

# Making the Thing

<https://gerolds.github.io/textbook/textbook/posts/making-the-thing/>

# Contents

1. Why coherence
2. The sequence
3. Find your governing commitment
4. Derive your core loop from the commitment
5. Diagnostic checkpoints
  - 5.1 Is there a signature primitive?
  - 5.2 Is the mood holding?
  - 5.3 Is the promise legible?
  - 5.4 Is there friction that doesn't pay?
  - 5.5 Are stakes clear?
  - 5.6 Is cadence alive?
  - 5.7 Is there something that converts?
  - 5.8 Are there handles?
  - 5.9 Is there a hinge?
6. Protect the commitment through production
7. The shape of the journey
8. Quick reference
9. Other frameworks
10. Related essays

# Making the Thing

This essay is about a practical problem in game development: how to make decisions that add up to a coherent work. It assumes some mix of ideas, constraints, collaborators, and deadlines. The question is how to keep those pressures from turning the project into an accumulation of locally reasonable but globally inconsistent choices.

---

## 1. Why coherence

Of all the things you could optimize for, why coherence?

Because coherence is the property that makes betrayal *legible*. When a game coheres around a commitment, decisions that violate it have a readable cost. The team can discuss the tradeoff. Playtesters may feel the mismatch even if they cannot name it. You can still argue about whether the cost is worth paying, but at least the cost is visible.

Compare this to other candidates:

**Quality.** “We will make every part as good as possible.” Quality provides no direction. When you must cut, it cannot tell you what to cut. Every part is “as good as possible” until it isn’t, and then you are negotiating without a compass.

**Fun.** “We will make it fun.” Fun is an outcome, not a constraint. You cannot derive decisions from it because fun emerges from coherent systems, not from direct pursuit. Optimizing for fun alone often produces feature soup: every part locally enjoyable, nothing accumulating.

**Vision.** “We will realize the director’s vision.” Vision locates authority in a person rather than in the work. When the person is unavailable, inconsistent, or simply mistaken, there is no self-correcting mechanism. Coherence can be checked against the work itself.

**Originality.** “We will do something new.” Originality is orthogonal to whether the result holds together. Many original games are incoherent; many coherent games are not particularly original. Originality might get you noticed. Coherence is what survives contact.

**Scope.** “We will ship what we can.” Scope tells you when to stop. It does not tell you what to protect. A game cut for scope alone becomes a random sample of what was buildable, not a deliberate artifact.

Coherence works differently. A coherent game has a commitment at its center, and every part either serves that commitment or is in tension with it. When you face a hard decision, coherence gives you a question: *Does this preserve or erode the commitment?* The answer is not always obvious, but the question remains available. That is more than the alternatives provide.

Coherence is not sufficient. A coherent game can still be dull, broken, or unmarketable. But incoherence is a specific kind of failure: the parts do not add up. The player senses that the game does not know what it is. No amount of polish fixes this. Only finding the commitment and protecting it.

So coherence is not the only thing that matters. But it is a useful place to start because it makes the cost of betrayal visible.

---

## 2. The sequence

This essay gives you a sequence:

1. Find your **governing commitment**
2. Derive your **core loop** from it
3. Use **diagnostic checkpoints** to see what’s emerging
4. **Protect the commitment** through production

The sequence is not a checklist. It is a way of working that treats coherence as an outcome of repeated decisions rather than as a final polish pass.

---

### 3. Find your governing commitment

The governing commitment is the thing you are unwilling to betray. Not a feature, not a genre, not a market position. The experience you are protecting.

#### Why this must come first:

Without a commitment, there is no betrayal, and without betrayal, there is no legible cost. You can still make decisions because scope forces you to, but every cut is negotiated on local terms: “This feature is expensive.” “This asset isn’t done.” “Players didn’t like this.” None of these arguments reference what the game *is*. They reference what is convenient. Convenience produces drift; drift produces incoherence.

A commitment changes this. Once you have one, decisions have weight. Cutting a feature that serves the commitment *costs* something visible. Cutting a feature that doesn’t serve the commitment is usually relief, not loss. The commitment turns “what should we cut?” from a political question into a structural one.

