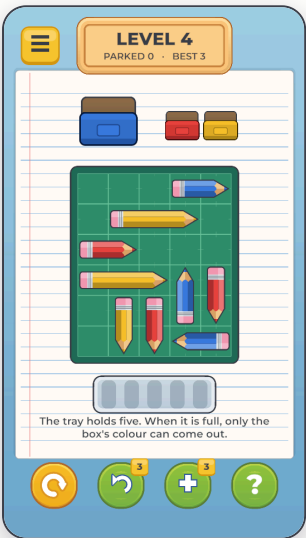


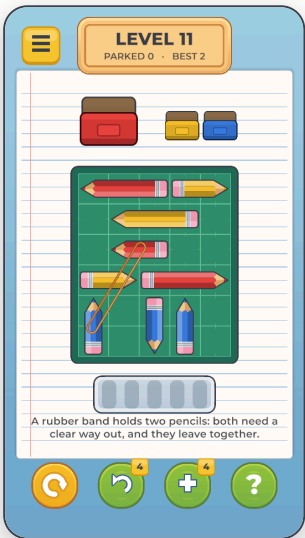
Pull-out

Tap the coloured pencils out of the pile and into the box of their colour. The boxes come one at a time, and the pencils that don't fit the box at the front wait in a tray of five.

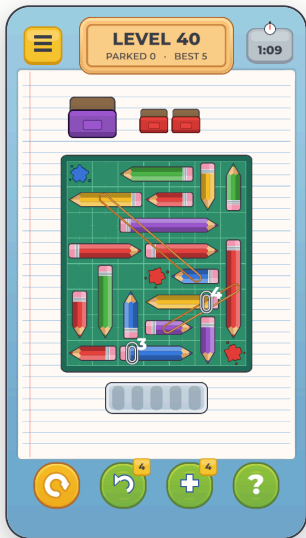
AT A GLANCE	
Genre	Hyper-casual tap-away puzzle with a queue of colour boxes
Platforms	Portrait phones (Android) and the browser; Unity 6, URP
Controls	Tap a pencil: if its way out is clear, it slides out
Session	A level is 6 to 18 pulls; par is pencils parked, 0 to 6
Content	40 levels in 4 chapters of 10, 7 of them timed; up to 5 colours on a 7×8 mat
Status	Playable prototype 0.2.0: web build and Android APK; not yet tested with players



Level 4: the tray starts to matter



Level 11: the first rubber band



Level 40: everything together

The core loop

- **Read:** which pencils are free, and which box comes next?
- **Pull:** a free pencil slides out the way it points.
- **Box or park:** into the front box if it matches, otherwise into the tray.
- **Roll:** a full box rolls away and waiting pencils hop into the next. All boxed wins; stars for the fewest parked.

Who it is for

Players of tap-away and sort puzzles who like a short, calm plan. The queue adds the sort genre's question (what goes where) to the tap-away question (what comes out first), and the tray is the budget that joins them.

Design pillars

- **Readable at a glance.** A pencil points the way it will go; the next boxes are always on show.
- **One new idea per chapter,** brought in on an easy level.
- **Fail only against the clock.** A dead end is named and restart is free; only timed levels can be lost.
- **No decoration.** A level's newest mechanic must change its par or the tray it needs; the generator checks it.

RULES

Pencils, boxes, the tray and three twists

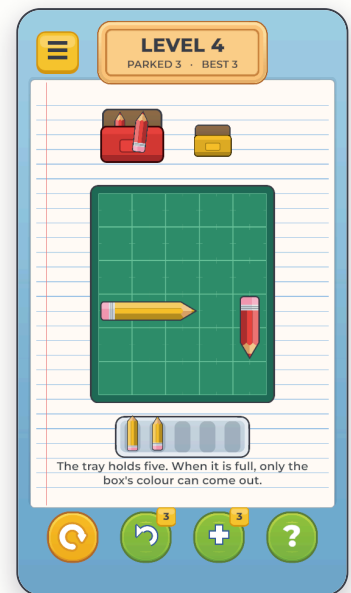
Pulling

Pencils lie on a cutting mat, each two to four cells long and pointing one way. Tap one: if nothing stands between its tip and the edge of the mat, it slides out that way. A pencil in the way flashes and the tapped one nudges back.

