

InZeDoku

The player's manual

Version 1.33 — 1 octobre 2026

Everything the game can do, mode by mode, setting by setting.

InZeMobile

TABLE DES MATIÈRES

Foreword.....	3
Partie 1 — Getting to grips with the game	
Chapter I — Discovering InZeDoku	7
Chapter II — The home screen	10
Chapter III — Your first grid	13
Partie 2 — The three basic modes	
Chapter IV — Sudoku	19
Chapter V — Calcudoku	22
Chapter VI — Colodoku.....	25
Partie 3 — The boards and what goes on them	
Chapter VII — The boards.....	31
Chapter VIII — The Calcudoku that teaches counting.....	36
Chapter IX — The thematic sets	44
Partie 4 — Playing well	
Chapter X — Difficulty	49
Chapter XI — The keypad aids.....	51
Chapter XII — The physical keyboard.....	56
Chapter XIII — Timer, games in progress and game codes.....	59
Partie 5 — Settings, accessibility and privacy	
Chapter XIV — The settings.....	69
Chapter XV — Accessibility.....	71
Chapter XVI — Privacy.....	73
Chapter XVII — Free, subscription, lifetime	75
Chapter XVIII — The web demo.....	78
Appendices	
Appendix A — The game code, letter by letter	81
Appendix B — Every key on the keyboard	86
Appendix C — Sizes, modes and limits	89
Resources	92
Lexique.....	94
Table des figures	
Index thématique	100
Index alphabétique	102

Foreword

This manual describes **InZeDoku** in full : the three basic modes, the sixteen thematic sets, the five board shapes, the generation options, the playing aids, the settings and the two versions of the application. It is written to be read in order by anyone discovering the game, and dipped into by anyone who already knows it.

The screenshots are not mock-ups : they were taken in the application itself, grid by grid, screen by screen. What you read there is exactly what you will see.

Every grid shown carries its **game code** — the short string displayed under the buttons. You can type any of these codes into the game to find the grid of the illustration again, cell for cell. It is also a way of checking, by playing, what this manual says.

One clarification so that nothing comes as a surprise : the codes **quoted in the text** are given in their shortest form — `G5CS62-b-FG` where the screen shows `G5CS62-b-FG` . The prefix names the generator, the last two letters form a check key, and neither is needed to find the grid again. Appendix A says why.

The vocabulary of the game is gathered in the **glossary**, at the end of the book. The terms listed there are set in small capitals at their first appearance in each chapter.

Partie 1

Getting to grips with the game

Chapter I — Discovering InZeDoku

An ordinary sudoku fits in one sentence : nine rows, nine columns, nine blocks, and never the same digit twice in any of them. The sentence is so short that one forgets what it really contains. It speaks neither of digits, nor of squares, nor of the number nine : it speaks of **groups of cells that must all differ**. The rest — the shape of the board, what one writes in a cell, the number of symbols — is scenery.

InZeDoku is built on that remark. A single engine solves and builds every grid here, and it knows nothing of a row or a column : you declare groups to it, and it takes care of the rest. Changing the shape of the board means declaring other groups to it ; changing mode means changing what a cell contains. That is why the game can offer a hexagon, a triangle or a completely irregular tiling without being three different games.

1.1 — What you have in your hands

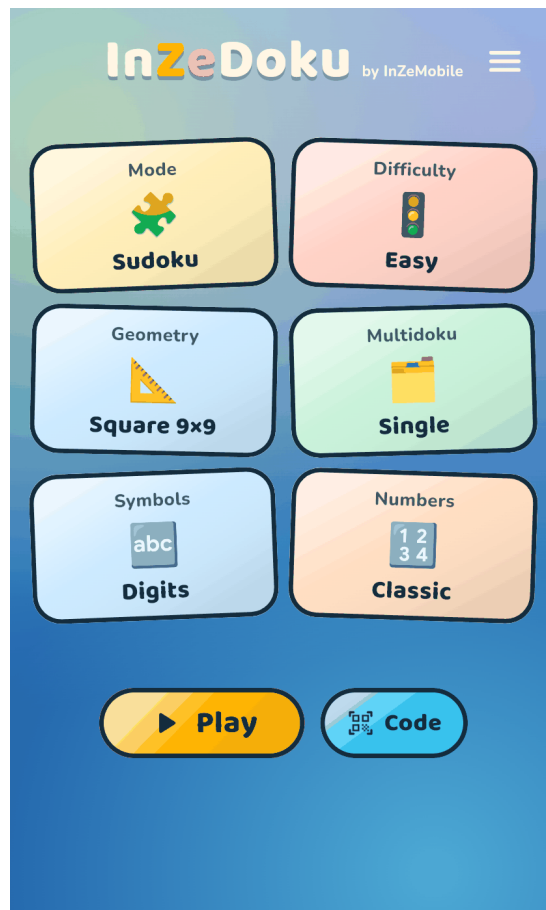


Figure 1.1 — Home : five tiles showing the current configuration, and one button to play

The game comes in three basic modes, detailed by chapter IV and the two that follow :

- **Sudoku**, the classic, where a cell carries a digit or a letter ;
- **Calcudoku**, where the grid is cut into *CAGE*¹s displaying a calculation to be found ;

- **Colodoku**, the signature mode, where a cell carries a **colour** and the only operation is mixing.

To those three are added sixteen **thematic sets** — chemistry, physics, biology, astronomy, characters, vehicles, shapes — where a cage is no longer a sum but a **recipe** : place the right pieces and they compose the thing asked for. They share the mechanics of Calcudoku and are chosen from the same menu.

These modes are played on five board shapes : the square from 2 to 26 a side, the hexagon, the triangle, the mosaic that blends the two, and the Voronoi tiling, irregular and different in every game. Not every combination is open — some would make no sense, and chapter VII explains which — but the vast majority are.

1.2 — Every grid is built on the device

No grid is downloaded. Nor are they stored : they are **built the moment you press Play**, from a number called the SEED, by a generator that proves the solution is unique before handing back control. It is that proof which guarantees a grid can be deduced — that there is never any guessing.

Two consequences matter for what follows.

First, the same seed always gives the same grid, on any device. That is what makes a game **shareable in a short string** : the GAME CODE carries the mode, the geometry, the size, the level, the options and the seed, and nothing else is needed to rebuild the grid identically (chapter XIII and appendix A).

Second, building takes a little time, and all the more as the grid is large or the level demanding. A square of nine is instantaneous ; a 20×20, a Voronoi or a Mystery Calcudoku take a few seconds. A small window announces it, with a button to interrupt.

1.3 — Offline, no account, no collection

InZeDoku asks for **no network permission**. This is not an intention but a property checked at every release : the published package is inspected for the absence of any internet access permission. There is therefore no ad network, no audience measurement tool, no crash report, no account, no synchronisation.

Your games stay on the device. Chapter XVI explains why it is built that way, and chapter XVII how a game that talks to nobody is funded all the same.

1.4 — How to read this manual

The three chapters of this first part are enough to play : they present the home screen, then the playing screen. The following parts can be read in order or separately.

Part 4 deserves a word of its own. It gathers the settings that make Calcudoku a **tool for learning arithmetic** : the choice of operations, the decoupling of the numbers from the size of the board, the result ceiling, the two-cell cages, the fractions and the decimal notation. A child can work on multiplication tables there on a grid five cells a side, or on fractions on a grid of six, without the difficulty of the sudoku having anything to do with that of the arithmetic.

1. Terms in small capitals refer to the glossary at the end of the book, where the full definition and reference context are given.

Finally, three appendices give the tables : the game code character by character, every key on the keyboard, and the limits of each combination of mode, geometry and size.

Chapter II — The home screen

Home asks no questions : it **shows the game you will get** if you press Play. Every setting is a coloured tile carrying its name, its emoji and its current value ; a tap opens the small window where that value is changed. So you can set off without touching anything, or set everything and never have to wonder what is selected.

The chosen configuration is **remembered** : it survives coming back from a grid as well as restarting the application.

2.1 — The setting tiles

The tiles are not always the same : **only those that make sense** for the current mode appear. The table below gives them all.

Tile	What it sets	When it appears
 Mode	Sudoku, Calcudoku, Colodoku, thematic sets	always
 Difficulty	Easy, Medium, Hard, Expert	always
 Geometry	board shape and size	except entity sets, whose 6×6 is fixed
 Multidoku	1 to 5 overlapping grids	square, and hexagon of radius ≤ 3
 Operations	cage operations, Mystery, cage settings	Calcudoku only
 Numbers	what the symbols are worth : integers, negatives, fractions	Sudoku and Calcudoku

When the number of tiles is odd, the last one takes the full width. That is the only layout effect to know about, and it avoids a hole on the right.

The mode picker

Twenty entries in a list would be unreadable. The picker therefore arranges them in three **families**, in a column on the left : **Classics** (Sudoku, Calcudoku, Colodoku), **Science** (Chemodoku, Phydoku, Biodoku, Astrodoiku) and **Kids** (the Pictoku, Whodoku, Geodoku). The window opens straight away on the family of the current mode, and the right-hand panel lists its members.

Colodoku has only one entry, but it carries a switch : **CMY** or **RGB**. The black or white dot shows the target of the mix — that is the subject of chapter VI. The switch does not select the mode and does not close the window : it sets, and you watch the setting move. It is the mode row itself that chooses and closes.

The geometry picker

The shape is chosen at the top — square, hexagonal, triangular, mosaic, Voronoi. A shape the current mode does not accept stays visible but greyed out: you see what exists, you do not wonder where it went.

Below come the settings specific to the chosen shape: a size slider for the square, a choice of radius for the hexagon, of side for the triangle and the mosaic, the number of symbols and the **outline** for the Voronoi, and for the mosaic the choice between a **Regular** layout and a **Surprise** layout. Part 3 takes them one by one.

The difficulty picker

Four levels: **Easy**, **Medium**, **Hard**, **Expert**. They are not decorated with --, -, +, ++ symbols: the names are enough, and a coloured dot tells them apart at a glance. What the level really changes is explained in chapter X — it is not only the number of given cells.

2.2 — Resuming a game

If games are in progress, a **Resume** card is added above the tiles. It leads to the list of saved games — up to twenty — where each shows its mode, its configuration, its timer and its progress. Chapter XIII details this list, including the two buttons that let a game **travel** from one device to another.

2.3 — The hamburger menu

The ☰ circle at the top right opens a drop-down menu. It contains nothing concerning the grid to come — everything about that is on the tiles.

- ⚙️ **Settings** — playing and display preferences (chapter XIV).
- ⓘ **About** — the version, the publisher's name, the licences.
- 📖 **Game guide** — the built-in presentation, in sixteen sections.
- ✉️ **Your feedback** — prepares a message; the application sends nothing by itself.
- 🌐 **Language** — the entry carries the **flag of the active language**, or a globe if the application follows the phone.

Changing language

The **Language** entry opens a window of pills in two columns: each language is written **in its own language**, with its flag — English, Français, Español, Português, Italiano, Deutsch, Nederlands. A full-width pill, marked with a globe, hands control back to the phone.

The flags are **drawn by the application**, not borrowed from the system font: many Android devices have no flag emoji font at all and would show two small letters instead.

The choice of language is here, and not in the Settings, because it is the first setting one needs — and because, as long as it stays hidden, the application silently follows the system and nothing tells the player that six other languages exist.

2.4 — Playing

The **Play** button, in yellow, launches the building of the grid. A small window announces the work in progress and offers to interrupt it — useful on large boards, where a few seconds can go by.

To its right, a second pill, in sky blue : **Code**. That is where a game from elsewhere comes in — you **type** a code there, you **paste** a string, or you **scan** a QR (chapter XIII). It is there and not in the menu because it belongs to the same family as Play : it is a **departure**, not a preference. Resuming a game you already have is something else again, and that is done from the **Resume** card.

The two pills have exactly the same height, and Play tightens up when the screen is narrow ; the circle, for its part, never shrinks below the size a finger aims at without hesitating.

If the chosen configuration goes beyond what the free version allows, a message says so **before** the game, naming every reason. Nothing is blocked for all that : the grid opens and plays. Chapter XVII explains exactly what changes.

Chapter III — Your first grid

The playing screen fits in three bands : the controls at the top, the board in the middle, the keypad at the bottom. There is no Android-style title bar — space is too precious for a strip that merely repeats a name.

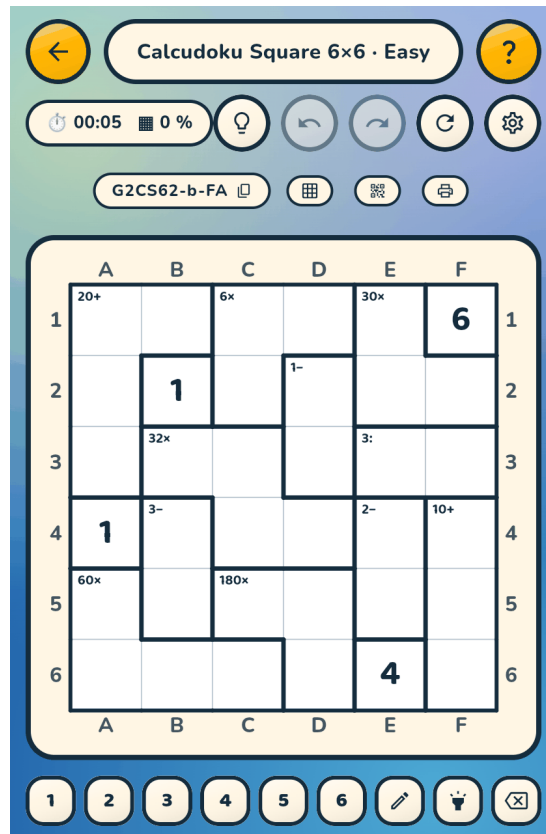


Figure 3.1 — The playing screen, here a 6×6 Calcudoku at Medium level

3.1 — The top band

The first row carries, from left to right, the yellow ← circle that returns to home, a cream pill recalling **what you are playing** — mode, geometry, size, level — and the yellow ? circle, which opens the *Rules of this grid* sheet.

That sheet is worth opening at least once per new configuration. It is not a fixed text : it is **assembled** from the options actually in force. A Voronoi game reads the neighbourhood rule there, a game in fractions reads how the targets are written, a game in unusual numbers reads which values are in play. None of that can be guessed from the board, and that is precisely why it is written down.

The second row gathers the timer, the progress and the tools :

Item	What it does
🕒 00:05 · 🧩 0 %	the time elapsed and the share of cells to fill already filled

Item	What it does
↶ Undo	undoes the last move, back to the very first
↷ Redo	redoes what has just been undone
↻ Restart	empties the grid and resets the timer
⚙ Settings	the same settings as from home, without leaving the game
👁 Preview	opens the solution read-only (chapter XI)

Three of these buttons are **optional**: Undo and Redo, the Preview, and the two keypad tools. Each is switched on or off in the Settings, and a switched-off button disappears from the row. The preview in particular is **off by default** — you have to go and fetch it to have it at hand.

The percentage counts **only the cells you have to fill**. The given cells say nothing about where you stand: counting them would give a flattering and useless figure.

The timer runs as long as you are on the board. It **pauses** as soon as you leave the screen or the application goes into the background, and the game is saved at that moment.

3.2 — The code pill

Below the tools, a small cream pill shows the **game code** — `G5CS62-b-FG` in the illustration. A tap copies it to the clipboard.

To its right, a second button, marked with a grid, copies **the whole game**: the code, then what you have written, cell by cell, candidates included. That is what it takes to pick up exactly where you are on another device; chapter XIII details the two formats.

Two buttons close the row: one shows the game as a **QR**, to be photographed from the other device, the other **prints** it. The four gestures go together — they all take the grid out of here, and it is again chapter XIII that takes them one by one.

3.3 — The board

The board sits on a cream panel outlined in ink. Tapping a cell selects it; the cells **linked** to it light up — its row, its column, its region, its neighbours, depending on the geometry. That is the surest way of understanding the rules of a board you are discovering: instead of reading what a row is on a triangle, you touch a cell and you see it.

On a **single-grid square**, the board also carries its **coordinates**: the column letters above and below, the row digits on either side, as on a chessboard. `D1` is the fourth column, first row.

They serve as soon as one has to **say where**: showing a cell to someone sitting opposite, following a correction, or reading the application's own answer, which names cells when it replies to you. Their size is taken from that of the cells — a small square full-screen shows them large, a 26×26 tightens them — and they zoom with the board, since they are part of it.

Elsewhere there are none, and that is no oversight : on a hexagon, a triangle, a mosaic or a Voronoi, there is **neither row nor column** to name ; on a Multidoku, the grids overlap and one and the same **D3** would designate two cells at once. A cell is then designated by its **rank**.

Two-finger zoom works everywhere, on every size. It changes nothing until you have pinched : at scale 1 the board is exactly fitted to its frame and does not slide. But a small square does not always fit comfortably on a narrow phone, and there are grids — the colours, the pictograms, the chemical formulas — that one wants to look at closely.

3.4 — The keypad at the bottom

The keypad fits on a **single line**. The keys are calibrated to the available width, and if even the smallest legible size does not fit, the line scrolls horizontally rather than wrapping.

In the screenshot, nine value keys, then three tools : the **pencil** () , the **torch** () and the **eraser** () . The first two switch the keypad into another mode ; chapter XI is devoted to them. The eraser always empties the selected cell.

Beyond nine symbols, the keys shrink a little and the line scrolls. In Mystery Calcudoku, an **extra row** appears above the digits : the operations to be placed on the cages.

3.5 — Winning

The game is won when every constraint is satisfied — not when the grid resembles a solution stored somewhere. The nuance has no effect on an ordinary grid, where the solution is unique, but it counts as soon as a value can be written in two ways : **1/2** and **4/8** are the same value, and either one is right.

The victory dialogue shows the time and offers three ways out : **Grid**, which simply closes and lets you look at what you have just finished, **Play again**, which goes straight on to a fresh grid with the same settings without passing through the home screen, and **Home**, which goes back to the menu.