#### Forms it can take:

A verb the player must feel:

“The player excavates.” “The player intrudes.” “The player tends.”

A constraint you will not violate:

“No UI that breaks the fiction.” “Every death is the player’s fault and they know it.” “The world does not pause.”

A promise about what the experience delivers:

“The worst moment will also be the most beautiful.” “Mastery feels like composure, not power.” “By the end, the player will have built something they don’t want to leave.”

A refusal:

“No quest markers.” “No tutorials.” “No safe places.”

**How to find it:**

*Ask: If I had to cut everything except one thing, what would I protect?*

*Ask: What would make me walk away from this project if I were forced to remove it?*

*Ask: When I imagine someone playing this and feeling nothing, what specific feeling are they missing?*

The commitment is not the game. The game is hundreds of decisions. The commitment is what makes those decisions point the same direction.

**Test it:**

Write the commitment down. Show it to your team, or say it aloud to yourself. If it is too vague to constrain decisions, it is not a commitment yet. “Make a great RPG” is not a commitment. “The player lives with the consequences of who they chose to trust” is.

**When finding is hard:**

Sometimes the commitment doesn't arrive easily. You have too many ideas, or your candidate is too vague to constrain, or you commit to something specific but wrong. See *Finding the Commitment* (<https://gerolds.github.io/posts/finding-the-commitment/>) for sources to look for and tests to run before committing resources.

---

## 4. Derive your core loop from the commitment

The core loop is what the player does most of the time. It needs to serve the commitment, or the commitment is mostly rhetorical.

**Why the loop must derive from the commitment:**

The loop is where the player spends their hours. If it doesn't serve the commitment, the player's actual experience, the thing they repeat, optimize, and build muscle memory around, has little to do with what the game claims to be protecting.

This is where legible cost matters most. When someone proposes a loop mechanic that is “fun but off-theme,” you can now see the cost: every hour the player spends in that loop is an hour *not* spent feeling the commitment. The betrayal is measurable in playtime. A detour loop doesn’t just fail to help; it competes for the hours where the commitment should land.

Deriving the loop from the commitment prevents this. You ask: *What repeated action would make the player feel the commitment?* The answer constrains your design space. Some loops are disqualified not because they are bad, but because they serve a different game.

### **The derivation question:**

*What repeated action would make the player feel the commitment?*

If your commitment is “the player excavates,” the loop might involve uncovering, interpreting, and placing. If your commitment is “mastery feels like composure,” the loop might involve reading threats, timing responses, and recovering poise. If your commitment is “the player tends,” the loop might involve observing needs, making small adjustments, and watching consequences unfold.

### **What the loop needs:**

- **A decision.** The player chooses something. If there is no choice, there is no loop, only animation.
- **Feedback.** The world responds in a way the player can read. They learn whether their choice was good.
- **Variation.** The same loop produces different situations. The player is not solving the same problem forever.
- **Depth.** The loop supports getting better. There is a difference between playing it adequately and playing it well.

### **Prototyping:**

Build the loop as cheaply as possible. Placeholder art, no menus, no progression. Just the action and the feedback. Play it. Does it serve the commitment? Does it feel like the verb you are protecting?

If yes, you have something. If no, try a different loop. The commitment stays; the implementation is negotiable.

### **When prototyping is hard:**

“Build cheap and test” hides craft. What goes into a loop prototype? How do you test honestly? When do you stop iterating? See *Prototyping the Loop* (<https://gerolds.github.io/posts/prototyping-the-loop/>) for the method.

---

## **5. Diagnostic checkpoints**

Once you have a loop that serves the commitment, you start building. As you build, use these questions to see what is emerging. They are lenses, not mandates.

### **Why checkpoints, not requirements:**

The commitment tells you what to protect; it does not tell you what to build. Two games with the same commitment might have completely different features, aesthetics, and structures. The checkpoints help you see *whether what you’re building is serving the commitment*, without prescribing *what* to build.

Each checkpoint names a property that, if missing, creates a specific cost. The cost is legible: you can point to it, argue about it, decide whether to pay it. Some costs are worth paying. The point is to pay them knowingly rather than by accident.

### **5.1. Is there a signature primitive?**

The signature primitive is the smallest unit of play that carries the game’s identity, the thing people will mime when describing the game.

