

Catan board diagrams in pure L^AT_EX/TikZ

Contents

1

6	Pieces, ports, and scene elements	5
7	Options and customization	6
8	Gallery	6
8.1	Icons off	6
8.2	Silhouette with all 54 vertex spots	7
8.3	Probability view	8
8.4	An opening position	8
8.5	Highlighting a placement pair	9
8.6	Annotating with low-level coordinates	9
9	Legality and scope	10
10	Roadmap	10
11	License	10

1 Introduction

Board contents are declared in game terms (hexes, tokens, settlements, ports) and the package computes all geometry. The interface is built around three decisions:

1. **Data-driven boards.** A board is a list of $\{q\}\{r\}\{\text{resource}\}\{\text{number}\}$ entries handed to a *renderer callback*. The same board data can therefore be drawn in full color, desaturated, as a bare silhouette, or with selected hexes highlighted, simply by swapping one macro name.
2. **Named coordinates.** Pieces and ports are placed by hex plus a compass name ($\{0\}\{-1\}\{N\}$), never by raw lengths; the board is addressed the way a player points at it.
3. **One scale knob.** Every dimension derives from `\CatanS`, the hex circumradius in centimetres. Redefining it (globally or inside a group) scales the entire figure coherently, line widths, tokens, and text included.

All artwork is original flat geometry; no official Catan art is reproduced.

2 Installation and requirements

Requirements: a standard L^AT_EX distribution with TikZ (any recent T_EX Live or MiK_TE_X). Nothing else.

Manual installation (works everywhere, including Overleaf): place `tikz-catan.sty` in the same directory as the main `.tex` file (on Overleaf, the project root) and load it with `\usepackage{tikz-catan}`.

T_EX tree installation: place the file in `TEXMFHOME/tex/latex/tikz-catan/` (`kpsewhich -var-value TEXMFHOME` locates the directory).

3 Quick start

A complete document:

```
\documentclass[tikz,border=8pt]{standalone}
\usepackage{tikz-catan}
\begin{document}
\begin{tikzpicture}
  \catanframe % sea frame
  \standardboard{\hexfull} % 19 hexes, full color
  \standardports{\port} % nine ports
  \robber{0}{0} % robber on the desert
```

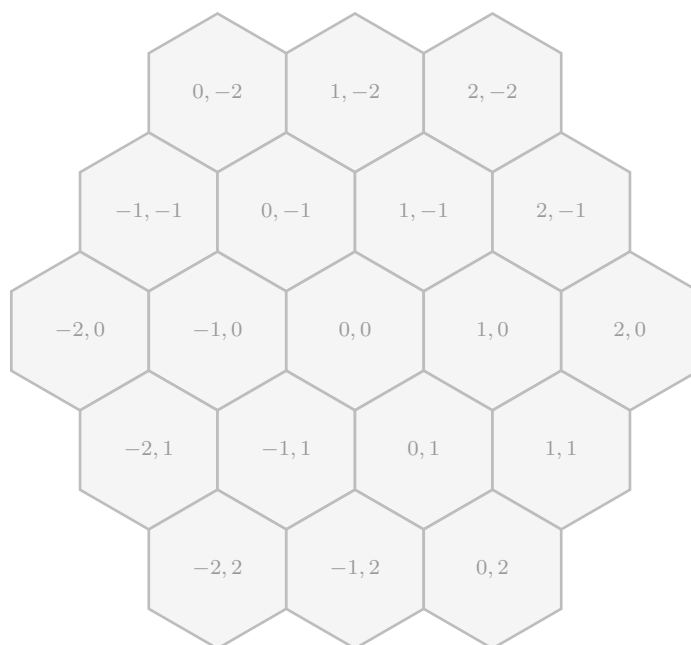
```
\end{tikzpicture}
\end{document}
```

That produces the board on the title page. `\standardboard` and `\standardports` are a built-in *legal* example layout: correct tile and token multisets, no two red (6/8) tokens adjacent, a token-free desert, and a standard-style port arrangement. Section 5 shows how to supply a different layout.