The grid seen again can no longer be changed — with one exception, and it matters : **Undo** still works. A finished game is not a dead game, and going back over your last move to see what you had written before is precisely the gesture one wants at that point. The game does not merely lift the lock : it **recomputes** the victory. Undo a move and the grid becomes incomplete again, hence playable, the timer restarts and the game takes its place back in the list ; redo it and it is won again, and the dialogue congratulates you afresh. It stays in the list once finished, moreover, under the **Finished** tab — chapter XIII.

3.6 — What the game flags, and what it does not

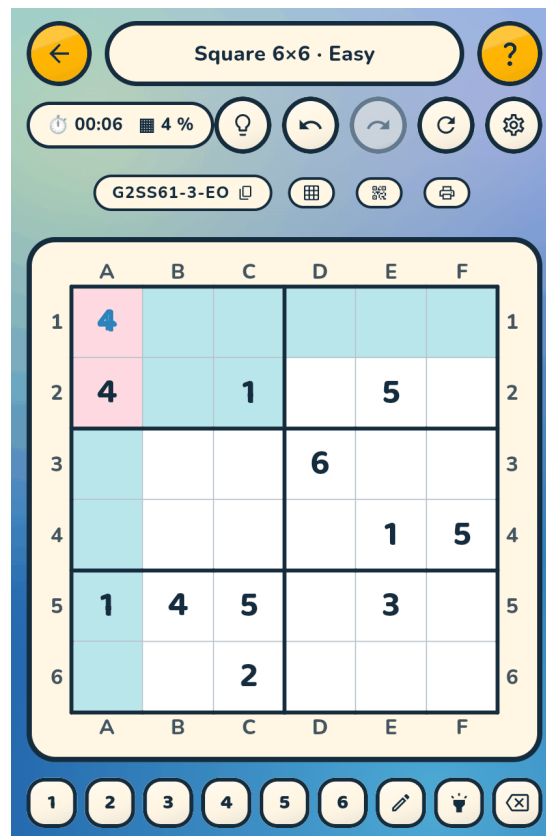


Figure 3.2 — The 4 placed at the top contradicts the given 4 just below : both are marked

The game can flag **conflicts** in real time : placing a value that contradicts an already filled cell marks it immediately. This is a setting, and it is **off by default** — so one plays without a net until it has been switched on.

On the other hand, the game **never** tells you that a value is wrong with respect to the solution, as long as it contradicts nothing. A cell can be wrong and perfectly legal at the time ; that is the very nature of a failed deduction, and announcing it would amount to playing in your stead. The preview does show everything — but that is a gesture you ask for explicitly.

One exception, in Mystery Calcudoku : an **operation** contradicted by the values already written turns red. This is not a hint — the target, the values and the operation are all three before your eyes, it is a calculation you can redo in your head — and without it a single wrong operation would make the game impossible to finish without knowing where to look.

Partie 2

The three basic modes

Chapter IV — Sudoku

Sudoku is the mode where a cell simply carries a symbol, and where all the deduction comes from the groups : rows, columns, regions. There is nothing to calculate. It is also the mode that best shows what the engine of the game can do, because it is played on **every** board shape, including those where the word « row » no longer means much.

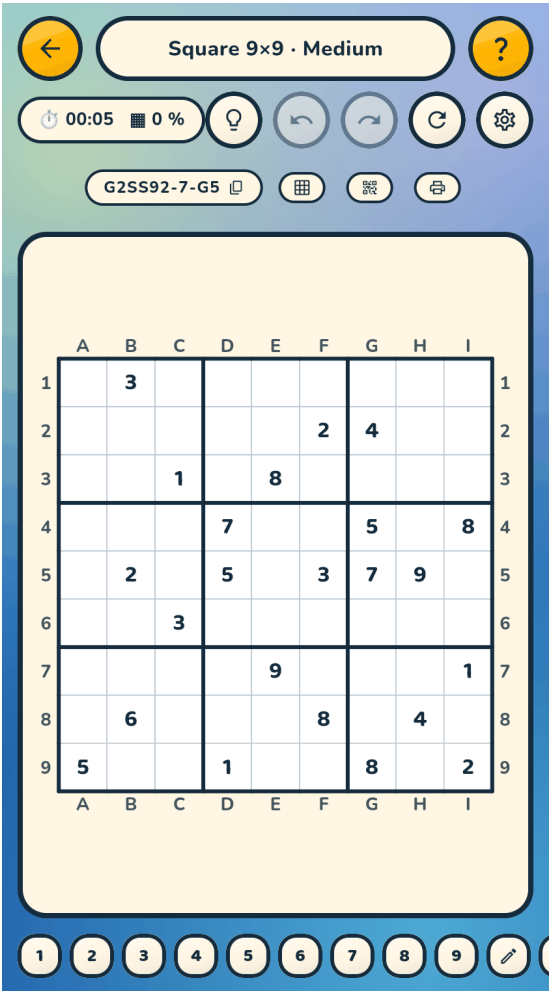


Figure 4.1 — A 9×9 square Sudoku, Medium level — code `G5SS92-7-GB`

4.1 — The rule, and where it applies

Fill every empty cell so that the same symbol never appears twice in the same group. What the groups are depends on the board :

Board	Groups
Square	rows, columns, and blocks when the size allows them
Hexagonal	the lines of the three directions ; no regions
Triangular	regions of nine, rows of the three directions, and adjacency by edge

Board	Groups
Mosaic	rows of the three directions ; a hexagon belongs to every one it touches
Voronoi	families of zones, plus the neighbourhood rule

When in doubt on a new board, the simplest thing remains to **select a cell**: everything linked to it lights up. Part 3 comes back to each shape.

4.2 — Digits, then letters

Up to nine symbols, the cells carry the digits 1 to 9. **Beyond nine a side**, Sudoku moves to the letters **A to Z**: one cell, one character.

This is no typographic whim. On a 16×16, writing «13» in a cell takes two characters where there is room for only one; the numbers become illegible well before the grid stops being playable. A letter settles the matter, and the keyboard follows — beyond nine, it is the **A - Z** keys that place the symbols, with nothing to compose.

Calcudoku, for its part, keeps two-digit numbers: it needs them, one does not calculate with letters. That is the only difference in treatment between the two modes on large grids.

4.3 — The numbers of Sudoku are only labels

The solver never sees values: it handles symbols numbered from zero to $k-1$, and knows no more what a «7» is than what a «square» is. That is what lets the  **Numbers** tile appear in Sudoku too: a Sudoku can be made to carry negative numbers, or numbers drawn at random between 1 and 20, without the grid changing one jot.

For an adult it is a curiosity. For a child learning to read negative numbers it is an exercise in reading without arithmetic on top: the grid remains an ordinary sudoku, only the symbols change. Chapter VIII details the five families.

The same indifference of the solver opens a third door, and it is the most useful in class: **you can write the symbols yourself**. The  **Symbols** tile offers, after the digits and the letters, a **Custom** entry — one syllable per line or separated by commas, MA, PA, LO, then later whole words, HOUSE, LORRY. The two forms mix, and the surrounding spaces are trimmed: you write as you think, without having to reformat a list you already had. The grid never contains those words: it contains ranks, and your ordered list renames them. The sudoku is therefore rigorously the same, and the same grid can be replayed with other words — enough to serve today's grid again next Monday without anyone recognising it.

And there is a family of symbols one does not think of first: **accented letters**. Nothing obliges a list to be made of syllables; it can carry é, è, ê, à, ç in French, ä, ö, ü in German, ñ in Spanish, ã and õ in Portuguese, à, è, ì in Italian. The child then has nothing to copy out: they must **tell apart**, and a sudoku never asks for anything else — two symbols cannot sit twice on the same line, so e and é become two different things that have to be seen. That is what makes the game multilingual down to that level: each language brings the characters that belong to it, and the same mechanism puts them to work.

The case is the one you type: café stays café, and does not become CAFÉ. That is no detail when the accent is precisely what one wants recognised — it is on the small letter that the child learned it, and it is the small letter they read in their book.

You need **at least as many as the grid has cells a side** : nine symbols for a 9×9 . The entry screen says so, but the check does not stop there — if you enlarge the grid *after* writing your list, the tile flags what is missing and the game refuses to start rather than falling back on digits without a word.

The symbols travel with the game code and with its QR, which is the whole point : the teacher writes the syllables the class has seen this week, projects the QR, and thirty pupils have the same grid. A code whose list is too short is in fact refused on reading, like any malformed code — that is precisely where the mistake would cost most. The *teaching dossier* makes a lesson of it.

4.4 — What combines with Sudoku

Sudoku accepts every board option : **Multidoku**, where two to five grids overlap, and the **bonus windows**, extra shaded groups whose pattern is drawn at random in each game. Chapter VII is devoted to them.

It also accepts every size of the square, from 2 to 26, and it is the mode where the structural limit of very large grids shows best : below a certain proportion of given cells, a large Sudoku **stops having a unique solution**. At side 26 that floor is about half the grid, and no amount of computing time would change it. Chapter VII gives the figures.

Chapter V — Calcudoku

Calcudoku keeps the rule of Sudoku — never the same value twice in a row or a column — but takes away the blocks and adds something else: the grid is cut into **CAGES**, and every cage announces a calculation.

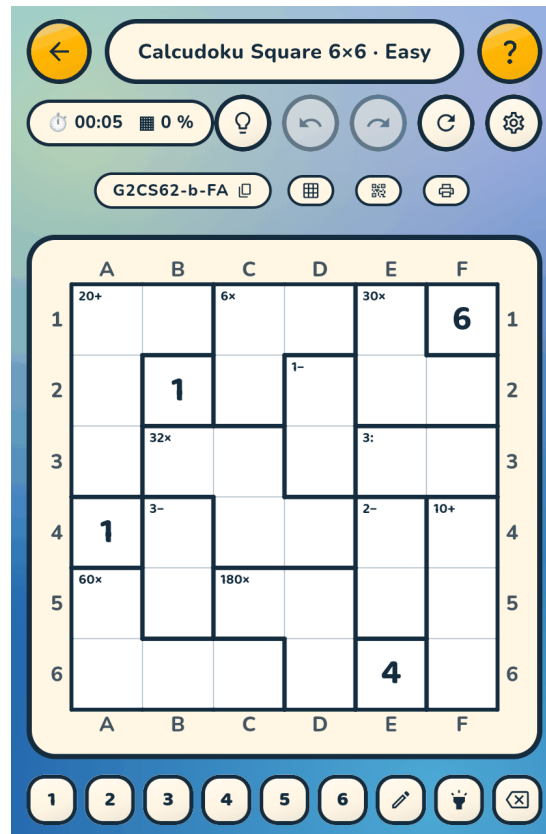


Figure 5.1 — A 6×6 square Calcudoku, Medium level — code `G5CS62-b-FG`

5.1 — Reading a cage

A cage is a group of neighbouring cells, outlined by a thick border. In its upper left corner, a label gives the **result** then the **operation**: **20+**, **32×**, **3−**, **3:**. The values placed in the cage must produce that result by that operation.

Four operations are possible. Addition and multiplication apply to as many cells as you like: **20+** on three cells means three values whose sum is twenty. Subtraction and division only make sense on **two** cells — and by default, on positive numbers, the smaller is subtracted from the larger and the larger is divided by the smaller, so that the order of the two cells does not matter at all.

A remark on the division sign: the application writes **:** and not **÷**. Small and bold, a **÷** reduces to a horizontal bar flanked by matter — that is, to something very close to a **+**. A colon looks like nothing else, and it is moreover the sign taught at school in continental Europe.

A **one-cell cage** has no operation : it gives its value directly, like a sudoku clue. In the illustration, the cells carrying a large 6 , 1 or 4 are of that kind.

5.2 — No blocks, and deliberately so

There is **no block** in Calcudoku, whatever the size. The rows and the columns carry the distribution constraint ; the cages carry the deduction. Adding blocks on top would make most cages useless.

That is also what frees up the sizes : a prime side, which would give a Latin square without blocks in Sudoku, poses no problem at all here. Calcudoku is played from 2 to 26 a side without exception.

5.3 — Calcudoku is not reserved for the square

The cage engine ignores the shape of the board : a cage is merely a group of neighbouring cells. Calcudoku is therefore also played on the hexagon, the triangle, the mosaic and the Voronoi.

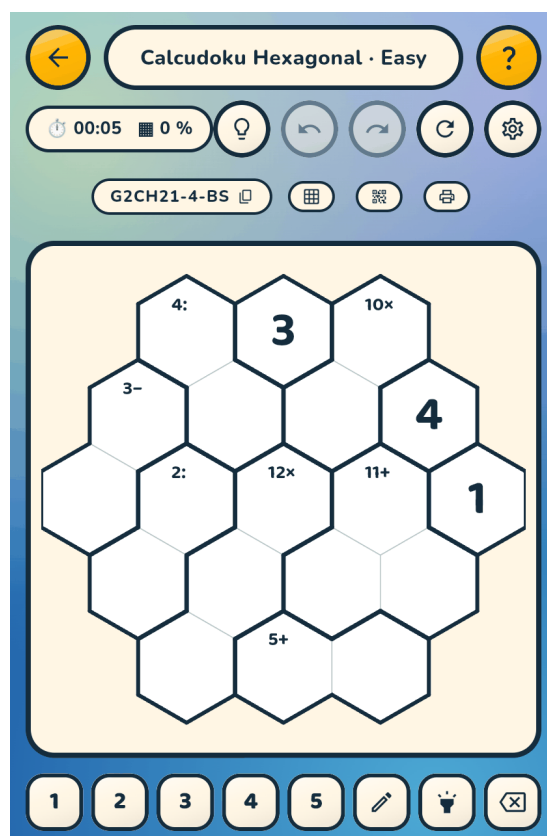


Figure 5.2 — A hexagonal Calcudoku of radius 2 — code `G5CH21-4-BY`

On cells that are not square, the cage label is not placed in a corner : the corner of the bounding box of a hexagon or a triangle falls **outside** the cell. It is anchored near the centre of the cell, which explains why it looks more « in the middle » than on the square.

5.4 — The mode that grows with the child

Calcudoku is the only mode whose **arithmetic** can be set independently of the difficulty of the sudoku. That is what makes it, among other things, an exercise book.

The  **Operations** tile on the home screen opens all these settings at once :

- **Which operations appear** — from addition alone to all four together (chapter VIII).
- **Mystery Calcudoku** — cages give the number, never the operation (chapter VIII).
- **Two cells at most** — every cage becomes a domino (chapter VIII).
- **Maximum result** — no cage result goes beyond the chosen value (chapter VIII).

And the  **Numbers** tile decides what the symbols are worth : the integers 1 to k as always, integers around zero, negatives, numbers drawn at random from an adjustable range, or **fractions** (chapter VIII).

These settings are independent of one another and independent of the difficulty level. Nothing prevents an « Easy » 5×5 Calcudoku from carrying numbers from 1 to 20 and allowing only multiplication : that is a tables exercise on a trivial sudoku, and that is exactly the point.

5.5 — How to go about it

The basic technique is **counting the possibilities**. A 3– cage on two cells of a 6×6 admits only three pairs : {1,4}, {2,5}, {3,6}. A 180× cage on three cells often admits only one. You note the candidates in pencil (chapter XI), cross them with the rows and the columns, and the grid closes up.

One-cell cages are a free starting point, but a good generator gives few of them : the game even deliberately seeks the fewest possible. It tries several seeds and keeps the one whose grid is the least filled — otherwise a grid opening with fourteen given cells out of eighty-one would have little left to deduce.

Chapter VI — Colodoku

Colodoku is the signature mode of the game, and its definition fits in one sentence : *a Calculodoku with additions only, and colours in place of numbers*. A cell carries a colour, a cage announces a colour, and the only operation is **mixing**.

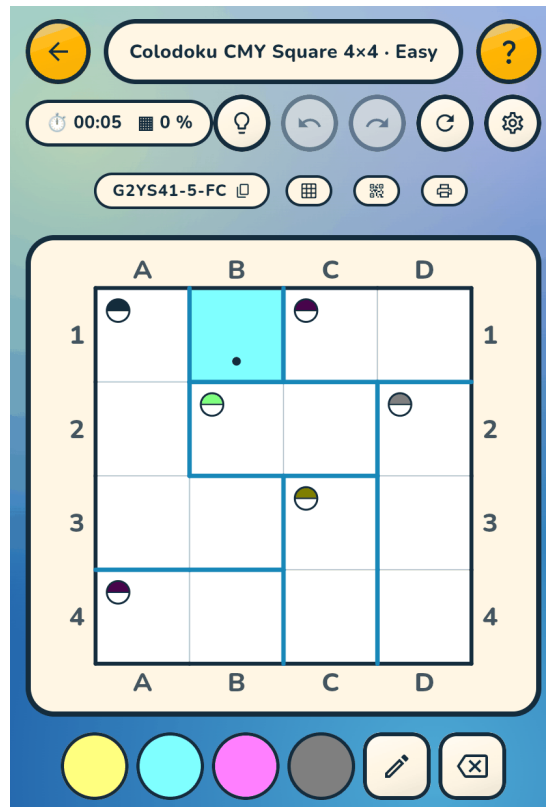


Figure 6.1 — A 4×4 CMY Colodoku, Easy level — code `G5YS41-5-FI`

6.1 — Two ways of mixing

The switch placed next to the mode, in the picker, chooses between two physics of mixing.

CMY — target black. This is the mixing of **ink**: cyan, magenta, yellow. Each colour removes light ; superimposing them all gives black. It is the default setting, and it is the one we know from painting and from the printer.

RGB — target white. This is the mixing of **light**: red, green, blue. Each colour adds some ; bringing them all together gives white. Red and green make yellow, red and blue magenta, blue and green cyan.

The same board is played in both regimes. This is no cosmetic variation : what mixes into what changes completely, and it is the first place in the game where one learns something about the world rather than about sudoku.

6.2 — The rule that makes the mode readable

The colours are not labels : they are **doses**, counted in twelfths on three channels. And the palettes of the game all obey an invariant : **the sum of a whole palette makes exactly the target**.

