

Learning to count with InZeDoku

A file of teaching proposals

Version 1.33 — 1 octobre 2026

Proposals for the classroom — you are master in your own house.

InZeMobile

TABLE DES MATIÈRES

Foreword.....	3
Partie 1 — Before you begin	
Chapter I — What this file proposes	7
Chapter II — What you need.....	9
Chapter III — One class, a single grid	12
Partie 2 — One lesson, four stages	
Chapter IV — Together, at the board	19
Chapter V — In groups	22
Chapter VI — Alone, in class.....	24
Chapter VII — As homework, and the correction	26
Partie 3 — The worksheets, by notion	
Chapter VIII — Additions	31
Chapter IX — Subtractions	33
Chapter X — Multiplication tables.....	36
Chapter XI — The four operations	40
Chapter XII — Fractions	43
Chapter XIII — Decimal numbers	46
Chapter XIV — Without a single digit	49
Appendices	
Appendix A — All the worksheets at a glance.....	55
Appendix B — What the free version allows	58
Appendix C — Where to look in the player's manual.....	61
Appendix D — The rules of deduction, one by one	63
Lexique	69
Table des figures	
Index thématique.....	73
Index alphabétique	74

Foreword

This file is addressed to you, who teach. It is not addressed to the child.

InZeDoku is a game, and it has its manual — *The player's manual*, two hundred and twenty-four pages describing every mode, every geometry, every setting. This file does not do that again. It answers another question : **what does one do with it, with twenty-five pupils and fifty minutes ?**

Three things sum it up.

Nothing that follows is an instruction. You know your class, its habits, its real level, and what worked last year. This file proposes a progression because one must propose one ; you will take it from the middle, backwards, or not at all. Where it writes « one does », read « one could ».

Everything described here is played in the free version. Not one worksheet asks you to buy anything. That is not a convenience of drafting : a school has no budget for an application, and a file that forgot it would be of no use. The rare places where the full version would open something more are pointed out once, without insisting.

The grids of this file can be reproduced identically. Every worksheet carries a GAME CODE[^lex-convention] : a short string of characters that the pupil types, and which gives them *exactly* the grid of the worksheet. Not a grid of the same kind — the same one. That is what makes it possible to correct at the board, to set homework, to compare two papers. These codes were produced by the game itself and are checked at every version.

The game works offline, asks for no account, shows no advertising and collects nothing. There is therefore nothing to prepare on the administrative side, and nothing for parents to sign.

Partie 1

Before you begin

Chapter I — What this file proposes

1.1 — A sudoku grid where one has to count

InZeDoku is a sudoku game, and the mode that concerns us here is called **Calcudoku**. The principle holds in two sentences. As in a sudoku, every row and every column carries every number once and once only. But the grid is further cut into small groups of cells, each forming a **CAGE** — and each announces a result and an operation : **7+** means that the numbers of the cage make 7 when added together.

The child therefore has two things to do at the same time, and they lean on each other : **counting**, to know what could go in a cage, and **deducing**, to know which of the possibilities stands up in the row and the column. This is not a counting exercise disguised as a game : it is a problem in which the counting is good for something, and in which a mistake in the counting shows — the grid ends up contradicting itself.

1.2 — Two difficulties, two knobs

Here is what makes this game a classroom tool rather than a pastime, and it is the point to remember from this whole file : **the difficulty of the counting and the difficulty of the deduction are set separately.**

The level — Easy, Medium, Hard, Expert — touches only the deduction : how much information the grid gives, how far one has to reason. The numbers are set elsewhere : which operations appear, which numbers the cells carry, what size the cages are allowed to reach.

Concretely, that means one can give a seven-year-old a grid whose **arithmetic is exactly within reach** — additions of two numbers between 1 and 5 — on a board small enough for them to finish. And that one can give the same child, three months later, exactly the same board with multiplications. The sudoku has not moved ; what it works on has changed.

That is also what explains the shape of this file : the worksheets of the third part are sorted **by mathematical notion**, and not by age. The order in which one learns the operations is roughly the same everywhere ; the age at which one learns them is not. Every worksheet therefore gives the notion first, and an age in brackets, as a landmark and nothing more.

1.3 — The real lesson : thinking backwards

There is one thing this mode does that an exercise sheet does not, and it is worth saying before all the rest.

Faced with a **7+** cage of two cells, the child does not compute $3 + 4$. They look for **which pairs make 7**. Faced with **15×**, they do not compute 3×5 : they look for **which factors give 15**. It is the operation read backwards — decomposition rather than addition, factorisation rather than product.

That holds for all four operations, and it is **more demanding** than direct counting : there is not one answer but a family of answers, and one has to hold them all in mind long enough to see which one the row allows. That kind of concentration is precisely what makes things stick. A child who has looked ten times for which pairs make 12 knows their tables better than one who has written « $3 \times 4 = 12$ » ten times.

That is also why the same tables, worked on here, do not feel like a repetition — and why a child asks for more.

1.4 — One setting to know straight away

Since it has just come up : « **Two cells at most** ».

It is a switch of the $\div$ tile, and it forces every cage to hold in a pair. A multiplication cage then becomes exactly what one is after — $21 \times$ on two cells is 3×7 and nothing else. For a lesson on tables or on simple additions, it is the setting that changes everything.

It is announced here, from the very first chapter, because a teacher who looks for it and does not find it closes the file concluding that the game cannot do it — whereas it does it very well. The worksheet of chapter X goes into the detail.

1.5 — The progression this file suggests

The second part describes one lesson in four stages : the grid is shown **together** at the board, then done **in groups**, then **alone in class**, then **at home**. These are not four compulsory lessons ; it is a trajectory, covered in an hour with a class that follows, or over three weeks with a class that is discovering.

There is nothing original about that division. What is original is that the game allows it to be done with **the same grid** from one end to the other, and whatever the support : projected on the board, opened on a tablet, or photocopied. Any one of the four stages can therefore do without a screen — all four, if you wish — without the class changing exercise. Chapter III explains why, and chapter II what is needed for it : less than one thinks.

1.6 — What this file does not do

It prescribes nothing. You are master in your own house, and a class is not steered from a book.

Nor does it replace the player's manual : as soon as a question is about *how the game works* rather than *what to do with it*, this file refers to the manual instead of copying it out. Appendix C gives the table of cross-references.

And it does not claim that the game replaces the teaching of arithmetic. A child who has never set out an addition will not learn to here. What the game does well is **training** : repetition that does not look like repetition, and a check that comes from the grid and not from the adult.

Chapter II — What you need

2.1 — The game exists in three forms

There is **the Android application**, the one installed on a phone or a tablet. It works entirely offline : once installed, it no longer needs a network, and the school has nothing to open in its firewall.

There is **the web demo**, which opens in a browser, on any computer, with nothing to install. That is the one to project. Its name of demo is deserved : the page **freezes while it makes a grid**, because a browser does not give it a separate thread of computation. On the small boards this file uses — four and five cells a side — that lasts a fraction of a second and goes unnoticed. On a board twenty cells a side, it lasts, and the screen looks stuck.

And there is **the same game, projected from a tablet**, by mirroring its screen onto the projector. That is the solution if your room has no computer, and it has one advantage : the tablet never freezes.

All three give **the same grid for the same code**. That is not a supposition : the two engines were compared grid by grid.

2.2 — Projecting, the first time

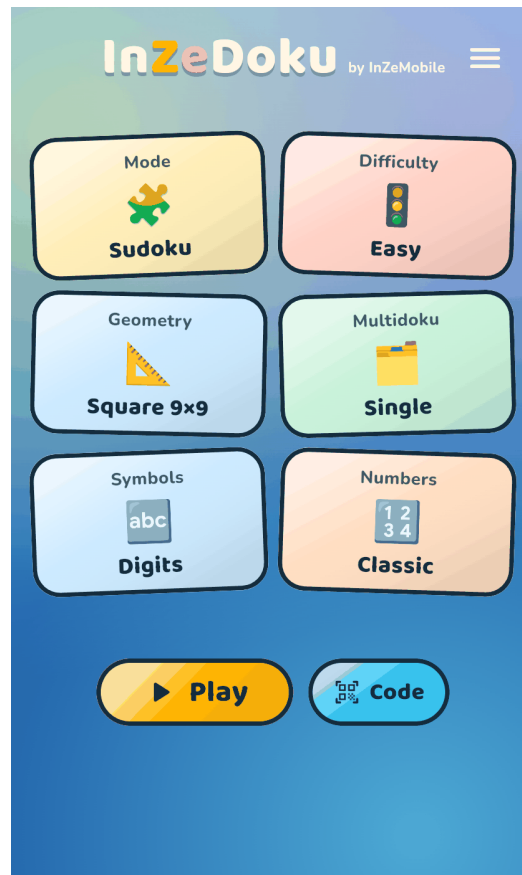


Figure 2.1 — The home screen : every tile is a setting, and displays its value

The home screen is made of coloured tiles. Each is a setting — the mode, the difficulty, the size of the board, the operations, the numbers — and each **displays the value chosen** without having to be opened. That is convenient at the board: the class sees what is being set before playing.

For a first projection, there is nothing to prepare. One opens the demo, types a code in the  menu (chapter III), and the grid is there.

One piece of advice that looks like nothing : **make it bigger**. The board is pinched with two fingers on a tablet, and zoomed in the browser on a computer (`Ctrl` and `+`). A grid five cells a side, projected large, can be read from the back of the room; the same one at screen size cannot be read from the third row.

2.3 — The keyboard, if you have one

On a computer — and on any tablet to which a keyboard is plugged — one plays with the keyboard. The digits place a value, the arrows move the cell, `Space` switches to the pencil, `Del` erases, `Ctrl+Z` undoes.

That changes the demonstration at the board: one does not hunt for a cell with a finger on a mirrored screen, one types. And `Ctrl+Z` is precious in front of a class — a deliberate mistake, shown and then undone, is worth more than a long speech.

2.4 — What you do not need

No account, no registration, no e-mail address — neither yours nor the pupils'. The game asks for nothing and sends nothing.

No connection during the lesson, except to load the page of the demo the first time.

No school licence : the free version is enough for everything this file describes. Appendix B says exactly where the border runs, so that you do not have to discover it in the middle of a lesson.

And above all, **no device for the pupils**. The point deserves to be stated plainly, because one hardly expects it from a file built around an application : the grids can be printed, and a whole class can do the lesson in pencil, on a photocopy, without a single child touching a screen. All that is then needed is one machine — yours — to prepare the grids, and a photocopier.

This is not a makeshift for a poorly equipped room. The counting and the deducing happen in the head, not in the device ; the screen is only a convenient way of handing out the grid and checking it. What one loses in going over to paper is real but short — the immediate correction, the pencil that erases without a trace, the step backwards — and it is replaced by what existed before it : a rubber, and you.

The grid, in any case, travels as a code of a few characters, which is the subject of the next chapter — where we shall also see how to get these sheets out.

