

How to Write a Fun Puzzle

Jay Lorch

Abstract

The primary goal of writing a puzzle is to entertain the solver. This article describes tips for writing puzzles that will be fun to solve.

1 Introduction

This article presents guidelines and suggestions for how to write a puzzle. The aim is to teach you how to write a puzzle that solvers will find fun. After all, a puzzle is a form of entertainment, so your goal should be to make your solvers have fun.

The article describes how to write a puzzle, and important principles of puzzle design, including:

- Your puzzle should be fun for everyone.
- Your puzzle should be solvable by everyone.
- You should use intrinsic hints, rather than relying on explicit ones.
- Keep flavor text minimal.
- Test on many independent solvers.
- Use tester feedback to repeatedly refine the puzzle.
- Talk to your mentor early and often.

2 Fun

The #1 rule in writing a puzzle is that you should make it fun. It's also the #2 and #3 rules! People solve puzzles because they want to be entertained, so it's your job when writing a puzzle to make it fun. People will remember your puzzle fondly or not depending essentially on whether they had fun solving it.

In contrast, your goal is *not* to torment, educate, fool, stump, tease, dominate, or evaluate the solver. You're not writing an entrance exam or interview regimen. You're designing a fun activity that will entertain your solvers.

2.1 Choosing fun activities

To this end, you should ensure that most of the time the solver is doing fun activities. One way to figure out ideas for fun activities is to look at existing puzzles, particularly ones you enjoyed, and see what activities constitute the core fun part of them. Some good sources of existing puzzles are Games Magazine, Nikoli, and Panda Magazine.

Indeed, a good way to start designing a puzzle is by thinking of a fun activity, then figuring out how to construct a puzzle that will require the user to engage in that activity. Ideally (but extremely rarely), that activity will be one that no one's ever designed a puzzle around. Much more often, you'll start with a fun activity that you've seen used in a puzzle before, but you'll make it unique with a new, interesting, or thematic twist.

2.2 Avoiding un-fun activities

Part of making sure the solver is spending the bulk of their time on fun activities is making sure they spend as little time as possible on un-fun activities. Un-fun activities include:

- decoding encoded information, like Braille or Morse;
- searching the Internet for information;
- indexing, i.e., finding the n th letter in a string;
- counting things.

Such un-fun activities can be useful in letting a solver extract an answer from a puzzle, so you'll often see them even in well-designed puzzles. However, the writers of such well-designed puzzles make sure that solvers spend as little time as possible on such activities. In other words, feel free to use such un-fun activities sparingly, but don't make them the core of your puzzle. Few people would find a puzzle fun if its sole activity were, say, indexing.

Be particularly careful not to require solvers to search the Internet a lot to solve your puzzle. It's OK if they have to search a tiny bit, to fill in small gaps in their knowledge. However, they shouldn't be spending more than a little time on this. People enjoy retrieving information from their brains, including and even especially if it's tricky to find that information in their brains. But, they don't enjoy retrieving information from the Internet.

3 Solvability

An important component of making your puzzle fun is making it solvable. A person will enjoy a puzzle much more if they eventually manage to solve it. Your primary goal shouldn't be to make the solver, after finishing the puzzle, think, "Wow, the author was really clever!" You want the solver to say, "Wow, *I'm* clever!"

Mark Gottlieb made the astute observation that “a puzzle is a hard-fought battle between the constructor and the solver that the constructor must lose.” The constructor has all the advantages: they understand deeply how the puzzle works, they built every piece of it, and they know the answer. Yet, ultimately the solver must prevail by solving the puzzle. The constructor must insert clues into the puzzle as necessary to allow it to be defeated by the constructor.

If there’s a trick the solver must discern before they can make progress, it’s important to give them a *hook*. A hook is a clue that’s easier than the rest, and whose solution suggests more clearly than the rest what needs to be done. You should know which clue or clues are the hooks for your puzzle, and put at least one of them in a prominent position. The best place for a hook is early in your puzzle: among the first 3–5 clues, but not at the very beginning. Having a clue or two before the hook gives the solver the experience of wonder, which is then relieved by the hook that allows them to make progress.