Boxes and the tray

Boxes queue at the top, three pencils to a box; the front box and the next two show. A pencil goes into the front box if it is that colour, otherwise into the tray. A full box rolls away with a red tick, the next arrives, and any pencils of its colour waiting in the tray hop in. The tray holds five; a pull that would overfill it is refused. When no pull is possible the game says so, rings the tray and pulses restart.

Why par counts pencils parked. Every win takes the same number of taps, one per pencil, so moves can't score a level. What a good line saves is the tray: the solver's par is the fewest pencils parked, and stars are three at par, two within two of it, one otherwise.



A box rolls away; the waiting reds hop in

Rubber bands (chapter 2)

Two banded pencils leave together. Each needs a clear way out (its partner never counts as in the way), and both land, the tapped one first; if the second couldn't land, the pull is refused.

Paper clips (chapter 3)

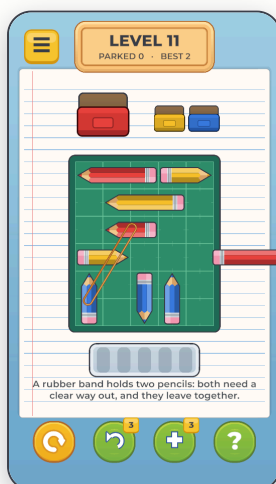
A clipped pencil can't move until that many pulls, of any pencils, have been made. The clip shows the count and drops off at none.

Ink blots (chapter 4)

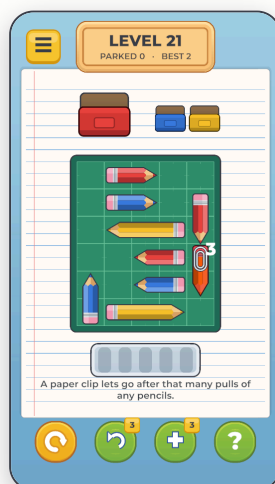
A blot stops any pencil passing over it until a pencil of the blot's colour is out, boxed or parked.

Boosters and the clock

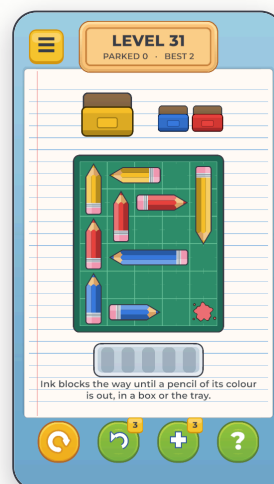
Two boosters, three of each to start and one more of each for finishing a chapter: **an extra tray slot** for the level, and **undo**, which takes back a pull but caps that win at two stars (undo makes par reachable by trial and error). Levels 10, 15, 20, 25, 30, 35, 40 are timed: 15 seconds plus 3 a pencil, the clock starting with the first pull. Out of time offers a retry or 15 more seconds once, the slot a rewarded ad would fill.