From this follows what is seen on screen. A complete line makes precisely the target — black in CMY, white in RGB. An overshoot is therefore a mistake, and a mistake **visible before finishing**, which is rare in a deduction game.

The 3×3 is the ideal entry level : red + green + blue = white, cyan + magenta + yellow = black, without a single fraction. The rule « a complete line makes precisely the target » is seen there in three cells.

The PALETTES run from three to eight colours :

Colours	Palette
3	R, G, B at full dose
4	R, G, B at half dose, plus a grey at half dose
5	a neutral, R G B at a third, a grey at two thirds
6	six hues at a third
7	the six thirds, plus a neutral
8	six hues at a quarter, a grey at a quarter, a neutral

The 8×8 is the last step where the eight colours remain distinguishable. At nine, three almost identical greys would be needed — and a grid one cannot read is not a grid. That is also why Colodoku stops earlier than the other modes on each geometry.

6.3 — The cages and their badge

The grid is entirely cut into cages of **two to four cells** — never more. Inside a cage, the same hue cannot serve twice.

Every cage carries **one round BADGE**, anchored in the upper left corner of its leftmost cell. It is cut in two :

- the **upper half** is the mix to be reached ;
- the **lower half** is the mix you have placed so far.

When the cage is complete **and correct**, the badge becomes **solid**, with no dividing line. That is more readable than comparing two almost identical shades — beyond a certain point the eye no longer decides, and the game stops asking it to.

A one-cell cage, as in Calcudoku, is a clue : its colour is already placed, and it carries a **black dot** at the centre.

6.4 — The marks on the board

Colodoku cannot afford the usual marks : a coloured frame around a cell would hide the border of its cage, and a green wash would erase precisely what one is reading. The signals are therefore **shapes**, drawn at the centre of the cell :

Mark	Meaning
+ at the centre	the cell is selected
a large red cross	this colour is already in the row or the column
black dot	the colour is given, it does not change

For the same reason the **torch** does not exist in Colodoku : it would light up cells whose colour is precisely the information being sought. The pencil, on the other hand, works : candidates are noted there as small discs.

6.5 — Where Colodoku is played

Board	Sizes
Square	3 to 8 — without blocks, it is a Latin square
Hexagonal	radius 2 and 3 ; no radius 4 (nine indistinguishable colours)
Triangular	on its « Latin » variant, side 5 to 7
Mosaic	side 6 and 7
Multidoku	yes, from two to five grids

Voronoi is set aside : its palette invariant assumes complete lines, and there is no line left on an irregular tiling. The bonus windows are set aside too — they would add groups that reading the colours could no longer follow.

Partie 3

The boards and what goes on them

Chapter VII — The boards

The square is only one of the six shapes of the game. The others do not change the rule — every group contains every symbol once and once only — but they change what a group is, and hence the way of looking at the board. This chapter reviews them, from the most familiar to the strangest, and ends with what can be superimposed on all of them : several grids at once. The sizes, the permitted combinations and the block tables are gathered in appendix C rather than repeated here.

7.1 — The square, from 2×2 to 26×26

A slider sets the side from **2 to 26**. The 2×2 is a four-cell toy ; the 26×26 counts six hundred and seventy-six and takes several sittings.

The BLOCKS follow a simple rule : the block is the **divisor closest to the square root**, the height not exceeding the width — hence 2×3 at six, 3×3 at nine, 4×4 at sixteen. A **prime side** has no divisor : there are therefore no blocks at all, and the grid is a **LATIN SQUARE**, held by its rows and columns alone. This is not a degraded size but another flavour, where deduction is longer and more linear. Calcudoku, for its part, never has blocks : it is its cages that carry the deduction.

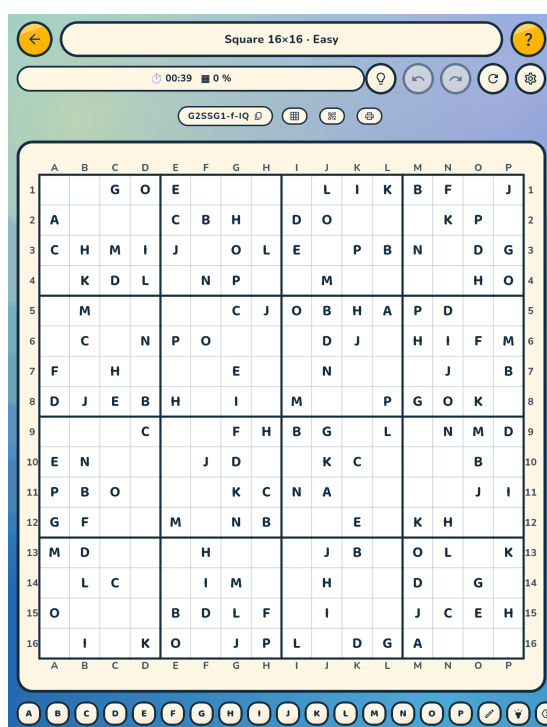


Figure 7.1 — A 16×16 Sudoku : the symbols are letters — code `G2SSG1-f-IW`

Beyond nine a side, two things change. The symbols first : Sudoku moves to the letters A to Z, one per cell, while Calcudoku keeps the numbers — chapter IV explains why. The screen next : a grid of sixteen does not fit comfortably on a phone, and the first time you go beyond nine, the game tells you that the web demo, on a computer, will be more pleasant. The message does not come back, and never appears if you are already in a browser.

The difficulty, for its part, changes mechanism : beyond a certain size, looking for a seed rated at the right level would cost minutes, and so it is **how far the generator empties the grid** that the level drives. One word deserves to be taken literally in the table of the appendix : **floor**. Below a certain proportion of given cells, a large Sudoku stops having a unique solution — half the board at side 26 — and a grid with two solutions is no longer a grid, it is a guessing game. No amount of computing time changes that : it is a property of the grids, not a limit of the program, and the « 10 % of given cells » one sometimes sees announced elsewhere **do not exist** for a sudoku of that size. Calcudoku, for its part, goes far lower, because its cages carry part of the deduction.

7.2 — The hexagon and the triangle

These two shapes share a property : they have neither a horizontal row nor a vertical column in the ordinary sense. The groups run in **three directions**.

The **hexagon** is set by a radius from 2 to 4, that is 19, 37 or 61 cells for 5, 7 or 9 symbols. It has **no region**, and that is no aesthetic choice : these three numbers are **prime**, and a board whose number of cells is prime cannot be cut into equal regions. It is therefore the equivalent of a Latin square, with three families of lines instead of two. A detail of generation explains what you will see : difficulty there is **bimodal** — the grids are almost always plainly easy or plainly expert, and a « Medium » asked for on a hexagon may yield something slightly easier or slightly harder than on a square.

The **triangle** is a large triangle cut into small triangles, of side 6 or 9, with nine symbols in both cases. It is the densest board of the game, because three rules are superimposed there : **regions** of nine small triangles ; **rows** in the three directions, gathering the triangles of one band that point the same way ; and **adjacency**, which forbids two triangles touching by a side from carrying the same symbol. That last rule has no equivalent on the square, and without it the board would need many more clues. Side 4 is not offered : measured, it is unsolvable even with all its constraints.

The triangular Colodoku, for its part, is played on a **Latin** variant of the board — the rows of the three directions alone, without regions or adjacency, the exact analogue of the square without blocks — at a side of your choice from 5 to 7.

A reading remark applies to both shapes : on a square, the label of a cage sits in the upper left corner of its first cell ; on a hexagon or a triangle, that corner falls **outside** the cell. The labels there are therefore anchored near the **centre**. It is the same rule, applied to cells that have no usable corner.

7.3 — The mosaic and the Voronoi

The last two shapes go further. One mixes two cell shapes on the same board ; the other abandons the idea of a regular network.

The **mosaic** starts from a triangular board, then **fuses** certain ROSETTES of six small triangles into a hexagon. The rule is that of the triangle, lightened : the rows of the three directions only, no regions and no adjacency. A hexagon belongs to **all** the rows of its six original triangles — it is the most constrained cell of the board, and often the one to start from. The picker offers two layouts : **Regular**, laid out to a deterministic pattern, or **Surprise**, drawn from the seed of the game, therefore different in every grid and rebuilt identically on resuming. The two are equal in difficulty.

The **Voronoi** is the most disorienting board : no hidden network, a tiling drawn from the seed, cells of three to ten sides, and never twice the same board. Rows and columns disappear there — there cannot be any — and are replaced by two **ZONE FAMILYS** : the first

cuts the board up entirely into connected zones, the second lays a few over the top. A cell therefore belongs to one zone, sometimes two, never three. Each family has **its own frame colour** — one per family, not one per zone — so that the frames read like a map of regions. The OUTLINE of the board is your choice : square, hexagon, octagon or triangle.

One rule remains, and it is the most important :

Two cells that touch, even along a very small side, never carry the same symbol.

It appears nowhere, because no drawing could show it: it holds between all pairs of neighbouring cells, everywhere. It is therefore **written** on the *Rules of this grid* sheet, the only place you will read it — and it is what makes the Voronoi Sudoku playable, by bringing the necessary clues down from nearly 86 % to about 30 %.

To find your way, **touch a cell** : the cells of its zones are tinted, in two cumulative shades if it belongs to two zones. This is not a convenience but the reading tool of the board, and it therefore does not depend on the « highlight linked cells » setting. Finally, the level sets the **number of zones of the second family** there : it modifies the geometry itself, which explains why a bare seed would never be enough to find a Voronoi grid again.

7.4 — Multidoku and the bonus windows

Two options are added to the board without changing its shape. The first lays several grids on it, the second adds groups to it.

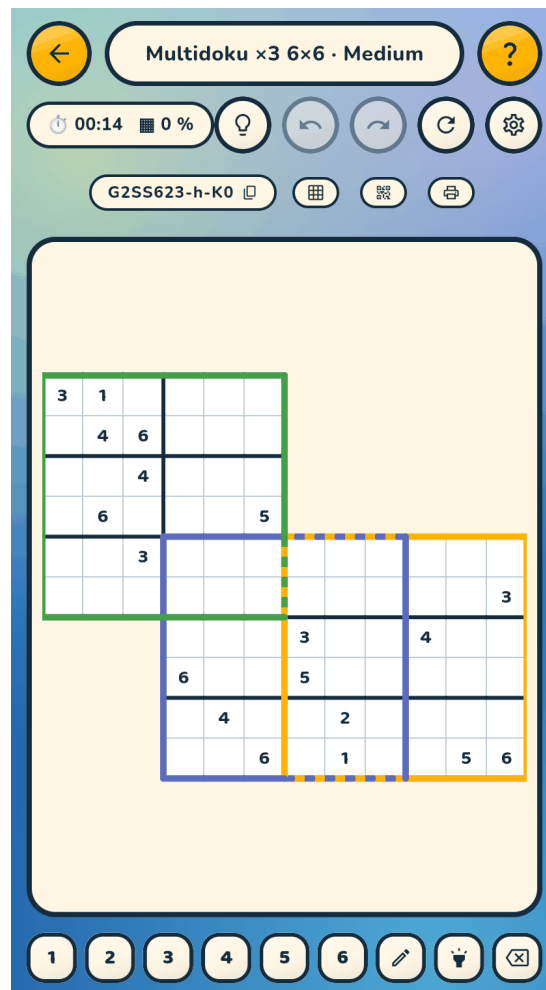


Figure 7.2 — Three overlapping 6×6 grids — code `G5SS623-h-K6`

MULTIDOKU makes **two to five grids** overlap. Each keeps its rows, columns and blocks ; the shared cells belong to several grids and must satisfy the rules of all of them. The famous layouts — samurai, butterfly, windmill — differ only in the arrangement of the grids : rather than delivering a handful, the game **draws the layout from the seed**, so that every game is a new map which resuming rebuilds exactly.

Each grid has **its own colour frame**, all of the same thickness — the colour is enough to tell them apart. The intensity of the stroke, however, does say something : a **bright** frame signals a real boundary underneath (edge of a block, of a cage or of the board), a **very pale** frame signals that it is crossing something that continues, and an **alternating** segment is shared by several grids.

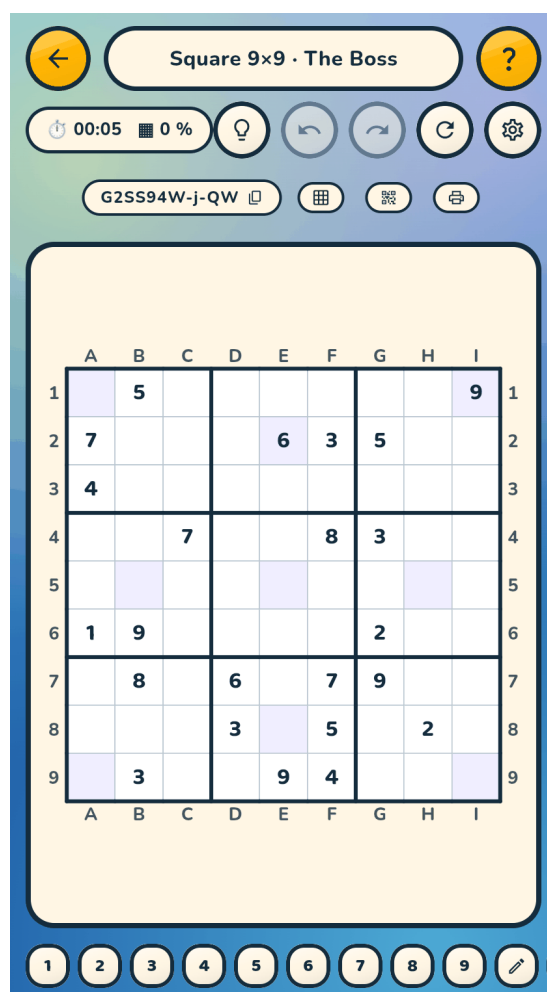


Figure 7.3 — The shaded cells form extra groups — code `G5SS92W-j-QQ`

The BONUS WINDOWS add **extra all-different groups**, shaded on the board. There is only one switch on the player's side : the pattern is drawn from the seed — diagonals, windows in the manner of Windoku, block centres, asterisk, girandole — and the generator offers only those that make sense on the board in play. The shading is **uniform** whatever the pattern : the game does not ask you to recognise a figure, only to see which cells are inside. They combine with Multidoku, but are avoided in Colodoku, where they would add groups that reading the mixes could no longer follow.

Chapter VIII — The Calcudoku that teaches counting

Calcudoku has a peculiarity that Sudoku does not : the arithmetic can be set **independently** of the size of the grid. A 5×5 can drill the tables up to 20, or fractions, or negative numbers, without becoming any harder as a sudoku. It is this decoupling that makes the mode a school tool, and this chapter says what exists and where it is to be found — two tiles on the home screen,  **Operations** and  **Numbers**, are enough to set everything.

8.1 — Choosing the operations

An ordinary Calcudoku mixes the four operations. That is fine for an adult, and unusable for a six-year-old who has just learned to add. The  **Operations** tile therefore allows keeping only some of them : four pills to switch on and off, of which at least one must be kept among $+$, $\times$ and $-$.

The sequence that follows the order of school is this one :

Setting	What the child works on
$+$	addition alone
$+$ $-$	addition and subtraction
$+$ $-$ $\times$	the first three
$\times$	the multiplication tables, and nothing else
$+$ $-$ $\times$ $:$	everything

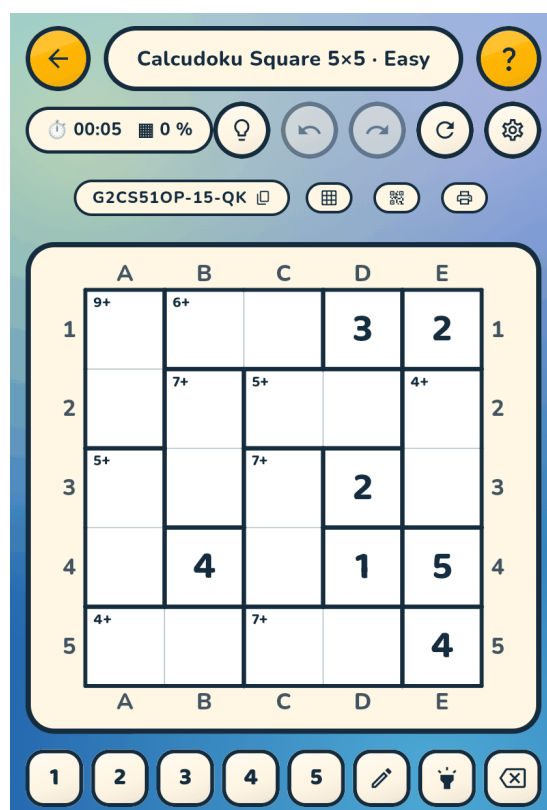


Figure 8.1 — A 5×5 Calcudoku in additions only — code `G5CS51OP-15-QQ`

It is that progression which the **free version** recognises, along with multiplication alone ; the other combinations open and play, but only fill up to a share of their cells, as chapter XVII explains.

One setting deserves a pause, because it does not do what one expects: **subtraction alone**.