*Why it matters:* Without a signature primitive, the game has no center of gravity. Players cannot point to what makes it *this* game rather than another. The cost is **diffusion**: a collection of systems, not a thing with identity. When you must cut, you have no tiebreaker; every system feels equally central because none is.

*Check:* Can you describe what the player does in one sentence without referencing genre or story? Is that action deep enough to support variation, mastery, and expression?

*If missing:* Your loop may be too diffuse. Look for the moment inside the loop where the player makes the most meaningful choice. That moment is a candidate. Build around it.

## 5.2. Is the mood holding?

Mood is the aesthetic-cognitive climate, what makes the game a *place* in memory.

*Why it matters:* Mood is coherence made sensory. When mood breaks, the player is reminded they are using software. The cost is **presence**: if the player cannot stay in the fiction, emotional stakes weaken. A game that cannot hold mood will struggle to hold tension, melancholy, wonder, or dread.

*Check:* When you playtest, do players slow down? Do they look around? Do they comment on how it *feels*? Or do they sprint, skip, mute, optimize?

*If weak:* Look for contradictions. Is the UI breaking the fiction? Is the pacing fighting the tone? Are reward popups interrupting emotional states? Mood is killed by inconsistency.

## 5.3. Is the promise legible?

Legible promise means the player understands the contract in the first hour.

*Why it matters:* If the player doesn't know what they're signing up for, they cannot evaluate whether the game is delivering. The cost is **trust**: players who expected one thing and got another may feel misled, even if what they got is good. Worse, *you* lose signal. You cannot tell if players are bouncing because the game is bad or because the promise was unclear.

*Check:* After the first session, can a player describe what the game is? Do they know what they are working toward? Do they know what "getting better" looks like?

*If weak:* The opening may be teaching the wrong thing, or nothing. Redesign the first hour to demonstrate the mature loop, not a tutorial parody.

#### 5.4. **Is there friction that doesn't pay?**

Friction is acceptable if it purchases meaning. Otherwise it wastes time.

*Why it matters:* Friction that serves the commitment is load-bearing; friction that doesn't is parasitic. The cost is **patience**: every unnecessary click, every unskippable animation, every obtuse menu encourages the player to disengage. Eventually they may stop trusting that your decisions are intentional. Once that trust is gone, even intentional friction can read as incompetence.

*Check:* Are players alt-tabbing to guides? Complaining about "quality of life"? Developing workarounds and exploits? These are signs of unintended friction.

*If high:* Audit repeated actions. Each one should teach, test, or express. If it does none of those, cut it, automate it, or compress it.

#### 5.5. **Are stakes clear?**

Stakes clarity means the player knows what matters right now.

*Why it matters:* Stakes are how the commitment becomes felt. If the player doesn't know what they're risking, they cannot care about outcomes. The cost is **investment**: unclear stakes produce cautious, disengaged play. The player hedges instead of committing, save-scums instead of living with consequences. The emotional effect weakens.

*Check:* Can you watch someone play and tell what they are trying to do? Can they tell you what they are risking? If the player is confused about what "good" looks like, stakes are unclear.

*If muddy:* Reduce currencies. Make consequences visible. Ensure the type of consequence is predictable even if the magnitude is not.

## 5.6. Is cadence alive?

Cadence is the alternation of novelty, consolidation, payoff, and rest.

*Why it matters:* Cadence is how the commitment unfolds over time. A commitment that arrives too early has no build; one that arrives too late tests patience past breaking. The cost is **stamina**: if the middle dies, players leave before the commitment pays off. They remember the game as “promising but boring.” The coherent structure existed, but they never saw it.

*Check:* Is the middle dying? Are players enthusiastic early and bored later? Does the game feel like work?

*If failing:* You may be frontloading novelty and then coasting. Introduce wrinkles at a measured rate. Respect fatigue. Cut padding that looks like content but feels like delay.

## 5.7. Is there something that converts?

Conversion is the durable change the game produces in the player.

*Why it matters:* Conversion is one test of whether the commitment landed. A player who finishes unchanged may still have been entertained, but the work will usually leave a shallower mark. The cost is **memorability**: games that do not convert are often consumed and forgotten. If you care whether your work outlasts the play session, you care about conversion.