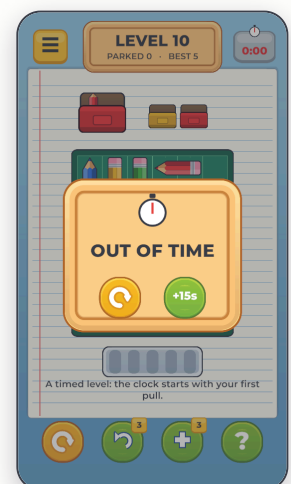
A banded pair leaves together



A clip refuses: 3 pulls to go



Ink in the way flashes



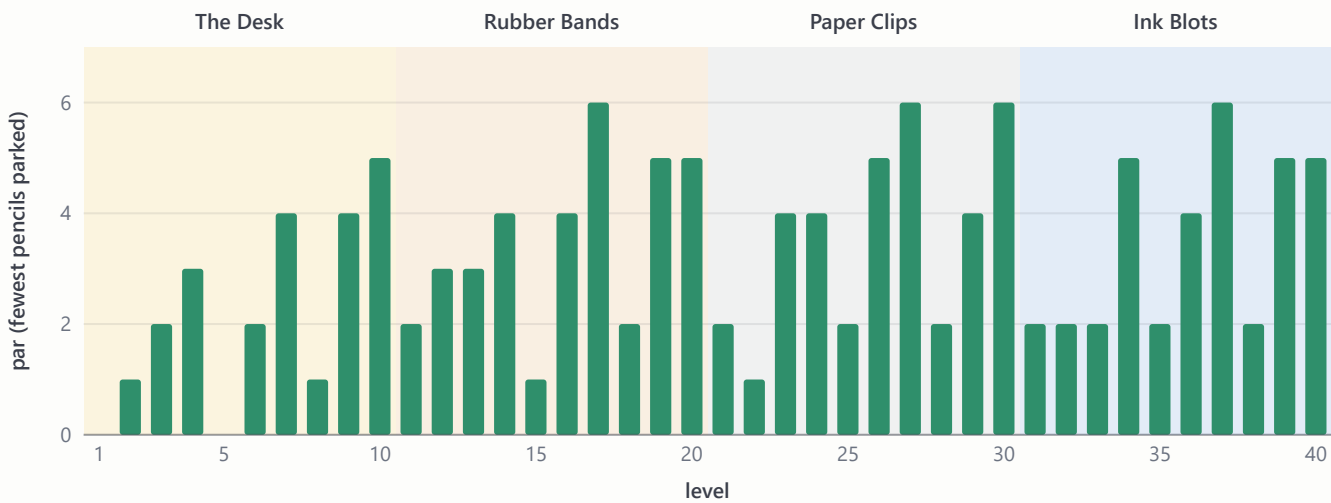
Out of time: retry or +15 s

Four chapters, one new idea each

Each chapter brings in one idea on an easy level, then builds on it. Within a chapter par climbs in a sawtooth, with a breather after the peaks; every fifth level from 10 is timed.

CHAPTER	NEW IDEA	MAT	PENCILS	COLOURS	PAR	TIMED
1. The Desk	Pulling, the box queue and the tray	4×5–6×7	6–15	2–4	0–5	10
2. Rubber Bands	Rubber bands	5×5–6×8	9–15	3–5	1–6	15, 20
3. Paper Clips	Paper clips	5×5–6×8	9–18	3–5	1–6	25, 30
4. Ink Blots	Ink blots, then everything together	5×5–7×8	9–18	3–5	2–6	35, 40

The difficulty curve: par per level



Par is a stand-in for difficulty, not a measure of it. The workbook adds the tray each level needs (the smallest that still wins): levels 10, 17, 26, 27, 30, 34, 37, 39 need all five, so a single wasted park ends them. Breathers (a drop of two or more): levels 5, 8, 11, 15, 18, 21, 25, 28, 31, 35, 38.

Teaching without a tutorial

A new idea gets a one-line tip on the level where it first appears, and level 1 shows a hand on the first pencil to tap.

LEVEL	THE ONE-LINE TIP
1	Tap a pencil to slide it out the way it points.
2	Pencils go in the box of their colour. Any other waits in the tray.
4	The tray holds five. When it is full, only the box's colour can come out.
6	Stuck? + opens a sixth tray slot; the arrow takes back a pull (two stars at most).
7	Check the boxes coming next before you pull.
10	A timed level: the clock starts with your first pull.
11	A rubber band holds two pencils: both need a clear way out, and they leave together.
21	A paper clip lets go after that many pulls of any pencils.
31	Ink blocks the way until a pencil of its colour is out, in a box or the tray.

How the levels are made and proven

1 • The spec: what I decide

Each level is one line in the plan: the size of the mat, the pencils of each colour and their lengths, how many start blocked, how many bands, clips and blots, the range its par must fall in, and its focus: the newest mechanic.

2 • The generator deals it

It lays pencils at random, shuffles the boxes, and keeps a deal only if:

- no pencils block each other in a circle, and enough start blocked,
- the solver's par falls inside the range,
- the focus mechanic matters: switched off (no bands, no clips, no blots), the level's par or the tray it needs changes.

3 • The solver proves it

A search over (pencils pulled, pencils boxed) by cost: boxing is free, parking costs one, a banded pair can cost two, so it keeps one queue per cost (Dial's algorithm) and the first win it reaches parks the fewest. The tray, the clips and the blots all follow from those two numbers, which keeps the search small. The same solver gives hints in the game.