Chapter III — One class, a single grid

3.1 — The problem it solves

A sudoku game that draws a grid at random is unusable in class. Every pupil has their own, nobody can help anybody else without getting the wrong board, and correction at the board no longer means anything.

InZeDoku does not store its grids : it **makes** them from a number, and the same number always gives back the same grid, on any device whatever. That is what makes it possible to designate it by a short string instead of carrying it about.

3.2 — What a code looks like

Take the one of the first worksheet of this file :

```
G4CS410P-a30y6s-IU
```

It reads from left to right. **G1** first : the **generator version**, that is to say the way of making grids that produced this one. Then **C** for Calcudoku, **S** for a square board, **4** for four cells a side, **1** for the Easy level, and **OP** — the operations block, here addition alone. After the hyphen, **b** is the number of the grid. And **RM** , after the second hyphen, is a **check key**.

The version at the head looks like a detail and is not one. The game does not store its grids, it remakes them — and a grid number only gives back the right one with **the generator that made it**. By carrying it, the code tells the application which one to use instead of making it guess. And since the earlier ways of making them are kept, a code never stops working : the one you hand out this year will open the same grid in five years.

The check key deserves a word because it was added with exactly your situation in mind : *if one mistypes the code, one would not want to get another grid*. Without it, a typing mistake that unluckily lands on a valid code opens a neighbouring grid, in silence, and the pupil works on something other than their classmates. With it, **any mistake on one character and any swap of two neighbouring characters are refused**. The game says « code not recognised », which is infinitely preferable.

The key is optional : codes that do not carry one remain valid. But all the codes of this file carry one, and that is why.

3.3 — Where one types it



Figure 3.1 — The **Code** button, to the right of Play: the only way in for a game

On the home screen, to the right of the Play button: the **Code** pill. One types, and **the configuration is displayed in plain words under the field** as one goes — mode, geometry, level. The pupil therefore sees what they are about to open before opening it, which catches the typing mistake even better than the key does.

The button is in that place for a reason that concerns you: entering a code is a **start**, on the same footing as Play, and not a setting. Buried in a menu, it became impossible to find for an eight-year-old to whom a code is being dictated at the board.

Two pills complete the field: **Paste a game**, for a code received by messaging, and **Scan a QR**.

3.4 — The QR, when the screen is projected



Figure 3.2 — The QR displayed on the board ; the class scans it, everyone has the grid

This is the most efficient gesture in this whole file, and it is worth repeating once to make it stick.

You open the grid on the projected screen. Under the board, a button displays its **QR**. The pupils press **Code** → **Scan a QR**, aim at the board, and the grid opens **directly**: a QR has no typing mistake to catch, so there is nothing to confirm. Twenty-five pupils in one minute, with no dictating, no spelling out.

The camera is asked for **only at that moment**, never at installation, and a pupil who refuses is not stuck: the code written on the board still works.

And you do not even need to project anything: **every worksheet of this file carries the QR of its grid**, next to its code. A photocopied page, or the book open under the visualiser's camera, is enough to hand the grid to the whole class.

3.5 — Passing on a grid already begun

There is a second form of the code, longer, which carries **what is already written in the grid** — the values placed and the pencilled candidates. It is the STATE CODE, and one gets it by the grid-shaped button, next to the code.

In class, it serves one precise purpose : **taking up a pupil's work where they have got to**. They send you their string, you open it on the projected screen, and the class looks at their reasoning rather than at a blank grid.

It travels both ways, and that is where the flexibility of the lesson comes from. A grid begun together at the board is finished at home : if time ran short, the lesson is not lost and the class does not fall behind.

A game with no pencilled candidates at all holds in about ten characters — it is the game code itself. It is the candidates that weigh, and they alone : a grid of nine cells pencilled all over comes close to nine hundred characters. Beyond what a camera can read back, the game refuses to display a QR and sends you back to the string, rather than showing an unreadable mosaic.

Appendix A gathers all the codes of this file on one page, ready to print.

3.6 — When there is neither screen nor telephone

That page is one we print for you. But the application can do as much with any grid, and that is what remains when the room has no projector, when the telephones stay in the schoolbags, or when the homework has to go out on paper.

A grid to photocopy. Open the grid, then the last button of the code row, under the board. A small window opens. **Blank grid** is already switched on there, and that is what you need : the sheet comes out with nothing filled in. One photocopy per pupil, and the whole class has the same board.

The answer sheet, only if you ask for it. The second switch, **Add the solution**, is off. Switch it on and a second page comes out with the grid solved — your copy, not the pupils'. It never switches itself on, which avoids the answer sheet slipped by mistake into the pile one hands out.

A pupil's path. A third switch appears as soon as a game counts at least one move : **Player's path**. It brings out a page that takes up the moves **in the order in which they were made**, each with the rule that applied at the time, and the grid at the head with the wrong cells marked. It is the tool for the correction : chapter VII explains how to read it, and above all what it does not say.

The sheet of codes. If you have prepared several grids in advance, « Games in progress » prints them all at once : one thumbnail per game, with its code and its QR. That is the sheet one keeps beside oneself during the lesson — the code to dictate, the QR to have scanned, with nothing to find again in the application in the middle of the class.

That list is also your filing cabinet. A **finished** grid no longer leaves it : it switches to the **Finished** tab, which lets you read it again or print it long after the lesson. Every game accepts a **comment** — the pupil's name, the date, what got stuck — which is displayed directly in the list, without opening anything ; and a **padlock** puts it out of harm's

way, including from deletion in bulk and from the ceiling of twenty games. Enough to keep the papers of one lesson without fearing that an unlucky gesture will carry them off.

One detail that counts : the QR of that sheet carries the **blank grid**, and never your game in progress. What your pupils get is the board, not your attempts.

Every grid sheet also carries its code and its QR. The pupil who prefers to work on the screen therefore finds exactly the grid of their photocopy, and you have nothing to copy out on the board.

Partie 2

One lesson, four stages

Chapter IV — Together, at the board

What one is aiming at : that the rule be understood. Not that a grid be finished.

4.1 — Beginning with a sudoku, with no counting

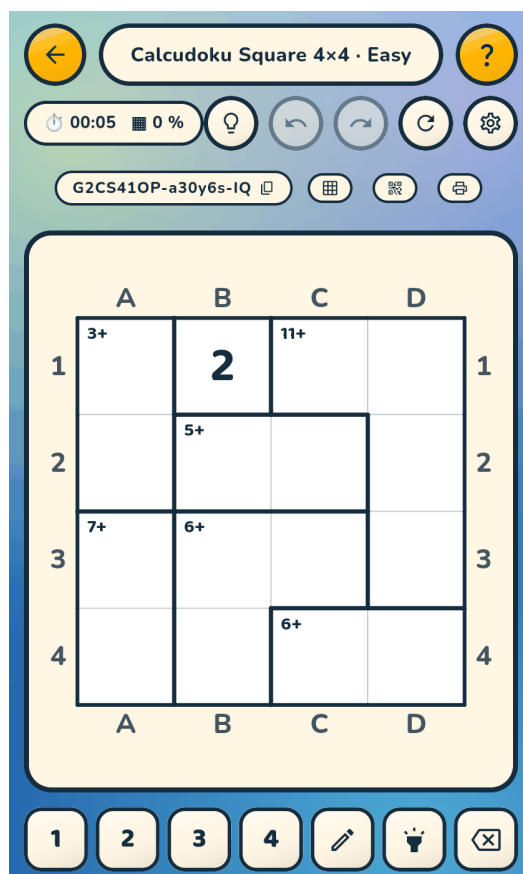


Figure 4.1 — Four cells a side : the rule holds in one screen

Before the counting, the rule of the sudoku. A board of **four cells a side** is enough, and it has one advantage over the classic 9×9 : one goes round it aloud in three minutes.

Grid	Code	To scan
Sudoku, 4 × 4, Easy, twelve cells to fill	G4SS41-d28-j6x-3B	

Project it and put the question to the class rather than to one pupil: *where are we sure of what to put?* There is always a cell where three of the four numbers are already visible on the row, the column or the block. One fills it in, and one starts again. In twelve cells, everyone has seen the mechanism.

The board carries its **landmarks** — the letters of the columns, the numbers of the rows — which spares you pointing a finger at a screen that not everyone sees from the same angle: one asks *and in C2?*, and the class looks in the same place. It is also the notation of appendix D and the one the application uses when it answers a pupil who is stuck.

If you have only one moment, skip this step and go straight to the Calcudoku: the rule of the sudoku explains itself along the way.

4.2 — Then the same thing, with cages

Grid	Code	To scan
Calcudoku, 4 × 4, additions only, Easy, fifteen cells	G4CS410P-a30y6s-IU	

This grid gives **only one cell**. All the rest comes from the cages, and that is what one wants: the child cannot begin by copying, they have to count.

Show first what a label means. 7+ on two cells: two numbers that make 7. Ask which — on a board of four, they are 3 and 4, and there is no other possibility. That is the moment when the class understands that **the cage narrows the choices**, and that is all one needs to have understood in order to play.

Then take a cage of three cells and let them search: there are several possible answers, and it is the row or the column that decides. The second idea is there, and there is no third one.

4.3 — Making a mistake in front of everyone

Deliberately place a wrong value, and leave it. Two cells further on, the grid contradicts itself — the same number twice in a row, and the game **crosses out both**.

That is the most useful moment of the lesson. It is not you who corrects, it is the grid; and the pupil who makes a mistake later on will find out the same way, without having to put up a hand. Then show `Ctrl+Z`, or the undo button, and set off again.

4.4 — Two mistakes not to make

Do not finish the grid. A grid finished at the board is a grid the class no longer wants to do. Stop after five or six cells.

Do not explain the settings. The operations, the numbers, the levels : none of that is of any interest to a pupil, and all of it is your work. At this stage, the only thing that counts is the rule of the game.

Chapter V — In groups

What one is aiming at : that the reasoning be said aloud.

5.1 — Why the group, here

A Calcudoku lends itself badly to silent work and well to work in twos or threes, for a reason that lies in its structure : **the two halves of the reasoning divide up naturally**. One looks for what the cage allows — « 7 in two cells is 3 and 4 » —, the other looks for what the row forbids. The two answer each other, and they have to talk.

Two or three pupils per device, no more. With four, two are watching.

5.2 — The same grid for every group

Grid	Code	To scan
Calcudoku, 5×5 , + - , Easy, twenty-four cells	G4CS510PM- edxmz1-3R	

This is where the code takes on its full meaning. All the groups work on **exactly the same grid**, and therefore :

- a group that is stuck can be sent to see another group, and what it finds there will genuinely be of use ;
- a remark made to one group holds for the whole class, and you can announce it aloud without giving away anything useless ;
- the correction, at the end, is done on the board that everyone has before their eyes.

The QR displayed on the board hands out the grid in one minute (chapter III). Failing that, the code written in chalk : eleven characters.