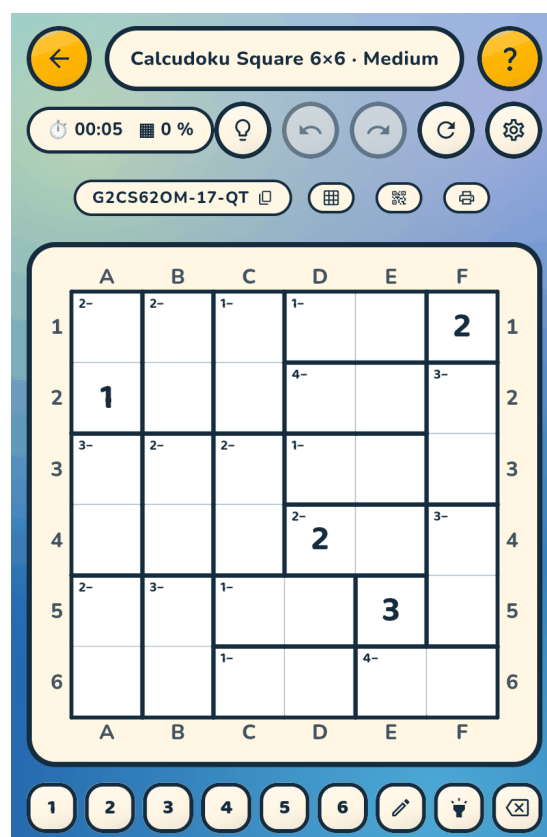


Figure 8.2 — In subtractions only, every cage is a domino — code `G5CS62OM-17-QZ`

Subtraction is defined on **two cells** only — there is no «subtraction of three numbers» in this game. If it is the only operation available, a three-cell cage would therefore have no possible value, and the generator draws the consequence: **every cage becomes a DOMINO**. The board then changes character — small and numerous cages, each saying little, and a deduction that rests far more on the rows and the columns than on calculation. It is an interesting complication, not a simplification.

Division alone, on the other hand, is the only combination the game refuses, and the measurement is clear: most pairs of numbers do not divide one another into a whole number, so that 64 % of cells would be left isolated against 4 % in subtraction — that is to say a grid half filled before the first clue is even removed. Accompanied, in $-$: for example, division regains its full place.

8.2 — Decoupling the numbers from the size

Until now, the size of the board **dictated** the arithmetic : a 5×5 necessarily carried the numbers 1 to 5. A child who wanted larger numbers had no choice but to face a larger sudoku, whereas these are two perfectly independent difficulties.

The **Numbers** tile separates them : it decides what the symbols are **worth** — the ALPHABET of the grid — without changing its size or its level in any way. A grid carries one **VALUE FAMILY**, and only one.

Family	What it produces	Example at k = 9
Classics	1 to k, as always	1 ... 9

Family	What it produces	Example at $k = 9$
Around zero	k contiguous integers centred on zero	$-4 \dots 4$
Negative	$-k$ to -1	$-9 \dots -1$
Random	k values drawn from the seed within an adjustable range	$-7, -3, 0, 2, 5, 8, 11, 14, 19$
Fractions	k fractions drawn from the seed	see below

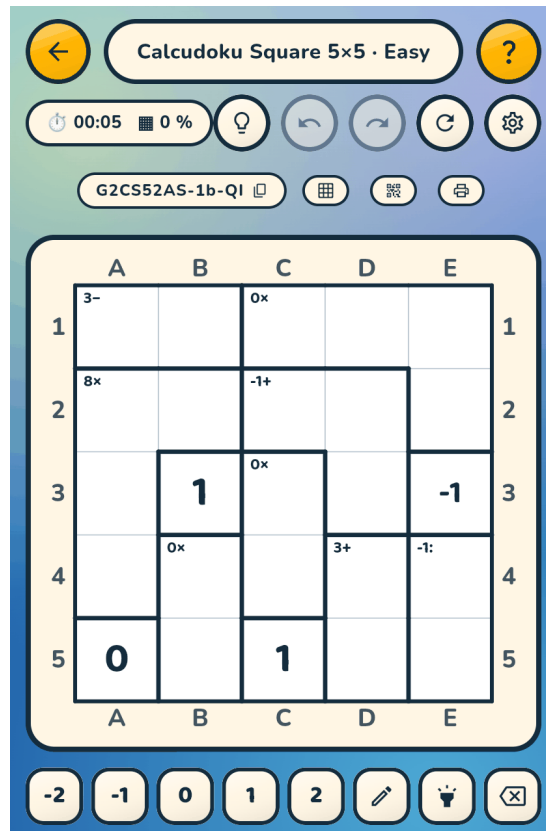


Figure 8.3 — A 5×5 Calcudoku in values centred on zero — code `G5CS52AS-1b-QO`

Around zero keeps the zero when k is even: at six, the family yields $-3 \dots 2$ rather than excluding zero, which is too interesting to be sacrificed to symmetry. *Random* is the one that truly decouples: a double slider bounds the range — at unit steps near zero, wider afterwards, with an entry field for the values that are not a step — and the game draws k different values in every grid. « Five numbers between 1 and 20 on a board five a side » is set that way in two gestures.

The sign changes **nothing** in the calculation, and that is the whole point: $-$ yields the **difference** between the two cells, $:$ divides the larger in magnitude by the smaller. The two rules are symmetrical, so the order of the cells of a cage does not matter and nobody needs to know which one is « the first ». Two consequences show on the board: a subtraction target is never negative — a difference is not — and a label such as $-7+$ brings together two signs that must be told apart, the short hyphen of the number in front, the longer minus of the operator behind.

The on-screen keypad has one key per value, so there is nothing to compose ; the **physical** keyboard types values and not ranks, and chapter XII details that mechanism. Finally, the  tile also appears in **Sudoku**, where it only changes the labels : for a child learning to order negative numbers, it is an exercise in pure reading, without arithmetic on top. The free version covers the positive values from 1 to 10 — the scope of the multiplication tables.

8.3 — Setting the cages

Two switches in the same window do not bear on the operations but on the **cages** : their size and the magnitude of their results. Both are off by default, and both aim at the same use — giving the child calculations to his measure without touching the size of the grid.

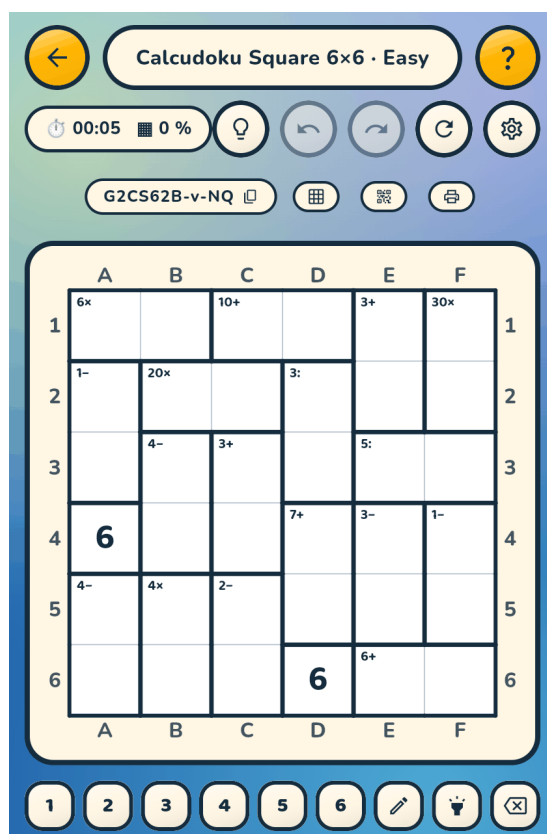


Figure 8.4 — Every cage fits in two cells — code `G5CS62B-v-NW`

Two cells at most forces every cage to fit in two cells, independently of the choice of operations. One thing is surprising and worth saying: dominoes do not fill the grid, they **empty** it — on a square of 9, free cages leave 11 % of given cells, dominoes 5 %, and the grid is built four times faster. Many small cages constrain the board more finely than a small number of large ones, each of which admits a great many internal combinations.

The maximum result caps the absolute value of every target, from 2 to 360 or without limit. It is the setting to raise gradually : 10, then 20, then 50, on a grid whose size does not move. It bites only on addition and multiplication, the two operations that make numbers grow, and its price depends greatly on the size : tight on a large grid, it leaves plenty of pairs without a possible operation, and each falls back to a given cell ; on a 5×5 or a 6×6 — the regime aimed at — it costs almost nothing. Rather than forbidding, the game shows beneath the slider the **share of cages that will remain viable** : you see the price before starting.

Under a very low cap, finally, the same calculation necessarily comes back — nine possible targets do not suffice for thirty cages — and one must then raise the cap or open one more operation.

Together they give the recipe for a tables exercise : square Calcudoku 5×5, × alone, two cells at most, random numbers from 1 to 10, Easy level. Every cage is a line of a table to be found, and the child checks his products against the rows and the columns of the grid. The two settings apply on **every** board of the game, large squares and Voronoi included : a three-cell cage with « two cells at most » switched on would be a defect, and the code of the game is enough to report it.

8.4 — Calcudoku in fractions

The fifth family replaces the integers with **fractions**, on the square board and with all four operations. It is no decorative variant : the aim is to have children learn them, and every setting that follows derives from that.

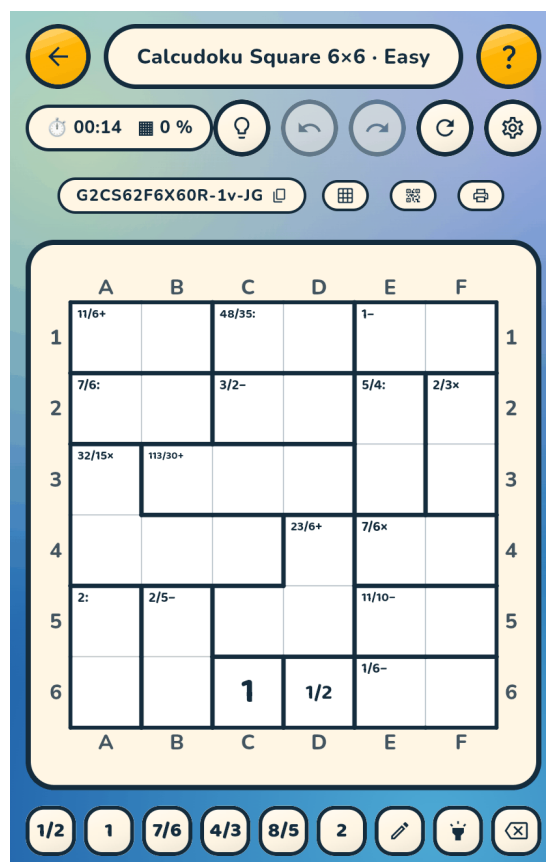


Figure 8.5 — A 6×6 Calcudoku in reduced fractions, denominators 6 and 60 — code `G5CS62F6X60R-1v-JM`

Two sliders, **entirely independent**, set it. The denominator of the **cells** gives the pool : every denominator from 2 up to the chosen value, primes included. That of the **results** is a bound : an operation is retained only if the reduced denominator of its result does not exceed it — with an « unlimited » step at the end. Nothing couples the two, and that is no negligence : $1/8 + 3/8 = 1/2$ reduces to a half, so a result can be **simpler than its operands**, and demanding that the bound be a multiple of the pool would needlessly condemn calculations such as $10/9 - 2/7 = 52/63$.

Beneath the sliders, a line gives the share of **compatible cages**. That is the figure to watch : when it collapses, the cages without an operation fall back to given cells and the grid arrives half filled. In practice, the higher the denominator of the **cells**, the more generous the bound on the results must be ; beyond 6 in cells, it is the unlimited step that gives the least filled grids, at the price of targets with no kinship between them.

A switch at the head of the window, **on by default**, imposes reduced fractions : $1/2$, and never $4/8$. Switch it off, and that is where the lesson lies — recognising that $4/8$ is a half is part of what one learns, and the target is then written on the common denominator : $4/8 + 1/3 = 10/12$ shows exactly the work that $5/6$ was hiding.

The k fractions of a grid are **drawn from the seed**, spelling included : two games do not carry the same ones, and resuming finds its own again identically. The game does not choose an «easy» alphabet — sorting the fractions settles nothing, and making them all agree would come down to renamed integers ; what it bounds is the **cages**. It also discards partitions that repeat a same calculation, so far as arithmetic allows. The game code carries all this at the end — `F<cells>X<results>` , plus a final `R` for reduced fractions — and appendix A gives the grammar.

8.5 — Decimal notation

A fraction and a decimal number are two ways of writing the same thing, and passing from one to the other is an exercise in itself. A slider gives the **proportion** of values written as decimal numbers — cells and targets without distinction. It is not a yes/no, and that matters : the two notations coexist on the same grid, and that is where the lesson lies.

Still, the value has to be writable short. The game only moves to decimals if the repeating part is **at most two digits** and there are **at most three** before it : $3/2$ gives 1.5 , $17/8$ gives 2.125 , $1/3$ gives 0.3 topped with a **bar** saying that the 3 repeats — but $1/7$ and $1/256$ stay fractions. One further rule watches over legibility : if two values of the same grid give the same string of digits, neither goes decimal, since 0.3 and 0.3 with a bar differ only by a stroke. The separator follows the language of the application — only English writes the point.

Like the alphabet, this choice is drawn from the seed, but from a separate stream : it is a setting of **pure display**, which touches neither the grid, nor the cages, nor the solution, and the spelling of a game already begun does not move.

8.6 — Mystery Calcudoku

MYSTERY CALCUDOKU is a harder variant, and its rule fits in one sentence : **the cages give only the number, not the operation**. A label there shows $12?$ where an ordinary Calcudoku shows $12\times$. It is up to you to fill the cells **and** to place the operation. It is not a separate mode but a switch in the **÷ Operations** window, and it therefore combines with everything else — the five geometries, Multidoku, the bonus windows, the restricted operations, the number families.

A row of operation keys is added above the digits : select a cell, press an operation, and it is placed on the **cake** of that cell ; pressing again erases it. The four keys stay visible, but only the candidates are active — on a three-cell cage, $-$ and $:$ are off. The label follows, $12?$ becoming $12+$, and a **÷ n** counter is added to the timer to say how many cages are still waiting. The game is won only if the cells **and** the operations are right ; a one-cell cage asks nothing, it is a clue.

Two things are worth knowing before starting. The operation is not always **unique**: a two-cell cage holding 2 and 2 with target 4 reads as $+$ just as well as $\times$, and both are accepted — what is unique is the grid. And a **complete** cage whose operation does not give the target turns **red**. This is not a hint: the target, the values and the operation are all three before your eyes, it is a calculation you can redo in your head. Without that signal, a single $6-$ placed on 5 and 1 where $6+$ was needed would make the game impossible to finish without knowing where to look.

A Mystery grid is built about four times more slowly than an ordinary one — the cage is weaker, so uniqueness takes longer to prove — and the mode is outside the free version, in the sense of chapter XVII. One combination, finally, deserves a warning: **subtractions only + Mystery** is reachable but degenerate, since a domino in subtraction has only one candidate operation. There is nothing to guess, and yet the same key must be pressed some forty times. In $- :$, Mystery keeps all its meaning.

Chapter IX — The thematic sets

Beyond the three basic modes, the same mechanics serve sets in which a cell carries neither digit nor letter, but a chemical species, a component of an experiment or a piece of a drawing. The principle is that of Colodoku, generalised : a cage is no longer a sum but a **RECIPE** — place the right pieces in it and they compose the thing asked for. The Latin square does not move ; rows and columns keep their rule.

Two families gather them in the mode picker, and both carry at their head a mention in red : **experimental — in development**. It forbids nothing, everything plays. It says only that these sets are not settled — the species chosen, their recipes and even the alphabets may move, and a grid started today will not necessarily replay identically tomorrow. The three basic modes, for their part, are stable.

9.1 — The sciences

Chemdoku composes chemical species from ions and molecules, in three sets — acid-base, oxidation-reduction, and a second acid-base with more varied ions. The label of a cage gives the formula to obtain ; at Hard and Expert levels, only the **name**, which adds a question of vocabulary. From Medium level onwards, the keypad shows only the bare symbol and asks you for the **charge** of the species you place : right answer, it is placed ; wrong, nothing.

Phydoku offers two very different sets. In *mechanics*, a cage is not a formula but an **experiment to set up** — « Build a pendulum », « Measure the fall of a body » — and the necessary laboratory objects must be gathered in it. In *nuclear*, a cage is a nuclide, and the accounting bears on protons and neutrons.

Biodoku takes up the idea of the experiment with a garden vocabulary — to sow, to pollinate, to germinate without soil — and it is the most accessible to the youngest. **Astrodoku** composes objects of the sky : the ringed planet, the star and its planets.

9.2 — For the children

The **Kids** family gathers the sets in which the cells carry **pictograms** — objects a child recognises before knowing how to read. One finds there the seven **Pictoku** (characters, land and air vehicles, three farm sets, economy), the portraits of **Whodoku**, and **Geodoku**, its six elementary geometric shapes — the most suitable for the very first players.

They fall into two models, and the difference leaps to the eye. In the **ENTITY** model — characters, land vehicles, Whodoku — the grid is a 6×6 cut into 2×3 regions, and **each region composes one complete entity** made of its six pieces. Elsewhere, the cages are smaller and form combinations that are already whole things.

9.3 — What you need to know to play them

One property common to all these sets contradicts a habit of sudoku : **the alphabet may be larger than the grid**. Twelve pieces for six cells per region, nine components for a board of seven — a row therefore contains pieces that are all different, but not all the pieces. It is the natural widening of the rule, and it is what creates the deduction : knowing that a pointed hat is already taken elsewhere tells you where it cannot be.

The badges are small and a formula or an assembled scene fits into them badly. **Hold your finger on the badge** of a cage and its target appears very large over the board. For sets with a textual target, a «Target» banner also gives the goal when a cell is selected. A black dot under a pictogram signals a given piece ; a green ring around the badge, a cage complete and correct.

They are all played on a **square board without blocks**, from 4 or 5 up to 6 depending on the set. They accept neither Multidoku, nor bonus windows, nor the other geometries : their composition rules have not been measured there, and the game does not offer what it has not checked.

Partie 4

Playing well

Chapter X — Difficulty

Four levels : **Easy**, **Medium**, **Hard**, **Expert**. These are not labels stuck afterwards on a grid drawn at random : the level takes part in the **making** of the grid, and it does so in two different ways depending on the board.

10.1 — The search for a seed

On most boards, the game builds a grid, has it **rated by a solver**, and starts again if the rating does not match the level asked for. It thus tries several dozen seeds before keeping the right one.

Two refinements are added in Calcudoku.

