mark[w, t-square]{F1}
\mark[b, t-circle]{F1}
\mark[n, t-square]{F1}
\end{goban}
```

Board

```
\pgfdefinegoban{pgfgo}{  
  background = {  
    fill = {  
      color = pgfgo,  
      opacity = 1},  
    line = {  
      color = pgfgo!10!black,  
      opacity = 1},  
    shadow = {true},  
    rounded corners = {  
      inner = 0pt,  
      outer = 2pt  
    }},  
    grid = {  
      line = {  
        color = pgfgo!10!black,  
        width = 1pt}},  
      label = {  
        color = black  
      },  
      arrow = {color = red!80!black},  
      line = {color = red!80!black}  
    }  
  }
```

Players

The colors `wainshade`, `wainwhite` and `wainblack` are those of the profile `wain`, below.

We declare all the players and start by defining the look of the neutral one.

```
\pgfdefineplayer{pgfgo-neutral}{  
  stone = {  
    line = {  
      true,  
      color = black,  
      width = 0.3pt,  
      opacity = 1,  
    },  
    fill = {  
      true,  
      color = gray,  
      opacity = 1,  
    },  
    radius = 0.17cm,  
    scale = 1,  
    shadow = {true, opacity = 0.5, offset = 1pt, angle = -45},  
    shine = {true, angle = 45, color = wainshade}  
  },  
  move = {  
    label = {  
      color = black,  
      format = arabic,  
    }  
  },  
  mark = {  
    line = {  
      color = black,  
      width = 1pt,  
    },  
    fill = {true, color = gray},  
    label = {  
      color = black,  
      format = arabic  
    }  
  }  
}
```

Some things, such as the line width, opacities and so on, I want kept in every stone, so I copy the template of the neutral player.

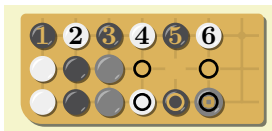
```

\pgfdefineplayer[from = pgfgo-neutral]{pgfgo-white}{
  stone = {
    fill = {color = wainwhite},
    shine = {color = white}
  },
  move = {
    label = {
      color = black
    }
  },
  mark = {
    line = {color = black},
    label = {color = black}
  }
}

\pgfdefineplayer[from = pgfgo-neutral]{pgfgo-black}{
  stone = {
    fill = {
      color = wainblack
    }
  },
  move = {
    label = {
      color = pgfgo
    }
  },
  mark = {
    line = {
      color = pgfgo,
    },
    label = {color = black}
  }
}

```

13.3 pgfgosmooth



```

\pgfgoset{profile = pgfgosmooth}

\begin{goban}[partial = A1:G3]
  \moves[from = 1]{A:F 3}
  \stones[w]{A1:2}
  \stones[b]{B1:2}
  \stones[n]{C1:2}
  \stone[w, mark = circle]{D1}
  \mark[w, circle]{D2}
  \stone[b, mark = circle]{E1}
  \mark[b, circle]{E2}
  \stone[n, mark = circle]{F1}
  \mark[n, circle]{F2}
  \mark[w, t-square]{F1}
  \mark[b, t-circle]{F1}
  \mark[n, t-square]{F1}
\end{goban}

```

Board

```
\definecolor{pgfgsmooth-grid}{HTML}{C7A253}

\pgfgdefinegoban[from = pgfgo]{pgfgsmooth}{
  background = {
    line = false,
    rounded corners = {
      inner = 2pt,
      outer = 8pt
    },
    shadow = {true}},
  grid = {
    line = {color = pgfgsmooth-grid}
  }
}
```

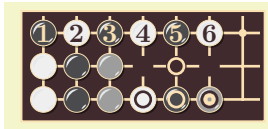
Players

```
\pgfgdefineplayer[from = pgfgo-neutral]{pgfgsmooth-neutral}{
  stone = {
    line = false,
    radius = 0.19cm,
    shine = {opacity = 0.5}}
}
```

```
\pgfgdefineplayer[from = pgfgo-black]{pgfgsmooth-black}{
  stone = {line = false, shine = {opacity = 0.5}}
}
```

```
\pgfgdefineplayer[from = pgfgo-white]{pgfgsmooth-white}{
  stone = {line = false}
}
```

13.4 wain



