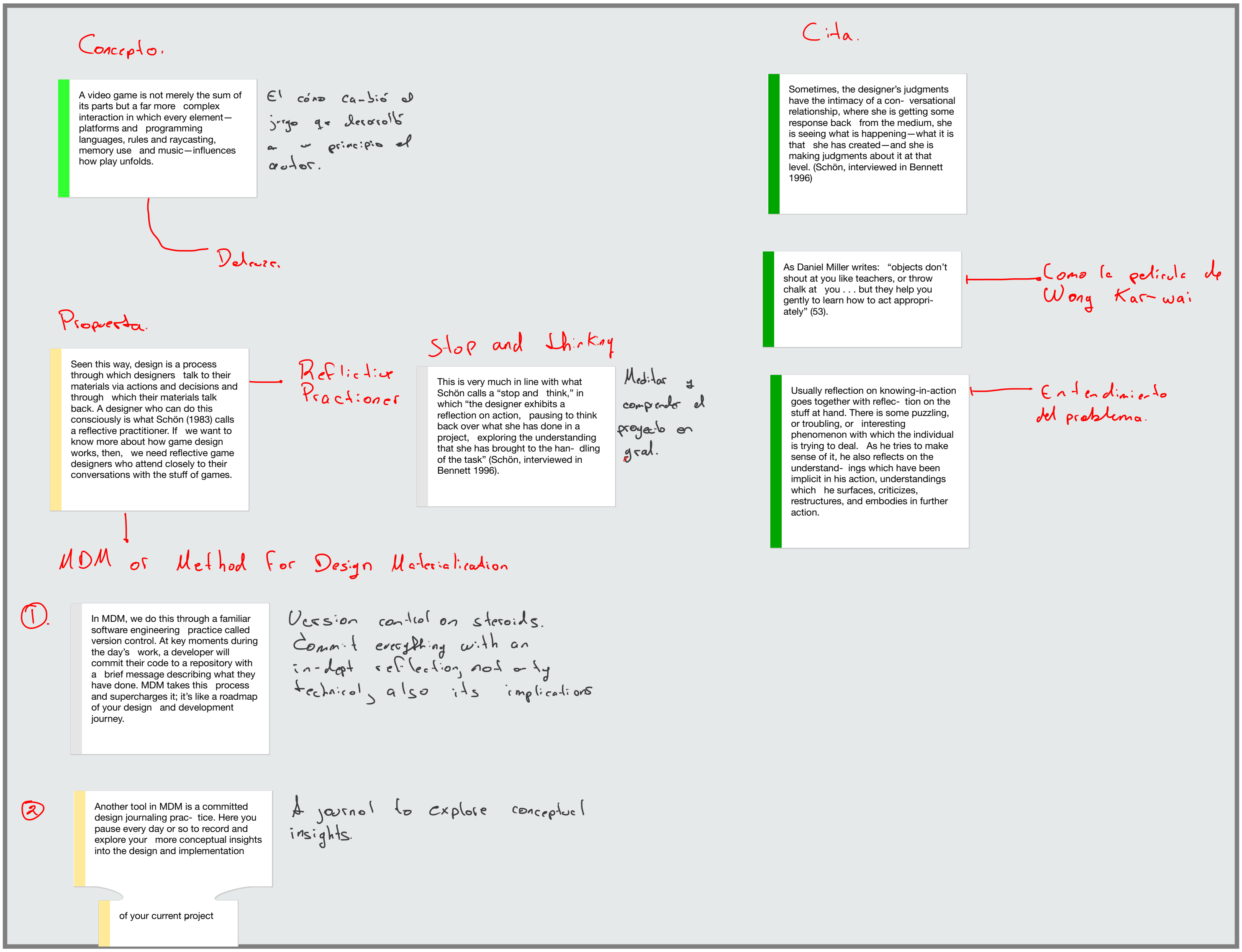
Workspace 'Workspace 1' in 'The Stuff Games Are Made Of'

Page 1 (row 1, column 1)



Concepto.

A video game is not merely the sum of its parts but a far more complex interaction in which every element—platforms and programming languages, rules and raycasting, memory use and music—influences how play unfolds.

El cómo cambió el juego que desarrolló en un principio el autor.

Deleuze.

Propuesta.

Seen this way, design is a process through which designers talk to their materials via actions and decisions and through which their materials talk back. A designer who can do this consciously is what Schön (1983) calls a reflective practitioner. If we want to know more about how game design works, then, we need reflective game designers who attend closely to their conversations with the stuff of games.

Reflective Practioner

Stop and thinking

This is very much in line with what Schön calls a "stop and think," in which "the designer exhibits a reflection on action, pausing to think back over what she has done in a project, exploring the understanding that she has brought to the handling of the task" (Schön, interviewed in Bennett 1996).