5.3 — What you do meanwhile

You go round, and you do not give values. Three questions are almost always enough, and they work on any grid whatever :

- *Which is the most constrained cage?* — the one of two cells with a large or a very small result. That is always where one gets in.
- *What does this row forbid?* — the question the groups forget, because they are looking at the cages.
- *Are you sure, or are you trying?* — and if it is an attempt, it has a place to live : the pencil.

5.4 — The pencil, to be shown now

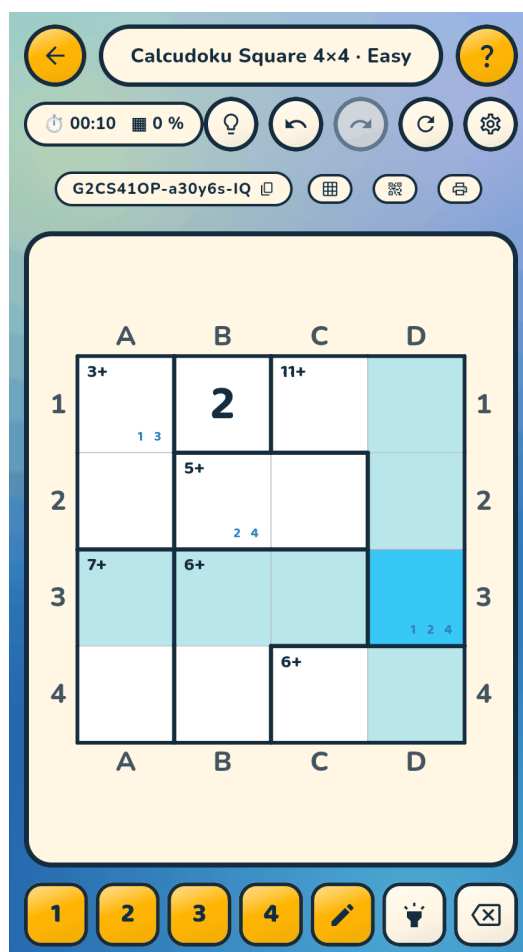


Figure 5.1 — The candidates are written small in the corner of the cell

The pencil button switches the keyboard over: a digit typed no longer fills the cell, it is **noted small** in it. Type it a second time and it disappears.

It is the tool of group work. « 3 or 4 » written in a cell is shared information, which both pupils see and discuss; the same information in the head of one of them is of no use to anybody. Show it at the board before starting the groups — thirty seconds is enough.

A long press on a cell enters the pencil and allows, by sliding the finger, **noting the same candidate in several cells at once**. Useful when one wants to mark « these three cells are the only ones that can carry the 5 ».

Chapter VI — Alone, in class

What one is aiming at : that everyone goes right through one grid, once.

6.1 — Finishing counts more than finishing fast

This is the moment when one discovers that the class does not move at the same pace, and that is normal : the Calcudoku mixes two skills, and a pupil who is quick at counting may be slow at deducing — and the other way round too.

Grid	Code	To scan
Calcudoku, 5×5 , + - , Medium, twenty-three cells	G4CS520PM-dc93mz - WS	

This board is the same as the one in groups, one notch harder. The continuity is deliberate : the pupil finds a grid whose shape they know, and no longer has to learn the interface at the same time as the reasoning.

The game permanently displays **the time elapsed and the percentage filled**. The percentage counts only the cells the pupil has to fill — the given cells do not enter into it, they say nothing about how far they have got. It is a useful landmark for you : at a glance at a screen, you know where a pupil stands without speaking to them.

Do not compare the stopwatches aloud. The clock is for the one who is playing, not for a ranking.

6.2 — What to do with a pupil who is stuck

Three levers, from the lightest to the heaviest.

The pencil, first : a pupil who is stuck has almost always stopped writing their candidates. Asking them to note, in three cells, what is still possible unblocks them more often than a hint does.

Then an easier grid : the same board at the Easy level is another code, given in appendix A. Changing level is not a failure, it is a setting — and the pupil does not lose face, since they change grid, not category.

The eye, as a last resort. The eye-shaped button shows the complete solution, read-only : while it is active, **the whole keyboard is disabled** and nothing can be modified. That is deliberate. One looks, one closes it again, one takes up one's grid untouched. Tell the pupils so : it is not cheating, it is an answer sheet — but an answer sheet one cannot copy out cell by cell without having closed it and found one's own grid again.

6.3 — What to do with those who have not finished

Nothing special : **the game saves itself** as soon as one leaves the screen. It reappears on the *Resume game* card of the home screen, with its clock and its progress, and opens again exactly where it was left. The game keeps twenty at a time.

That is what makes it possible to finish the lesson on time without anyone losing their work — and to take up the same grid the following week.

6.4 — The same lesson, on paper

Everything above assumes one screen per pupil, and many classes do not have that. This stage works just as well on a **photocopy** : you bring out the blank grid once, you reproduce it, and everyone solves it in pencil. Chapter III details the printing window ; here it is enough to know that it is possible, and that it is not a cut-price version.

The course of the lesson does not change. The candidates are written small in the corner of the cells, exactly where the game puts them — the application invented nothing, it took over the gesture. The answer sheet is the second page, the one printed only for oneself. And the pupil who wants to carry on at home finds **the code and the QR at the foot of their sheet** : they find their grid unchanged, without your having anything to copy out on the board.

What one loses comes down to three things, and it is better to know them before the lesson : the grid no longer points out the mistake at the moment it is made, the eye no longer shows the solution along the way, and there is neither clock nor percentage to tell you at a glance where the class stands — you go round the rows again, as for any other exercise. In exchange, twenty-five papers stay on the desks until the correction, and that is something no screen does.

Chapter VII — As homework, and the correction

What one is aiming at: that work done at home can be checked without copying out twenty-five grids.

7.1 — Setting the homework

Grid	Code	To scan
Calcudoku, 5×5 , + - $\times$, Medium, twenty-four cells	G4CS520PMT-6bgf1z-4A	

Homework can hold in one line in the exercise book : **the code**. The pupil types it at home, on the family telephone, and gets the grid of the whole class. Nothing to print, nothing to hand out, nothing to find again at the bottom of a schoolbag.

Dictate it, or have it copied from the board, or send it by whatever channel you already use. The check key does its work : a code copied crookedly is refused instead of opening something else.

Homework is not necessarily a fresh grid. You can set **the one you began in class**, at the exact point where you stopped : the state code passes it on as it stands, filled cells and pencilled candidates included.

For the first grids, this is a way of **grading the progression**. The moment when one really has to search is at the start, while the grid is still empty ; past that moment, it is finished with what has just been learnt. So you open the grid with them, you cross the difficult passage together, and you entrust them with the rest — the part they can finish alone.

Two precautions are worth telling families, once :

- **the game works without a connection** — no internet is needed at home, only installing it once ;
- **there is nothing to pay** : every grid in this file can be finished in the free version.

7.2 — The same homework, on paper

All that assumes there is a device at home, and that is not the case everywhere — nor in every family of the same class. The homework is therefore just as well set **on a photocopy** : the blank grid, one per pupil. It is then a mathematics exercise like any other, done in pencil, collected and corrected on the second page — the one carrying the solution, which one prints only for oneself.

The simplest thing is not to choose at all. The printed sheet carries **its own code and its QR**: hand it out to everyone, and each family takes what it has. Those who have a tablet scan the corner of the page and find exactly the grid of the photocopy; the others get out a rubber. Nobody has different homework from their neighbour, and you have had to ask nobody what they own.

What remains is what a paper copy does not bring back, and that is precisely the subject of what follows.

7.3 — Correcting without collecting

This is where the second format, the state code of chapter III, changes the nature of the exercise.



Figure 7.1 — Every game keeps its clock and its progress

The pupil copies their game — *copy* button on the line of the list — and sends you the string. You paste it on the projected screen, and **their grid appears just as they left it**: their values, their pencilled candidates, their empty cells.

What you see then is not a mark, it is a **piece of reasoning**. The pencilled candidates say where they hesitated; the empty cells say where they gave up. A paper copy already shows part of that — the crossings-out and the pencil speak —, but never the **order** in which it all happened, and that is where the screen adds something.

Even then you see only the **state of arrival**. The path that led there, the application keeps as well, and can give back to you : the **Player's path** switch, in the printing window, brings out a page that takes up the pupil's moves **in the order in which they made them**, each with the rule that applied at that moment. The page opens on their grid, with their wrong cells marked.

Three verdicts can be read there, and they must be told apart in front of the class. When the rule is named, the reasoning has been reconstructed. When the sheet says *right, but the reasoning could not be identified*, the pupil **is right** : they simply saw further, or otherwise, than the machine looks. And when it says *mistake*, it says so only on one strict condition : the value placed makes the grid impossible to finish. So it is not « this does not look like the answer sheet » — and, in case of doubt, the sheet keeps quiet rather than accuse.

One last point, which changes the way one reads it : a wrong move is **set aside** for what follows. Without that, a single unlucky cell would close the grid and everything after it would be marked wrong in a block, whereas a pupil very often goes wrong in one place and reasons perfectly on the rest. That is precisely what one wants to see. What the sheet will never say, on the other hand, is **why** they believed it : it says what is wrong and from where, the rest belongs to the conversation you will have with them.

In practice, two or three copies are enough for a class correction : one takes the one that stopped earliest, one asks everybody what should have been seen, and one goes on.

7.4 — Why a transmitted game is never laid down crookedly

The question deserves to be asked, because it has a history. The game does not carry the grid about : it **remakes** it from the code, then lays the values of the string on top, one per cell to fill. It still has to be the same grid — and another version of the game does not make exactly the same ones.

That is why **the code carries the generator version** : the first two characters, **G1** . The application therefore knows which way of making to use, instead of guessing it ; and since the earlier ones are kept, an old code always opens its original grid. A string it would not know how to replay is **refused** — you will see a message, never a grid silently shifted by one cell.

Appendix A tells why that marker exists. In two words : without it, the only recourse was to **count** the cells to fill and check that the count came out right. A count is not a proof — two ways of making can empty the same number of cells —, and a game has already been found rebuilt on another grid, wrong from one end to the other although the player had not made a single mistake. That counting still exists, but it is no more than a second barrier behind the first.

7.5 — After the homework

The natural sequel is not a harder grid, it is a grid that works on **something else**. That is the subject of the third part : the same board five cells a side serves in turn for additions, for subtractions, for tables, for fractions. The sudoku does not change ; the child has been all round it, and can at last think about the counting.