If you’re constructing a logic puzzle, it’s not sufficient to just prove that there’s a single solution to your puzzle. You need to construct the puzzle so that there’s a logical path to the solution, and so that the solver will discover that path as they solve. In other words, you should understand how solvers will solve your puzzle.

To guide solvers to the solution, it’s useful to give them somewhere to put their intermediate progress. Solvers particularly welcome sequences of blanks, or grids, where they can fill in answers and thereby get confirmation that they’re entering correct information.

4 Approachability

A solver can’t solve your puzzle if they never pick it up. And, even if you get them to pick it up, you need to give them enough sense of progress so that they keep working on it rather than set it aside in frustration.

To get solvers to pick up your puzzle, you should make it attractive. A well-produced puzzle is a signal of attention, which a solver will (consciously or unconsciously) take as evidence that you’ve spent a lot of time on it and therefore that it will be fun and solvable. There are various ways to make a puzzle appear well-produced, including:

- Use color rather than pure black-and-white.
- Use background fill and drop shadows.
- Include pictures as components of your puzzle.
- Use an attractive, professional font.
- If your puzzle has a list of things, format each element of that list identically.
- Format the puzzle consistently with other puzzles in whatever set or event it’s part of.
- Leave the solver a generous amount of space to fill in whatever they need to fill in.

Another element of making your puzzle approachable is giving the solver somewhere to start. In other words, include components that the solver recognizes immediately as presenting something to do, like:

- a set of pictures that need to be identified,
- a set of clues to answer, or
- a recognizable puzzle type like a crossword or sudoku.

And, as discussed in §3, it’s important to have a hook early in your puzzle. That way, they’ll feel a sense of progress and not discard your puzzle out of frustration.

5 Mentoring

It’s valuable to have multiple perspectives when authoring a puzzle, which is why it’s useful to have a mentor.

One of the roles of a mentor is someone to talk over ideas with you at an early stage, before you do too much work on them. This way, they can give you feedback before it becomes onerous to rework the entire puzzle to take it into account. It will also help you avoid wasting the effort of trying to write the puzzle if they can see some fundamental problem with it that makes it wholly unworkable. They may also be able to suggest an improvement to the structure of the puzzle that’s much easier to incorporate at the outset than after the puzzle is written.

Another role of a mentor is to look at early drafts before they go out to testers. This helps make better use of testing resources, by avoiding wasting a playtest on a buggy version. It also helps you stay on good terms with your testers, since giving them a broken puzzle may discourage them from testing the next puzzle you ask them to test.

A mentor can also represent your puzzles at a planning meeting. Keep them up to date on where you are in the puzzle development and testing process, and they can pass that information on in meetings when called for. They can also watch out for group-wide decisions that might impact your puzzles.

6 Structure

Once you’ve figured out an idea for a puzzle, and the fun activity that will form its core, you should devise the structure for the puzzle. That structure should be as streamlined as possible; if you can’t explain the steps necessary to solve your puzzle in a short paragraph, it’s likely too complicated.

Usually, the structure of a puzzle includes an aha, an activity, and an extraction. The aha is the step of figuring out the nature of the puzzle and what there is to do. The activity is the fun core of the puzzle that takes most of the solver’s time and is enjoyable; recall from

§2 that the activity the solver spends the most time on must be fun. The extraction is the step of obtaining a final answer from what they've done.

Although the above is a typical structure, an often-preferable alternative is to swap the order of the activity and the aha. The nice thing about making this swap is that the solver has the immediate feeling of making progress on your puzzle and so is less likely to set it aside and not work on it. The downside of this structure is that it can lead the solver to the annoying experience of doing a lot of work and then getting stuck. So, it's good to alleviate this by having the aha be an easy one if you decide to swap the order this way.