4 The coordinate system

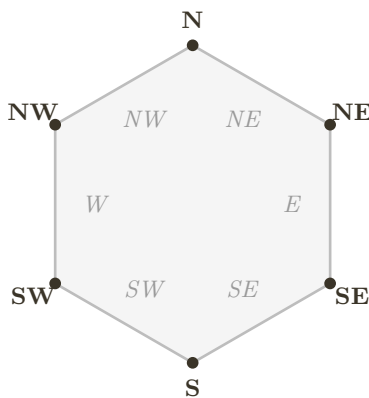
4.1 Hexes: axial coordinates

Hexes are pointy-top and addressed by axial coordinates (q, r) : q grows to the right, r grows downward with a half-step rightward slant. The 19 board hexes are all (q, r) with $|q| \leq 2$, $|r| \leq 2$, $|q + r| \leq 2$; the desert of the built-in layout sits at $(0, 0)$.



4.2 Corners and sides: compass names

Each hex has six *corners* (vertices, where settlements go) and six *sides* (edges, where roads go), named by compass direction:



Corner names (bold, outside) address vertices; side names (italic, inside) address edges. A side’s two endpoints are its neighbouring corners: side E runs from corner NE to corner SE — so a road on side E connects to a settlement on either of those corners. Every vertex of the board belongs to up to three hexes and can be addressed through any of them; $\{0\}\{-1\}\{N\}$ and $\{0\}\{-2\}\{S\}$ are the same physical spot.

5 Drawing boards

5.1 Renderers

A board renderer is any macro taking four arguments $\{q\}\{r\}\{resource\}\{number\}$, where **resource** is one of wood, brick, sheep, wheat, ore, desert, and **number** is the token value (0 = no token, used for the desert). The package ships five renderers:

`\hexfull{q}{r}{res}{n}` — colored hex, resource glyph (if icons are on), number token with probability pips, 6 and 8 in red.

`\hexdim...` — desaturated gray hex with a muted token; the backdrop for highlight figures.

`\hexshape...` — bare silhouette (ignores resource and number).

`\hexprob...` — neutral cream fill with full-color tokens; a “probability view” that isolates the number layer.

`\hexauto...` — selector-aware: renders `\hexfull` when the hex matches a highlight declaration (Section 5.3), `\hexdim` otherwise.

Drawing the built-in layout means passing a renderer to `\standardboard: \standardboard{\hexdim}`. A custom board is the same call list with different data:

```
\newcommand\myboard[1]{%
  #1{0}\{-2\}{wood}\{11\}#1{1}\{-2\}{ore}\{3\}% ... one entry per hex
}
\myboard{\hexfull}
```

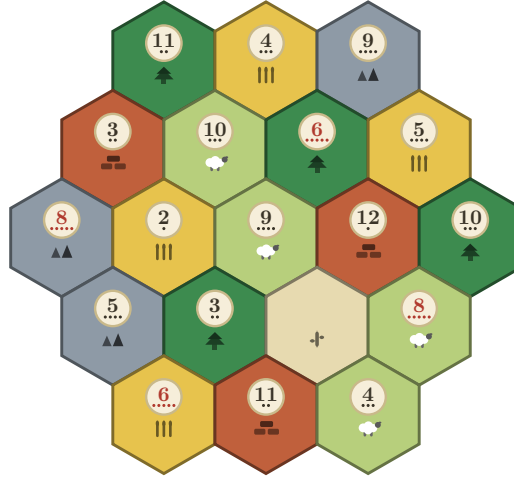
Renderers are ordinary macros, so custom ones are one definition away; the coordinate map in Section 4 is drawn by a four-line callback that combines `\hexshape` with a text node.

`\hexdots{q}{r}{.}{.}` is a sixth, additive callback: it marks the hex’s six corners with dots. Used after a shape pass (`\standardboard{\hexshape}\standardboard{\hexdots}`), it displays all 54 settlement spots.

5.2 Row input

Nineteen coordinate tuples are tedious to write by hand. `\catanrows` accepts a board the way it is read from the table (five rows, top to bottom, each row left to right) and assigns the axial coordinates automatically:

```
\catanrows{\hexfull}{%
  wood 11, wheat 4, ore 9;
  brick 3, sheep 10, wood 6, wheat 5;
  ore 8, wheat 2, sheep 9, brick 12, wood 10;
  ore 5, wood 3, desert, sheep 8;
  wheat 6, brick 11, sheep 4}
```



Format: exactly five rows of 3, 4, 5, 4, 3 entries, separated by four semicolons (no trailing semicolon); entries are comma-separated “resource number” pairs; the desert may be written without a number. The first argument is the renderer, exactly as for `\standardboard`.