Meditar y comprender el proyecto en gral.

MDM or Method for Design Materialization

①

In MDM, we do this through a familiar software engineering practice called version control. At key moments during the day's work, a developer will commit their code to a repository with a brief message describing what they have done. MDM takes this process and supercharges it; it's like a roadmap of your design and development journey.

Version control on steroids. Commit everything with an in-dept reflection, not only technical, also its implications

②

Another tool in MDM is a committed design journaling practice. Here you pause every day or so to record and explore your more conceptual insights into the design and implementation

of your current project

A journal to explore conceptual insights.

Cita.

Sometimes, the designer's judgments have the intimacy of a conversational relationship, where she is getting some response back from the medium, she is seeing what is happening—what it is that she has created—and she is making judgments about it at that level. (Schön, interviewed in Bennett 1996)

As Daniel Miller writes: "objects don't shout at you like teachers, or throw chalk at you . . . but they help you gently to learn how to act appropriately" (53).

Como la película de Wong Kar-wai

Usually reflection on knowing-in-action goes together with reflection on the stuff at hand. There is some puzzling, or troubling, or interesting phenomenon with which the individual is trying to deal. As he tries to make sense of it, he also reflects on the understandings which have been implicit in his action, understandings which he surfaces, criticizes, restructures, and embodies in further action.

Entendimiento del problema.

1 INTRODUCTION: WHAT IS ALL THIS STUFF?

Imagine a game made purely out of points.

You'd play the game on the scoreboard itself, clicking a plus button next to your score to increase it by a single point each time. You could push down other players in the rankings by clicking a minus button next to their entry. Just a wall of names and scores vying to be number one. The points would be all there was.

Como los Clickers.

Well, I made that game back in 2014. I was interested in the idea of a game literally *about* points rather than other accomplishments like collecting coins, scoring goals, or shooting faces. I called my game *Leaderboarder* (Barr 2014), put it up as a website, told my meager Twitter following about it, and started to play, clicking my plus button as fast as I could.

At first everyone played as I'd envisaged, increasing their own score a click at a time and sometimes decreasing someone else's. Soon, though, things changed. Things got weird. As tens and then hundreds of players joined the fray, the competition became fierce. Fierce and automated. Some players, not content with the finger's pace of their scoring, applied *auto* *clickers*—software that can simulate thousands of mouse clicks per second—to the problem. I watched as scores raced up the rankings like performance sports cars.

Then the auto clickers were old news. *Leaderboarder* was written in JavaScript and JavaScript is not a particularly secure programming language. Any player could view the source code quite easily in their web browser, and that is exactly what some of them did. They looked through the program, understood how the scoring system worked, and then ran their own code in the game. Now they could set their score to arbitrary values and go from last to first in an instant. Some players began to covet the lowest score. Some started giving themselves fractional scores. One player named themselves "Nanananananananananana" and gave themselves the score of "BATMAN."

The endgame came when a few enterprising players wrote code that monitored the leaderboard for changes in real time and automatically adjusted their score accordingly. This meant a player could write a simple bot that would increase their score if any other player surpassed it, ensuring they'd stay on top. More than one player had this idea and there were epic battles between automated scoring bots, each flipping in and out of the lead at an outrageous rate. It was something to behold. In a note to myself, I wrote: "O brave new world!"

Leaderboarder ran deeper than I had initially understood. While I had intended to create a game about points, it turned out to also be about web browsers, about JavaScript, about players in conversation with the very stuff games are made of. A video game is not merely the sum of its parts but a far more complex interaction in which every element—platforms and programming languages, rules and raycasting, memory use and music—influences how play unfolds. Each component part is worthy of contemplation, and that's my guiding principle in this book. *What can we learn when we really pay attention to the stuff of games?*

I mostly chose the title *The Stuff Games Are Made Of* because I like the way it sounds. The term "stuff" is rather hard to pin down, and I don't plan to try. Anthropologist Daniel Miller (2010, 1), who studies diverse cultural uses of everyday objects and who literally wrote a book called *Stuff*, takes the same line. He opens his discussion of the idea with: "don't, just don't, ask for or expect a clear definition of 'stuff'." And yet, as Miller's work emphasizes again and again in studies of key artifacts from diverse countries, it's the stuff all around us—from saris to cellphones—that shapes how we think and live. In India, a sari is foremost an item of clothing, but it can do so much more. It can signal social status or ceremony, can help carry a hot pot or act as a pacifier for a crying baby, can serve as an improvised purse, and the list goes on. All these complex functions require "a continued engagement and conversation with its wearer, and a constant pressure to respond to changes in one's surrounding social environment" (30–31). And as for cellphones, in the Philippines they're the very stuff of life: "a relationship is not really a relationship without between 20 and 120 texts a day." Crucially, Miller points out, this cellphone-based love metric "isn't an inherent capacity [of the cellphone], but then it's not really a Filipino tradition either. It's something new" (112).