First, the game does not keep the first seed that comes out right : it tries several at the right level and keeps the one whose grid is the **least filled**. The difference shows — on a square of 9 in multiplication only, one grid went from fourteen given cells to seven by that single choice.

Second, a **clue budget** : a grid where more than 45 % of the cells are given is set aside, even if its rating is good. An almost full grid is not an easy grid, it is a grid of no interest.

10.2 — The ceiling on emptied cells

On the boards where the search for a seed would cost too much — the large squares, the triangle in Sudoku, the Voronoi — the level directly drives **how far the generator empties the grid**.

This is no stopgap. On a 20×20, removing the clues to the end would take hours : the cost of a complete removal is multiplied by 4.4 at each step of the side. With a ceiling, the same grid is built in less than a second. Chapter VII gives the scales.

On the **Voronoi**, the level goes further still : it determines the number of zones of the second family, that is to say **the geometry of the board itself**. A Voronoi grid on Easy and the same on Expert are not the same grid filled differently : they are two different boards.

10.3 — What it gives in practice

Level	On a 9×9
Easy	about a third of the cells given
Medium	slightly fewer
Hard	markedly fewer
Expert	as few as the grid can bear

The word **bear** should be taken seriously : one GIVEN less means one more deduction to make — but below a certain number of clues, a grid stops having a unique solution. The generator never goes below, because a grid with two solutions can no longer be deduced. That is also why very large grids have a floor that does not go down — see chapter VII.

10.4 — The level shown is not always the one asked for

Here is the point that surprises, and that is worth explaining : the pill of the HUD, at the top of the playing screen, does not show the level you asked for. It shows the **rating the solver gives to the grid delivered**.

The two almost always coincide. When they differ, it is because the generator could not reach the level asked for and returned the closest grid. This happens in two cases.

Bimodal boards. In Calcudoku, the hexagon, the triangle and the mosaic produce roughly only two kinds of grids : very easy or expert. The middle levels there are **out of reach** — that is not a defect of the generator, it is a property of those boards. Asking for « Medium » on a hexagon in Calcudoku will therefore often give you Easy, and « Hard » will give Expert.

Costly grids. On a Mystery Calcudoku, or on a tight combination of options, the generator stops after a certain number of attempts rather than keeping you waiting indefinitely, and returns the best candidate it has found.

In both cases, the pill tells the truth about the grid before your eyes. That is more useful than a reminder of what you had ticked.

10.5 — Difficulty is not the only lever

It must be said again here, because it is the easiest misreading to make : in Calcudoku, **the level does not set the difficulty of the arithmetic**.

An « Expert » with two-cell cages and numbers from 1 to 5 is a very hard sudoku and a trivial arithmetic. An « Easy » with fractions of denominator 12 is a sudoku without difficulty and a serious calculation exercise. The two axes are independent, and deliberately so : mixing them would make an « Expert » both more tiresome to compute **and** easier to deduce.

The calculation settings are in the  **Operations** and  **Numbers** tiles — part 4 of this manual.

10.6 — The levels and the free version

The **Hard** and **Expert** levels are outside the free version. As always, they open and play, but the grid only fills up to a share of its cells. See chapter XVII.

Chapter XI — The keypad aids

The keypad at the bottom has **three exclusive modes**. In normal mode, a key places a value. Two buttons switch into the other two, and pressing the same button again returns to normal mode.

Mode	Button	What a key does
Filling	—	places the value in the selected cell
Pencil		notes the value small, as a candidate
Highlighting		lights up every cell carrying that value, as long as you hold

On the physical keyboard, the **space** bar cycles through all three. **Escape**, for its part, exits — from the pencil or the highlighting first, then from the grid — and never goes round in circles : that would be the opposite of what one expects from Escape.

The last two modes can be switched off in the Settings, and a switched-off mode is simply skipped by the cycle.

11.1 — The pencil

The pencil notes **several CANDIDATES in the same cell**. A value typed once appears, typed a second time disappears.

Candidates only hold for an **empty** cell : placing a value supersedes them. They are **saved** with the game, they go through Undo and Redo, and they travel with the state code of a game.

The  button only switches the mode : it is $\otimes$ that empties the cell, in pencil as elsewhere. The gesture thus stays predictable.

Where the candidates are placed

They are anchored in the **lower right** corner of the cell — the upper left corner being already taken by the cage label or the badge — and read from left to right, the block growing leftwards then upwards. On cells that are not square, where the corner of the bounding box falls outside the polygon, the block is **centred**.

When the symbols are **wide** — fractions, two-digit numbers — the candidates stack in a **column**, one per line, aligned to the right. Side by side, $8/7$ and $5/3$ would read $8/75/3$: two candidates would become a single number.

The symbol is noted as it is written : a digit or a letter in Sudoku, a small disc in Colodoku, a pictogram in the thematic sets.

Noting several cells at once

This is the most useful function of the pencil, and it cannot be guessed.

Long press then drag : hold your finger on a cell, then move it. Every cell crossed enters the selection, and a typed value is placed in all of them.

The direction of the gesture is decided on the **first** cell : if it was free, the finger **takes** what it crosses ; if it was already taken, it **releases**. The direction no longer reverses until you let go — going back over a cell therefore undoes nothing, and leaving the board does not either.

In normal mode, the long press **enters the pencil** and starts the group from that cell : it is the gesture that builds a group, and it would have been absurd to demand switching the mode first.

An **ordinary tap** always brings the selection back to the single cell tapped.

With a mouse and a keyboard, two shortcuts do the same thing without waiting :

Gesture	Effect
Ctrl + click	takes or releases one cell, immediately
Shift + click	takes everything between the anchor and the cell clicked
Shift + arrow	extends the selection by one cell

All of them enter the pencil, without resetting the selection — their reason for being is to add.

The range of **Shift** + click is **geometric** : every cell whose centre falls in the rectangle joining the two cells. On a square, that is the rectangle of a spreadsheet ; and it is the only definition that still holds on a hexagon, a triangle or a Voronoi, where there is neither row nor column to run along.

Two points about the group. Toggling a candidate bears on the **whole group** : as long as one targeted cell does not have it, it is placed everywhere ; it is removed only when all of them have it. Without that rule, one and the same press would place here and erase there. And **leaving the pencil folds** the group back onto its last cell : placing a value, pasting, setting a cage operation are single-cell gestures.

11.2 — The torch

The second button turns the keys green (or the colour you have chosen). **Hold a key** : every cell carrying that symbol lights up, over the selection and its peers. Let go, and it goes out.

It is the tool for « where are all the 7s ? », and it changes nothing.

The torch does **not exist in Colodoku** : the colour there *is* the value, and a green wash would erase exactly what is being sought.

Choosing the colours

The colours of the pencil and of the torch can be set (Settings → *Pencil colour*, *Torch colour*). Five hues are offered, and it is not a free palette : each is calibrated to the same lightness as the blue of the selection. A paler hue would read as a disabled state.

Five rather than one, too, for **colour blindness** : green and orange merge in protanopia, purple and amber do not.

11.3 — A helping hand

The pencil, the torch and the colours have this in common : they know nothing. They tidy your reasoning, they do not do it for you. The  bulb of the HUD, on the other hand, knows the answer — and the whole question is how much of it it says.

It says as much as you ask of it, in **three steps**, and you choose where to stop.

Step	What you get
1	how many cells can be found right now
2	the rule to use, without saying where
3	the cell and its value

The first gives no answer : it gives a **number**. That is little, and it is exactly what is needed — « 3 cells » tells you that you are looking at the right scale and that you are not looking in vain, which a mere « there is something » would not say. If there is none, it says so too : the grid then calls for a finer rule than those that show at a glance.

The second names the rule — *a value now has only one place in its row, its column or its block* — without naming the cell. It is a lesson, not an answer : you are handed the technique, you keep the search.

The third gives the cell and its value. You must ask twice to get there, and that is intended.

Its big sister is the **solving steps list**, which does not rescue one cell but tells the whole grid, from the first to the last. It prints, it addresses the teacher rather than the player, and it is reserved for the full version — chapter XIII. The bulb, for its part, is **free** and available in both versions.

It **never lies**. If what you have written makes the grid impossible to finish, it tells you instead of advising in the void — but it does not name the faulty cell : the game never flags a wrong value, and it is not for the help to start.

And when there is nothing left to find, it **disappears**. On a finished grid — as during the preview, where the solution is already on screen — the button fades away instead of offering a helping hand that has lost its object. Undo a move and it comes back, since the grid becomes a grid to finish again.

There remains the last step, the one the bulb does not cross : seeing everything at once.

11.4 — The preview

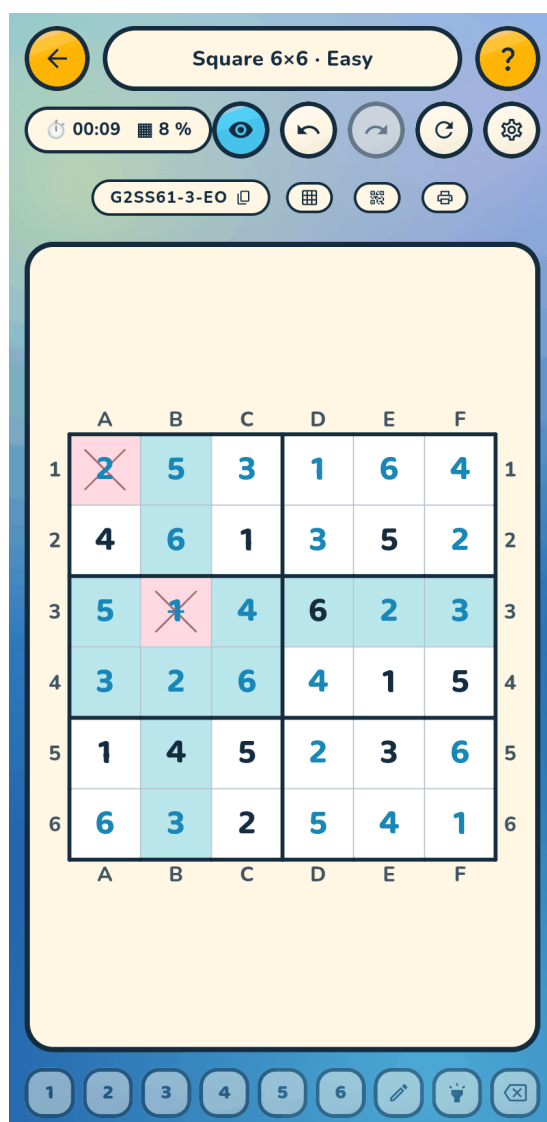


Figure 11.1 — The preview shows the solution ; the keypad is off

The PREVIEW is **off by default** : it must be enabled in the Settings (*Preview mode*) for the **eye** button to appear in the HUD. It is a tool one chooses to have at hand, not a tool one finds there without having asked.

Once switched on, the eye button opens the preview. It shows the **whole solution**, and it is **read-only** : the entire keypad at the bottom is disabled, tools included, and the physical keyboard is too. One looks, one does not touch. The bulb disappears for the duration of the preview — offering a hint when the solution is on screen would make no sense.

During the preview, the cells you had filled **wrongly** are marked : red background, plus a **fine cross** — the cross because a signal resting on colour alone is worth nothing to some players. Colodoku cells have no red background, the colour there being the value ; they keep the cross.

Real-time conflicts are **suspended** during the preview : they bear on your game, not on what is displayed.

Nothing you see there enters the game. Close the preview, and you find your grid exactly as you left it.

11.5 — Undo, redo, restart

The history is **complete** : from the very first move, candidates and cage operations included. A gesture on a group of cells counts as **one single move**.

Restart empties the grid and resets the timer — it is the same grid, taken up from the beginning.

Undo works **even on a finished grid**, where everything else is frozen : you can always come back to your last move, and the grid then becomes a game in progress again — chapter III.

The Undo and Redo buttons can be switched off in the Settings, for whoever wants to play without a reverse gear. The keyboard shortcuts follow the same setting.

11.6 — Copying a cell

On the keyboard, **Ctrl+C**, **Ctrl+X** and **Ctrl+V** copy, cut and paste the **contents** of a cell : its value if it carries one, otherwise its candidates. The mode of the keypad changes nothing there, and pasting **replaces**, in a single move of the history.

Three safeguards : a **given** cell can be copied but neither cut nor overwritten ; an **empty** cell does not replace what you are holding — an unfortunate copy would otherwise turn pasting into an eraser ; and none of this works in **preview**, where it would be a « cheat » button.

Chapter XII — The physical keyboard

The whole game is driven from the keyboard, and **not only on a computer**: plug a keyboard and a mouse into a phone or a tablet, and everything that follows works. Nothing here is reserved for the desktop.

Appendix B gives the complete table, to be consulted at a glance. This chapter explains the two or three mechanisms that cannot be guessed.

12.1 — The basic keys

Key	Effect
1 ... 9	places the symbol in the selected cell
A ... Z	places the letter — Sudoku of more than 9 a side
← ↑ → ↓	moves the selection
Space	cycles through the three keypad modes
Escape	exits — the mode first, the screen next
Del , ∅	empties the candidates, otherwise clears the cell
Ctrl+Z	undoes · Ctrl+Y or Ctrl+Shift+Z redoes
Ctrl+C X V	copies, cuts, pastes the contents of a cell
+ - * /	places the operation of a cage, in Mystery Calcudoku

On a Mac, ⌘ replaces Ctrl. The **numeric keypad** counts as much as the top row, operations included. And on an **AZERTY** keyboard, where the digits require the Shift key, the game also reads the character produced : the key is never mute.

Escape **exits any screen, never the application**. It unstacks : you come back to where you came from, exactly like the ← button or like the Back key on Android. Settings opened from a grid lead back there, not to the home screen. On the grid it is a cascade — the pencil first, the grid next — and it can be repeated until you find yourself at home. Leaving a grid asks for no confirmation : it saves itself.

12.2 — The arrows on a board without rows

A Voronoi has neither row nor column ; a triangle has rows in three directions at once. What should the « right » arrow do ?

The arrows therefore reason not about rows but about the **centres of the cells** : the arrow looks, within a cone of about $\pm 57^\circ$ around its direction, for the most aligned cell, then the nearest. It is the only rule that works on all five geometries — and on a square it gives exactly what one expects.

12.3 — Writing a two-digit number

The large Calcudoku goes up to 26 and keeps numbers : a letter would mean nothing in the operation of a cage. But the keyboard has only ten digit keys. So one **composes** : 1 then 2 , within the second, write twelve.

Three things are worth knowing.

Nothing is written until the number is known. The game waits for the end of the typing. Placing the first digit at once would be wrong : a key **toggles** — typing it again erases — so in a cell holding the candidates 1 and 12, removing the 12 would start by removing the 1.

One only waits when there is an ambiguity. A digit is put on hold only if it can **begin** a number of this grid. In a grid of twelve symbols, only the 1 waits ; in a grid of twenty-six, the 1 and the 2 ; **below ten symbols, none.** The keyboard is then exactly what it has always been, without the slightest delay.

The 0 exists only as a second digit. Alone it places nothing — but without it, 10 and 20 would be impossible to write.

The waiting digit is written when the delay expires, or sooner if something else comes in between : an arrow, the space bar, a cage operation, a tap on the board, leaving the screen. Two exceptions, and they are logical : the **eraser** drops it — 1 then ⌫ must not write a 1 only to erase it at once — and **Ctrl+Z** does too, otherwise it would fall back after the undo and restore what has just been undone.

The composed number counts as **one single move of the history** : 1 2 is undone by a single Ctrl+Z . And it is written in the cell where it was **typed**, not in the current selection — a second is enough for that to have moved.

Finally, a **held** key repeats nothing on digits and letters : « typing again erases », so the burst would empty the cell every other repetition. The arrows repeat normally.

12.4 — Typing on an alphabet that is not 1...k

When the grid carries negative numbers or numbers drawn at random (chapter VIII), the keyboard types **values**, not ranks.

- A number that does not belong to the alphabet of the grid **places nothing**.
- The **-** key opens a negative number.
- The question « can this digit begin a number of this grid ? » comes before « is it worth anything on its own ? », because on an arbitrary alphabet 1 may be only the beginning of 15.

In Mystery Calcudoku, the **-** key keeps its role as an operation : the on-screen keypad has one key per value, so there is no conflict.

12.5 — What the digits do according to the mode

Mode	A digit...
Filling	places the value ; typing it again erases
Pencil	notes the candidate in all the cells taken ; typing it again removes it

Mode	A digit...
Highlighting	lights up the cells carrying that symbol, as long as the key is held

Highlighting does not wait for the end of a typing : it writes nothing, it lights up, and a second digit corrects. Letting go switches it off.

12.6 — The mouse

Two shortcuts serve to build a multiple selection in the pencil, without the delay of the long press :

- **Ctrl** + click takes or releases a cell, immediately ;
- **Shift** + click takes everything between the anchor and the cell clicked.

Both enter the pencil without resetting the selection. Chapter XI details the mechanism.

12.7 — What the keyboard does not do

The **thematic sets** are excluded from the digits : their keypad carries pictograms, and on the chemistry sets a key opens the charge quiz that a shortcut would bypass. The arrows, the eraser, the history and the clipboard work there normally.

The large **Sudoku** does not compose : beyond nine a side it plays in letters, one key per value. It is the large **Calcudoku** that composes, because its values enter into operations.

And the **on-screen** keypad never composes : it already has one key per value.

Chapter XIII — Timer, games in progress and game codes

An InZeDoku game is never lost, and it can travel from one device to another without account or synchronisation. This chapter explains how.

13.1 — The timer and the progress

The timer runs as long as you are on the board. It **pauses** as soon as you leave the screen or the application goes into the background — that is also the moment the game is written. Victory stops it, and the dialogue shows the time.

Beside it, a percentage of progress. It counts only the cells **you** have to fill: the given cells say nothing about where you stand.

Both also appear on each line of the list of games in progress — that is what lets you recognise a game without opening it.