5.3 Semantic highlighting

Highlight declarations mark what `\hexauto` (and `\portauto`, below) should render in full color. Three selectors exist, and a hex lights up if it matches *any* of them (union semantics):

`\hlhex{q}{r}` — one hex by coordinate.

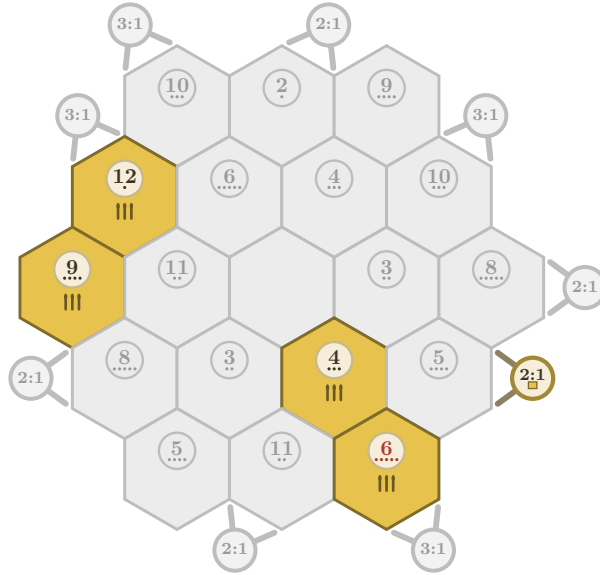
`\hlres{resource}` — every hex of a resource.

`\hlnum{n}` — every hex with number token n .

Declarations are local to the current `tikzpicture`. Intersections (e.g. ore hexes *on* 6 or 8) are expressed with explicit `\hlhex` coordinates.

`\portauto{q}{r}{side}{type}` extends selection to harbors: a resource port renders in full color when its resource is declared via `\hlres`, and desaturated (`\portdim`) otherwise. A board’s “wheat economy”, every wheat hex plus the wheat port, is then three lines:

```
\hlres{wheat}
\standardboard{\hexauto}
\standardports{\portauto}
```



6 Pieces, ports, and scene elements

`\settlement{q}{r}{corner}{color}`, `\city{q}{r}{corner}{color}` — a house / a larger house with tower at the named vertex. Player colors `pred`, `pblue`, `porange`, `pwhite` are predefined; any `xcolor` expression works.

`\road{q}{r}{side}{color}` — a road along the named edge, drawn with a darker outline for contrast on any terrain.

`\robber{q}{r}` — the robber figure, offset beside the token position.

`\port{q}{r}{side}{type}` — a harbor badge off the named *coastal* side of a land hex, with piers to the two vertices it serves. `type` is `generic` (3:1) or a resource name (2:1 with a colored ring and swatch).

`\portdim...` / `\portauto...` — desaturated port / selector-aware port (Section 5.3); same arguments as `\port`.

`\standardports{<port renderer>}` — the nine ports of the built-in layout (four 3:1, five 2:1) through a callback: `\standardports{\port}`, `\standardports{\portauto}`, etc.

`\catanframe` — the hexagonal sea backdrop; drawn first.

`\catanlegend{y}` — a resource color legend centered at height `y` (in cm; -6.6 sits below a default-scale framed board).

`\vertexXY{q}{r}{corner}` / `\hexXY{q}{r}` — low-level: set `\vx`, `\vy` / `\hx`, `\hy` to the point's coordinates, for annotations such as distance paths (see the last gallery figure).

7 Options and customization

`\CatanS` — hex circumradius in cm (default 1.05). Redefinable globally or locally: `{\renewcommand\CatanS{0.8} \begin`

`\cataniconsfalse` / `\cataniconstrue` — switch the resource glyph layer off/on. Icons help when introducing the game; on busy boards (pieces, highlights, robber) turning them off reduces visual noise and lets color alone carry the resource.