```
\pgfgset{profile = wain}

\begin{goban}[partial = A1:G3]
  \moves[from = 1]{A:F 3}
  \stones[w]{A1:2}
  \stones[b]{B1:2}
  \stones[n]{C1:2}
  \stone[w, mark = circle]{D1}
  \mark[w, circle]{D2}
  \stone[b, mark = circle]{E1}
  \mark[b, circle]{E2}
  \stone[n, mark = circle]{F1}
  \mark[n, circle]{F2}
  \mark[w, t-square]{F1}
  \mark[b, t-circle]{F1}
  \mark[n, t-square]{F1}
\end{goban}
```

Colors

```
\definecolor{waingoban}{HTML}{402A2D}
\definecolor{waingoban-s}{HTML}{493739}
\definecolor{waingrid}{HTML}{FAE0B2}
\definecolor{wainblack}{HTML}{4B4B4D} %PIEDRA NEGRA
\definecolor{wainshade}{HTML}{E6E7E8}
\definecolor{wainwhite}{HTML}{EFEFEF} %PIEDRA BLANCA
```

Board

```
\pgfdefinegoban[from = default]{wain}{
  background = {
    fill = {
      color = waingoban,
      opacity = 1},
    line = {
      color = waingrid,
      opacity = 1}},
  grid = {
    line = {
      color = waingrid,
      width = 1pt}
    },
  label = {color = waingoban},
  arrow = {color = waingrid},
  line = {color = waingrid}
}
```

Players

```
\pgfdefineplayer[from = default-neutral]{wain-neutral}{
  stone = {
    line = {
      true,
      color = waingrid,
      width = 0.2pt,
      opacity = 1,
    },
    fill = {
      true,
      color = wainblack!50!wainwhite,
      opacity = 1,
    },
    radius = 0.17cm,
    scale = 1,
    shadow = {true, opacity = 0.5, offset = 1pt, angle = -45},
    shine = {true, angle = 45, color = wainshade}
  },
  move = {
    label = {
      color = black,
      format = arabic,
    }
  },
  mark = {
    line = {
      color = waingoban,
      width = 1pt,
    },
    label = {
      color = waingoban,
      format = arabic
    },
  },
}
```

```

\pgfdefineplayer[from = wain-neutral]{wain-white}{
  stone = {
    fill = {
      color = wainwhite
    }
  },
  move = {
    label = {
      color = waingoban
    }
  },
  mark = {
    line = {
      color = waingoban,
    },
    label = {color = waingoban}
  }
}

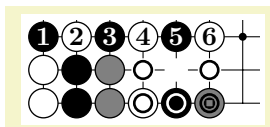
```

```

\pgfdefineplayer[from = wain-neutral]{wain-black}{
  stone = {
    fill = {
      color = wainblack
    }
  },
  move = {
    label = {
      color = waingrid
    }
  },
  mark = {
    line = {
      color = waingrid,
    },
    label = {color = waingrid}
  }
}

```

13.5 simple



```

\pgfset{profile = simple}

\begin{goban}[partial = A1:G3]
  \moves[from = 1]{A:F 3}
  \stones[w]{A1:2}
  \stones[b]{B1:2}
  \stones[n]{C1:2}
  \stone[w, mark = circle]{D1}
  \mark[w, circle]{D2}
  \stone[b, mark = circle]{E1}
  \mark[b, circle]{E2}
  \stone[n, mark = circle]{F1}
  \mark[n, circle]{F2}
  \mark[w, t-square]{F1}
  \mark[b, t-circle]{F1}
  \mark[n, t-square]{F1}
\end{goban}

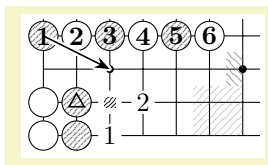
```

Definition

```
\pgfdefineprofile[from = default]{simple}{
  goban = {
    background = {line = false},
    grid = {line = {width = 0.4pt}},
    arrow = {width = 0.8pt},
    line = {width = 0.6pt}
  },
  all = {stone = {line = {width = 0.5pt}}}
}
```

13.6 draft

Black and white, to print or photocopy: thin lines, no border on the background, and lines where there is a fill. The black stones, the filled marks and the territory use `fill/pattern` (p. 62); the territory, with lines in one direction for each player, is seen with `draw = fill` or `draw = region`.