13.2 — The games in progress

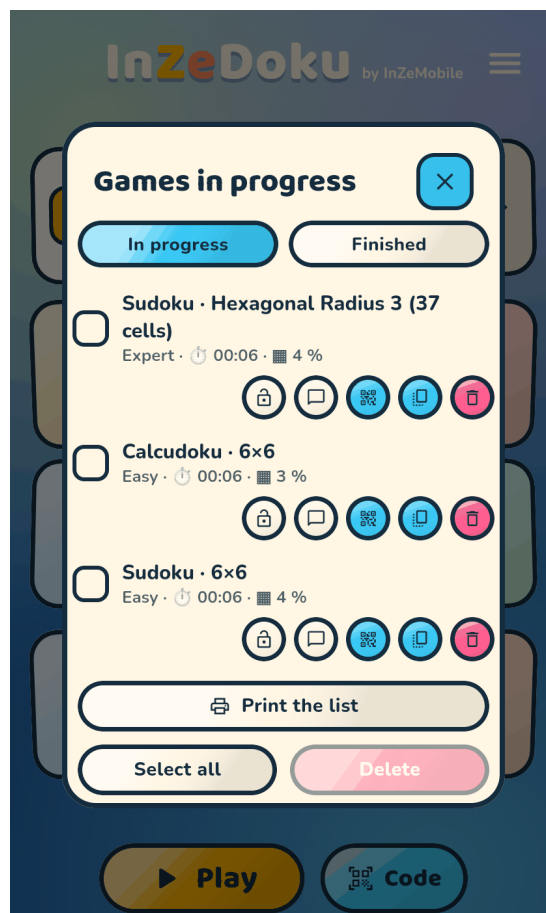


Figure 13.1 — The list: in progress or finished, with timer, progress, padlock, comment and ways out

The game keeps **up to twenty games** at a time. They open from the **Resume** card of the home screen.

A game does not leave the list when you win it: it **stays** there, marked finished. That is what makes it possible to read it again, to print it or to pass it on afterwards, whereas formerly everything that had been played disappeared at the very moment it became interesting. Two pills at the head of the window separate the two families — **In progress** and **Finished** — and you arrive on the first, which is the one you come looking for. The **Resume** card on the home screen, for its part, skips finished games: reopening a completed grid is not resuming it.

Each line carries a tick box on the left, the name of the game and its progress, then, on the line below, five buttons aligned to the right: a **padlock**, a **comment**, a **QR**, a **copy**, and finally the **bin** in pink. That order is not arbitrary: throughout the application the harmless action is on the left in sky blue and the irreversible action on the right in pink. At the foot of the window, a **Print the list** pill takes the full width, and beneath it two buttons — **Select all** in sky blue, **Delete** in pink — follow the same order.

The **padlock** exists for what one especially does not want to lose. A locked game resists the bin on its line, deletion in bulk, and even the ceiling of twenty: it is always an unlocked game that is discarded when the list overflows. A second press removes it — it protects against an unfortunate gesture, it locks nothing away for ever.

The **comment** is a free field, which the game never interprets. It appears **under the title of the game**, in italics, without your having to open it: that is precisely what one wants when one has to find *one's own* game among twenty. A teacher writes « Lea, Tuesday » there, a pupil « stuck on column C ».

This window is **local**, and that explains what it contains: it shows the games you have, and lets you resume them, throw them away, or **take them out** — by the clipboard, by a QR, or on paper. Bringing a game *in* from elsewhere is another gesture, and it has its own place: the **Code** button on the home screen, which is discussed below.

What a saved game contains — and what it does not

A saved game does **not contain the grid**. It contains the mode, the configuration, the seed and your entries. The grid is **rebuilt** on resuming, from the seed, identically.

That is what makes saved games tiny — and it is also why the generator must remain rigorously deterministic. If a new version of the game changed the way grids are made, a saved game would rebuild on **another** grid, and your entries would fall out of place.

The game protects itself in two ways. It records the version of the generator and discards a saved game that does not come from it; and on rebuilding, it **checks** that the rebuilt grid has the same number of cells to fill as the one that was saved. If the count does not come out right, the game is refused rather than filled out of place.

13.3 — The game code

Every grid has a **GAME CODE**, a short string — `G5CVS72-1a2b3c-V0` — that carries everything needed to rebuild it: the mode, the geometry, the outline, the size, the level, the options and the seed.

It appears **under the tools**, on the playing screen, in a small cream pill. A tap copies it.

Type in someone else's code — **Code** button, to the right of Play on the home screen — and you get exactly their grid.



Figure 13.2 — The three ways of bringing a game in, in the same place

This window is **the only place through which a game comes in**, and it offers three paths to the same field : **typing** it by hand, **pasting** it from the clipboard, or **scanning** it on the screen of another device. All three are the same gesture — bringing a game from elsewhere — and they all end at the same place.

That is also why they are not in « Games in progress » : that list is local, it shows what you have. Resuming is not importing.

The field checks the entry **live** : as soon as the code is valid, the configuration appears in plain words below — mode, geometry, level. You therefore know what you are about to start before starting it, wherever the code comes from.

Why a bare seed would not be enough

That is the question one asks on seeing a ten-character code where one expected a number. The answer is simple : **the same number gives a different grid depending on the mode, the geometry, the size and the level**. A seed alone designates nothing.

On a Voronoi board it is clearer still : the level determines the number of zones, that is to say the very geometry of the board. Without the level it is not the same grid — it is not the same board.

The decoding is **strict**. An unexpected character, a combination that does not exist, an option in the wrong order : the code is refused. Better « code not recognised » than a grid next door to your correspondent's.

Two tolerances, all the same: the code is insensitive to case and to spaces, and punctuation stuck **at the ends** is trimmed. A code travels by messaging, by e-mail, by clipboard, and often comes back with a full stop or a slash hooked on the end. Inside, on the other hand, a stray character remains an error — absorbing it would silently give your neighbour's grid.

Appendix A gives the complete grammar, character by character.

13.4 — Passing on a game in progress

The game code passes on a **blank** grid. To pass on a game **as it stands**, with what you have written, there is a second format, the STATE CODE.

From the playing screen, the small button marked with a grid, to the right of the code pill, copies everything. From the list of games in progress, it is the **copy** button on each line.

The string looks like this :

```
G5SS41-2-EC:1.(24).....
```

It is the game code, a colon, then **one token per cell to fill**, in the order of the cells of the board :

Token	Meaning
a symbol (3 , A ...)	the value you have placed
.	a cell left empty
(24)	the candidates noted on an empty cell

Given cells are skipped : they come from the seed, it would be pointless to carry them.

On the other device, this string comes in through the **Code** button of the home screen, where the **Paste a game** pill drops it into the field. The configuration then appears in plain words just below — you see what you are going to open before opening it — and **Play** launches the grid.

The pasting **checks** that the number of tokens matches the cells to fill of the rebuilt grid. If it does not — because the code comes from another version of the game, for instance — it is refused rather than placing your values one cell out.

That is how a game passes from one phone to another, or from a phone to the browser, with the clipboard as its only intermediary and without any data leaving the device other than by your own gesture.

One limit to know about: the operations placed on the cages of a **Mystery Calcudoku** do not travel with the state. The values and the candidates do.

13.5 — The QR, when both devices are at hand

Copying a string from a phone to a computer is tedious, and reading it aloud in a classroom is more so still. When both devices are there, the QR makes the trip for you.



Figure 13.3 — The QR carries the blank grid or the game in progress, as you choose ; the code stays legible underneath

Showing. The QR button on the playing screen — the third of the code row, after the pill and the button that copies the grid — opens a small window. In « Games in progress », each row has its own.

A switch there decides what the QR takes away. By default, the **blank grid** : the game code, and nothing else. That is what one wants most of the time, for a QR is shown, and often to several people — what is then handed out is the grid, not the cells one has just filled oneself. Switch it off and the QR carries the game **as it stands**, exactly the state code. The switch does not even appear when it would serve no purpose : on a grid where nothing is yet written, both positions give the same code.

Note the code in plain sight under the image : it is always there, even when the QR is not. It is the safety net for whoever has no camera, and a tap on it copies what the QR carries — the two outputs of this window always say the same game.

Scanning. From the **Code** button, the **Scan a QR** pill opens the camera. A recognised QR **opens the game directly** : there is no typing mistake to correct on a scanned code, and the check key already refuses any damaged code. A QR that is not one of ours is simply **dropped into the field**, where « code not recognised » appears — saying nothing would leave one thinking the scan had failed.

Access to the camera is asked for **at the moment of the scan**, never at installation, and a refusal is no dead end — the string always works. The image is decoded on the device : nothing leaves it, and the decoder is embedded rather than downloaded, so that scanning works without a network like the rest of the game.

When the game refuses to show a QR

It happens, and it is intended. A QR that is too dense is no longer readable by a camera : beyond a hundred or so squares a side, the lens would have to be pressed against it. Past that limit, the game says so and refers you to the string, rather than displaying a mosaic nobody will manage to photograph. There is never a burst of several QRs.

That refusal can only strike the **game in progress** position. A bare game code always fits, however large the grid — the default position is therefore also the one that is never refused.

What weighs is not what one imagines. A game **without candidates** fits in eleven characters, whatever its size : with nothing noted, the state code is the game code, and a placed value costs only one more. It is the **pencil candidates** that weigh, and they alone.

Game	Length	Fits ?
any at all, without candidates	11 characters	yes
9×9 calcudoku entirely pencilled	903	yes
16×16 sudoku, five candidates per cell	1,118	yes
26×26 calcudoku, five candidates per cell	3,897	no

In other words : you will only meet the refusal by heavily annotating a very large board — the one that is not carried in a single gesture, and for which the clipboard remains entirely suitable.

13.6 — Printing

The clipboard demands a second device, the QR a camera. There remains paper, which demands nothing — and it is often paper one wants when the same grid must pass between several hands at once.



Figure 13.4 — What is printed, whether one wants the solution, and — in the full version — the solving of the grid or that of the player

Two buttons lead there. Under the board, the last of the code row prints **the grid before your eyes**. In «Games in progress», the **Print the list** pill brings them **all out at once**: one thumbnail per game, with what it is, its code and its QR.

The sheet reproduces the board as it is on screen, **coordinates included** when the grid has them — the letters and the digits all around. That is what allows several people to talk about it without leaning over the same page: one says **D1**, and everyone knows where to look.

On a grid, a small window opens before printing. **Blank grid**, on by default, prints the board without anything you have written: it is the sheet one photocopies, and it is what makes it possible to play — or to have a whole class play — **without the slightest device**, in pencil, the application serving only to prepare the grid. Switch it off and the grid comes out as it stands, your values in grey and the given cells in black — but with no conflict flagged, for a sheet handed to someone has no business telling them where they went wrong. **Add the solution** is off, and stays so until you touch it: the solved grid then comes out on a second page, never by accident.

A third switch appears in the **full version**: **Solving steps**. Off by default too, and for the same reason — a solution slipped into the pile one hands to children would be an incident. It adds a page taking the grid **cell by cell, in the order in which they can be found**, each with the rule that unlocks it: *in row 1, 0 has no other place*, or, when several are needed, the cell at the head and the rules below.

It is **one** path among others, and the sheet says so at its head : at each step, several techniques may serve. It may also **stop before the end** — on a grid that logic alone does not suffice to solve, which is the very definition of the Expert level, the last line announces how many cells remain to be found. Inventing the rest would be worse than stopping.

A **fourth** switch follows it as soon as the game counts at least one move played : **Player's path**. It turns the question round — instead of telling how the grid *could* be solved, it tells how you *actually* did it. The game replays your moves in order, places itself each time on the board as it was **then**, and names the rule that applied there, in the same words as the solving steps : the two sheets are read side by side, the machine on one side, the pupil on the other.

Each move receives one of **three verdicts**, never two. Either the rule is named. Or the value is right but the reasoning could not be identified — which **is not a mistake** : you may have chained two deductions where the game looks for only one. Or it is a **mistake**, and the word is used under one strict condition only : the value placed makes the grid impossible to finish. It is therefore not « this does not look like the solution » — on a grid where two spellings are worth the same, that would be false — and, in case of doubt, the game keeps quiet rather than accusing.

The page opens on **your grid, its faulty cells marked**. That is exactly the opposite of the sheet one hands out, which tells the pupil nothing of his mistakes ; here one reads the game again **with** him, and that is what one came for. One mistake does not condemn the rest : the faulty move is **set aside** and everything that comes after is judged without it, failing which a single cell would close the grid and the twenty following moves would be marked wrong in a block. The sheet says so at the foot of the list, so that one is not surprised. And a mistake you have corrected yourself disappears from the grid — the **line**, however, stays : it is the history of the game, and one does not rewrite it.

What the sheet never does is say **why** you thought that. It says what is wrong and from where ; the rest belongs to the conversation one will have afterwards.

In the free version, these two switches **do not exist** — rather than being there, greyed out, showing boxes one cannot tick. To be rescued on one cell, the  bulb of the HUD is free : chapter XI.

On the list, there is nothing to set and the sheet comes out at once. Its QR carries the **bare code**, never the state : it is a sheet one hands out, and those who receive it have no use for the half-filled game of whoever printed it. That is the whole difference from the QR button of the same list, which does carry the game in progress.

Then it is the **printing dialogue of the system** that takes over : choice of printer, of format, and « Save as PDF », which covers sharing without the game having to invent one more export.

Like everything else, printing adds **no permission**. The document is made on the device and handed to the printing service ; nothing goes out on the network.

Partie 5

Settings, accessibility and privacy

Chapter XIV — The settings

The settings do **not** concern the grid to come — everything about that is on the tiles of the home screen. They concern the way you play and the way the game is displayed.

They are reached from two places : the ☰ menu of the home screen, or the ⚙ circle of the HUD during a game. In both cases, the Escape key or the ← button lead back **to where you came from**.

14.1 — The theme

Three positions : **System**, **Light**, **Dark**. By default, the game follows the phone.

The dark theme is not an inversion of the colours. The washes of the board, the frame tints and the strokes are all **recalibrated** : a wash set for a white background is quite simply invisible on a black one, and a dark ink would no longer make any border. That is why the dark theme has its own values rather than a filter applied to the light one.

14.2 — The reading aids

Highlight the selected cell's line and region — on by default. It is the most useful aid for discovering a board : the cells linked to the one touched light up, and one sees at a glance what a « row » is on a triangle or a « zone » on a Voronoi.

One exception : on a **Voronoi** board, the tinting of the zones does **not** depend on this setting. It is always active, because it is not a convenience but the reading tool of the board — without it, one cannot follow a zone with the eye.

Flag conflicts in real time — off by default. Once on, placing a value that contradicts an already filled cell marks it immediately. Off, the game says nothing : it is up to you to read yourself again.

Note carefully what this setting does not do. Even switched on, it never flags a value that is simply **wrong** with respect to the solution : it flags a value that contradicts another already placed. A failed deduction stays silent until it runs into something.

14.3 — Colour blindness

A switch, then a **type** — that is the subject of chapter XV.

14.4 — The keypad tools

Four switches decide what appears in the HUD and in the keypad. Each removes a button when switched off.

Setting	By default	What it adds
Undo / redo buttons	on	the two circles of the HUD, and the <code>Ctrl+Z</code> / <code>Ctrl+Y</code> shortcuts
Preview mode	off	the eye circle, which opens the solution read-only

Setting	By default	What it adds
Pencil (candidates)	on	the  button of the keypad, and its place in the Space cycle
Symbol highlighting	on	the  button of the keypad, and its place in the cycle

Switching off a tool also removes it from the **cycle of the space bar**: it never offers a mode that does not exist.

The preview is off by default because it is the only tool that shows the solution. It is there to understand a grid one is stuck on, not to be permanently within thumb's reach.

14.5 — The colours of the pencil and the torch

When the pencil or the highlighting are on, a row of dots allows their colour to be chosen. Five hues, and the button, the keys and the lit cells all carry the same.

The choice is not a free palette: the five hues are calibrated to the same **lightness** as the blue of the selection. A paler colour would read as a disabled state. And five rather than one also serve colour blindness — green and orange merge in protanopia, purple and amber do not.

14.6 — What is not here

Two settings one would naturally look for in this screen are elsewhere, and deliberately so.

The **language** is in the  menu of the home screen. As long as it stays buried in a settings screen, the application silently follows the system and nothing tells the player that six other languages exist.

The **grid** settings — mode, size, difficulty, operations, numbers — are on the tiles of the home screen, where one sees at a glance the game one is about to start.

Chapter XV — Accessibility

A game whose signature mode rests on colours must take colour blindness seriously. InZeDoku treats it as a **type**, not as a switch — because colour blindnesses are not compensated for in the same way.

15.1 — Choosing your type

The **Colour blindness** setting is first of all a switch. Once on, a line shows the type chosen and opens the choice :

Type	What merges
Deuteranopia	red and green
Protanopia	red and green, with reds appearing dark
Tritanopia	blue and green, yellow and pink
Achromatopsia	no colour is perceived — black and white

Each type is named **and** described in full words, with the colours concerned shown as dots. This is no affectation : « deutan » means nothing to anybody, and a child, or the parent setting up the device, must be able to recognise their own without consulting a dictionary.

The default type, when the switch is turned on, is deuteranopia — it is the most widespread form.

15.2 — What the setting changes

In **Colodoku**, the cells carry a **symbol over the colour**. The colour remains the value, but it is no longer alone in carrying it.

In **Multidoku**, the palette of the frames swings onto an axis that the chosen type separates further :

Type	Axis of the frames
Red-green (protan, deutan)	blue and yellow
Tritanopia	red and green, plus a difference in lightness
Achromatopsia	no colour at all : each grid receives a line pattern

In achromatopsia, the frames therefore turn ink-coloured and are told apart by their stroke : solid, long dashes, dots, dash-dot, medium dashes. A segment shared by two grids alternates their patterns section by section, exactly as it would alternate their colours.

15.3 — The marks that depend on no colour

Some marks of the game rest on colour in **no** setting. They were drawn that way from the start, and it is worth knowing them.