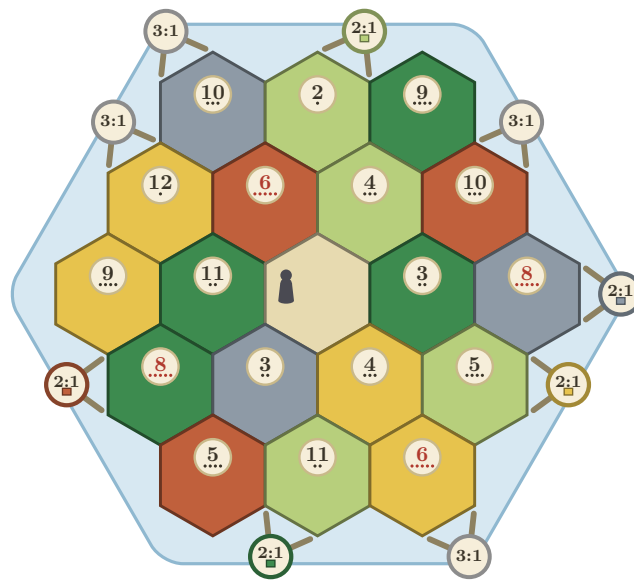
Palette — all colors are named `xcolor` definitions and can be redefined before the figure: `reswood`, `resbrick`, `ressheep`, `reswheat`, `resore`, `resdesert`; `seafill`, `seaborder`; `tokenfill`, `tokenborder`, `tokenred`; `dimfill`, `dimborder`, `dimtoken`; `legendtext`; `robbergray`; and the player colors `pred`, `pblue`, `porange`, `pwhite`.

8 Gallery

Each figure states the code that produces it (frame, board, ports as in the quick start are abbreviated to “...” where repeated).

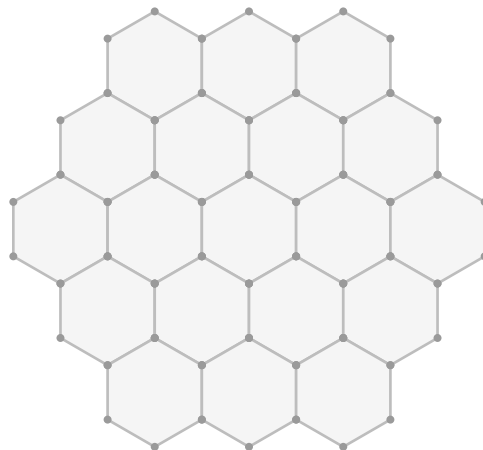
8.1 Icons off

```
\cataniconsfalse \catanframe \standardboard{\hexfull} \standardports{\port}
```



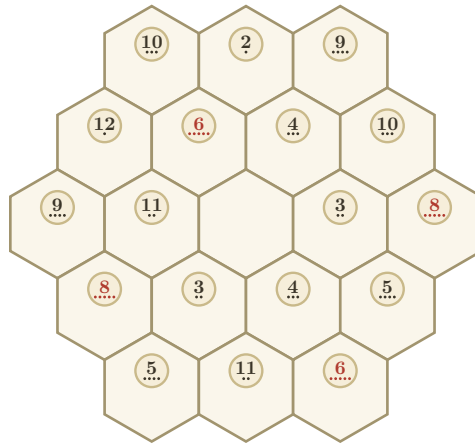
8.2 Silhouette with all 54 vertex spots

```
\standardboard{\hexshape} \standardboard{\hexdots}
```