Partie 3

The worksheets, by notion

Chapter VIII — Additions

Notion	adding two or three numbers, and breaking a total apart
Age landmark	6 to 8 years
Assumes	knowing how to add two numbers up to 10
Length	10 to 20 minutes per grid

8.1 — The settings

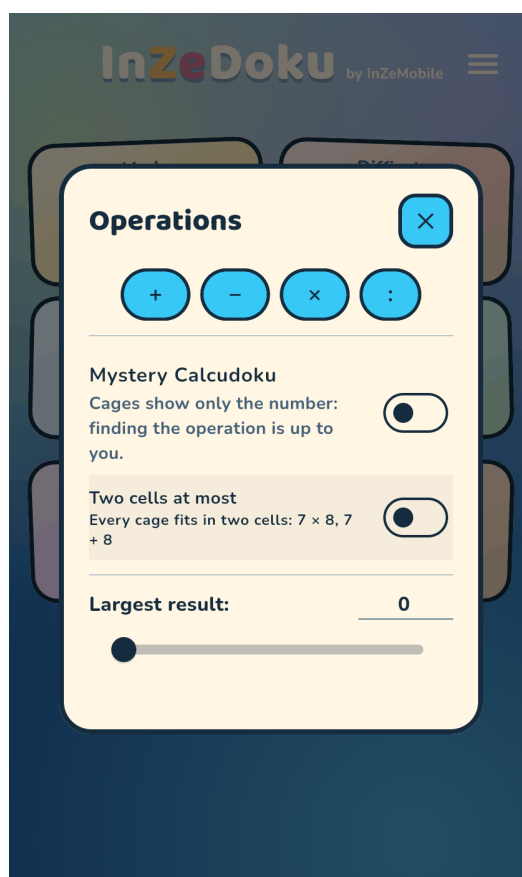


Figure 8.1 — The $\div$ tile : four pills, and addition alone switched on

Calcudoku mode, **square** board, and in the $\div$ **Operations** tile, one single pill switched on : $+$. Every cage will then be an addition.

It is the barest setting in the game, and it is the right one to begin with. The child has only one operation in mind, and all their attention goes to what is really new : the fact that a cage limits the possibilities.

8.2 — The grids

Grid	Code	To scan
4 × 4, Easy, fifteen cells to fill	G4CS410P-a30y6s-IU	
5 × 5, Medium, twenty-four cells to fill	G4CS520P-7fobas-N8	

The first gives **only one cell**: all the rest is deduced from the cages. Its labels run from 3+ to 7+, on cages of two or three cells — so sums the child does in their head.

The second goes up to five cells a side, therefore to numbers up to 5 and sums up to 12, and it gives two cells to start with.

8.3 — What the child really works on

Not addition : **decomposition**. Faced with 7+ on two cells of a board of five, the question is not «how much are 3 and 4» but «which pairs make 7». There are two : 3 + 4 and 2 + 5. It is an exercise in mental arithmetic run backwards, the one rarely done on a worksheet because it has no single answer — and here, the absence of a single answer is precisely what drives the game.

To this is added, without anyone having to name it, **commutativity** : 7+ does not say which of the two numbers goes where, and it is the column that decides.

8.4 — Easier, harder

Easier — stay at 4 × 4 and go back to the Easy level. The board does not change in nature, it simply gives more to hold on to.

Harder — two directions, and they are not equivalent :

- raising the **level** makes the deduction longer, the counting stays the same. That is the right direction for a child at ease with counting ;
- adding **subtraction** (next chapter) changes the counting and leaves the deduction where it is. That is the right direction for all the others.

The fact that one can choose is exactly what makes this mode useful in class.

Chapter IX — Subtractions

Notion	subtracting, and recognising a difference
Age landmark	7 to 9 years
Assumes	addition ; knowing that one takes the smaller from the larger
Length	15 to 25 minutes per grid

9.1 — The settings

In the  **Operations** tile, two pills switched on :  and .

One remark about the way it works, because it explains the shape of the grids : subtraction is only defined **on two cells**. Three numbers have no difference. Subtraction cages are therefore always pairs, while addition cages continue to go up to four cells. A grid in   thus naturally mixes small two-number cages and larger ones — which gives the board a more varied look than the previous one.

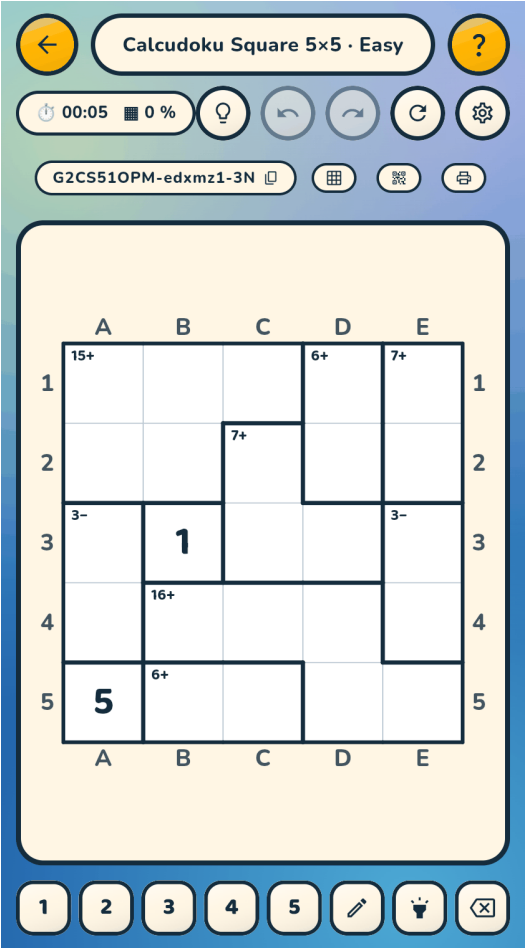


Figure 9.1 — The pairs carry a `–`, the larger cages a `+`

9.2 — The grids

Grid	Code	To scan
5 × 5, Easy, twenty-three cells to fill	G4CS510PM-edxmz1-3R	
5 × 5, Medium, twenty-three cells to fill	G4CS520PM-dc93mz-WS	

These are the two grids of the second part : the one for group work and the one for individual work. If you have followed that progression, the class already knows them.

9.3 — What the child really works on

The **difference**, and it is a subtler notion than a subtraction set out on paper. 2- on two cells of a board of five means four possible pairs : 1 and 3, 2 and 4, 3 and 5 — and their mirror images, since the label does not say in which order. The child learns that a difference designates a **family** of pairs, not one pair.

It is also the first setting in which the order of reading does not matter : the game accepts the difference whichever way round, as long as the numbers stay positive. A child who writes « 3 minus 5 » is not punished for it.

9.4 — Easier, harder

Easier — go back to additions alone for the space of one grid, or drop to 4×4 .

Harder — multiplication, in the next chapter.

And one variant to know about : one can remove addition and keep only $-$. All the cages then become pairs — a difference cannot be worked out on three numbers — and the grid becomes a pure exercise in differences. It is free, like any operation taken on its own.

It is also the case in which the **deduction** becomes the harshest in this whole file : when all the cages say the same thing, they overlap less, and a difference constrains far less than a sum. The grid remains perfectly playable, but it calls for patience.

The remedy is not to give it up, it is to **go down one level** : a grid in $-$ alone at the Easy level is worth a grid in $+-\times$ at the Medium level. It is a very good exercise for a fast pupil who is bored, provided they are warned that it will take a while.

Chapter X — Multiplication tables

Notion	the products of two numbers from 1 to 10
Age landmark	8 to 10 years
Assumes	knowing what a product is ; knowing a few tables
Length	15 to 25 minutes per grid

This is the worksheet this mode was designed for, and it calls for three settings instead of one. They are worth it.

10.1 — The setting that changes everything

On a board five cells a side, the cells normally carry the numbers 1 to 5 — and a multiplication table that stops at 5 is not of much interest.

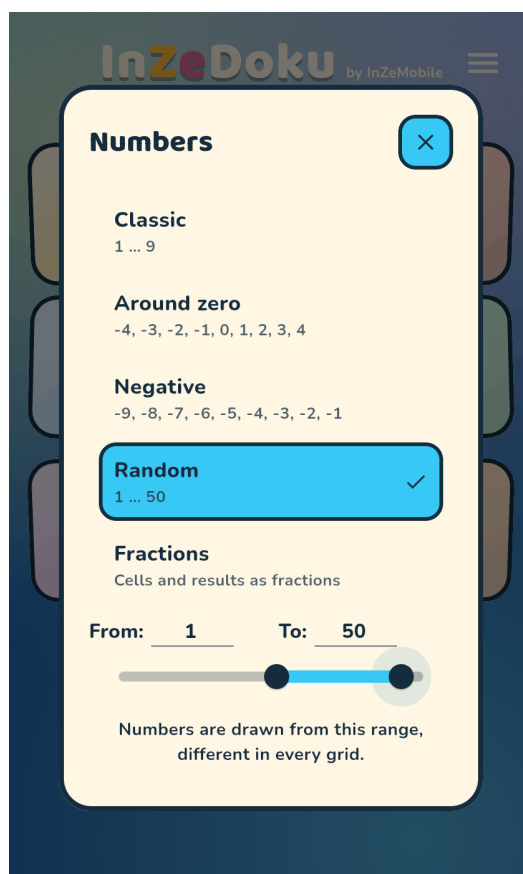


Figure 10.1 — The  tile : « Random », between 1 and 10

The  **Numbers** tile undoes that link. By choosing the **Random** family and setting the two sliders to **1** and **10**, one asks for *five numbers drawn between 1 and 10* — different in every grid. The board stays a small 5×5 that a child finishes, and the products met there are those of their tables.

This is the point this file has been repeating from the start, and it has never been so plain : **the size of the board and the difficulty of the counting have nothing to do with each other.**

10.2 — Two cells at most

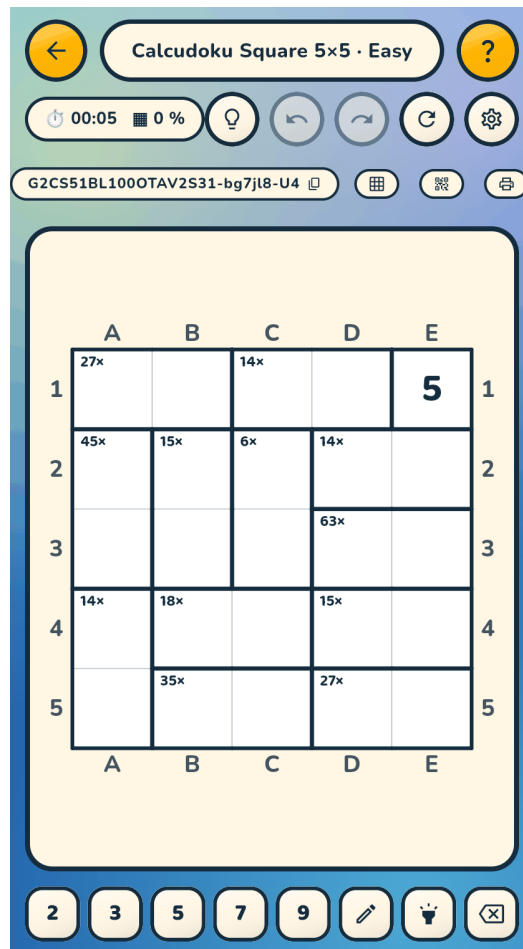


Figure 10.2 — Every cage holds in two cells : each one is a product

In the  tile, the **Two cells at most** switch forces every cage to hold in a pair. A multiplication cage then becomes exactly what one is after : $21\times$ on two cells is 3×7 and nothing else.

