SAMPLE

Building Tomorrow's Game Developers

The Loop's Keeper in the Machine Era

Aiman JoeHanz

Free sample: Introduction & Chapter 1

putrabooks.github.io

Buy the full book:

Carousell carousell.sg/p/building-tomorrow-s-game-developers-ebook-pdf-1471386053

Ko-fi

Payhip

Contents

Introduction

Chapter 1 — The Loop

Chapter 2 — The Territories

Chapter 3 — The Way In

Chapter 4 — The Working Day

Chapter 5 — The Loop

Chapter 6 — The Crunch

Chapter 7 — The Conversation

Chapter 8 — The Instruments

Chapter 9 — The Year Whole

Chapter 10 — The Long Path

Introduction

A game is finished not when somebody runs out of ideas but when somebody decides, on evidence, to stop changing it.

This book is about the work behind that decision: what game developers actually hold on to, how the roles differ, what the career path rewards, and what changes when the tools change. It is written for people considering the trade, for people already inside it and wondering whether they have understood it correctly, and for the people who commission games without having made one.

The claim is narrow and arguable. Game development is not the accumulation of features and it is not the execution of a vision. It is the operation of a loop: something is built, somebody plays it, something is observed, one thing is changed, and the cycle repeats until the observed behaviour matches the intended experience. Everything else in the trade, the engines, the art, the audio, the marketing, the patch notes, exists to serve that cycle or to survive its absence.

Two folklores stand in the way. The first is the auteur, for whom the game is a personal vision and the players are an audience to be brought to it. The second is the spreadsheet, for whom the numbers are the truth: retention, conversion, session length, and any experience that does not move them is a cost. Both remove the thing the loop requires, which is attention to what actually happened when somebody played. The auteur stops watching, and the spreadsheet stops seeing anything that was not instrumented.

The chapters follow the collection's order. The keeper comes first, then the territories the trade divides into, then the routes in, the working day, the craft of the loop itself, the strain it puts on people, the conversations it requires, the instruments, a composite year, and the long arc.

One warning belongs in the introduction rather than the conclusion. This trade has a documented habit of consuming the people who are most committed to it. The chapter on strain is not a digression from the craft; it is part of an accurate description, because the pressure that produces long hours comes from the same place as the pressure that produces good games, which is that the work is made of judgement applied to things people care about.

Nothing here assumes a platform, an engine, or a studio size. The mechanisms described hold for a two-person team and for a team of six hundred, and where they diverge the text says so.

The machine era arrives as a change in the instruments. Tools now generate assets, draft code, and produce variations at a rate nobody anticipated. What they do not do is play the game and notice that it is not fun yet, or notice that the third level teaches the wrong thing, or decide that the feature everyone has spent a month on should be cut. Those remain the work, and the loop is the arrangement by which they are done.

Chapter 1 — The Loop

What the Loop Is

The loop is the smallest complete unit of game development, and it has four parts that must all be present for it to work.

Something is built, in a form crude enough to be produced quickly. Somebody plays it, ideally somebody who did not build it and who is not being polite. Something is observed, which is a narrower activity than listening to an opinion. One thing is changed, and then the whole cycle begins again with the new build.

The fourth element is the one that distinguishes the loop from ordinary iteration, and it is the deliberate limit of one change at a time. When three things change between two builds and the game feels worse, nothing has been learned. When one thing changes, the observation attaches to a cause. The trade's practitioners know this, violate it constantly under deadline pressure, and pay for the violation in confusion.

The loop is not a methodology to be adopted at a studio level. It is a discipline of the individual and small group: the designer who runs a build every afternoon rather than every fortnight, the programmer who plays their own feature before handing it over, the artist who looks at the asset in position rather than on its own. Studios that have elaborate processes and no fast loop make poor games slowly. Studios with almost no process and a fast loop occasionally make good games.

The Two Folklores

Every trade has stories that make its work legible from outside and misleading from inside, and this one has two that are unusually well known to players.

The first is the auteur. The story says that games are made by a visionary whose intent is realised by a team, that the vision is the valuable thing, and that the audience either receives it or does not. It is attractive because it is a good story, because it is occasionally true, and because it flatters the person with the idea. It fails because a vision that has not been tested against players is a guess, and games are complex enough that even experienced designers are wrong about their own work most of the time.

The second is the spreadsheet. The story says that games are products optimised against numbers: retention, conversion, session length, daily activity. It is attractive because the numbers are real, because they are legible

to people who fund games, and because a studio that ignores them can run out of money. It fails because the numbers describe what happened and not why, and because a team optimising a metric will eventually produce something that moves the metric and has no other reason to exist.