Mark	Where	Why a shape and not a hue
large red cross	Colodoku : colour twice in a line	a coloured frame would hide the cage border
+ at the centre	Colodoku : selected cell	same reason
black dot	Colodoku, sets : given value	reads without distinguishing the colour
fine cross	preview : wrong cell	doubles the red background, which would not suffice
solid badge	Colodoku : cage correct and complete	a shape, not a shade to compare

That last one deserves a special mention. A Colodoku badge is cut in two — the target above, your mix below — and **becomes solid when the cage is right**. Comparing two almost identical shades is difficult for everybody, and impossible for some : the solid badge is a binary answer, which is seen without discussion.

15.4 — The zoom

Two-finger zoom works on every board, without exception and without condition. It does not merely serve the large grids : a Colodoku, a chemical formula or a pictogram is looked at closely, and a small square does not always fit comfortably on a narrow phone.

At normal scale, the board is exactly fitted to its frame and does not slide : the zoom therefore changes nothing until you have pinched.

15.5 — The touch targets

Every control of the game — HUD circles, keypad keys, pills, rows of the setting windows — is at least 44 pixels a side, including when the visible drawing is smaller. The button that copies the state of a game, for instance, draws a small grid of 14 pixels within a target of 44.

15.6 — What remains to be done

It must be said honestly : **screen readers** are not yet supported. The accessibility labels for TalkBack and VoiceOver, following the system text sizes and the systematic checking of targets at 48 points are on the programme of version 1 and are not finished.

Chapter XVI — Privacy

InZeDoku is **entirely offline**. This chapter says what exactly that covers, how it is verified, and why the game is built that way.

16.1 — No network permission

The published package declares **no network access permission**. This is not a promise but a verifiable property : the check is part of the release procedure, and a package declaring a network permission would not be published.

Everything else follows from that absence. Without network permission, there can be :

- no advertising network — an ad library needs the network ;
- no audience measurement or usage statistics tool ;
- no crash report sent anywhere ;
- no account, no identifier, no synchronisation ;
- no in-app purchase, which would require a payment library and a receipt to be checked online.

These are not functions the game has chosen not to use : they are functions it **cannot** use.

16.2 — Your games stay with you

The games in progress, the home configuration and the settings are saved on the device, and nowhere else. Nothing is sent, nothing is backed up online, nothing follows an account.

The other side of that must be known : **if you uninstall the game, your games disappear with it**. There is no restoring, because there is nowhere to restore from. If a game matters to you, copy its state code (chapter XIII) and keep it where you like.

16.3 — The only two outward links

Two places in the game can lead outside the application, and neither of them accesses the network itself.

The name **InZeMobile**, in the About box, is a link to the publisher's site. The  **Your feedback** entry of the menu prepares an e-mail.

In both cases the game hands over : it asks the system to open a browser or a mail application, and it is that application that accesses the network. The game still has no permission at all — it needs none to hand an address to somebody else.

If no browser is available, the opening fails silently.

16.4 — The feedback form sends nothing

The  **Your feedback** entry deserves a few lines, because it is the only place in the game where you are asked to write something.

It **transmits nothing by itself**. It prepares a message : a category (Generator, Suggestion, Other), an optional subject, your text, and a block of **technical context** — version, language, platform, configuration, and above all the current **game code**, which is what allows the very grid you are talking about to be replayed.

That block is in an **editable field**. You see it in full, you can correct it, you can empty it. In an application that collects nothing, attaching anything without showing it would be too much.

Two ways out, and you choose : **Copy it all** puts the address, the subject and the body into the clipboard — a single paste suffices — or **Open my mail** hands over to your mail application with the message prefilled. If the opening fails, the message is copied and the game tells you so.

The dialogue does not close by itself after either one : you leave by the cross when you have finished. A failed send must not oblige you to write everything again.

No category is ticked in advance, which would be a way of biasing the sorting.

16.5 — Why it is built this way

This is not a peculiarity of this game. It is the way **InZeMobile** builds everything it publishes : no tracking, no profiling, no data leaving the device, in every product, without exception and without a version where the rule would be relaxed.

A logic game has no need whatever to know who you are. And a company that collects nothing has nothing to lose, nothing to let leak, and nothing to sell.

It is a deliberate wager : to build a reputation on the originality and the quality of what one makes, and on an uncompromising position on privacy — rather than on what one might learn from the people who play.

Chapter XVII explains how that choice is funded : one single application, three ways of playing it, and a network that serves the purchase alone.

Chapter XVII — Free, subscription, lifetime

There is **one application**, and it plays in three ways : free, by **subscription**, or after a **lifetime purchase**. This chapter says exactly what each one allows — and why the line falls there rather than elsewhere.

17.1 — The free version locks nothing

That is the most important point, and it is counter-intuitive : the free version **forbids nothing**. Every mode, every geometry, every size, every level, every setting opens and plays. A game code from a player of the full version opens too.

What it limits is the act of **finishing**. Outside its range, a grid only fills up to a **share** of its cells to fill ; beyond that, the keypad places nothing more.

In other words : you can try a Voronoi, a 26×26, a Mystery Calcudoku, a eight-colour Colodoku — look at them, understand them, do a good part of them. You simply will not be able to finish them.

17.2 — The free range

Board and mode	Free up to	Beyond
Square · Sudoku	9×9	25 %
Square · Calcudoku	5×5	25 %
Square · Colodoku	4×4	35 %
Square · Pictoku	5×5	50 %
Square · science sets	5×5	30 %
Hexagonal	radius 2	35 %
Triangular	side 6	35 %
Voronoi	5 symbols	35 %
Mosaic	—	25 %

Calcudoku stops at **5×5** and not at 9×9, and that is no tightening : it is the direct consequence of the decoupling of the numbers from the size (chapter VIII). A child now works on the tables up to 20 **on a grid five cells a side** — the original justification, « for children's multiplication tables », is therefore served, not abandoned.

Five more things fall outside the free range, independently of the size :

Outside the free range	
the Hard and Expert levels	25 %
the mosaic geometry	25 %

Outside the free range	
Mystery Calcudoku	25 %
operations outside the four progressions $+$, $+-$, $+-\times$, $+-\div$:	25 %
values outside 1 to 10	25 %

That last line deserves an explanation. The free version covers the **positive values from 1 to 10** — « multiplication tables, that's all ». It is not a question of grid size : since the numbers were decoupled from it, a 5×5 can carry $-9\dots 9$ or $1\dots 20$, and it is that which is bounded. The *Around zero* and *Negative* families fall outside by construction ; the *Random* family stays inside as long as its range fits in $1\dots 10$.

Note in passing that a large square in **classic** values falls there too : a 16×16 carries the values 1 to 16, hence beyond 10.

Fractions are not on that list, and deliberately so : they follow the size rule of Calcudoku, like the integers. Up to 5×5 a grid in fractions plays entirely in the free version, decimal notation included ; beyond that it is the size that limits, not the fractions.

When several reasons apply, **the lowest share wins**.

17.3 — The game warns you three times

Nothing comes as a surprise.

On closing a picker, as soon as the share changes, a message says so. You therefore see the price of a setting at the moment you choose it.

On opening each affected game, a dialogue announces the permitted share and names every reason that limits it. That is the only sure moment : a game resumed, or launched by a game code, goes through no picker.

When the wall is reached, a third message announces it. It had to be said : a key that does nothing would pass for a defect.

17.4 — What the wall stops, and what it does not

Only the **adding** of a value is refused. Everything else keeps working :

- erasing a cell, correcting a cell already filled ;
- undoing and redoing ;
- the pencil and the candidates ;
- the preview ;
- the timer, the saving, the game code.

The principle is that a grid outside the range **cannot be finished** — not that it cannot be touched.

17.5 — Unlocking loses you nothing

A game begun for free is not lost, and there is nothing to do to keep it. The subscription or the purchase lift the limit **in the application you are already playing in**: your games in progress, your candidates, your codes and your history are where they were, and the grid you were stuck on can be finished.

That is the first reason for having only one application.

17.6 — Without a network

A subscription runs **one year from the purchase**, and it plays **offline until it ends**: the game knows the date, it has nothing to ask. A **lifetime** purchase is never checked again.

There is only one moment when you must be online, and it is not a check: **changing phone, or reinstalling**. The Play Store must then be able to give back, once, what you bought.

That check is about the purchase and about nothing else. The network serves only it: your grids are made on the device, your games stay there, and nothing of what you play goes through it — that is the subject of chapter XVI.

17.7 — Why one application, and the purchase inside it

There were two applications, and the reason held in one sentence: a game with no advertising, which collects nothing and talks to no server, must still be funded. **Advertising** requires a library that profiles you — and that is why you will never see a single advert here.

But the split was paid for elsewhere: you had to know which one to install, you could end up with both on the same phone, and the more visible one — the free application — was a dead end from which games had to be moved by hand.

The serious objection remains: a purchase in a game children play. InZeDoku takes no money and never sees a means of payment: it is **Google** that shows its own window, and that window obeys the account settings — a password for every purchase, parental approval on a supervised account. A slip opens a screen; it takes nothing. The same screen is reached by opening the Play Store.

What remains, because it was true before and must stay so: **one single place** to buy, never during a game, and no reminders. Nowhere else will the game ask you for anything.

Chapter XVIII — The web demo

The game also runs in a browser, at <https://inzedoku.inzemobile.com>. Every geometry, every mode, every level, the rules sheet and the seven languages are there.

It is the comfortable way of playing the **very large squares**: a 20×20 on a big screen, with a mouse and a keyboard, has nothing to do with the same board on a phone.

18.1 — What differs from the application

It is a **foretaste**, not the delivered version, and three differences are worth knowing.

The page freezes while a grid is being made. A browser has no separate thread of computation: while the generator works, nothing responds, and the *Cancel* button has no effect. On a large board or a Mystery Calcudoku, that can last several seconds.

The games are saved in this browser. They follow neither your phone, nor another browser, nor another computer. Clear the site data and they disappear.

The first load is heavy. The game embeds its rendering engine; the following times are fast.

18.2 — A game code gives the same grid on both sides

This is the point that matters most, and it has been **measured grid by grid**, not assumed: a game code produces exactly the same grid in the browser and on Android. The two platforms do not, however, compute in the same way — in JavaScript an integer is a floating-point number — and the generator was written so that it changes nothing.

You can therefore begin a game on a phone, copy its state, and continue it in the browser, or the other way round. It is chapter XIII that explains the format.

18.3 — Two good uses

Discovering the game before installing it. Everything is there.

Playing the large grids. The application says so itself: the first time you go beyond nine a side on a phone, a message tells you that those sizes play better on a computer screen. The message never appears if you are already in a browser — offering the web demo to somebody who is already in it would make no sense.

Appendices

Appendix A — The game code, letter by letter

A game code carries everything needed to rebuild a grid : the mode, the shape of the board, its size, the level, the options, and the seed. It does **not** carry the grid — that is recomputed.

G5CVS72-1a2b3c

					the seed, in base 36
					the level, 1 to 4
					the size
					the silhouette — Voronoi only
					the geometry
					the mode (1 or 2 letters)
					the generator version

A.1 — The generator version, at the head

The **G5** prefix identifies the engine that builds the puzzle. The manual examples use engine 5 ; the teaching worksheets use engine 4. The current app supports both engines. The same code rebuilds the same puzzle with its specified engine.

Older **G1** , **G2** and unversioned examples have been replaced. Do not change their prefix by hand : the engine and checksum are part of the code. Use the complete code from this edition of the guide.

A.2 — The check key

A code displayed by the game ends with a second hyphen and **two characters** :

G5SS92-7-GB

This key exists for one precise case, and it is the only one where this format could lie. Strict decoding catches impossible codes — but a typing mistake that happens to land on a **valid** code used to designate another grid, without a word. In a classroom where a code is dictated, that is the most ordinary of errors, and the hardest to understand.

The key is computed on the whole code, **taking into account the position** of each character. That is what makes it able to catch :

- **any mistake on one character**, whatever it is and wherever it is ;
- **any swap of two neighbouring characters** — the commonest typing error, and the one a plain sum would be blind to.

The rest gets through with one chance in one thousand two hundred and ninety-one, that is, less than 0.08 %.

The checksum is optional when reading a code. The updated examples include it, except for the format diagram above. Copy the entire code shown by the game.

A.3 — The letter of the mode

Letter	Mode	Letter	Mode
S	Sudoku	PP	Pictoku · Characters
C	Calcudoku	PT	Pictoku · Land vehicles
R	Colodoku RGB	PA	Pictoku · Aircraft
Y	Colodoku CMY	FC	Pictoku · Farm — horse
WD	Whodoku · Portraits	FA	Pictoku · Farm — animals
GD	Geodoku · Shapes	FE	Pictoku · Farm — machines
HM	Phydoku · Mechanics	EC	Pictoku · Economy
HN	Phydoku · Nuclear	MA	Chemdoku · Acid-base
BJ	Biodoku · Garden	MR	Chemdoku · Redox
AS	Astrodoku · Sky	MB	Chemdoku · Acid-base II

No one-letter code is the prefix of a two-letter one : the reading is therefore unambiguous, and a test checks it.

A.4 — The geometry, the silhouette and the size

Letter	Geometry
S	square
H	hexagonal
T	triangular
M	mosaic
V	Voronoi

On the **Voronoi only**, a silhouette letter follows : S square, H hexagon, O octagon, T triangle.

The **size** fits on a single character : the digits 2 to 9 stand for themselves, then the letters take over.

Char.	Size	Char.	Size	Char.	Size
2 – 9	2 to 9	E	14	L	21
A	10	F	15	M	22
B	11	G	16	N	23

Char.	Size	Char.	Size	Char.	Size
C	12	H	17	O	24
D	13	I	18	P	25
		J	19	Q	26
		K	20		

Depending on the geometry, this number is the side of the square, the radius of the hexagon, the side of the triangle or of the mosaic, or the number of symbols of the Voronoi.

Then comes the **level**, from 1 (Easy) to 4 (Expert).

A.5 — The options, in the canonical order

The order is **imposed**. A code presenting the same options in another order is refused : without that, two different codes would designate the same game.

Rank	Block	Meaning
1	a digit 2 - 5	multidoku : number of grids
2	W	bonus windows
3	R	regular mosaic
4	M	Mystery Calcudoku
5	B	two cells at most (pair)
6	L<n>	maximum result
7	O + letters	allowed operations
8	A + family	family of values
9	F<cells>X<results>[R][V<n>]	fractions

Then the hyphen, and the seed in base 36.

The operations block

0 followed by one letter per operation, in the order + - × ÷ :

Letter	Operation
P	+
M	-
T	×
D	÷

The block only appears if the four operations are not all allowed, and only in Calcudoku.

The **M** of *Mystery* and the **M** of *minus* are not confused : the operation letters are only read **after** the **O** . In the same way, the letter of the « two cells at most » setting is **B** and not **D** — **D** is already the division.

The values block

A followed by a family letter :

Letter	Family
S	Around zero
N	Negative
V	Random

The classic family is not written — it is the default case.

For **V** only, the two bounds follow, **on two characters each, in base 36 shifted by +99**. The shift is not a flourish : a **-** in the options would cut the code in the wrong place, since that is what separates the seed. So **-99...99** becomes **0...198**, that is **00** to **5I** .

The fractions block

F , the denominator of the cells, **X** , the denominator of the results. Then, if need be, **R** for *reduced fractions only* and **V<n>** for the proportion of decimal writing.

The **X** separates the two numbers : without it, **F1260** would read (12, 60) as readily as (126, 0). The final **R** cannot be confused with the **R** of the regular mosaic — that one comes **before** the **F** block, which ends the word.

A.6 — A few codes read out

Code	Reading
G5SS92-7-GB	Sudoku, square 9×9, Medium
G5CS62-b-FG	Calcudoku, square 6×6, Medium
G5YS42-5-F0	Colodoku CMY, square 4×4, Medium
G5SVH52-c-L3	Sudoku, Voronoi, hexagonal silhouette, 5 symbols, Medium
G5SSG1-f-IW	Sudoku, square 16×16, Easy
G5SS623-h-K6	Sudoku, square 6×6, Medium, 3 grids
G5SS92W-j-QQ	Sudoku, square 9×9, Medium, bonus windows
G5CS62M-r-P1	Calcudoku, square 6×6, Medium, Mystery
G5CS62B-v-NW	Calcudoku, square 6×6, Medium, pairs
G5CS52L200PT-11-CB	Calcudoku 5×5, Medium, result ≤ 20 , operations + and ×

Code	Reading
G5CS52AN-1h-RE	Calcudoku 5×5, Medium, negative values
G5CS52AV2S3B-1n-E0	Calcudoku 5×5, Medium, values drawn between 1 and 20
G5CS62F6X60R-1v-JM	Calcudoku 6×6, Medium, fractions 6/60, reduced
G5CS62F6X60RV60-1z-4Y	ditto, with 60 % of decimal writing

A.7 — The state code

The code of a game **in progress** adds a colon and one token per cell to fill, in the order of the cells of the board :

```
G5SS41-2-EC:1.(24).....
```

Token	Meaning
a symbol	the value placed
.	cell left empty
(...)	the candidates of an empty cell, sorted

The **given** cells are skipped : they come from the seed. With nothing written, the encoder returns the bare game code.

On pasting, the number of tokens must fall exactly on the cells to fill of the rebuilt grid. Otherwise the paste is refused — better a refusal than a grid filled one cell out of step.

The form that travels in a QR

In a QR, the same string is written slightly differently : **all in capitals**, and the parentheses of the candidates replaced by **+** and *****.

```
G5SS41-2-EC:1.+24*.....
```

This is not an encoding — it is the same string, still readable by eye, and the game accepts either writing indifferently. The reason is mechanical : a QR packs digits and capitals twice as tightly as the rest, and the parentheses forced it to change regime at every candidate. The gain is two to seven notches of size depending on the grid — that is to say, very concretely, the difference between a code a camera reads and a code it does not.