6.1 Ahas

Sometimes a puzzle will have more than one aha. This will make the puzzle harder, but will often make it more enjoyable for solvers wanting a harder experience.

Each aha must produce a clearly non-random result, to verify to the solver that what they've done is correct. After all, the solver doesn't know that the aha is correct, and may have a variety of hypotheses as to what's going on. If the solver tries to apply an aha and gets something that looks random, the solver won't persist. The solver will rationally decide that the aha wasn't right and will move on.

This is why the result of following an aha shouldn't be a sequence of letters that must subsequently be anagrammed. You've probably heard the now-well-understood puzzle-design principle that a puzzle should never end in an unclued anagram. You can now see that this is just a special case of a more general principle: no step of a puzzle should end in an unclued anagram, because the results of an aha must produce non-random-looking results.

It's not always sufficient for the sequence of letters produced after an aha to just be in order. That order furthermore has to start with something that looks correct. After all, if the solver gets what looks like garbage when they first try to apply the aha, they're not going to keep going. For instance, if the clue phrase is GM'S DHIVYA, a solver will start out getting GMSDH, which looks like too many consonants in a row, and probably conclude they're off track. So, make sure that the extracted message starts out looking valid; if it doesn't, preface it with an intro. For instance, if the answer is KUMQUAT, a possible message is ANSWER KUMQUAT. And, if an intermediate extraction needs to tell them to use J Crew catalog numbers, consider NOW USE J CREW CATALOG NUMBERS instead of just J CREW CATALOG NUMBERS.

6.2 Extraction

Most puzzles require a final step to convert what the solver obtains to an answer. Ideally, this extraction step

should be natural, i.e., it shouldn't require a tricky aha to find. After all, typically the solver has already put a lot of work into the puzzle by this point and feels they deserve to know the answer. They can get frustrated to feel they've completely solved the puzzle but can't get its answer.

The extraction step should also be quick to apply. For instance, it shouldn't require a lot of busy work like decoding or counting. After all, the extraction mechanism is only necessary because the puzzle has to resolve to a short answer; it's rarely one of the fun steps of the puzzle. So, by the principle that the bulk of the solver's time should be spent on fun activities, you should ensure the solver doesn't have to spend more than a trivial amount of time on extraction.

It's common for the extraction step to be a recursive application of a key earlier step of the puzzle. For instance, if the puzzle is a bunch of movie-identification clues, the final step could be to solve one more movie-identification clue. However, unlike all the other clues in the puzzle, you must make sure this final clue is solvable by everyone. After all, the solver can work around not being able to solve a few of the initial clues, but they have to be able to solve the final clue to get the answer. So, you have to make sure that the final clue is easy enough that anyone can get it.

Similarly, if the final clue is a piece of trivia, it must be gettable by everyone. Ideally, people will know the answer to the trivia question without consulting the Internet, but it's OK if sometimes they need the Internet for the final clue. However, if you're relying on the Internet to make the clue solvable, you must make sure that it's obvious to solvers what to search for and that the result appears prominently among the search results.

When the extraction is just reading a sequence of letters in order, the puzzle can often be short-circuited. That is, the solver can get the answer without solving most of the puzzle. This isn't always a bad thing; solvers often have a happy reaction to figuring out the answer with less work than seems to be expected. However, sometimes solving like this will spoil the solver's fun because it takes away the chance to solve more fun clues. So, it's sometimes helpful to have a slightly more elaborate extraction mechanism to prevent such short-cutting. For instance, you can have the extraction provide not only a sequence of letters but also the order in which they go. Don't overdo this elaboration, though. After all, it's natural for there to be some parts of the puzzle the solver can't get, and you should make sure the puzzle can be solved without those parts. You don't want the solver to have the frustrating experience of trying in vain to solve the last few clues that, for some reason, they have extreme difficulty solving.

7 Hints

You must hint every aha in the puzzle.