Against all expectation, this setting **empties** the grid instead of filling it : the game gives half as many cells with it as without. Smaller cages constrain better, so fewer clues are needed.

The third setting, **Largest result**, bounds the size of the labels. It is useful for a child who has not yet met large numbers, and it costs nothing as long as one stays on a small board.

10.3 — Multiplication alone, and what it costs

In the $\div$ tile, one can switch on $\times$ only. All the cages are then multiplications, and the worksheet becomes a lesson on tables and nothing else. That is what the grid below does, and **it is free** — like any operation taken on its own.

One operation alone makes the deduction harder.

The sudoku is not deduced from the cages alone: it is deduced from the variety of the constraints. When all the cages say the same thing, they overlap less, and the child has to reason further before being sure.

Subtraction alone is the extreme case: with no $+$ and no $\times$, all the cages necessarily become pairs — a difference cannot be worked out on three numbers — and a difference constrains far less than a sum. The grid remains perfectly playable, but it calls for patience.

*That is not a reason to do without it: it is a reason to **go down one level** when trying it. A grid in $-$ alone at the Easy level is worth a grid in $+-\times$ at the Medium level.*

Multiplication alone, for its part, is among the gentlest of the isolated operations: a product constrains strongly, especially when bounded by a largest result.

10.4 — The grids

Grid	Code	To scan
5 $\times$ 5, Easy, twenty-four cells	G4CS51BL1000TAV2S31- bg7jl8-U8	
5 $\times$ 5, Medium, twenty-four cells	G4CS52BL1000- TAV2S31-9368qy-ZK	

These codes are long because they carry the three settings; they are scanned as a QR like the others, and that is where the QR saves the most time.

The first grid draws the numbers **2, 6, 7, 9 and 10**, and **all** its cages are products: $12\times$, $14\times$, $18\times$, $42\times$, $54\times$, $70\times$, $90\times$... The same configuration gives other numbers on another grid number: that is what makes it possible to do the exercise again without doing it again.

10.5 — What the child really works on

Factorisation, more than multiplication. Faced with $15\times$ on two cells, they do not compute 3×5 : they look for which two numbers of the board give 15. It is the table read backwards, the one that serves for division.

And they check it themselves, immediately. As soon as a cage is **complete** and its values do not give the announced result, the game **circles it in red**. Not the row, not the cell: the cage, at the exact place where the counting is wrong.

It is not a hint about the solution — the target, the two numbers and the operation are all three before the child's eyes, it is a sum they can do again in their head. But it is what lets them correct themselves without putting up a hand, and lets you go round without having to check twenty-five grids.

10.6 — Easier, harder

Easier — bring the second numbers slider back to **5**, or lower the Largest result. The grid keeps its shape, the products get smaller.

Harder — raise the slider to **12** or **20** to go beyond the usual tables. Careful: beyond 10, one leaves the free version, which keeps to the values from 1 to 10 — « multiplication tables, and that's all ».

Chapter XI — The four operations

Notion	choosing the operation, and dividing exactly
Age landmark	9 to 11 years
Assumes	the four operations set out on paper ; the notion of a divisor
Length	20 to 30 minutes per grid

11.1 — The settings

None. It is the game's original setting : the four pills switched on in the $\div$ tile, which is why the codes of this worksheet are the shortest in the file.

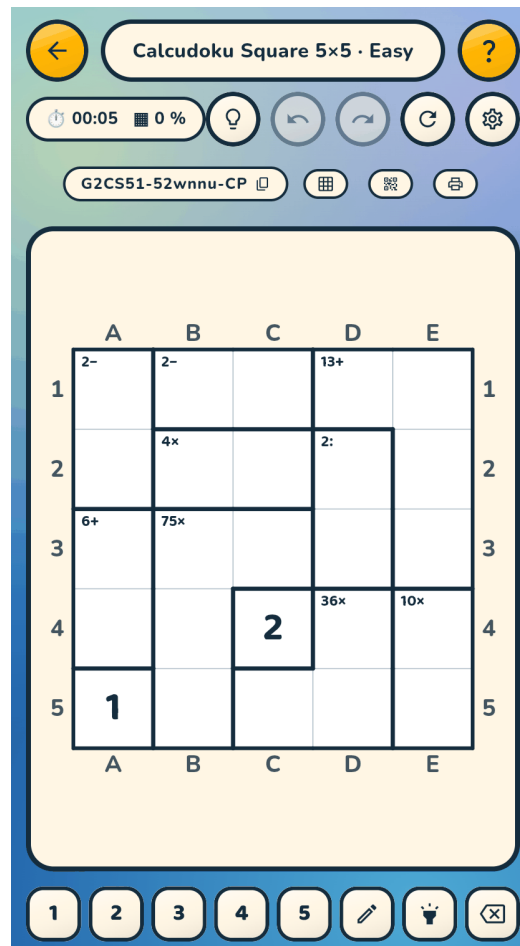


Figure 11.1 — The four signs sit side by side ; division is written « : »

One detail of display that matters in class : **division is written :** and not $\div$. Small and in bold, the sign $\div$ is hard to tell from $+$ — both come down to a horizontal bar surrounded by matter. And $:$ is in any case the sign taught in a good part of Europe.

11.2 — The grids

Grid	Code	To scan
5 × 5, Easy, twenty-three cells to fill	G4CS51-52wnnu-CT	
5 × 5, Medium, twenty-four cells to fill	G4CS52-8bf4qk-2R	

Twenty-four cells out of twenty-five: these two grids give almost nothing. All or nearly all of it comes out of the cages, which is possible because four operations constrain far more than two — and it is also what makes them harder than they look, in spite of the Easy level.

If the class gets stuck, it is not the level that should be lowered but the number of operations : a grid in $+-\times$ from the previous chapter is a better step.

11.3 — What the child really works on

The choice of the operation, which is a skill in its own right and which no written-out counting exercise calls for. 12 on two cells could be a sum, a product or a quotient ; the label says which, but one has to have read the sign, and above all to have in mind what each operation makes possible.

Divisibility, next. A 3: cage on two cells calls for a whole quotient — the game never proposes a division that does not come out exactly. The child therefore looks for pairs in which one divides the other, which is exactly the question of multiples and divisors.

11.4 — Easier, harder

Easier — remove division. $+-\times$ remains a rich board, and division is by far the most constraining operation.

Harder — add **Two cells at most**, seen in chapter X: the four operations on pairs only make a dry, fast grid, in which every label is a small closed problem.

There is one notch above that, the **Mystery Calcudoku**, where the cages give the result **without** the operation : it is up to the child to find which one fits. It is a fine idea for a pupil who is very much at ease, and it calls for the full version.

Chapter XII — Fractions

Notion	adding, multiplying and dividing fractions
Age landmark	9 to 12 years
Assumes	knowing what a fraction says ; the notion of a common denominator
Length	20 to 30 minutes per grid

The cells no longer carry 1, 2, 3, 4 but $\frac{1}{2}$, 1, $\frac{3}{2}$, 2 — and all the rest of the game does not move an inch. That is what makes this worksheet interesting : the child already has the mechanics of the board, they have only the numbers left to deal with.

12.1 — Up to five cells a side, it is free

To be said at once, because it is often the first question : fractions **are not a paying option**. They follow the same rule of size as the whole-number grids, and the Calcudoku is entirely free up to five cells a side. The two grids of this worksheet can therefore be played right through.

12.2 — The two sliders

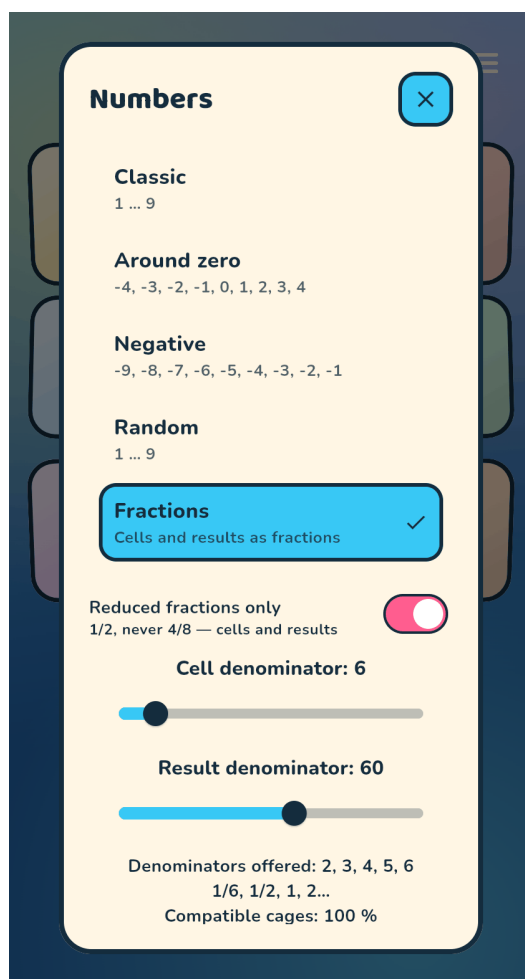


Figure 12.1 — Two independent sliders : the cells, and the results

In the  **Numbers** tile, the **Fractions** family opens two sliders, and one must have understood that they are **independent** :

- the first bounds the denominator of the **cells** — at 2, the child will see only halves ; at 4, halves, thirds and quarters ;
- the second bounds the denominator of the **results**, that is to say of the cage labels.

Nothing couples them, and that is deliberate : $1/8 + 3/8 = 1/2$ has a simpler result than its operands. One can therefore ask for complicated cells and well-behaved results, or the other way round.

A third setting, **Reduced fractions only**, is switched on by default : the child sees $1/2$ and never $4/8$. By switching it off, one gets the unreduced writing — and one lesson more, since one then has to recognise that $4/8$ is a half.

12.3 — The grids

Grid	Code	To scan
4 × 4, Easy, fourteen cells — halves	G4CS41F2X6R-aan1-j5-57	
4 × 4, Easy, fifteen cells — thirds and quarters	G4CS41F4X12R-g31zwm-KS	

The first carries only $\frac{1}{2}$, **1**, $\frac{3}{2}$ and **2**. It is the way in: four values, one single denominator, and labels like $3+$ or $\frac{9}{4}\times$. A child who knows that two halves make a whole can begin it.

The second carries **$\frac{1}{4}$, $\frac{3}{4}$, $\frac{4}{3}$ and 2**: quarters *and* thirds, therefore real reductions to a common denominator from the very first addition — its $\frac{14}{3}+$ cage is a good example. That is the jump, and it is a clear one.

