PORTFOLIO / DESIGN DOCUMENTS

System & Screen Design — Design Documents

Dorian Souc

Application portfolio — System Planner

This portfolio shows system design, screen design and design documentation, produced on personal projects, freelance contracts, recruitment tests and self-directed studies.
Work done under employment — including the console UI/UX work at Kylotonn — is covered by NDA and is described in writing instead.

CONTENTS

01 Professional experience (described in writing — NDA)
02 Skills, languages & tools
03 Aös — published board game, system design
04 Forever Untold — tactical RPG in development, system design
05 Ascension! — game proposal
06 (Re)Construction — game proposal
07 Pre-battle unit placement — personal study
08 Herd — freelance contract
09 Kanji Rush — screens for a Japanese-learning game
10 Ship design screen — recruitment test

Candidate

Name	Dorian Souc
Current role	Bandai Namco Entertainment Europe Junior Associate Producer
Education	MSc Game Design & UX (CNAM-ENJMIN) / BSc Cognitive Psychology
Languages	French (native) / English (business) / Japanese (JLPT N2)
Portfolio	https://www.doriansouc.com

Professional experience

Everything produced during employment is covered by confidentiality and is not shown in this portfolio. The roles are described in writing below.

2025 — present

Bandai Namco Entertainment Europe

Junior Associate Producer

- Coordinating Bandai Namco Europe's internal teams on titles led from Japan — developed in Japan, often outsourced to external studios
- Milestone and build tracking; keeping progress visible across teams
- Coordination around Live Ops operations
- Aligning Japanese and European teams across language and working methods

> Note: title names and internal documents cannot be disclosed.

2021 — 2023

Kylotonn (KT Racing)

UX Designer

- Led core game experience design on an endurance racing title — competitive benchmarking, gameplay flow and progression structure
- HUD information design, menu flows and player navigation
- Wireframes and prototypes (Figma / Adobe XD), through to implementation with engineers and artists
- Writing specifications and balancing references to align the team

> Note: deliverables cannot be shown — unreleased titles.

2020 — 2021

Ubisoft Paris

User Researcher

- Designing and running playtests grounded in cognitive psychology
- Quantitative and qualitative data analysis
- Feedback to production teams on game feel and clarity

> Note: titles studied and test results cannot be disclosed.

2023 — 2024

teamLab Planets (Tokyo)

Customer operations / international support (working holiday)

- Worked in Tokyo: multilingual visitor support and on-site operations
- Professional experience in a Japanese-language workplace

Skills & languages

Only tools and areas actually used in professional or personal work are listed.

Tools

- Figma (wireframes, screen flows, prototypes)
- Adobe XD
- Unreal Engine 5 (Blueprint, AI-assisted C++)
- Unity
- Excel (data handling, balance work)
- Jira / Confluence / Git

Design areas

- Screen and information design (HUD, menus, navigation)
- Writing specifications (GDD, screen specs)
- Playtest design and analysis
- Benchmarking and comparing existing titles

Languages

- French — native
- English — business level
- Japanese — JLPT N2 (preparing N1)
- One year working in Tokyo

What this portfolio does and does not include

— Everything here was produced on a personal project, a freelance contract, a recruitment test or a self-directed study.

— The design work I did professionally — notably the console UI/UX work at Kylotonn — is under NDA and cannot be shown. This portfolio stands in for it: the same kind of work, done on my own.

— Some documents use existing games as their subject. I did not work on any of those titles.

— A few projects were deliberately left at exploratory stage; the page concerned says so.

— A detailed PDF for each project (Japanese and English) is available at https://www.doriansouc.com.

PERSONAL PROJECT — PUBLISHED

Aös

A negotiation board game built on the prisoner's dilemma — designed, balanced and published solo

Overview

A 4-to-10-player negotiation board game. Each round, players are paired with a neighbour and secretly choose to ally or betray; the payoff matrix drives an influence economy. Designed from 2016 and released in 2022.

My role

- Game design, rule design and balancing
- Rulebook writing and layout direction
- Production follow-up and manufacturing
- Crowdfunding campaign and fulfilment

Format & figures

4 to 10 players (best at 6 or 7) — 30 to 120 minutes

300 copies funded and sold through Ulule. IP fully owned.

Why it matters

No engine, no VFX, no code to hide behind. If the rules do not produce an experience, it shows at the first playtest.

A second edition, redefined as a party game, was proposed to a publisher and is currently at proposal stage.



Cast and card art. Each character carries an attribute and a mercenary pool used during the alliance phase.



Board, tokens and event cards in play.



First edition, as shipped to backers.

System breakdown

How one round works, and the two rules that carry the whole design.

01 Alliance phase ▶ 02 Event phase ▶ 03 Scoring & round end ▶ 04 Elimination

PAYOFF MATRIX — STANDARD RULES

	partner ALLIES	partner BETRAYS
I ALLY	+1 / +1	−1 / +2
I BETRAY	+2 / −1	±0 / ±0

Mutual trust pays, but it can always be beaten. An optional harsher ruleset (+1/−3, +3/−3) raises the stakes without changing the shape of the decision.