When I say “hint” I mean something *within* the puzzle that guides the solver toward the aha, not something external to the puzzle that’s provided to the solver if they’re having trouble solving it. Some puzzle events offer the possibility of providing hints to solvers who are behind, or of letting solvers spend tokens for hints. However, solvers far prefer to solve a puzzle on their own, without receiving external hints, even if they have to pay tokens for those hints. So, if testing reveals that the hint you’ve included in the puzzle for an aha isn’t sufficient for some solvers, you shouldn’t respond by saying that you’ll make an external hint available for such solvers. You should correct the issue by changing the hint internal to the puzzle to one that’s more useful to solvers.

Often when someone thinks of a hint they think of flavor text or the title, or an intermediate extracted message. But, the best type of hint is an *intrinsic* hint. By this I mean a pattern to the clues or their answers that suggests what to do. And, if the pattern is among the answers rather than the clues, then there must be sufficiently many easy clues that the pattern can be discerned from the answers to those clues. For instance, if all the clues’ answers have two X’s with another letter in between, then it’s pretty clear that this is important and the likely place to look for the next step is the letters in between.

If an intrinsic hint is, or turns out to be, too subtle, it can be helpful to include further hinting in text. Such text hints can be part of the flavor text or title. In this case, the text hint is acting as a *confirmer* of an intrinsic hint. That is, the solver isn’t expected to discern the aha from the confirmer. Instead, the solver is expected to discern the aha from the intrinsic hint and, if they’re not sure they’re on the right track, look at the text and find confirmation that indeed they are.

An example of an intrinsic hint plus a confirmer is in my puzzle *Cosmopolitan*, available online at <https://jaylorch.net/puzzles/Cosmopolitan/>. Here, a lot of the answers are the names of cocktails, intrinsically hinting that cocktails are relevant. However, this intrinsic hint is somewhat subtle because not all of the answers are the names of cocktails. The title helps a solver who’s thinking that perhaps cocktails are relevant see that indeed they are. If someone is already thinking of cocktails, they’ll recognize that *Cosmopolitan* is a cocktail and thus feel more certain that they’re onto something. But if someone isn’t thinking of cocktails, especially in the context of a puzzle set themed about magazines, they likely won’t see this meaning of the title. Thus, this title is merely a confirmer, not a direct hint.

Sometimes you can’t think of a way to include an intrinsic hint and you have to fall back on including

the hint in text. This can be the title, or flavor text, or an intermediate message produced in the course of solving. Be careful in this case that such hints are sufficiently expressive to guide the solver to the appropriate aha. I recall once playtesting a puzzle where the word *ROYALTY* was meant to guide the solver to consider the numbers in the puzzle as dates during the reign of queens of England; this was far too big a leap. If you’re trying to clue something elaborate, you’ll need a lot of words to clue that elaborate concept.

Since flavor text often contains confirmers and/or direct hints, you should keep flavor text to a minimum. It’s frustrating to comb through paragraphs of flavor text looking for the small tidbit that is the relevant material. Furthermore, you’d be surprised how much solvers can read into the benign material you include as flavor text without intending it to have meaning. It’s even more frustrating to pursue useless avenues because one was guided down those avenues by extraneous flavor text. The ideal flavor text length is no flavor text at all, with all hints confined to just the title. If you must have flavor text, try to keep it to a sentence or two.

In some cases, the best hint to an aha is an explicit instruction giving away the aha entirely. If there’s an aha that expert solvers will immediately deduce but which novice solvers will have difficulty with, it’s usually best to just give it away with an explicit instruction. For instance, if your puzzle is a cryptic crossword then experts will immediately recognize it but novice solvers may be completely unfamiliar with the form. So, provide solvers with an explicit indication that it’s a cryptic crossword and perhaps a link to a page with solving tips.

8 Presentation

An important principle of puzzle design is that everything must be clued twice. The most recognizable example of this is in a crossword, where each letter is clued twice, once in an across word and once in a down word. The reason for this principle is that it’s very difficult to clue something in a way that doesn’t admit alternate answers. Cluing everything in two different ways makes clear to the solver that unintended alternate answers are wrong. Supplying an enumeration (letter count) for an answer helps confirm it, but isn’t sufficient to count as having clued the answer a second time. The only time you don’t have to clue a letter in an answer twice is when that letter is meant to be extracted as the final message. Its inclusion in that message, which must read sensibly, is sufficient to count for the second cluing.