The level file

Plain text, readable and editable by hand. Level 11, the first rubber band:

```
=== 11 | Banded | par 2 | tray 5 | boxes r y b |
focus bands
grid 5 6
r 1 3 < 2
b 3 1 ^ 2
r 4 2 > 3
r 0 5 < 3
y 1 4 < 3
b 0 1 ^ 2
y 1 2 > 2
y 4 5 > 2
b 2 0 v 2
band 0 5
```

Tests that guard the design

- Every level's par equals what the solver finds; every level is well formed (no overlaps, a box for every three).
- Every focus mechanic changes the par or the tray needed; timed levels are every fifth from 10.
- The rules: blocking, boxing, parking, a full tray, a box rolling with hops, bands, clips, blots, the extra slot, the undo star cap, a dead end.
- Playthroughs win levels 1, 4, 10, 11, 21, 31 and 40 in the real scene; others check undo, the extra slot, the clock running out and the map.

What the solver and the bots found

Pull-out has not been tested with players yet, beyond my own play of each build. These findings come from the solver and from bots playing the real rules; each changed the design.

1 • Clips and blots tied to boxes did nothing

First version: a clip held a pencil until a number of boxes had rolled away, and a blot lasted until its colour's box had. In 300 random deals per layout neither ever changed a par or the tray a level needs. The box order is fixed, so what they hold back is either needed before it is let go (the level can't be won) or it isn't (it costs nothing).

Change: released by the player's own pulls instead: a clip after that many pulls, a blot once a pencil of its colour is out. Now they changed the par in roughly one winnable deal in nine to one in five, and the tray needed about as often: enough to generate from.

2 • The measure was wrong as well

A delay, or a smaller tray, rarely makes a player park more pencils; it makes more of them wait at once. Par alone missed that.

Change: the focus check also counts the tray a level needs, the smallest that still wins. Rubber bands mattered from the start: the tray needed changed in 30 of 48 winnable deals.

3 • A banded pull can park two

The C# solver priced a banded pull by its first pencil only, and both solvers used a search that assumes every step costs nothing or one. The tests caught the first (level 11's par was 1 in C#, 2 in Python).

Change: both solvers now search by cost with one queue per cost; level 14's par had been one too high and the level was dealt again.

Watch first in playtests

A greedy bot (it boxes a pencil when it can, otherwise pulls at random) dead-ends most often on levels 17 (98%), 27 (94%), 34 (99%), 37 (96%). Does a band read as holding two pencils? Is a clip's count noticed? Is 3 seconds a pencil the right clock?

Juice, art, monetisation and the roadmap

Game feel

- A pulled pencil shoots off the mat, then arcs into its box or the tray, standing upright.
- A refused pull nudges back and flashes what is in the way; a full tray shakes; a clip wiggles.
- A full box gets a teacher's red tick and rolls away; the next slides in and waiting pencils hop across.
- A band snaps when its pair leaves; washed ink fades; a win rains confetti and the stars pop in.

Art and audio

The buttons and panels are cut from Coffee Beans Studio's Cute Cartoonish UI Pack (free demo, modification allowed). The pencils, boxes, tray, cutting mat, notebook page, bands, clips and ink are drawn in code in its style. Type is Montserrat; sounds, the pointer and the stars are Kenney's.

A kit tried and dropped. A stationery sticker kit fitted the theme best, but its free PNGs had no transparency and its licence allowed changes only in the paid version, so the notebook page and red marks are drawn instead.

Built with

Unity 6 (URP). The rules live in a pure C# library with no engine code, mirrored in a Python tool that generates, probes and measures the levels. Built with an AI coding assistant (Claude): I set the rules, the mechanics, the chapter plan and the tuning, reviewed every build and decided each change.

Monetisation (a plan, not built)

The prototype has no ads or purchases. As a hyper-casual release it would follow the genre:

- **Rewarded rescues** where the game already offers one: the extra slot at a dead end, +15 s when time runs out.
- **Interstitials** between levels, never mid-level, not in the first chapter.
- **Booster packs** and **remove ads** as purchases.

What to measure first

- Dead ends and boosters used per level: are the bots' walls walls for people too?
- Refused pulls by reason (blocked, clipped, ink, tray full): is each rule understood?
- Timed levels: wins, retries and +15 s taken.
- D1 retention and the level where players stop.

Roadmap

- Playtests with first-time players; tune the tray, the clock and the boosters from what they show.
- A visual level editor in Unity on top of the generator and solver.

Credits

Buttons and panels: Cute Cartoonish UI Pack by Coffee Beans Studio (coffee-beans-studio.itch.io). Sounds, pointer and stars: Kenney (kenney.nl, CC0). Font: Montserrat (SIL Open Font License).