How we built ten animated timers where the picture is the time

Peabody blog · Habits · updated 2026-10-29

A gorilla eats a pizza while your class works. Here is the one rule that lets a picture be a clock, and what we refused to use to draw it.

A timer a five-year-old can read has to say the time in something other than digits. Ours says it in pizza: eight slices, one for each eighth of however long you set, and the last slice goes as the clock reaches zero. That only works if the picture is drawn from the time itself and never from its own animation — which turned out to be the single rule that made the other nine pictures possible too. You can try all ten at Peabody Timers, free through the end of 2026, with no ads and no sign-in.

We wrote about the 45 free timers we opened before building ours, and we ranked ours among them with the disclosure that we made it. This post is the other half: how the thing is put together, for anyone who wants to build one or who wonders why ours behaves the way it does.

The rule: one number, and the picture is a function of it

Most animated timers are an animation playing next to a clock. The animation runs on its own frame loop at whatever speed the laptop manages, and the clock runs on the clock. They agree at the start and drift apart by the end, which is why so many of them look right for thirty seconds and wrong at the finish.

Ours has one number, progress, which goes from 0 to 1 and means "how much of the set time has gone". Every picture is a pure function of it. The gorilla scene is handed 0.5 and answers: four whole slices, one half-eaten, a gorilla a little fuller than he was. It is not told how many frames have passed, what time it is, or how fast the machine is. It cannot be, because it has no way to ask.

That sounds like a restriction and is actually the whole feature:

- A sleeping tab cannot break it. The clock is measured as the difference between two readings of the real time, so a laptop that slept through four minutes of a five minute timer wakes up showing one slice left.
- Adding thirty seconds mid-run is honest. Press the right arrow and the pizza grows back by the right amount, because the picture is re-derived rather than rewound.
- A shared link opens the same picture. The length, the scene and the direction ride in the address after the #, so a teacher can bookmark "seven minute gorilla" and get the same thing tomorrow.

- Reduced motion still works. Someone who has asked their computer for less movement gets the picture standing still at the right point, rather than no picture at all.

Anything that genuinely is decoration — a chew, a blink, a wing beat, snow falling, the fish swimming — runs free and carries no information. If you cannot tell the time from it, it is allowed to be loose.

Counting up is not the film run backwards

Every timer here goes both ways, because a tidy-up is not a deadline: the job is to keep going until the time is up, not to beat it. The naive version of counting up is to play the countdown in reverse, and it looks wrong immediately. Reversed, the popcorn timer starts with a full bowl and un-pops it; the dinosaur egg starts hatched and the chick climbs back in.

So each picture has two stories rather than one film. Counting up, the popcorn bowl fills pop by pop, the train waits for its track and pulls in when the last sleeper is down, the egg is built from its own fragments with no chick in sight, and the caterpillar stays a caterpillar while the leaf grows back. Ten scenes, twenty stories, each one drawn from the same single number.

What we refused to use

We have a 3D engine. We did not use it here. A classroom timer is a page a teacher opens on school hardware most days of the year, and the pages that run our 3D games are budgeted at over a megabyte of JavaScript. So every scene is plain graphics — SVG for nine of them, one canvas for the pond where fish have to crowd round the food — and each page loads only the picture it is showing, not the other nine.

The sounds are made in the browser out of oscillators and filtered noise: a burp for the gorilla, a plop for the pond, a whistle for the train, pops for the popcorn. Nothing is downloaded, so a sound costs a few hundred bytes of code instead of a megabyte of audio, and it still works on a laptop that has lost its network. There are three volumes, because a quiet classroom and a hall are not the same room, and a button that plays the sound to the teacher before it plays to thirty children.

The test that keeps it honest

The rule above is only true if nothing quietly breaks it, so it is a test rather than a note in a file. Every scene is drawn at twenty-one points of both its stories and the pictures are hashed against a recorded fixture. A refactor that moves a picture by a hair fails. A deliberate change re-records the hashes in the same commit and has to say why.

The same test asks the obvious questions a person would forget to: does the stuff only ever go one way, is the start exactly full and the end exactly empty, does every part stay on the screen, and does any scene file reach for a clock, a frame count or a random number. That last one is a search of the source, which is a blunt instrument and exactly right for a rule that

must not have exceptions.

What it does not do

It does not teach the clock by itself, though the kid version shades the slice of a real clock face the timer will use and says where the big hand will be at the end. It keeps nothing about a child: no score, no streak, nothing saved but the last picture chosen, in that browser. And nothing on Peabody starts a timer for a child — a timer here is one a child or a grown-up opens and points at their own job, which is a line we wrote down before we built it.

If you want the ten pictures, they are at Peabody Timers, and every free timer we checked is listed at the index of free online timers. If you would rather your child read something, that is what the rest of the site is for: the library is open, free through the end of 2026.

Free at peabody.kids/blog/how-we-built-animated-timers