Pay attention to the order you present things in your puzzle, since solvers (at least expert ones) will expect you to. If the order you present something is important, let the solver know this, e.g., by numbering them. More importantly, if the order you present something in is *unimportant*, make clear to the solver that it’s arbi-

trary. Ways to do this include (1) ordering them alphabetically by clue, (2) ordering them alphabetically by answer, (3) shuffling them in a bag you give the solver, or (4) randomly ordering them and *explicitly* saying in the puzzle instructions that they're in random order.

As discussed in §4, it's important to make the puzzle look nice. However, something you shouldn't do in making the puzzle attractive is include extraneous elements that may interfere with solving. As discussed in §7, you should keep flavor text to a minimum since any extraneous word can be misconstrued to have unintended meaning that leads solvers on a wild goose chase. Similarly, extraneous pictures or design elements can also be distracting. It's fine to use standard design flourishes like drop shadows, but don't go overboard lest solvers get distracted. If you want to include design flourishes, make sure to use them consistently on the full set of puzzles you present the solver, so that they can deduce the flourishes' irrelevance from their ubiquity.

9 Testing

Testing is critical to producing a high-quality puzzle. Indeed, a puzzle will usually require several test cycles before it's ready. The more feedback you can incorporate from testers, the better your puzzle will be.

A corollary of this is that you should design your puzzle from the outset for changeability. If your puzzle is so elaborately constructed that changing one part necessitates changing the entirety, you won't be able to modify it easily in response to feedback. Ideally, your puzzle will be constructed from modular, separable clues so that the problematic ones can be replaced entirely if need be.

If testing reveals that your hint for an aha isn't consistently cuing solvers what to do, you need to make the hint less subtle and test again. If this problem persists, the best way out of it is to just replace the aha with an explicit instruction.

This is why it's risky to write a "pure aha" puzzle, i.e., one in which the only fun steps are figuring out ahas. If you have to give away the aha of a "pure aha" puzzle by giving an explicit instruction, then it becomes not fun any more! Since you may have to give away ahas, it's best to design your puzzle from the outset so that the bulk of the solver's time is spent on fun activities that aren't ahas.

You should perform your testing *serially*, to make use of the limited population of testers you have available. In other words, you should give the puzzle to one solver (or one solving group who'll solve together), then solicit their feedback, then update the puzzle to reflect that feedback, then start that process again with the updated puzzle. In contrast, you should *not* give the puzzle to two different solvers at the same time. To do so would be wasteful because you'd miss the opportunity to test

on the second tester the fixes you make in response to the first tester.

Take every piece of feedback seriously, and ideally address it by changing the puzzle accordingly. There's always temptation to ignore feedback that requires changing the puzzle, so make special mental effort to resist that temptation. Sometimes you'll feel you have good reason to reject someone's feedback, and you may even be right. But take care not to fool yourself into thinking you're right; recognize that you're likely just unconsciously trying to avoid work or criticism. And, if you do reject someone's feedback but a later solver points it out again, recognize at that point that it's almost certainly something you have to fix.

Ideally, you should watch testers solve your puzzle, so you can learn as much as you can about the solving experience. However, typically this isn't possible since solvers prefer to solve the puzzle whenever they happen to have time. In this case, send them a list of things to keep track of during solving so that their emailed feedback report afterward will be as useful as possible. I suggest asking them to report (1) whether they had fun, (2) suggestions they have for making the puzzle more fun, (3) which clues they found most fun and least fun, (4) which clue or clues were the "hook" that cued them to how the puzzle was supposed to be solved, and (5) how many person-hours it took them to solve. After they send you this feedback, ask follow-up questions by email or, better yet, ask to debrief them by phone to learn even more about their experience.