ROUND STRUCTURE

01	Alliance phase	repeats until a player reaches 3
02	Event phase A	every player at exactly 3 draws
03	Phases B, C …	while anyone still sits at 3
04	Round ends	the moment a player reaches 5

Alliances only re-form when a player is eliminated or left without a partner. Otherwise pairs persist and move up the board.

WHAT THE SYSTEM PRODUCES

Success is what exposes you
The gate at 3 turns progress into risk. No coefficient was tuned to achieve it.

Score reads as a position
Players stall below the threshold and negotiate to push rivals into it.

Elimination removes power, not presence
A player at zero keeps arguing, so the table never goes quiet.

Every choice is paid for
Allying, betraying, advancing — each one costs something traceable to a decision.

INFLUENCE TRACK

0	1	2	3	4	5
recovery	start	safe	must draw	exempt	round won

Only a score of exactly 3 forces an event draw. A player at 4 is exempt.

Approaching victory is what exposes you, and the rule achieves it without a single balancing coefficient. Players stall under the threshold on purpose and negotiate to push rivals into it.

RECOVERY AT ZERO

REMOVED
- — Character attribute
- — Mercenary cards
- — A say in event choices
- — Presence on the board

RETAINED
- — A voice in every negotiation
- — Influence over other players
- — A return in the next round

A player dropping to zero goes into recovery until the end of the round. Mechanical power is removed, social power is not — which is how the game avoids the dead time that elimination usually creates.



Rulebook and components, first edition.

PERSONAL PROJECT — IN DEVELOPMENT

Forever Untold

A tactical RPG where the story branches on how battles are won, not on dialogue choices

Overview

A tactical RPG built in Unreal Engine 5. Its central idea is that narrative branches are opened by conditions met on the battlefield — clearing a map within a turn limit, reaching a character before they fall, holding a flank — rather than by dialogue options after the fact.

My role

- Game design, class design and battle system
- Branching model and chapter structure
- UI and screen design
- Prototyping in UE5 (Blueprint, AI-assisted C++)

Scope & status

Solo tactical RPG, 12–15 h main path — PC and consoles

In development. Battle system and branching conditions are implemented as a prototype; world and scenario design are in progress. Not released.

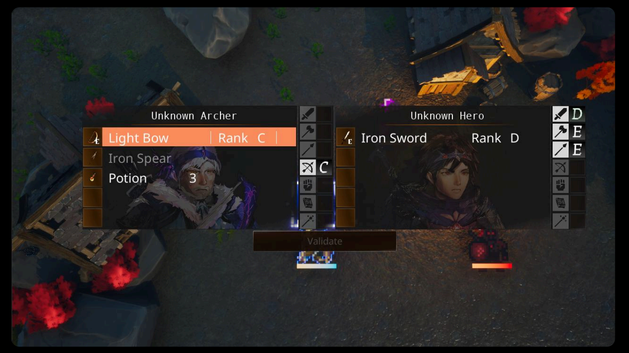
Why this project

Tactical RPGs usually resolve their drama in cutscenes and dialogue menus. This project asks what happens when the same drama is produced by the rules the player is already playing.

Designed and prototyped solo, with a small team forming around a vertical slice.



Combat prototype in Unreal Engine 5. Grid, turn order and the conditions that open a branch are already running.



Map and deployment view.



Menus and inventory.

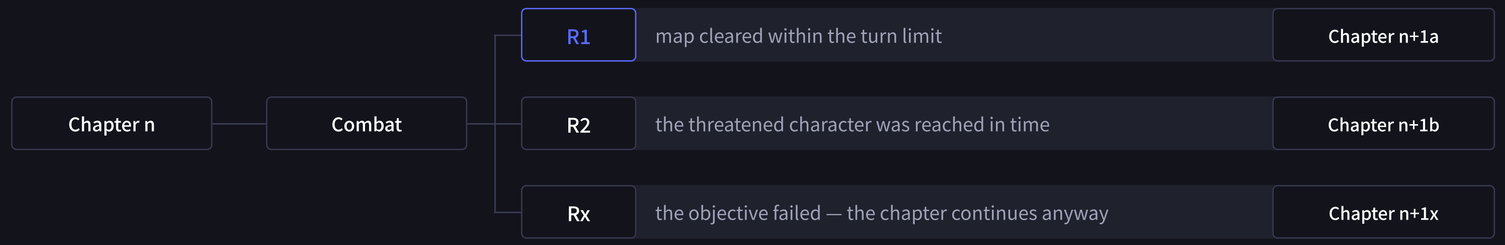
System breakdown

How a chapter branches, and what the player has to trade to open each path.

01 Introduction ▶ 02 Preparation ▶ 03 Combat ▶ 04 Turning point

BRANCHING MODEL — ONE CHAPTER

R = resolution — the outcome a combat settles into



Each combat resolves into one of several outcomes, and the outcome selects the next chapter. The player never picks a branch from a menu — the branch is the record of how the battle actually went.