Appendix B — Every key on the keyboard

Everything that follows works on a computer, in the browser, **and on Android** as soon as a keyboard is plugged in. On a Mac, `⌘` replaces `Ctrl`. The numeric keypad counts as much as the top row, operations included.

B.1 — The complete table

Key	Effect	Conditions
1 ... 9	places the symbol in the selected cell	not in the thematic sets
0	second digit only (10, 20) — alone it places nothing	ditto
A ... Z	places the letter	Sudoku of more than 9 a side
← ↑ → ↓	moves the selection from cell to cell	—
Shift + arrow	extends the selection to the cell reached	pencil mode
Escape	exits : the pencil or the highlighting first, then the screen	—
Space	cycles through the keypad modes	according to the settings
Del , ☒	empties the candidates, otherwise clears the cell — of the whole group	—
Ctrl + Z	undoes	if the undo/redo buttons are active
Ctrl + Shift + Z , Ctrl + Y	redoes	ditto
Ctrl + C	copies the contents of the cell	not in preview
Ctrl + X	cuts : copies, then empties	not in preview, not on a given cell
Ctrl + V	pastes into the cell	not in preview, not on a given cell
+ - * /	places the operation of the cage	Mystery Calcudoku
x X × : ÷ −	ditto — these are how one writes them by hand	ditto

B.2 — With the mouse

Gesture	Effect
click	selects a cell ; cancels a multiple selection
long press	enters the pencil and starts a group
long press, then drag	takes (or releases) the cells crossed
Ctrl + click	takes or releases a cell, immediately
Shift + click	takes everything between the anchor and the cell clicked
two-finger pinch	zooms, on every board

B.3 — Composing a two-digit number

On a board of more than nine symbols, in **Calcudoku** :

Rule	
1 then 2 within the second	writes 12
nothing is written before the end of the typing	a key toggles : writing too soon would break « typing again erases »
a digit only waits if it can begin a number of the grid	in a grid of 12, only the 1 ; below 10 symbols, none
the composed number	counts as only one move of the history
it is written in the cell where it was typed	not in the current selection

What closes a waiting keystroke :

Event	The waiting digit is...
the delay expires	written
arrow, Space , cage operation, Ctrl + C / X / V	written , before the action
tap on the board	written , in the targeted cell
leaving the screen, going into the background	written
Del / ⌫	dropped
Ctrl + Z / Ctrl + Y	dropped

A **held** key repeats nothing on digits and letters — the arrows do.

B.4 — The digits according to the mode

Mode	A digit...
Filling	places the value ; typing it again erases
Pencil	notes the candidate in all the cells taken ; typing it again removes it
Highlighting	lights up the cells carrying that symbol, as long as the key is held

Highlighting **never waits** for the end of a keystroke.

B.5 — What the keyboard does not do

- The **thematic sets** are excluded from the digits : their keypad carries pictograms, and on the chemistry sets a key opens the charge quiz. The arrows, the eraser, the history and the clipboard work there.
- The large **Sudoku** does not compose : it plays in letters, one key per value.
- The **on-screen** keypad never composes : it already has one key per value.

Appendix C — Sizes, modes and limits

Three summary tables, to settle things at a glance.

C.1 — Which mode on which board

Board	Sizes	Sudoku	Calcudoku	Colodoku	Sets
Square	2 – 26	✓	✓	3 – 8	4 – 6
Hexagonal	radius 2 – 4	✓	✓	radii 2 – 3	—
Triangular	side 6 or 9	✓	✓	Latin variant, side 5 – 7	—
Mosaic	side 6 – 9	✓	✓	sides 6 – 7	—
Voronoi	5 – 8 symbols	5 – 6	5 – 8	—	—

Option	Square	Hexagonal	Triangular	Mosaic	Voronoi
Multidoku	4, 6, 8, 9	radii 2 – 3	—	—	—
Bonus windows	✓	✓	✓	—	—
Fractions	✓ (Calcudoku)	—	—	—	—
Value families	✓	✓	✓	✓	✓
Mystery Calcudoku	✓	✓	✓	✓	✓
Cage settings	✓	✓	✓	✓	✓

C.2 — The blocks of the square

The block is the divisor closest to the square root, the height not exceeding the width. A **prime** side has no blocks at all : the grid is a Latin square.

Side	Blocks	Side	Blocks	Side	Blocks
4	2 × 2	12	3 × 4	20	4 × 5
6	2 × 3	14	2 × 7	21	3 × 7
8	2 × 4	15	3 × 5	22	2 × 11
9	3 × 3	16	4 × 4	24	4 × 6
10	2 × 5	18	3 × 6	25	5 × 5
				26	2 × 13

Calcudoku **never** has blocks, whatever the size.

C.3 — What the level sets, according to the board

Board and mode	What the level drives
Square ≤ 9 , hexagon, mosaic	a search for a seed rated at the right level
Square > 9	how far the generator empties the grid
Triangle, Sudoku	how far the generator empties the grid
Triangle, Calcudoku	a search for a seed
Voronoi	the number of zones of the second family — hence the geometry

Given cells on a **large square**, according to the level :

Level	Sudoku	Calcudoku
Easy	$\approx 55 \%$	$\approx 30 \%$
Medium	$\approx 48 \%$	$\approx 20 \%$
Hard	$\approx 42 \%$	$\approx 13 \%$
Expert	the floor	$\approx 6 \%$

The **floor** of Sudoku — below it, the solution ceases to be unique :

Side	10	12	16	20	26
Given cells	31 %	35 %	41 %	43 %	50 %

C.4 — The free range

Board and mode	Free up to	Beyond
Square · Sudoku	9×9	25 %
Square · Calcudoku	5×5	25 %
Square · Colodoku	4×4	35 %
Square · Pictoku	5×5	50 %
Square · science sets	5×5	30 %
Hexagonal	radius 2	35 %
Triangular	side 6	35 %
Voronoi	5 symbols	35 %
Mosaic	—	25 %

Outside the free range whatever the size, at 25 % : the **Hard** and **Expert** levels, **Mystery Calcudoku**, the sets of operations outside $+$, $+-$, $+-\times$ and $+-\times:$, and the **values outside 1 to 10**. Fractions, for their part, follow the size of Calcudoku : free up to 5×5 .

When several reasons apply, the lowest share wins. Nothing is forbidden : only the possibility of **finishing** the grid is bounded.

C.5 — The palettes of Colodoku

Colours	Palette
3	R, G, B at full dose
4	R, G, B at half dose, plus a grey at half dose
5	a neutral, R G B at a third, a grey at two thirds
6	six hues at a third
7	the six thirds, plus a neutral
8	six hues at a quarter, a grey at a quarter, a neutral

Each palette sums exactly to the target : a complete line therefore makes precisely black (CMY) or precisely white (RGB).

Resources

Download

- PDF format (manual_v1.33_en_20261001.pdf)
- EPUB format (manual_v1.33_en_20261001.epub)
- Markdown format (manual_v1.33_en_20261001.md)

Where to find the game

Resource	Address
Play in the browser	https://inzedoku.inzemobile.com
The publisher	https://inzemobile.com
Write to the team	sudoku@inzemobile.com

The game also carries its own documentation: the 📖 **Game guide** entry in the home menu opens a presentation in sixteen sections, available in the seven languages of the application and readable offline like everything else. This manual goes further — it details the settings one by one and gives the complete tables in the appendices — but the built-in guide remains what you open when the phone is in your hand.

Reporting something

The ✉ **Your feedback** entry in the same menu prepares a message: it sends nothing by itself. It attaches the version, the language, the configuration and above all the current **game code** — that is what makes it possible to replay exactly the grid you are talking about. The context block can be edited, and you may empty it.

Lexique

alphabet — The set of values the cells of a grid carry. It runs 1 to k by default, but can be made of negative integers, of numbers drawn at random, or of fractions — and it may hold more values than the grid has cells per row. → Ch. VIII, Ch. IX

badge — The small disc set on a cage in Colodoku and in the thematic sets : upper half the target, lower half what is currently placed. It turns solid when the cage is complete and correct. → Ch. VI, Ch. IX

block / region — A sub-rectangle of the square that must contain every symbol once. The block size is the divisor closest to the square root of the side. → Ch. VII

bonus windows — Extra all-different groups, shaded on the board. The pattern — diagonals, windows, block centres, asterisk, girandole — is drawn from the seed. → Ch. VII

cage — A group of neighbouring cells, outlined by a thick border, that announces a calculation (Calcudoku), a colour to obtain (Colodoku) or a recipe to compose (thematic sets). A one-cell cage has no operation : it gives its value directly. → Ch. V, Ch. VI, Ch. VIII

candidate — A value written small in an empty cell, in pencil. Candidates are saved with the game and go through the history. → Ch. XI

domino — A two-cell cage. The « two cells at most » setting produces only those ; subtraction alone forces them anyway, since « - » only makes sense on two cells. → Ch. VIII

entity — In the pictogram sets of the Kids family, a complete character or object composed of the six pieces of one 2×3 region. → Ch. IX

game code — A short string — `CVS72-1a2b3c` — carrying the mode, the geometry, the outline, the size, the level, the options and the seed. It is enough to rebuild a grid exactly. A bare seed would not be : the same value gives a different grid depending on the configuration. → Ch. XIII, App. A

given / clue — A cell whose value is already set when the game starts. It cannot be changed, and the progress percentage does not count it. → Ch. X, Ch. IV

Latin square — A grid held together by its rows and columns alone, without blocks. That is what a square of prime side is, and what a Calcudoku or a Colodoku always is. → Ch. VII, Ch. VI

Multidoku — Two to five grids overlapping on a single board, each with its own coloured frame. The layout is drawn from the seed of the game. → Ch. VII

Mystery Calcudoku — A Calcudoku variant in which cages show only the result, never the operation. The game is won only if the cells AND the operations are right. → Ch. VIII

outline — The contour of a Voronoi board : square, hexagon, octagon or triangle. The tiling inside is irregular and drawn from the seed. → Ch. VII

palette — The set of colours of a Colodoku grid. Every palette sums exactly to the target, so that a complete line makes precisely black (CMY) or precisely white (RGB). → Ch. VI

preview — A read-only view of the solution, which marks in passing the wrong values already placed. It does not touch the game, and the keypad is entirely disabled while it is open. → Ch. XI

recipe — In the thematic sets, what a cage asks you to compose : a chemical species, an experiment, a character. The equivalent of the target of a Calcudoku cage. → Ch. IX

rosette — A group of six small triangles fused into a hexagon on a mosaic board. The hexagon belongs to every row of its six original triangles. → Ch. VII

seed — The number the whole grid is derived from. The same seed always yields the same grid, on any device — that is what makes games shareable and saved games tiny, since they hold not the grid but the means of recomputing it. → Ch. I, Ch. XIII

state code — The game code followed by a colon and one token per cell to fill : it carries a game IN PROGRESS, pencil marks included. → Ch. XIII, App. A

value family — One of the five ways of building the alphabet : Classics, Around zero, Negative, Random, Fractions. They are mutually exclusive. → Ch. VIII

zone family — On a Voronoi board, a set of zones that never overlap one another — the analogue of « all the rows » or « all the blocks ». A board carries two, each with its own frame colour. → Ch. VII

Table of figures and tables

Figures

N°	Titre	Page
1.1	Home : five tiles showing the current configuration, and one button to play	7
3.1	The playing screen, here a 6×6 Calcudoku at Medium level	13
3.2	The 4 placed at the top contradicts the given 4 just below : both are marked	16
4.1	A 9×9 square Sudoku, Medium level — code G5SS92-7-GB	19
5.1	A 6×6 square Calcudoku, Medium level — code G5CS62-b-FG	22
5.2	A hexagonal Calcudoku of radius 2 — code G5-CH21-4-BY	23
6.1	A 4×4 CMY Colodoku, Easy level — code G5YS41-5-FI	25
7.1	A 16×16 Sudoku : the symbols are letters — code G5SSG1-f-IW	31
7.2	Three overlapping 6×6 grids — code G5SS623-h-K6	34
7.3	The shaded cells form extra groups — code G5SS92W-j-QQ	35
8.1	A 5×5 Calcudoku in additions only — code G5CS510P-15-QQ	37
8.2	In subtractions only, every cage is a domino — code G5CS620M-17-QZ	38
8.3	A 5×5 Calcudoku in values centred on zero — code G5CS52AS-1b-Q0	39
8.4	Every cage fits in two cells — code G5CS62B-v-NW	40
8.5	A 6×6 Calcudoku in reduced fractions, denominators 6 and 60 — code G5CS62F6X60R-1v-JM	41
11.1	The preview shows the solution ; the keypad is off	54

N°	Titre	Page
13.1	The list : in progress or finished, with timer, progress, padlock, comment and ways out	59
13.2	The three ways of bringing a game in, in the same place	61
13.3	The QR carries the blank grid or the game in progress, as you	
choose ; the code stays legible underneath	63	
13.4	What is printed, whether one wants the solution, and — in	
the full version — the solving of the grid or that of the player	65	

Index thématique

ARITHMETIC

decimal notation → Ch. VIII
dominoes → Ch. VIII
fractions → Ch. VIII
maximum result → Ch. VIII
multiplication tables → Ch. VIII, Ch. XVII
negative numbers → Ch. VIII
operations → Ch. VIII, Ch. V
repeating bar → Ch. VIII
subtraction only → Ch. VIII

BOARDS

blocks → Ch. VII, App. C
bonus windows → Ch. VII
hexagon → Ch. VII
large square → Ch. VII, Ch. X, Ch. XVIII
mosaic → Ch. VII
Multidoku → Ch. VII, Ch. XV
square → Ch. VII, App. C
triangle → Ch. VII
Voronoi → Ch. VII, Ch. X

MODES

Astrodoku → Ch. IX
Biodoku → Ch. IX
Calcudoku → Ch. V, Ch. VIII
Chemdoku → Ch. IX
Colodoku → Ch. VI, Ch. XV
Geodoku → Ch. IX
Mystery Calcudoku → Ch. VIII, Ch. XVII
Phydoku → Ch. IX
Pictoku → Ch. IX
Sudoku → Ch. IV, Ch. VII, Ch. X
Whodoku → Ch. IX

PLAYING

copying a cell → Ch. XI, App. B
difficulty → Ch. X, App. C
multiple selection → Ch. XI, Ch. XII
pencil → Ch. XI, Ch. XII
physical keyboard → Ch. XII, App. B
preview → Ch. XI
timer → Ch. XIII, Ch. III
torch → Ch. XI
undo and redo → Ch. XI, Ch. XIV
zoom → Ch. III, Ch. XV

SETTINGS

colour blindness → Ch. XV, Ch. XIV
conflicts → Ch. XIV, Ch. III
dark theme → Ch. XIV
languages → Ch. II, Ch. VIII

SHARING

game code → Ch. XIII, App. A
games in progress → Ch. XIII, Ch. II
seed → Ch. I, Ch. XIII
state code → Ch. XIII, App. A, Ch. XVII

THE APP

built-in guide → Ch. II
free version → Ch. XVII, App. C
full version → Ch. XVII
player feedback → Ch. XVI
privacy → Ch. XVI
web demo → Ch. XVIII, Ch. VII

Index alphabétique

A

Astrodoku → Ch. IX

B

Biodoku → Ch. IX

blocks → Ch. VII, App. C

bonus windows → Ch. VII

built-in guide → Ch. II

C

Calcudoku → Ch. V, Ch. VIII

Chemdoku → Ch. IX

Colodoku → Ch. VI, Ch. XV

colour blindness → Ch. XV, Ch. XIV

conflicts → Ch. XIV, Ch. III

copying a cell → Ch. XI, App. B

D

dark theme → Ch. XIV

decimal notation → Ch. VIII

difficulty → Ch. X, App. C

dominoes → Ch. VIII

F

fractions → Ch. VIII

free version → Ch. XVII, App. C

full version → Ch. XVII

G

game code → Ch. XIII, App. A

games in progress → Ch. XIII, Ch. II

Geodoku → Ch. IX

H

hexagon → Ch. VII

L

languages → Ch. II, Ch. VIII

large square → Ch. VII, Ch. X, Ch. XVIII

M

maximum result → Ch. VIII
mosaic → Ch. VII
Multidoku → Ch. VII, Ch. XV
multiple selection → Ch. XI, Ch. XII
multiplication tables → Ch. VIII, Ch. XVII
Mystery Calcudoku → Ch. VIII, Ch. XVII

N

negative numbers → Ch. VIII

O

operations → Ch. VIII, Ch. V

P

pencil → Ch. XI, Ch. XII
Phydoku → Ch. IX
physical keyboard → Ch. XII, App. B
Pictoku → Ch. IX
player feedback → Ch. XVI
preview → Ch. XI
privacy → Ch. XVI

R

repeating bar → Ch. VIII

S

seed → Ch. I, Ch. XIII
square → Ch. VII, App. C
state code → Ch. XIII, App. A, Ch. XVII
subtraction only → Ch. VIII
Sudoku → Ch. IV, Ch. VII, Ch. X

T

timer → Ch. XIII, Ch. III
torch → Ch. XI
triangle → Ch. VII

U

undo and redo → Ch. XI, Ch. XIV

V

Voronoi → Ch. VII, Ch. X

W

web demo → Ch. XVIII, Ch. VII

Whodoku → Ch. IX

Z

zoom → Ch. III, Ch. XV

One engine, five board shapes.

InZeDoku takes the rule everyone knows — never the same symbol twice in the same group — and runs it on five geometries, three ways of filling a cell and sizes ranging from a four-cell toy to a six-hundred-and-seventy-six-cell marathon.

This manual goes through them all : Sudoku, Calcudoku and its cages, Colodoku and its colour mixes ; the square, the hexagon, the triangle, the mosaic and the Voronoi tiling ; the fractions, the negative numbers and the decimal notation that turn Calcudoku into an arithmetic workbook ; the aids, the keyboard, the game codes and the settings.

Every illustrated grid carries its code : you can reopen it in the game and play what you are reading.

Every grid is built on the device, offline. No advertising, no account, no data collection.