```
\pgfset{profile = draft}

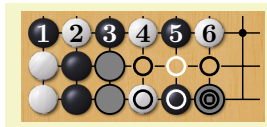
\begin{goban}[partial = A1:G4]
  \moves[from = 1]{A:F 4}
  \stones[w]{A1:2}
  \stones[b]{B1:2}
  \mark[filled square]{C2}
  \mark[b, triangle]{B2}
  \mark[sequence]{C1, D2}
  \territory[b, draw = fill]{F1:2, G1:2}
  \territory[w, draw = fill]{G3}
  \arrow{A4}{C3}
\end{goban}
```

Definition

```
\pgfdefineprofile[from = default]{draft}{
  goban = {
    background = {line = false},
    grid = {line = {width = 0.4pt}},
    arrow = {width = 0.6pt},
    line = {width = 0.4pt}
  },
  all = {
    stone = {line = {width = 0.4pt}},
    mark = {
      line = {width = 0.6pt},
      fill = {
        color = white,
        pattern = {name = {Lines[angle = 45, distance = 1.2pt, line width = 0.3pt]}}
      }
    }
  },
  black = {
    stone = {
      fill = {
        color = white,
        pattern = {name = {Lines[angle = 45, distance = 1.5pt, line width = 0.4pt]}, color = black!50}
      }
    },
    move = {label = {color = black}},
    mark = {line = {color = black}, label = {color = black}},
    territory = {
      fill = {
        color = none,
        pattern = {name = {Lines[angle = 45, distance = 2pt, line width = 0.4pt]}, color = black!50}
      }
    }
  },
  white = {
    territory = {
      fill = {
        color = none,
        pattern = {name = {Lines[angle = -45, distance = 2pt, line width = 0.4pt]}, color = black!50}
      }
    }
  }
}
```

13.7 shell

A wooden board with slate and shell stones, drawn with the images that come with PGFGO (p. 70). Each white stone uses one of five images, turned a little. The shadow needs pdfL^AT_EX or LuaL^AT_EX with PDF output.



```
\pgfset{profile = shell}

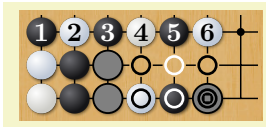
\begin{goban}[partial = A1:G3]
  \moves[from = 1]{A:F 3}
  \stones[w]{A1:2}
  \stones[b]{B1:2}
  \stones[n]{C1:2}
  \stone[w, mark = circle]{D1}
  \mark[w, circle]{D2}
  \stone[b, mark = circle]{E1}
  \mark[b, circle]{E2}
  \stone[n, mark = circle]{F1}
  \mark[n, circle]{F2}
  \mark[w, t-square]{F1}
  \mark[b, t-circle]{F1}
  \mark[n, t-square]{F1}
\end{goban}
```

Definition

```
\pgfdefineprofile[from = default]{shell}{
  goban = {
    background = {
      line = false,
      fill = {image = {file = pgfgo-board.jpg}}
    },
    arrow = {halo = false},
    line = {halo = false}
  },
  all = {
    stone = {
      shadow = {
        true,
        angle = -45,
        offset = 0.75pt,
        scale = 1.2,
        opacity = 0.6,
        image = {file = pgfgo-shadow.png}
      }
    }
  },
  black = {stone = {image = {file = pgfgo-shell-black.png}}},
  white = {
    stone = {
      image = {
        file = {pgfgo-shell-white-1.png, pgfgo-shell-white-2.png, pgfgo-shell-white-3.png,
          pgfgo-shell-white-4.png, pgfgo-shell-white-5.png},
        rotate = -15:15
      }
    }
  }
}
```

13.8 yunzi

Like `shell`, but with matte stones, without stripes, like the Chinese *yunzi*: two images for the white stones, without a turn, because the light of each image comes from one side. It starts from `shell` (`from = shell`): it only changes the images of the stones.