About the stuffs.
&
Antropological.
view

In conversation with our stuff, Miller concludes, we invent new habits, traditions, and practices. It's how we have new ideas and figure out new relations and ways of doing things. Video games are made of their own kinds of stuff, and through that stuff we can learn to understand them better. This is exactly what the players of *Leaderboarder* showed us. There was indeed a Miller-like engagement and conversation going on there that was neither inherent to the game nor the player but rather emerged out of their interaction. There is already a strong tradition of this kind of thinking about media, digital media more specifically. It ranges from the foundational ideas in Marshall McLuhan's (1994) *Understanding Media*, to Edward Tufte's (2004) hyper-specific *The Cognitive Style of PowerPoint*, to Lev Manovich's (2013) broader examination of various forms of media production software in *Software Takes Command*. However, to really get to know the stuff games are made of, we'll need to listen closely to them in particular.

I won't be theorizing video games as media here, though. While I've spent some time playing with theory in my academic career (focusing on triangles for some reason), in this book I'm looking at the stuff games are made of primarily from the perspective of a practitioner—I'm an experimental video game designer and developer. In some sense, I always have been. I grew up with HyperCard in the late 1980s, an exciting tool that allowed you to create animations and link different scenes with buttons. Then I moved on to Macromedia's Director before getting a degree in computer science that let me write my own programs from scratch. Throughout, I've always had an eye for how media, code, and more could combine into playful experiences, whether while attempting to recreate Hamlet in Flash (may it rest in peace) or while getting my computer to sing "Wouldn't It Be Nice" by the Beach Boys. I've stayed on that path—as I write this, I'm making a game about 3D models featured in video games, such as seagulls, supermarkets, and zombies. I'm arranging them into a virtual showcase and inviting players to join me in contemplating them. More on that later.

I'm not alone in my preoccupation with the stuff of games. From in-depth takes on retro gaming hardware (Custodio 2020; Montfort and Bogost 2009), to understanding games via art history (Flanagan 2009; Sharp 2015), to analyses of how video game tools influence game design (Nicoll and Keogh 2019b; Werning 2021), there's a lot of deep thinking out there. In many ways, though, I feel most kinship with what Bonnie Ruberg (2020) has called the "queer games avant-garde" in their interview-based book of the same name. In it, Robert Yang talks lovingly and thoughtfully about a urinal he created for his game *The Tearoom* and Andi McClure speaks of her collaboration with machines and algorithms. Amen.

When I'm designing and developing a video game, I don't follow a straightforward path from an idea ("What if I turned a famous performance art piece into a game?") to a finished work (*The Artist Is Present*). Rather, most of the design work takes place in conversation with stuff. A game programming framework like Phaser trumpets the virtues of making a game based on physics. User interface conventions admonish me to improve the usability of my menu system design. The Unity game engine whispers seductively about the beauty of 3D environments. When you're listening, the stuff games are made of is not quiet, not transparent, and not just there to be used. It's a close companion, offering up its own ideas about game design all along the way.

That experience of deeply engaging with the tools and media I use to make my games is what design theorist Donald Schön (1992) calls a "conversation with materials." I think this gives me permission to use the terms *stuff* and *material* interchangeably for the rest of the book. In Schön's work, the stuff is often that of architecture (steel girders, concrete beams,

Donald Schön → stuff → tools.

physical sites where construction will take place) but it might just as easily be programming languages, 3D models, and game mechanics, because Schön, more than anyone, captures for me how it feels to design and work with the stuff of games. Schön's underlying model of design acknowledges the close ties between designer and materials: Relación entre el diseño y materiales.

> Sometimes, the designer's judgments have the intimacy of a conversational relationship, where she is getting some response back from the medium, she is seeing what is happening—what it is that she has created—and she is making judgments about it at that level. (Schön, interviewed in Bennett 1996)

Propuesta

Seen this way, design is a process through which designers *talk* to their materials via actions and decisions and through which their materials *talk back*. A designer who can do this consciously is what Schön (1983) calls a *reflective practitioner*. If we want to know more about how game design works, then, we need *reflective game designers* who attend closely to their conversations with the stuff of games. As Daniel Miller writes: "objects don't shout at you like teachers, or throw chalk at you . . . but they help you gently to learn how to act appropriately" (53). If the stuff isn't going to shout, we'll have to lean in. To make the results accessible, we'll need ways to explicitly document such design work. It has long been a goal of mine to present a completed video game together with the design history of how it came to be. I first tried documenting this history in anxious blog entries: "Visions of clipping errors and a million unforeseen bugs crowd my peripheral vision along with the faceless horrors of maybe having forgetting [sic] to uncomment some bit of code" (Barr 2011a). Later, I turned to writing lengthy design reflections in private Evernote journals. These days, the process has become much more sophisticated, and it even has a name. I work with what my colleagues and I call