Both folklores share a structure: each replaces observation with a source of authority, one inside the author and one inside the dashboard. The loop is what remains when neither authority is trusted on its own.

What Is Actually Held

Ask a practitioner what they are responsible for and the answer will vary by role, and underneath the variation is a single thing being held.

It is the correspondence between what the player experiences and what the game was intended to produce. A designer holds it for a system, an artist for a moment, a programmer for a feel, an audio designer for a mood, a producer for the whole. None of them holds the game. Each holds a claim about what somebody will feel, and the loop is the only mechanism by which that claim can be checked while there is still time to change it.

This is why the trade's central artefact is not the design document. It is the build: a thing that can be played. A design document describes an intention and cannot be wrong in an interesting way. A build can be played, and playing it produces the specific disappointment that tells you what the intention actually was.

Practitioners who have understood this stop treating documentation as the design and start treating it as the record of what has been learned. The document that matters is the one that says what was tried, what happened, and what was concluded, which is a different document from the one that says what the game will be.

The Three Commitments

This book's argument rests on three commitments, offered as a description of good practice rather than a reform.

The first is to prototype the feeling before building the system. The experience that makes a game worth playing can usually be approximated in a week with crude assets and placeholder sound. If the approximation is not interesting, the system built around it will not be either, and the discovery is much cheaper at that stage than after six months of production.

The second is to tune by playing rather than by arguing. A design question that two people disagree about is almost always answerable by a build and twenty minutes, and the answer frequently surprises both. The trade's most expensive disagreements are conducted in documents and meetings about things that could have been tried.

The third is to ship the loop that survives its players. A game is not finished when it is released. It is played for a hundred hours by somebody who finds the seams, a thousand hours by somebody who optimises the fun out of it, and in a live game it is played for years by people who will discover every mistake. Designing for the first hour and hoping is the trade's most common failure at scale.

The Speed of the Loop

The loop's value is not that it exists but that it is fast, and the speed is set by the slowest step in the cycle.

For most teams the slowest step is not building. It is deciding what to build: the feature that is specified in a document, reviewed, estimated, scheduled, and implemented a month after the idea, by which time the idea has been described so many times that it has acquired a certainty it does not deserve. A team in this condition is running a loop measured in months, which means it can make perhaps eight decisions before the project ends.

The fastest teams run the same cycle in hours. Somebody has an idea, somebody builds a crude version, four people play it after lunch, and the idea is confirmed or dead by the end of the day. That team makes two hundred decisions in the same period, and the difference is not effort. It is the arrangement.

What makes a loop fast is three things: a build that can be produced without ceremony, an audience that is available without scheduling, and a decision-maker who is present. Teams that have all three frequently describe their process as chaotic and produce coherent games, because the coherence comes from the loop rather than from the plan.

The Slow Loop and Its Symptoms

A team running a slow loop exhibits a set of symptoms that are usually diagnosed as other problems.

The first is confidence about things that have not been tested. A team that has

not played the build in three weeks will describe the game's feel in the language of the design document, with adjectives nobody has verified. This is not dishonesty; it is the natural result of working from intention.

The second is disagreement that cannot be resolved. Arguments about whether a mechanic is fun, conducted by people who have not seen it, cannot be settled and so they recur. Teams in this condition describe themselves as having a design problem when they have an observation problem.

The third is the surprise at the end. A team that played the game monthly for two years knows roughly what it has. A team that played it quarterly discovers in the final month that a third of the game is not working, and the discovery is always described as sudden.

The fourth, and the most expensive, is that the cuts are made at the worst moment. Cutting in month six is a decision; cutting in month sixteen is a crisis, and the difference between the two is entirely a function of when the team found out.

Why the Loop Is Hard to Institutionalise

Every studio believes it runs a loop, and the practice is rarer than the belief for reasons that are structural rather than cultural.

The first reason is that a fast loop looks like indiscipline. Four people playing a build at eleven in the morning is not obviously work, and a manager under pressure to demonstrate progress will schedule something instead. The cost of the loop is immediate and visible; the benefit is deferred and invisible.

The second is that a playtest is a small humiliation. Somebody plays the thing you made and is bored, and the correct response is to treat that as information rather than as a verdict, which requires a composure that not everybody has. Teams where a bad session produces blame stop reporting what happened, and the loop starts producing flattering observations.

The third is that the loop requires somebody with the authority to decide. A team that plays, observes, and then takes the finding to a committee has the observation and not the decision, and the loop stalls at the point where it matters. The authority has to be in the room, and in a large studio it frequently is not.