```
\pgfset{profile = yunzi}

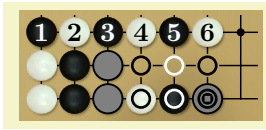
\begin{goban}[partial = A1:G3]
  \moves[from = 1]{A:F 3}
  \stones[w]{A1:2}
  \stones[b]{B1:2}
  \stones[n]{C1:2}
  \stone[w, mark = circle]{D1}
  \mark[w, circle]{D2}
  \stone[b, mark = circle]{E1}
  \mark[b, circle]{E2}
  \stone[n, mark = circle]{F1}
  \mark[n, circle]{F2}
  \mark[w, t-square]{F1}
  \mark[b, t-circle]{F1}
  \mark[n, t-square]{F1}
\end{goban}
```

Definition

```
\pgfdefineprofile[from = shell]{yunzi}{
  black = {stone = {image = {file = pgfgo-yunzi-black.png}}},
  white = {
    stone = {
      image = {
        file = {pgfgo-yunzi-white-1.png, pgfgo-yunzi-white-2.png},
        rotate = 0
      }
    }
  }
}
```

13.9 baduktv

A light board, of plain wood, with plain stones, one image for each color, without a turn; the soft shadow is that of `shell`. Like `shell`, it needs pdfL^AT_EX or LuaL^AT_EX with PDF output for the shadow.



```
\pgfset{profile = baduktv}

\begin{goban}[partial = A1:G3]
  \moves[from = 1]{A:F 3}
  \stones[w]{A1:2}
  \stones[b]{B1:2}
  \stones[n]{C1:2}
  \stone[w, mark = circle]{D1}
  \mark[w, circle]{D2}
  \stone[b, mark = circle]{E1}
  \mark[b, circle]{E2}
  \stone[n, mark = circle]{F1}
  \mark[n, circle]{F2}
  \mark[w, t-square]{F1}
  \mark[b, t-circle]{F1}
  \mark[n, t-square]{F1}
\end{goban}
```

Definition