*Check:* After playing, do testers describe a change in themselves? “I got better at reading situations.” “I didn’t know a game could make me feel that.” “I keep thinking about it.” Or do they just say “it was fun”?

*If weak:* Consider what kind of conversion you are aiming for:

- **Mastery:** The player becomes capable. Requires fair challenge and instructive failure.
- **Rupture:** The player is surprised by what the medium can do. Requires restraint before the hinge.
- **Mythic:** The player lives an archetype. Requires symbolic weight and negative space.

- **Identity:** The game becomes part of how the player describes themselves. Requires expressive choice and social recognition.
- **Habitat:** The game becomes a place to return to. Requires mood, low friction, and returnability without punishment.

You cannot manufacture conversion. You can create conditions where it is possible.

## 5.8. Are there handles?

Handles are the names, places, objects, and symbols that let players talk about what they felt.

*Why it matters:* Handles are how coherence becomes communicable. A game without handles cannot be discussed, recommended, or remembered in detail. The cost is **transmission**: players who loved the game cannot explain why. They gesture vaguely. The experience stays private and dies with the play session. Handles let the commitment propagate.

*Check:* When players describe the game, do they use specific names? “The moment in Midgar.” “The bonfire.” “GLaDOS.” Or do they speak in generalities?

*If missing:* You may have systems without landmarks. Give things names. Make places distinct. Let objects have histories. Handles emerge from specificity, not lore volume.

## 5.9. Is there a hinge?

A hinge is a recontextualization moment that changes how earlier play is interpreted.

*Why it matters:* A hinge is coherence revealed. The player thought they understood, then discovers a deeper structure. The cost of a missing hinge is **ceiling**: the game is exactly what it appears to be, no more. This isn’t always a problem, and many great games have no hinge, but if you are reaching for transformation, a hinge is one way the commitment can reveal a deeper structure.

*Check:* Is there a moment where the player’s understanding shifts? Do they play differently afterward, not just know a fact?

*If missing:* Not every game needs a hinge. But if you are aiming for one, plant seeds early, let the player stabilize into a model, then disrupt it meaningfully. The disruption should change behavior, not just add lore.

---

**Checkpoint summary:** Each of these nine properties, if absent, extracts a specific cost from the game. The costs are all legible, all arguable. Some you will pay deliberately. The checkpoints ensure you see the invoice before it's due.

---

## 6. Protect the commitment through production

Production will put pressure on your commitment. Scope, schedule, feedback, fatigue, and a hundred reasonable compromises will push you toward diffusion. This is normal.

### **Why protection requires the commitment:**

Without a commitment, there is nothing to protect. You just ship what you can. With a commitment, every compromise has a legible cost. The question shifts from “can we afford this?” to “can we afford this *and keep the commitment intact?*” Sometimes yes, sometimes no. But the question is askable, which means the answer is defensible. This is the only way to make hard decisions without politics consuming everything.

### **When you must cut:**

*Ask: Does this cut preserve the commitment or erode it? Cut ornament before you cut load-bearing structure. A smaller game that coheres beats a larger game that doesn't.*

The commitment makes this question answerable. Without it, “load-bearing” means “whoever argues loudest.” With it, you can trace the structure: this feature serves the commitment; that one is nice but optional. Cut the optional thing. If you must cut something load-bearing, know you are paying a cost and decide consciously whether it's worth it.

### **When the commitment seems wrong:**

Sometimes the game teaches you that your commitment was not quite right. The verb you were protecting is not the verb the game wants. This is discovery, not failure. Update the commitment. But do not abandon the *practice* of commitment. Shift to a better one and protect that.

The danger is using “the commitment evolved” as cover for drift. Real evolution is relatively rare and often feels like revelation: *oh, this is what the game was always trying to be*. Drift is common and often feels like relief: *we don't have to do the hard thing anymore*. Learn to tell the difference.

### **When the team disagrees:**

Different people will interpret the commitment differently. This is the work. Alignment is not assumed; it is maintained through repeated conversations about what the commitment means *in this specific decision*. The commitment is a negotiation, not a decree.