The Loop Across a Team

The loop scales by being repeated at different levels rather than by being

centralised, and the levels must be consistent with each other.

At the individual level, a designer loops on a mechanic: build, play, adjust. At the feature level, a small group loops on whether a system belongs: prototype, session, verdict. At the game level, the whole team loops on what the game is: a milestone build, a playtest with outsiders, a decision about scope.

The failure is when the levels disagree. A designer tunes a mechanic to feel good in isolation, and the feature group discovers it does not fit the game. A feature group builds a system that works, and the game-level playtest shows it makes the game worse. Each level did its job and the levels were not connected, which is why the studio's most senior designers spend their time in the middle level rather than at either extreme.

The connecting mechanism is the same build. If every level is played from the same build, the disagreements surface early and become information. If the levels are played from different builds, each is coherent internally and the incoherence is only discovered at integration, which is the most expensive place to find it.

The Observation That Is Not an Opinion

The loop's second step is where most teams substitute something easier, and the substitution is the reason their iteration does not converge.

An opinion is a reaction: it was fun, it was confusing, it felt slow. It is the cheapest thing to collect and the least useful, because it describes the person rather than the game. An observation is a description of what happened: on the third attempt the player stopped reading the prompt and walked into the wall.

Opinions are easy to solicit and impossible to act on. Observations are harder to collect, because they require somebody to watch and record, and they are the only material from which a change can be derived. A team that collects opinions discovers that its playtests produce disagreements rather than findings.

The second distinction is between what the player did and what the player said they did. People are poor reporters of their own behaviour, and a player's account of why they stopped is usually a rationalisation of the moment they got bored. The recording of the session, watched afterwards, is the primary source; the conversation is a supplement.

This is why the trade's best practitioners insist on silence during sessions and on notes taken by a person who is not playing. The arrangement is uncomfortable and it produces findings that nobody could have argued their way to.

The One Change

The fourth step of the loop is the disciplined one, and the discipline is entirely in the number.

Changing one thing per build is obvious in principle and violated constantly in practice, because a team under pressure changes everything it can think of and then finds that the game is better without knowing why. The improvement is real and it is unusable: nobody knows which change produced it, and the next regression cannot be attributed.

The discipline has a second function that is less obvious. A single change forces the team to decide what the most important problem is, and the decision is where design judgement is exercised. A build with six changes is a build that avoided a decision; a build with one is a build that made one, and the making is the skill.

The third function is psychological. A team that changes one thing and plays it sees a small, attributable result, which is motivating. A team that changes six things and plays it sees a different game and cannot tell whether to continue, which is discouraging, and the discouragement is the beginning of the end of the practice.

When the Loop Stops Working

There is a condition in which the loop produces no more learning, and it is worth recognising because the usual response makes it worse.

The condition is that the team can no longer see the game. They have played it four hundred times, they know where everything is, and their observations are the observations of experts. Changes that a new player would find decisive become invisible, and the team begins tuning details that nobody else will notice.

The symptom is that sessions produce agreement and no findings. When six people play and nobody is surprised, the loop has stopped being an instrument and become a ritual.

The remedy is outside input, and its scarcity is why studios maintain playtest

pools. A person who has never played produces forty observations in twenty minutes, most of them wrong and several of them decisive. The team's job is to distinguish, and the ability to distinguish is what four hundred sessions actually bought.

The Loop and the Machine

A tool that produces content very quickly does not produce a loop, and the distinction is the whole argument of this book applied to its newest instruments.

Generation compresses the first step. A level that took a week can be proposed in an hour, four variations in a morning. What generation does not compress is the second and third steps: somebody must play it, somebody must watch, somebody must decide. Those are bounded by human attention, which has not accelerated.

The arithmetic that follows is uncomfortable for teams that have not noticed it. If production becomes four times faster and observation stays constant, then three quarters of the new content can never be examined. A studio in that position is not making more game; it is making more unexamined game, and the trade already knows what unexamined content looks like at smaller scales.

The response is not to refuse the tools. It is to spend the gained time on the step that became the bottleneck: more sessions, more outside players, more comparison between what was proposed and what was experienced. The studios that get this right will look slower than their competitors in the first year and will ship games that are coherent, which is the only advantage that compounds.

ABOUT THE AUTHOR

In the Building Tomorrow's People series, Aiman JoeHanz treats each trade's core object as the thing it refuses to lose, and for the game developer that object is the loop: the thirty seconds a player repeats a thousand times. This volume follows the prototype, the playtest that misleads, the monetisation curve, the store review, the live operation that never ends, and the generative tools that can now produce a level faster than anyone can decide whether it is fun.