12.4 — What the child really works on

The **common denominator**, and they work on it in the only direction that counts: they have to find it, not to apply it. Faced with $\frac{5}{2}+$ on two cells, one of which could be $\frac{3}{2}$, one has to put down $\frac{3}{2} + 1$ and see that it does make $\frac{5}{2}$.

And the **multiplication of fractions**, which here has a rare advantage: it is easier than addition. $\frac{3}{4}\times$ on two cells is looked for by multiplying numerators and denominators, with nothing to reduce. Many pupils discover there that the rule they find complicated is in fact the simplest of the four.

12.5 — Easier, harder

Easier — the first slider at **2**, and nothing else. Four values, one denominator: it is the floor, and it is already of use.

Harder — raise the first slider, or move to 5×5 . Beware of the opposite reflex: **raising the denominators makes the counting harder but the sudoku easier**, because more varied values constrain better. A grid with complicated fractions may therefore be deduced faster than a grid with halves.

It is the same lesson again: two difficulties, two knobs.

Chapter XIII – Decimal numbers

Notion	going from a fraction to its decimal writing
Age landmark	10 to 12 years
Assumes	fractions ; decimal writing
Length	20 to 30 minutes per grid

13.1 – The two writings on the same board

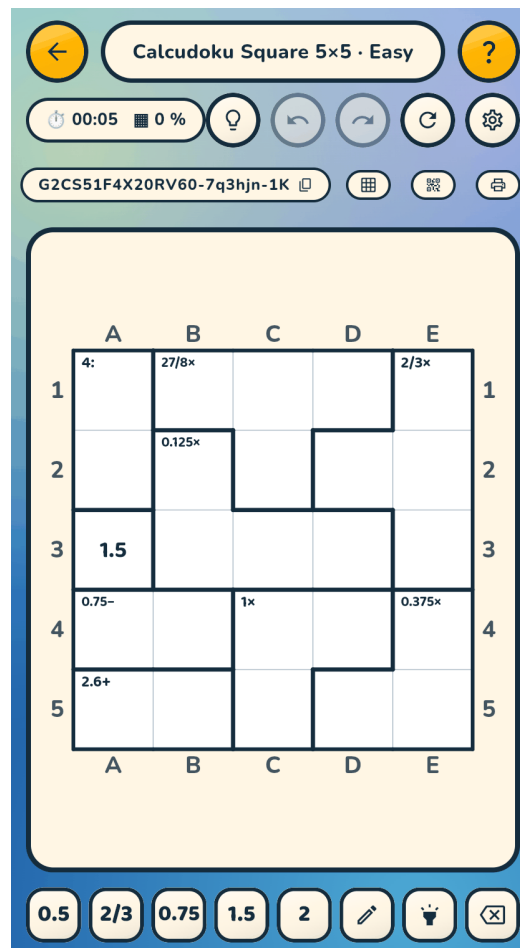


Figure 13.1 — `1.5` and `3/2` designate the same cell; one has to see it

Under the fractions sliders, a third one sets a **proportion of decimal writing**, from 0 to 100 %. At 60 %, a good half of the values and of the labels are written 1.5 instead of $\frac{3}{2}$ — and the other half stays in fractions.

It is the mixture that makes the exercise. An entirely decimal grid would only be a whole-number grid in disguise; an entirely fractional grid is the one of the previous chapter. Here, the child must constantly **translate** — recognise that the cell they have just filled with **1.75** is the one the neighbouring cage calls **7/4**.

13.2 — The grid

Grid	Code	To scan
5 × 5, Easy, twenty-four cells to fill	G4CS51F4X20RV60-7q3hjn-10	

Its five values are **1/2, 2/3, 3/4, 3/2 and 2**. Four of them have a decimal writing that comes out exactly — $0.5 \cdot 0.75 \cdot 1.5 \cdot 2$ —, and the fifth, $2/3$, is written **0.6 with a bar above it**: the 6 repeats endlessly. The grid therefore shows both cases in the same game, which is exactly the exercise.

13.3 — What the game converts, and what it refuses

Three safeguards, worth knowing in order to answer the questions.

A fraction that does not come out exactly stays a fraction. $1/7$ has no finite decimal writing, so it is written **1/7**. A child will never see a truncated decimal.

An unlimited but short decimal is accepted, with a **bar above the digit that repeats**: **0.3** with a bar above is a third. Beyond two repeated digits, or three digits before the repeating part, the game gives up and keeps the fraction — **0.00390625** would teach nobody anything.

Two values that would look too much alike both stay fractions. **0.3** and **0.3** with a bar differ only by a stroke: if the grid contained both, neither goes decimal. It is a precaution against an ambiguity one would only see afterwards.

Finally, the separator follows **the language of the application**: a full stop in English, a comma in French. Nothing to set.

13.4 — What the child really works on

Conversion in both directions, in a real situation. It is not a conversion exercise — there is no instruction saying «convert» — it is a constraint: the grid is mixed, so one has to translate in order to make progress.

And incidentally, the discovery that **some fractions have no finite decimal writing**. The question always comes from the class, when a pupil notices that $1/7$ is never converted. It is a good moment to talk about it; the game has asked the question for you.

13.5 — Easier, harder

Easier — lower the proportion to 20 or 30 %. A few decimals in a grid of fractions is already the exercise, in a gentler form.

Harder — raise it to 100 %, which does not mean « all of them » : the values with no short writing stay fractions, and the mixture remains. Or raise the denominator of the cells, which mechanically increases the number of unconvertible values.

Chapter XIV — Without a single digit

Notion	the logic of the sudoku, before numbers
Age landmark	from nursery school on
Assumes	nothing — neither reading nor counting
Length	5 to 15 minutes per grid

This chapter leaves counting behind, and it is probably where one should begin with the youngest. **A child who cannot yet count can play**: the rule of the sudoku does not call for numbers, only for recognising that two things are the same.

That holds from **nursery school** on, in a transition class, or in the very first days of primary school — before arithmetic, and even before digits.

14.1 — The Geodoku : shapes to recognise

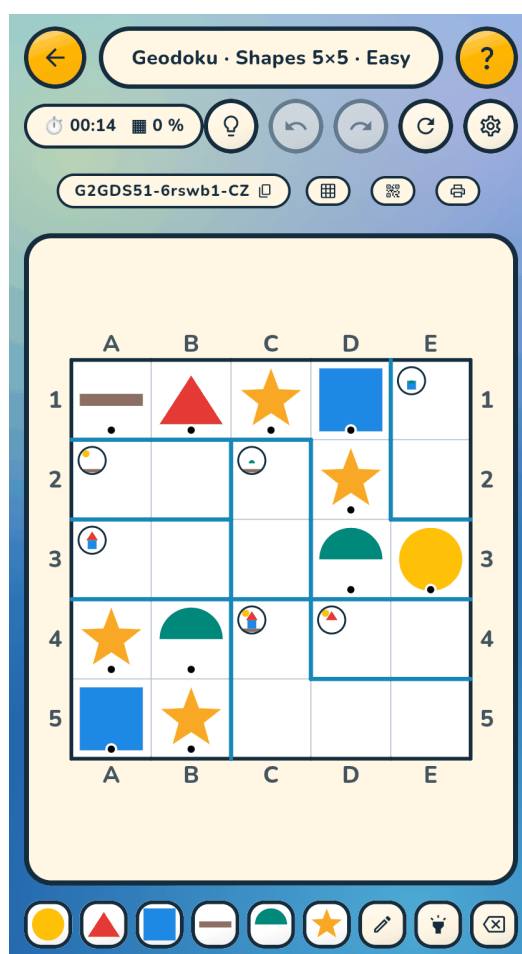


Figure 14.1 — Every region assembles a complete figure

The cells carry **geometric shapes** — square, triangle, circle, diamond. The rule of the sudoku is unchanged ; what changes is what one handles.

The educational interest is twofold, and it is real. **Recognising** the basic shapes, first: that is a nursery-school learning in its own right, and the game makes a child practise it dozens of times in a quarter of an hour without its being a lesson. **Naming** them, next : a child playing in a pair says their cells aloud, and there is no better exercise in geometric vocabulary than a game in which one has to say *which one* is being placed.

14.2 — How one runs it

At the board, exactly like a counting grid : one shows, one asks *where are we sure*, one fills two or three cells together and one stops.

Two remarks specific to the youngest.

In pairs, rather than alone. A four- or five-year-old playing alone gets discouraged at the first hard cell ; in pairs, they point out to each other what the other has not seen, and the rule settles in by imitation.

Do not count the time. The game's clock runs, but it has no meaning here : what one watches is whether the child looks for the cell *where they are sure* rather than filling at random.

14.3 — The grid

Grid	Code	To scan
5 × 5, Easy, fourteen cells	G4GDS51-6rswb1-D3	

14.4 — Your syllables, your words

Shapes are recognised before one can read ; what comes next is played just as the child is beginning, precisely, to read. The game then lets you **write the symbols of the grid yourself** — and it really is you who write them, not a ready-made list : nobody but you knows which syllables your class has met this week.

On the home screen, the **Symbols** tile offers a third kind after digits and letters : **Custom**. One types them one per line or separated by commas — MA, PA, LO, TI, RE, NU —, the two forms mixing with nothing to ask for, and the grid is played exactly like an ordinary sudoku. The rule does not change one jot ; all that changes is what is in the cells. Later in the year, the same list carries whole words : HOUSE , LORRY , BOAT .

There must be as many as the grid has cells on a side — six for a 6×6 — and the game sees to it from end to end : if you enlarge the grid after writing your list, it tells you what is missing and refuses to start the game rather than handing you back a grid of digits. A code that is too short is refused in the same way when read ; nobody will receive a grid other than the one you think you sent.

A cell displays a word without trouble up to 5×5 ; beyond that it becomes small, and the entry screen shows you precisely **a cell to scale** with your word in it, so that you may judge before the class rather than after. The two-finger zoom does the rest — and it is not a makeshift : having to move the grid to find again a cell one can no longer see exercises spatial bearings and memory, which is school work in its own right.

It is in a group that it takes on its full meaning. You share the grid by its QR, everyone has it before their eyes, and the exercise is done in threes or fours. To place a cell one has to name it : the child *says* the syllable, and hears it said by their classmates twenty times in a quarter of an hour. It is that repetition — spoken, heard, without anybody having had to ask for it — that makes decoding sink in, far more surely than a worksheet filled in in silence.

Two points of method, so that the lesson holds together. **Six symbols are enough** : a 6×6 with this week's syllables is better than a 9×9 bringing out last month's. And **keep the same grid** from one week to the next, changing only the list : the board is identical, the exercise is new, and the children do not recognise it.