```
\pgfdefineprofile[from = default]{baduktv}{
  goban = {
    background = {
      line = false,
      fill = {image = {file = pgfgo-baduktv-goban.png}}
    },
    arrow = {halo = false},
    line = {halo = false}
  },
  all = {
    stone = {
      shadow = {
        true,
        angle = -45,
        offset = 0.75pt,
        scale = 1.2,
        opacity = 0.6,
        image = {file = pgfgo-shadow.png}
      }
    }
  },
  black = {stone = {image = {file = pgfgo-baduktv-black.png}}},
  white = {stone = {image = {file = pgfgo-baduktv-white.png}}}
}
```

A What's new

- 0.2026.10.30:

- TikZ patterns in the fills of the stones, the marks, the territory and the background (`fill/pattern`, p. 62).
 - Coordinates on the border in lower case (`label/letter case = lower`, p. 58).
 - The profile `draft`: black and white, with thin lines, no border on the background, and lines where there is a fill.
 - `goban/arrow` and `goban/line`: the style of the arrows and lines is part of the profile of the board (p. 88). Every profile defines it: `pgfgo` and `pgfgosmooth` draw them in `red!80!black`.
 - `scale line width`: `scale` also multiplies the width of the lines.
 - `layout/valign = bottom`: the boards of a row of the grid end at the same height (p. 131).
 - `partial` of `\sgfdiagrams`, `\sgfproblem` and `\sgfanimation` also takes two corners (`partial = K10:T19`): the same part in every board (p. 120).
 - `\pgfgodefinesgf` takes `origin` and `transpose`: the corner where the point `aa` of the file goes and whether the first letter is the row, the eight symmetries of the board (p. 107).
 - `name` in `\sgfdiagrams` and `\sgfproblem`: each diagram is a named board (`var1`, `var2a...`), for `resume` and the views in the text (p. 120).
 - Columns of two letters after Z, for boards of more than 25 columns: `AA5`, `partial = AZ5:BG8` (p. 32).
 - `links` in `\sgfproblem`: with `hyperref`, the statement and its solutions link to each other (p. 127).
 - The option `pages`: one board per page, and `<document>-pages.lua` turns them into SVG, PNG or PDF (p. 31).
 - Sensei’s Library and L19 diagrams (`$$` lines): every SGF command reads them (p. 109). `sgf syntax` and `syntax` in `\pgfgodefinesgf` choose the syntax of the file.
 - The environment `pgfgosgf`: a file written in the document, in any syntax, which the SGF commands read as a file (p. 107).
 - `goban/blank`: intersections that are not part of the board, without lines or star points; they are the `_` of Sensei’s Library diagrams (p. 56).
 - The sizes of the board are in `pt`: they do not change with the size of the font of the document. One character of `mark/sequence` and one of `mark/custom` have the same size.
 - `grid/edge`: the edge of the grid, thicker or in another color.
 - `layout/align` is renamed `layout/halign`, and `layout/align captions` puts the captions of a row on the same line with the boards at the top (p. 131).
 - `notes/font`: the font of the notes.
 - `\sgfproblem` leaves out the games that the file does not have, with a single warning.
 - In Spanish, the colors are singular and the result also agrees with a name: “Gana Negro”, “Juega Blanco”, “Gana Ana por abandono”. In Portuguese, “Vitória de Ana”.
 - `\pgfgostone`: a single stone in the text, without a board, with a mark or a number (`\pgfgostone[b, number = 5]`, p. 106).
 - `\pgfgoview` with several parts (`{F15, G15, C3:E5}`): one view per part, joined with `view/separator` and `view/last` (“and” or its translation, p. 103).
- 0.2026.10.23:
 - Kernel and rules of the game:
 - * The computations (coordinates, numbering of moves, memory, etc.) have been separated from the drawing. They are done by a *kernel*, chosen with the package option `kernel` (p. 30). Besides the kernel written in L3, a kernel written in Lua has been added, which is the default with `LuaLATEX`.
 - * With the Lua kernel, captures have been added (`captures`, p. 92). With `captured marks = remove`, the marks on a captured stone go with it, and `\pgfgoprisoners` counts the prisoners of each player.

- * “N at M” notes have been added for numbered moves on points already played (`notes` and `\pgfgomovenotes`, p. 94). The numbered passes of an SGF are noted as “N: pass” (`notes/pass`).
 - * `\territory`, which marks the territory (with a polygon, or by itself with `auto` and the Lua kernel), and `\pgfgoscore`, which counts the points (p. 95), have been added. With `draw = region`, the territory of each color is a single polygon, and `\pgfgoregion` gives its measures.
- Named boards and views:
- * A goban with `name` keeps everything it draws, and another one starts with that board with `resume` (p. 101). `resume` brings the stones; `numbers`, `marks` and `territory` bring the rest. With `name` and `resume` in the same goban, variations are built.
 - * `\pgfgoview` draws a point or part of a named board in the middle of the text (p. 103), with the options of `view` (`marks`, `arrows`, `lines`, `numbers`, `territory`, `forgotten`, `raise`, `profile`).
 - * A view can come before its goban, like `\ref` (p. 106), and a board can be used in another document with `export` and `\pgfgoimport` (p. 107).
- SGF:
- * With the Lua kernel, `\sgf` reads games in SGF (p. 110): variations (`variation`), marks of the file (`marks`), the numbering of MN and `restart`, and the size of the board taken from the file (`sgf size`).
 - * Collections of games (`game`), the view (VW, `sgf view`), figures (FG) in `\sgfdiagrams`, variations as letters (`variations`), PL and files in Latin-1.
 - * The numbering asked for by PM (`numbers` and `restart = auto`), and the view (VW) when it changes during the game (`sgf view move`; `\sgfdiagrams` and `\sgfanimation` use it by themselves). `step` accepts a list, and `restart = {from =  $\langle n \rangle$ }` numbers each stretch from $\langle n \rangle$.