But the negotiation has a ground truth: the work itself. When two interpretations conflict, build both cheaply, playtest, and ask which one serves the commitment better. The commitment turns aesthetic disagreements into empirical questions. Not always, but often enough to matter.

### **When feedback confuses:**

Players will love things you thought were filler. They will ignore things you thought were the point. Ask: *Does this feedback serve the commitment or dissolve it?* Not all feedback is signal. Some is players wanting a different game than the one you are making.

The commitment lets you filter. Feedback that says “the commitment isn't landing” is actionable; you're failing at your own goal. Feedback that says “I wish you had a different commitment” is informative but not obligating. You can note it, consider it, and decline it. Without a commitment, all feedback feels equally valid, and you end up chasing an average that satisfies no one.

---

## 7. The shape of the journey

You protect a commitment. You derive a loop that serves it. You build, and as you build, you check what is emerging. You adjust when you learn. You protect the commitment through production. You ship something that is not the thing you first imagined but is, with luck, the thing the commitment was capable of becoming.

The mess is real. The difficulty is real. No framework eliminates them.

But this framework does one thing: it makes the cost of your decisions visible. When you betray the commitment, you can see the betrayal. When you protect it, you can see what you protected. Most projects weaken through a fog of reasonable compromises, each one hard to evaluate in isolation. Legible cost is one way to navigate that fog.

The path exists. Other people have walked versions of it. Your path will differ in the specifics, but the shape is familiar.

Then make the thing.

---

## 8. Quick reference

### **Sequence:**

1. Find your governing commitment (what you won't betray)
2. Derive your core loop (what repeated action serves the commitment)
3. Use diagnostic checkpoints (what is emerging)
4. Protect the commitment through production (cut coherently)

### **Diagnostic checkpoints:**

| Question                            | What you're checking                         |
|-------------------------------------|--|
| Is there a signature primitive?     | One irreducible action that carries identity |
| Is the mood holding?                | Presence is rewarded; players slow down      |
| Is the promise legible?             | First hour teaches the real contract         |
| Is there friction that doesn't pay? | Every repeated pain purchases meaning        |
| Are stakes clear?                   | Player knows what matters right now          |
| Is cadence alive?                   | Middle stays alive; novelty is paced         |
| Is there something that converts?   | Player changes durably                       |
| Are there handles?                  | Specific names players use to talk about it  |
| Is there a hinge?                   | A moment that reweights earlier play         |

## 9. Other frameworks

This essay is not the only way to think about game design. A few alternatives worth knowing:

**Pillars.** Many studios define 3-5 “pillars” that a game must support. Pillars are broader than a single commitment and can coexist in tension. The risk is diffusion: if everything serves multiple pillars, nothing has priority when you must cut.

**Lenses.** Jesse Schell's *The Art of Game Design* proposes 100+ lenses, each a question to ask about a design. Lenses are diagnostic rather than generative; they help you see what you have, not decide what to build. This essay's checkpoints are closer to lenses than to pillars.

**MDA.** The Mechanics-Dynamics-Aesthetics framework separates what the designer builds (mechanics), what emerges in play (dynamics), and what the player feels (aesthetics). MDA is useful for analysis but doesn't prescribe a workflow.

**Player experience goals.** Some teams define the target experience first, then derive everything else. This is close to “commitment” but often framed as a feeling rather than a constraint.

None of these are wrong. They emphasize different things. This essay bets on *coherence through a single commitment* because coherence makes the cost of betrayal visible. If another framework helps you see costs clearly, use it.

---

## 10. Related essays

- *Finding the Commitment* (<https://gerolds.github.io/posts/finding-the-commitment/>): the prior question—how to discover and validate a commitment.
- *Prototyping the Loop* (<https://gerolds.github.io/posts/prototyping-the-loop/>): testing whether a loop serves the commitment.
- *The Elements of a Good and Widely Appealing Game* (<https://gerolds.github.io/posts/wide-appeal/>): the analytical framework these checkpoints derive from.
- *When the Funnel Eats the Work* (<https://gerolds.github.io/posts/when-the-funnel-eats-the-work/>): cultural selection pressures.

---

*Drafting assistance: Claude Opus. All claims mine; errors my responsibility.*