Accents are an exercise in their own right, and it is perhaps the widest use of this tile. A list is not obliged to be made of syllables : it can be made of accented letters — é , è , ê , à , ç —, and the grid then becomes an exercise in pure discrimination. The child copies nothing out ; they must see that e and é are not the same symbol, otherwise the line contradicts itself. The correction is not yours, it is the grid's.

This does not hold for French alone. German has ä , ö , ü ; Spanish ñ ; Portuguese ã and õ ; Italian à , è , ì . Each language brings its own characters, and the same board puts them to work — it is at that level that the game becomes truly multilingual, far more than by translating its menus.

The case is the one you type, and here it matters : the accent is learned on the small letter, the one in the reading book, and not on its capital. café stays café .

One practical word, finally : an accented syllable travels very well by QR, but it is dictated badly at the board. If your class is working on mé , pè , lô , have them scan rather than copy.

14.5 — And there is more, for later

The game carries other sets without digits : pictograms to assemble, science sets, and the **Colodoku**, where the cells are colours one mixes — red and green give yellow, and a complete row always makes the same colour.

This file makes no worksheets of them, for a simple reason : a nursery-school child learns first **to recognise colours**, not to mix them. Mixing is a notion for much later, and the Colodoku will wait for that moment. You will find it without us when it is of use to you ; the *Player's manual* devotes its chapter 6 to it.

14.6 — What it prepares

None of this is a decorative preamble. A child who has done ten Geodokus has taken in, without a word of explanation, the two ideas on which all the counting worksheets of this file rest :

- a row contains each thing **once and once only** ;
- a region imposes a constraint that **narrows the possibilities** without fixing them.

When you give them their first grid of additions, they will only have the **+** left to learn.

Appendices

Appendix A — All the worksheets at a glance

This page is made to be printed and laid beside you. Every code is opened by the **Code** pill, to the right of the Play button on the home screen — either by typing it, or by **scanning the QR** of the last column with the *Scan a QR* pill. The pupils then have nothing to copy out : they aim at the page, the configuration is displayed in plain words, and **Play** starts the grid.

Photocopy it, or lay it under the visualiser's camera : a printed QR scans just as well as a projected one.

A.1 — The lesson of chapters IV to VII

Moment	Code	Grid	To fill	To scan
Showing the rule	G4SS41-d28j6x-3B	sudoku 4×4 , Easy	12	
Showing the cages	G4CS410P-a30y6s-IU	additions, 4×4 , Easy	15	
In groups	G4CS510PM-edxmz1-3R	+ - , 5×5 , Easy	23	
Alone in class	G4CS520PM-dc93mz-WS	+ - , 5×5 , Medium	23	
As homework	G4CS520PMT-6bgf1z-4A	+ - × , 5×5 , Medium	24	

A.2 — The worksheets by notion

Notion	Code	Grid	To fill	To scan
Additions, first step	G4CS410P-a30y6s-IU	4 × 4, Easy	15	
Additions, further on	G4CS520P-7fobas-N8	5 × 5, Medium	24	
Subtractions	G4CS510PM-edxmz1-3R	5 × 5, Easy	23	
Subtractions, further on	G4CS520PM-dc93mz-WS	5 × 5, Medium	23	
Multiplication tables	G4CS51BL1000TAV2S31-bg7jl8-U8	5 × 5, Easy	24	
Tables, further on	G4CS52BL1000TAV2S31-9368qy-ZK	5 × 5, Medium	24	
The four operations	G4CS51-52wnnu-CT	5 × 5, Easy	23	
The four, further on	G4CS52-8bf4qk-2R	5 × 5, Medium	24	

Notion	Code	Grid	To fill	To scan
Fractions, halves	G4CS41F2X6R-aanlj5-57	4 × 4, Easy	14	
Fractions, thirds and quarters	G4CS41F4X12R-g31zwm-KS	4 × 4, Easy	15	
Decimal numbers	G4CS51F4X20RV60-7q3hjn-10	5 × 5, Easy	24	
Geodoku	G4GDS51-6rswb1-D3	5 × 5, Easy	14	

All these grids can be finished entirely in the free version. That is not a statement of principle : it is checked automatically at every version of the game, at the same time as the codes themselves.

A.3 — The two characters at the end

A code ends with a second hyphen and two characters — ...-RM . They are not part of the grid : they **check** it.

Any mistake on one character, and any swap of two neighbouring characters, are refused : the game answers « code not recognised » instead of opening a neighbouring grid. That is exactly what is needed when twenty-five pupils copy out the same line from the board.

You will come across codes elsewhere that carry none — those of the *Player's manual*, for instance. They work : they are simply less protected against typing mistakes.

Appendix B — What the free version allows

B.1 — Nothing is locked

InZeDoku is **one single application**, and it plays in three ways : free, by **subscription**, or after a **lifetime purchase**. This appendix speaks only of the first, because the first is where this whole file fits.

The principle is worth understanding, because it is not the one people expect. The free version **forbids nothing** : all the modes, all the geometries, all the sizes, all the levels open and are played, and any game code whatever works.

What it limits is **finishing**. Outside the free range, a grid is only filled up to a share of its cells ; beyond that, the keypad places nothing more. Erasing, correcting, going back and the pencil all stay free.

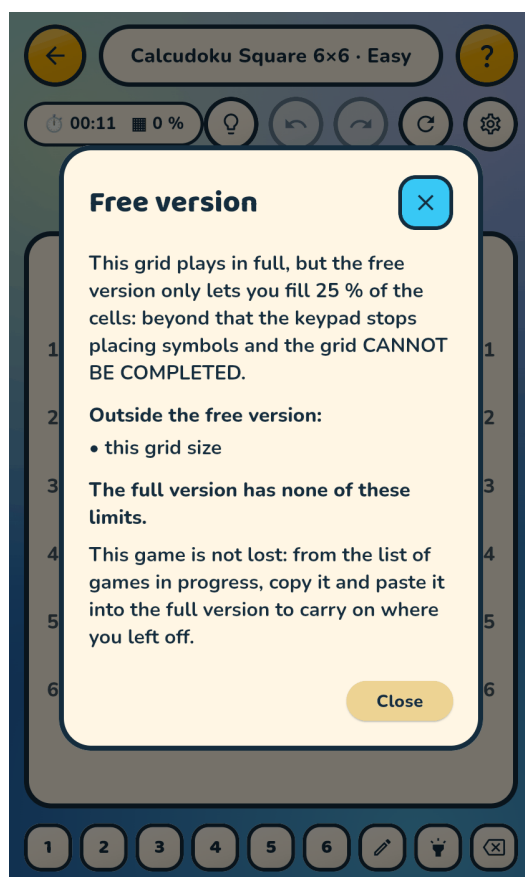


Figure B.1 — The game warns at the opening, with the percentage and the reason

And the game **warns at the opening of the game**, giving the percentage and the reason. So you will never discover the limit in the middle of a lesson : it is announced before the pupil starts.

B.2 — The free range

Board	Free up to
Square · Sudoku	9×9
Square · Calcudoku	5×5
Square · Colodoku	4×4
Square · Geodoku and other picture sets	5×5
Hexagonal	radius 2
Triangular	side 6
Voronoi	5 symbols
Mosaic	—

To this are added five limits independent of size :

Outside the free version
the Hard and Expert levels
the mosaic geometry
the Mystery Calcudoku (cages with no operation)
mixed sets of operations outside $+$, $+-$, $+-\times$, $+-\times$:
values outside 1 to 10

Two clarifications about that second-to-last line, because it reads badly.

One operation ALONE is free — $+$, $-$ or $\times$. What leaves the free version is the mixtures outside the learning progression: $+\times$ without subtraction, for instance. A lesson on tables in $\times$ only therefore costs nothing. ($\div$ alone does not exist: the game refuses a set with no $+$, $\times$ or $-$, because most pairs are not divisible and the grid would arrive half filled in.)

Fractions are not there either: they follow the size rule of the Calcudoku, like whole numbers. Up to 5×5 , a grid in fractions is played right through — decimal writing included.

B.3 — What that gives in class

Everything this file describes fits inside that range, and it is no accident: the worksheets were built inside it.

Concretely, without paying anything you have :

- the **Calcudoku up to 5×5** , which is the right size for a school lesson — a larger board does not make the counting more interesting, it makes the game longer ;
- the **Easy** and **Medium** levels, which are the two one uses ;
- the four progressions of operations, in the order in which they are taught — **and each operation taken on its own** ;
- the **values from 1 to 10**, therefore all the multiplication tables ;
- **fractions and decimals** ;
- the **Sudoku up to 9×9** , and the entry-level **Colodoku**.

What is missing, and is not missed here : the large boards, the exotic geometries, the Hard and Expert levels, the mixtures of operations outside the progression, and the values beyond 10.

B.4 — A grid begun is never lost

If you nevertheless come across a limited configuration, the game is not lost, and you have nothing to move : there is only one application, so the subscription or the purchase lift the limit **where you are already playing**, and the grid takes up exactly where it was.

Copying and pasting a game keeps its whole use, but it is the use of chapter VII : moving a grid from one device to another, from the class to the home.

Appendix C — Where to look in the player’s manual

This file does not describe the game, it describes what one does with it. As soon as a question is about how it works, the answer is in *InZeDoku — The player’s manual*, and here it is as a table of cross-references.

C.1 — The settings this file uses

What you are looking for	Player’s manual
What a cage is, how to read a label	chapter 5, <i>Calcudoku</i>
Choosing the operations, and what each choice changes	chapter 11, <i>Choosing the operations</i>
The families of numbers, the range sliders	chapter 12, <i>Uncoupling the numbers from the size</i>
« Two cells at most », the largest result	chapter 13, <i>Setting the cages</i>
The two fractions sliders, the unreduced writing	chapter 14, <i>The Calcudoku in fractions</i>
The proportion of decimal writing, the repetition bar	chapter 15, <i>Decimal writing</i>
Cages with no operation (Mystery)	chapter 16, <i>The Mystery Calcudoku</i>
What the four levels really change	chapter 19, <i>Difficulty</i>

C.2 — Playing, saving, passing on

What you are looking for	Player’s manual
The pencil, multiple selection, the eye	chapter 20, <i>The keypad aids</i>
Every key on the physical keyboard	chapter 21, and appendix B
The clock, the list of games, the codes, the QR	chapter 22, <i>Timer, games in progress and game codes</i>
The complete grammar of the game code	appendix A, <i>The game code, letter by letter</i>
Sizes, modes and limits, in tables	appendix C

C.3 — The rest

What you are looking for	Player's manual
The Colodoku : palettes, mixtures, badges	chapter 6, <i>Colodoku</i>
The Geodoku and the other picture sets	chapter 18, <i>For children</i>
The accessibility settings, colour blindness included	chapter 24, <i>Accessibility</i>
What the game collects — that is to say nothing	chapter 25, <i>Privacy</i>
Free, subscription, lifetime	chapter 26
The web demo and its limits	chapter 27

C.4 — And in the game itself

Two aids are within reach during the game, and it is worth showing them to the pupils once :

- the ? button of the game screen displays **the rules of the current grid**, assembled according to its options — so never a generic text ;
- the 📖 **Game guide** entry of the ≡ menu contains a short version of the manual, in sixteen sections, in the seven languages of the application.