 - * `variation = {R16, P17, pass}`: the variation can also be given with its moves, from the first one.
 - * Files in GBK, Big5, Shift_JIS, EUC-JP and EUC-KR are read (`CA` or `sgf charset`). If the font lacks the Chinese, Japanese or Korean characters of a text of the file, the package warns.
 - * Names for the files, with `\pgfgodefinessgf` (p. 107): the path, the game and the charset are written only once.
 - * The information of the game (`\sgfinfo`, also of a move with `move`). The names of the nodes and the annotations (TE, BM, GB, etc.) are written with the comments. `\sgfcomment` writes the comment of a move, and `chat = false` removes the chat of the servers (p. 115).
 - * `\sgfdiagrams` draws a game as a series of diagrams, as in books (p. 120). It writes the comments below each diagram, beside it or after all of them (`comments`), and the result, in words, and the score below the last one (`result`). With `variations`, it also draws the variations, each in its own diagram.
 - * `last mark` marks the last move, and `to = last` goes to the end.
 - * `\sgfproblem` has been added, which draws a problem (tsumego): the statement and one diagram for each variation of the solution, with its comments (p. 127). With `game = 1:20`, a problem sheet from a collection; with `verdict`, the caption says whether the variation is right.
 - * With the `animate` package, `\sgfanimation` steps through a game with buttons (p. 117). It builds the animation in layers (`layers`): each stone is drawn only once, and the PDF of a long game takes much less space.
- Drawing:
- * Arrows and lines have been added (`\arrow` and `\line`, p. 88), with `dash`, `shorten`, `bend` and `tips`, and the mark `dim`. `\sgf` also draws AR, LN, SL and DD.
 - * The board draws the star points (`hoshi`, p. 47).
 - * The coordinates of the border can be letters, numbers or kanji on each axis, counted from either side (`columns`, `rows`, `style = japanese`, etc.).
 - * `partial = auto` shows only the part of the board used by what is drawn (p. 56); also in `\sgfdiagrams`, `\sgfanimation` and `\sgfproblem`.

- * Images (PNG, JPG or PDF) on the stones (`stone/image`, p. 70), on the background and the grid of the board, on the star points and on the marks. With several files, each point always uses the same one, and `rotate` turns each stone or tile. Each image is kept only once in the PDF (p. 73).
- * Shadows and shines of your own (`shadow/image` and `shine/image`), and `shadow/scale` and `shadow/color`.
- * Marks of your own, with PGF code or with an image (`\pgfgodefinemark`, p. 82), and `mark/scale`.
- * Numbers in kanji for the moves and the marks (`format = kanji`).
- * Opacity of the stroke and of the fill of the marks, of the numbers of the moves, of the images of the stones and of the star points.
- * The TikZ coordinate system `go: (go cs:C3)` (p. 92).
- Styles and profiles:
 - * Profiles are *keys* (p. 135): `profile`, `goban profile`, `players profile` and `players/<player>/profile`. Before they are applied, the board and the players go back to the initial style. `\pgfgodefineprofile` defines a whole profile at once.
 - * Three new profiles: `simple`, and `shell` and `yunzi`, with images (p. 145): a wooden board with slate and shell stones, or matte ones. The images come with the package, next to its `.sty` files, and are used by their name.
 - * The same rule for the *keys* that turn something on or off (p. 29).
 - * The options of a `goban` accept those of `\pgfgoset`, only for that board, and `\pgfgoset{mark = {...}}` also sets the style of the marks of the three players.
- Layout: `layout` and the `pgfgolayout` environment lay out the diagrams in a grid of columns, with `fit`, `width = natural`, `per page` and `per frame` (p. 131).
- Languages: the texts of the package are translated into ten languages and follow `babel` or `polyglossia`; `lang`, `\pgfgotext`, `\pgfgodefinelanguage` and `\pgfgolanguagealias` (p. 30).
- Coordinates:
 - * Lines in any direction and stairs: `A1 -- E6` (in any direction), `A1 -|- E6` and `A1 |-| E6`.
 - * Rectangles are written `A1:C3`, as in `partial`, and stretches and rectangles are chained with the operators.
 - * `\moves` accepts every form of `\stones` and plays each one in its order (p. 38); `\moves` and `\marks[sequence]` warn if they have no fixed order.
- Behavior changes (the full list, with the old names of the *keys*, is in `CHANGELOG.md`):
 - * `\moves` numbers its moves; `numbers = false` leaves them without a number. `\move`, `\move*` and `\moves*` have been removed.
 - * `\stones` without a player draws neutral stones, like `\stone`.
 - * The only way to remember a board is `name` and `resume`; `goban/remember`, `goban/forget` and `resume` without a name have been removed.
 - * `\pgfgouseprofile` and the other `\pgfgouse...` have been removed; now `\pgfgoset{profile = wain}` is used.
 - * The operator `R` has been removed (`A1 R C3` is `A1:C3`), and `--:`, `:-|:` and `:|-:` are now `--`, `-|-` and `|-|`.
 - * The territory is drawn when the `goban` ends, with the last `\territory` that marked each point.
 - * A move number is never covered by a label of the file (LB) or by `last mark`.
- 0.2025.10.13: Bug fixes. Each `goban` starts at move 1 with Black, `\mark` and `\marks` only exist inside `goban`, and what falls outside a partial board is no longer drawn. The full list is in the file `CHANGELOG.md`.
- 0.2025.03.10:
 - The *keys* `inner` and `outer` have been added for the *rounded corner* of the board.
 - Three new coordinate operators have been added: `*` for every intersection, `L` for the whole column of coordinate `L` and `N` for the whole row of coordinate `N`.
- 0.2025.03.07: First release.