MDM, or the Method for Design Materialization (Khaled, Lessard, and Barr 2018; Khaled and Barr 2023). We have set out to explicitly encourage or even trigger what Schön (1983, 50) calls *reflection-in-action*:

Resolución de problemas

> Usually reflection on knowing-in-action goes together with reflection on the stuff at hand. There is some puzzling, or troubling, or interesting phenomenon with which the individual is trying to deal. As he tries to make sense of it, he also reflects on the understandings which have been implicit in his action, understandings which he surfaces, criticizes, restructures, and embodies in further action.

In MDM, we do this through a familiar software engineering practice called *version control*. At key moments during the day's work, a developer will commit their code to a repository with a brief message describing what they have done. MDM takes this process and supercharges it; it's like a roadmap of your design and development journey. While a standard message might be something as simple as "fix avatar color," an MDM practitioner would write an in-depth reflection not only on the technical implementation but also on its implications for and relationship to the greater design. Thus, for my game *Combat at the Movies* (Barr 2020c), commit messages are extensive reflections on the relationship between technical and design practices. Here's an example, drawing a connection between the player's input and the pathos of the *Citizen Kane* (Barr 2020b) deathbed scene:

MDM like a roadmap

Relación a un gran diseño.

> Citizen Kane: death on timer, tossing and turning. That is, the tank can now rotate while on the bed (though Kane himself doesn't appear to move much at all in the movie) as a way of tossing and turning and as a way of giving the player a little extra agency.

Another tool in MDM is a committed design journaling practice. Here you pause every day or so to record and explore your more conceptual insights into the design and implementation of your current project. In addition to a reflective commit message, the game's process journal would also contain something like the following:

> I think there's something to the idea that you press your shooting button to die instead of to kill, though I don't know that it makes sense in the context of *Citizen Kane* to say that he chose to die in that moment?
>
> An alternative would be to say that the objective of this game is to say Rosebud before you die? To "realize" or recall the importance of Rosebud before death takes you? In which case I would just have a timer running (visible or not?) and if you hit the action button you say Rosebud and then die, perhaps you get one point for saying it. (Barr 2020a)

This is very much in line with what Schön calls a "stop and think," in which "the designer exhibits a reflection on action, pausing to think back over what she has done in a project, exploring the understanding that she has brought to the handling of the task" (Schön, interviewed in Bennett 1996).

MDM almost *forces* me (and you, if you'd like) to act as a reflective practitioner. I do it gladly. I've now written hundreds of thousands of words about the games I've made in the last few years. It's these records of my conversations with the stuff games are made of that have made it possible to write this book. Without them, I'd be stuck reminiscing, trying to remember why I made everyone wear pants in *The Artist Is Present* or what really happened the day virtual water flooded my gallery in *v r 3*. These records are what allow me to talk to you about video game design *as it happens*. I know what I know about the stuff games are made of because I talk with it every day and I take notes. I don't see how we can hope to understand game design *without* documented reflection throughout the process. Tomar notas, documentar el proceso y/o reflexión.

That's what this book is about, my conversations with the stuff of video games. Some of the conversations are awkward, some of them are funny, some of them are provocative. They're all important to me, and I think they're worth overhearing.

In the following chapters I'll report on some of the outcomes of my conversations with stuff. In each chapter I'll take you on a deep dive into the design and development of one of my own games, focusing on the roles one distinct material played in each process. Each game is itself a meditation on that same material, which makes the discussion interestingly recursive. The first four chapters deal with practical, concrete stuff. In the very next chapter, we'll see *PONGS*, a set of thirty-six variations on the Atari classic *Pong*. Here we'll engage with the *rules* of a game, and how the tiniest change can change everything: a paddle becomes a ghost, a game becomes a painting. In chapter 3, we'll look at my game *Let's Play: Ancient Greek Punishment: CPU Edition!* and contemplate the stuff of *computation*, the fundamental fabric of software. We'll think about code, infinity, and existentialism in a game with no (human) players. In chapter 4, it will be time to talk about *graphics* through an examination of *v r 3*. We'll focus on the visual representation of water in video games and how to look at that water on its own terms rather than as part of the scenery. Finally, in chapter 5, we'll engage with the *user interface* of the game *It is as if you were doing work*. It's a game that is, in some sense, *all* interface, so there's a lot to be said.

After a short interlude during which I'll survey the ground we've covered to that point, we'll turn to more conceptual, but no less fundamental, stuff in the next four chapters. In chapter 6, *The Artist Is Present* will be our focal point for a reflection on *time* in games, particularly on the idea of *feeling*