Appendix D — The rules of deduction, one by one

When a pupil is stuck and touches the light bulb, the application answers in three stages : first **how many** cells can be found straight away, then **which rule** to use, and only last **which cell**. It is the second stage that calls for this appendix : the application says what the rule means, it does not name it. It is up to you to put the word on it when that is of use, and to have it practised.

The six rules hold in six pages. All the grids shown here are real grids from the game, taken at the exact moment when the rule comes into play.

D.1 — Reading the illustrations

The cells are named as on a chessboard : **the letter is the column, the number is the row**, and they are written all around the grid. D1 is therefore the fourth column, first row. It is a notation that is practised at the same time as the rest, and the occasion is a good one.

It is not peculiar to these pages : **the application carries the same landmarks around the board**, and so does the printed sheet. The pupil therefore reads D1 where they play, without having to transpose — and when the light bulb names a cell for them, they find it at a glance. On the boards that have neither row nor column — the hexagon, the triangle, the mosaic — there are no landmarks and the application gives the rank of the cell ; the grids of this appendix are all squares, where the notation holds.

The black badges carry the **number of the step** : the ❶ of the text is the ❶ on the grid. A cell belonging to two zones at once carries both colours in wide bands.

D.2 — The forced cell

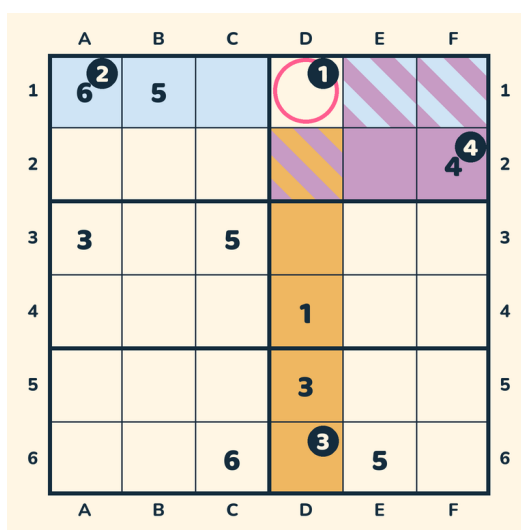


Figure D.1 — Blue, the row ; orange, the column ; plum, the block

1. One looks at the cell **D1**. **2.** Its row **1** already carries 6 and 5. **3.** Its column **D** already carries 1 and 3. **4.** Its block already carries 4. **5.** Only **2** is left : one writes **2** in **D1**.

It is the first rule a child finds on their own, and often the only one they need on an easy grid.

D.3 — The forced place

	A	B	C	D	E	F	
1	6 ¹	5	²	×	×	×	1
2						4	2
3	3		5				3
4				1			4
5				3			5
6			6		5		6
	A	B	C	D	E	F	

Figure D.2 — The crosses mark the cells where the 4 cannot go

1. One chooses a value — here the **4** — and a group : **row 1**. **2.** One crosses out the cells where the 4 cannot go. **3.** Only one is left : **C1**. One writes **4** in **C1**.

It is the reversal, and it costs something. The previous rule starts from the cell and asks «what can it carry?». This one starts from the number and asks «where can it go?». Making the difference felt is worth a whole lesson.

D.4 — The forced cage

	A	B	C	D	E	
1						1
2						2
3						3
4						4
5	20× 1	2				5
	A	B	C	D	E	

Figure D.3 — A cage of two cells that must make 20 by multiplying

1. One looks at the cage **20×**, on **A5** and **B5**. **2.** One looks for the two numbers that make 20 by multiplying. **3.** 2×10 and 1×20 go outside the grid, which only goes up to 5. **4.** Only 4×5 is left. **5.** Those two cells therefore carry a 4 and a 5 — one does not yet know in which order, but all the other values are eliminated.

It is here that the counting joins the deduction, and it is the heart of this file : the child counts in order to eliminate, not to find a result.

D.5 — The pair

	A	B	C	D	E	F	
1	6	5					1
2						4	2
3	3		5				3
4	1 2 4	2	2 4	1			4
5				3			5
6			6		5		6
	A	B	C	D	E	F	

Figure D.4 — In plum, the pair; in orange, the cells it frees

1. In **row 4**, the cells **A4** and **C4** can only carry **2** or **4**, either of them. **2.** One does not know which carries which — and it does not matter at all : the 2 and the 4 are both taken by those two cells. **3.** So no other cell of the row can carry them : they are removed from **B4**, **E4** and **F4**.

It is the first reasoning in which one concludes **without knowing**. Many pupils resist it ; it is worth installing slowly.

D.6 — The crossing

	A	B	C	D	E	F	
1	6	5	1				1
2		2				4	2
3	3		5				3
4				1			4
5				3			5
6			6		5		6
	A	B	C	D	E	F	

Figure D.5 — The 3 of column C is shut inside a single block

1. In **column C**, the **3** can only go in **C1** or **C2**. **2.** One does not know which of the two — but both are in the same **block**. **3.** So the 3 of that block is necessarily in column C : it disappears from the rest of the block, here from **B2**.

The reasoning travels from one group to another by their crossing. It is the rule that most often unblocks a grid that seems to be standing still.

D.7 — The trio

	A	B	C	D	E	F	
1	6	5					1
2	1	1 2 3	1 2 3	2		4	2
3	3		5				3
4				1			4
5				3			5
6			6		5		6
	A	B	C	D	E	F	

Figure D.6 — Three cells, three values, and the rest of the row is freed

1. In **row 2**, the cells **A2**, **B2** and **C2** carry only **1**, **2** or **3**. **2.** Those three values are therefore taken by those three cells. **3.** They are removed from the rest of the row : **D2** and **E2**.

It is exactly the pair, with one cell more. It is not a new idea — it is the same one, longer to see, and that is precisely what makes it harder.

D.8 — Using them in class

Two uses, and the second is the more fruitful.

Naming them, once. The six rules hold on the board in six lines. Writing them there once makes it possible afterwards to refer to them in a word — «that is a forced place» — and saves a great deal of explaining.

Making them say why. In front of a cell that has been found, ask not for the value but for the rule : «how do you know?» is worth more than «what did you put?». It is also what the printed solving list does : one line per deduction, with the rule that justifies it.

Two things to tell the pupils about it.

First, a few lines place no value at all : they **cross out** possibilities — «in this row, these two cells share the 3 and the 7, so neither of them is anywhere else». That is a step in its own right, and often the most instructive one.

Then, and it is what the sheet announces at its head, it is **one path among others**. That is not a stylistic precaution : on an ordinary grid, **more than one cell in two** can be justified by at least two different rules. A child who proposes one other than the printed one is therefore not wrong — they must be told so, and asked to defend it.

Lexique

cage — Group of neighbouring cells, surrounded by a line, carrying a label such as « 7+ »: the numbers of the cage must give that result by that operation. It is the constraint proper to the Calcudoku, the one that comes on top of the sudoku rule. → Chap. I, Chap. VIII

game code — Short string of characters — « G4CS41OP-a30y6s-IU » — which designates one precise grid: the generator version, the mode, the geometry, the size, the level, the options and the number of the grid. Typed on another device, it gives back exactly the same grid, because the game makes it from that information instead of storing it. → Chap. III

state code — Long form of the game code, which carries in addition what is already written in the grid: the values placed and the pencilled candidates. It allows a game in progress to be passed on, and therefore a pupil's work to be taken up just as they left it. → Chap. III, Chap. VII

Table of figures and tables

Figures

N°	Titre	Page
2.1	The home screen : every tile is a setting, and displays its value	10
3.1	The Code button, to the right of Play : the only way in for a game	13
3.2	The QR displayed on the board ; the class scans it, everyone has the grid	14
4.1	Four cells a side : the rule holds in one screen	19
5.1	The candidates are written small in the corner of the cell	23
7.1	Every game keeps its clock and its progress	27
8.1	The  tile : four pills, and addition alone switched on	31
9.1	The pairs carry a  , the larger cages a 	34
10.1	The  tile : « Random », between 1 and 10	36
10.2	Every cage holds in two cells : each one is a product	37
11.1	The four signs sit side by side ; division is written « : »	40
12.1	Two independent sliders : the cells, and the results	44
13.1	 and  designate the same cell ; one has to see it	46
14.1	Every region assembles a complete figure	49
B.1	The game warns at the opening, with the percentage and the reason	58
D.1	Blue, the row ; orange, the column ; plum, the block	63
D.2	The crosses mark the cells where the 4 cannot go	64
D.3	A cage of two cells that must make 20 by multiplying	65
D.4	In plum, the pair ; in orange, the cells it frees	65
D.5	The 3 of column C is shut inside a single block	66
D.6	Three cells, three values, and the rest of the row is freed	67

Index thématique

NOTIONS COVERED

addition → Chap. VIII
decimal writing → Chap. XIII
division → Chap. XI
fractions → Chap. XII
geometric shapes → Chap. XIV
multiplication tables → Chap. X
subtraction → Chap. IX

RUNNING A LESSON

at the board → Chap. IV
correction → Chap. VII
group work → Chap. V
homework → Chap. VII
individual work → Chap. VI

THE GAME

free version → Chap. II, App. B
game code → Chap. III
pencil → Chap. V
QR → Chap. III
web demo → Chap. II

Index alphabétique

A

addition → Chap. VIII
at the board → Chap. IV

C

correction → Chap. VII

D

decimal writing → Chap. XIII
division → Chap. XI

F

fractions → Chap. XII
free version → Chap. II, App. B

G

game code → Chap. III
geometric shapes → Chap. XIV
group work → Chap. V

H

homework → Chap. VII

I

individual work → Chap. VI

M

multiplication tables → Chap. X

P

pencil → Chap. V

Q

QR → Chap. III

S

subtraction → Chap. IX

W

web demo → Chap. II

Twenty-five pupils, fifty minutes, and the same grid for everyone.

InZeDoku is a sudoku game in which one has to count in order to deduce. This file explains what one does with it in class : one lesson in four stages — at the board, in groups, alone, as homework — and worksheets sorted by notion, from the first additions to fractions and decimal numbers.

Every worksheet carries a code of a few characters. The pupil types it, or scans the projected QR, and gets **exactly** the grid of the worksheet. That is what makes it possible to correct at the board, to set homework, to compare two papers — and to take up a pupil's work where they left it, hesitations included.

The game sets the difficulty of the **counting** and that of the **deduction** separately : a board five cells a side can carry the multiplication tables of a nine-year-old as well as the halves of a seven-year-old.

Everything described here is played in the free version. Offline, with no account, no advertising, and not the least data collected.

Nothing in this file is an instruction : you are master in your own house.