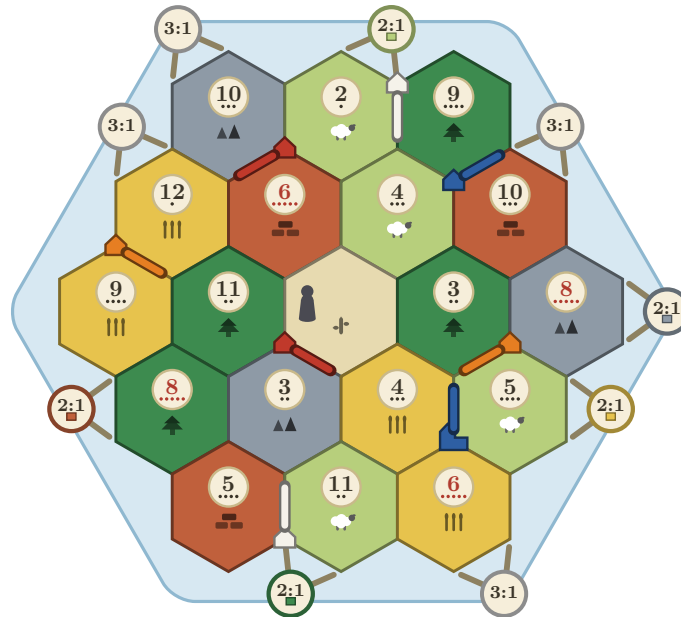
8.3 Probability view

`\standardboard{\hexprob}`



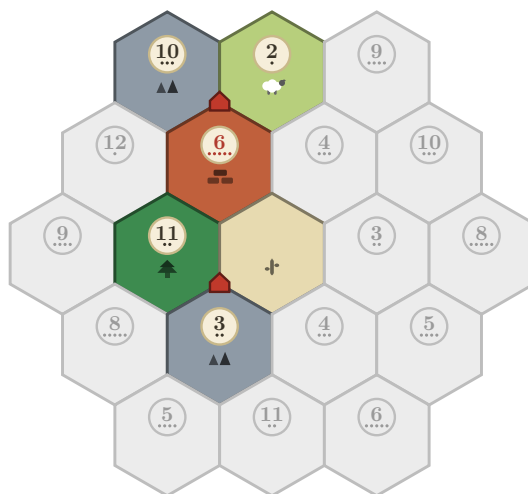
8.4 An opening position

Settlements, a city, roads, and the robber; the placement respects the distance rule.



8.5 Highlighting a placement pair

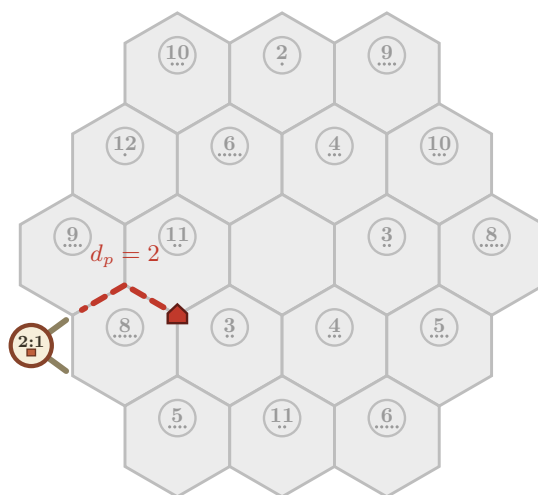
```
\hlhex{0}{-1}\hlhex{0}{-2}\hlhex{1}{-2}
\hlhex{-1}{1}\hlhex{-1}{0}\hlhex{0}{0}
\standardboard{\hexauto}
\settlement{0}{-1}{N}{pred} \settlement{-1}{1}{N}{pred}
```



8.6 Annotating with low-level coordinates

A settlement two roads from a brick port, with the path and distance labelled via `\vertexXY`:

```
\standardboard{\hexdim}
\port{-2}{1}{W}{brick} \settlement{-2}{1}{NE}{pred}
\vertexXY{-2}{1}{NE}\coordinate (pa) at (\vx,\vy);
\vertexXY{-2}{1}{N} \coordinate (pb) at (\vx,\vy);
\vertexXY{-2}{1}{NW}\coordinate (pc) at (\vx,\vy);
\draw[pred,line width=2.2pt,dash pattern=on 5pt off 3.5pt,
      line cap=round,shorten <=6pt,shorten >=4pt] (pa)--(pb)--(pc);
```



9 Legality and scope

The built-in layout guarantees: the base-game tile multiset (4 wood, 4 wheat, 4 sheep, 3 brick, 3 ore, 1 desert), the token multiset (one 2 and 12, two each of 3–6 and 8–11), no two red tokens adjacent, and a token-free desert. It is a *representative* legal layout, not the official beginner setup. The package draws the data it is given; user-supplied boards and placements are not validated.

10 Roadmap

Planned for future versions: the official beginner layout as a second built-in; a per-vertex production heatmap renderer; a longest-road trace helper; a 5–6 player frame; a game-state layer so pieces, like hexes, become data that renderers and selectors can act on. Issues and contributions are welcome at the repository.

11 License

MIT License, © 2026 Teodor Filip Leahu; see the `LICENSE` file in the repository.