CORE LOOP — SIX PHASES

01 Introduction
the chapter states what is at stake

02 Preparation
squad, classes and equipment are committed

03 Combat
the tactical layer, where conditions are met or missed

04 Conclusion
the outcome is recorded, losses are permanent

05 Turning point
the branch opens from that record

06 Interlude
the hub, where relationships move

Combat and turning point are the two phases carrying the design: one produces the record, the other spends it.

WHAT THE SYSTEM PRODUCES

Consequence without a menu

The player is never asked to choose a path. The path is what their play produced.

Losses are permanent

A character lost in combat is gone from the story, and the chapters that needed them change shape.

Replay reads differently

A second run is not a different dialogue option; it is a different battle, fought differently.

The cost is visible in advance

Turn limits, rescue windows and held positions are all on screen before the player commits.

POSITIONING



PERSONAL PROJECT

Ascension!

Roguelite boss rush — concept through core loop on a single page

Overview

A personal one-pager: concept, core loop, target audience and intended platforms on one page. A light follow-up explores class selection.

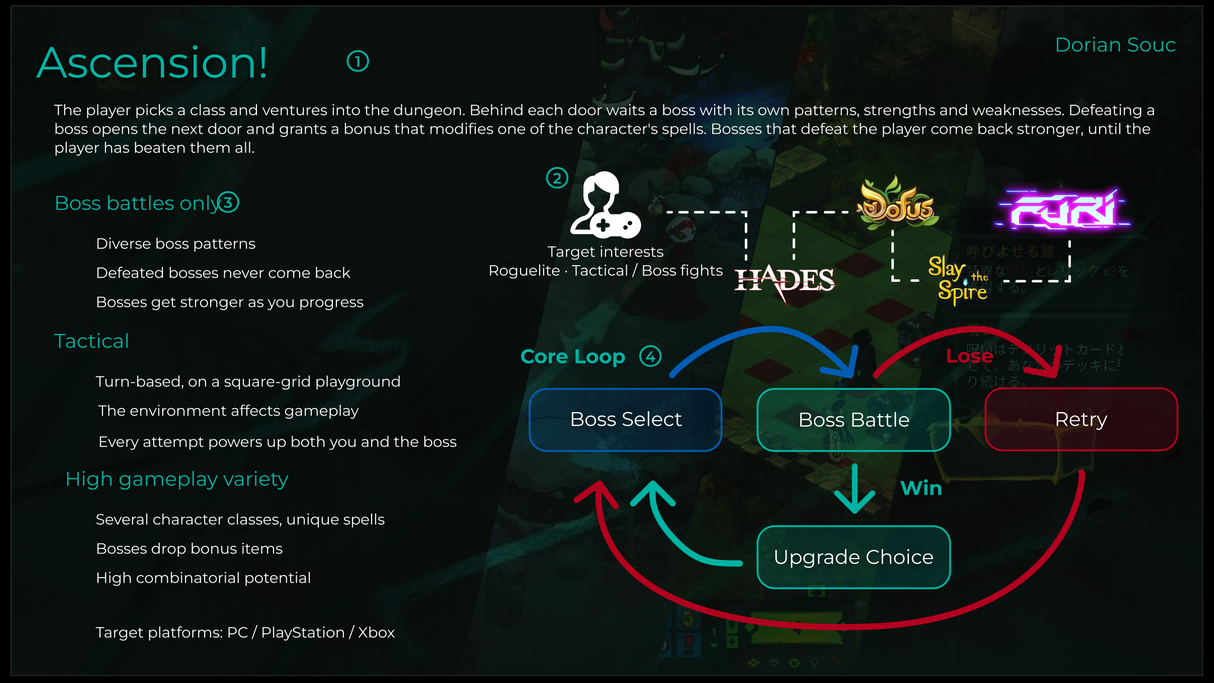
The pitch

- Nothing but boss fights, turn-based, on a square grid
- Every win powers up both the player and the bosses
- A defeated boss never returns, so every run differs
- Each class has its own spells and its own feel
- Intended platforms: PC / PlayStation / Xbox

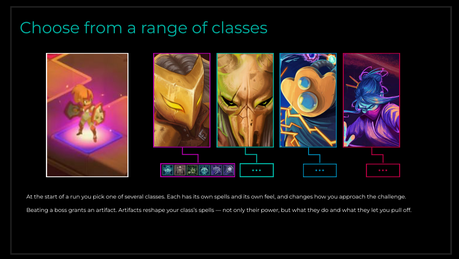
Note

This stayed at one-pager stage — no detailed specification, no implementation.

2-page PDF available at doriansouc.com.



The one-pager itself. Concept and pillars on the left, core loop diagram on the right.



Light follow-up after the one-pager: class selection.

PERSONAL PROJECT

(Re)Construction

Third-person action built on telekinesis — puzzle, combat and story on one page

Overview

A personal one-pager for a third-person action game set in a dreamlike fantasy world, where moving objects with your mind is how you open the way forward.

The pitch

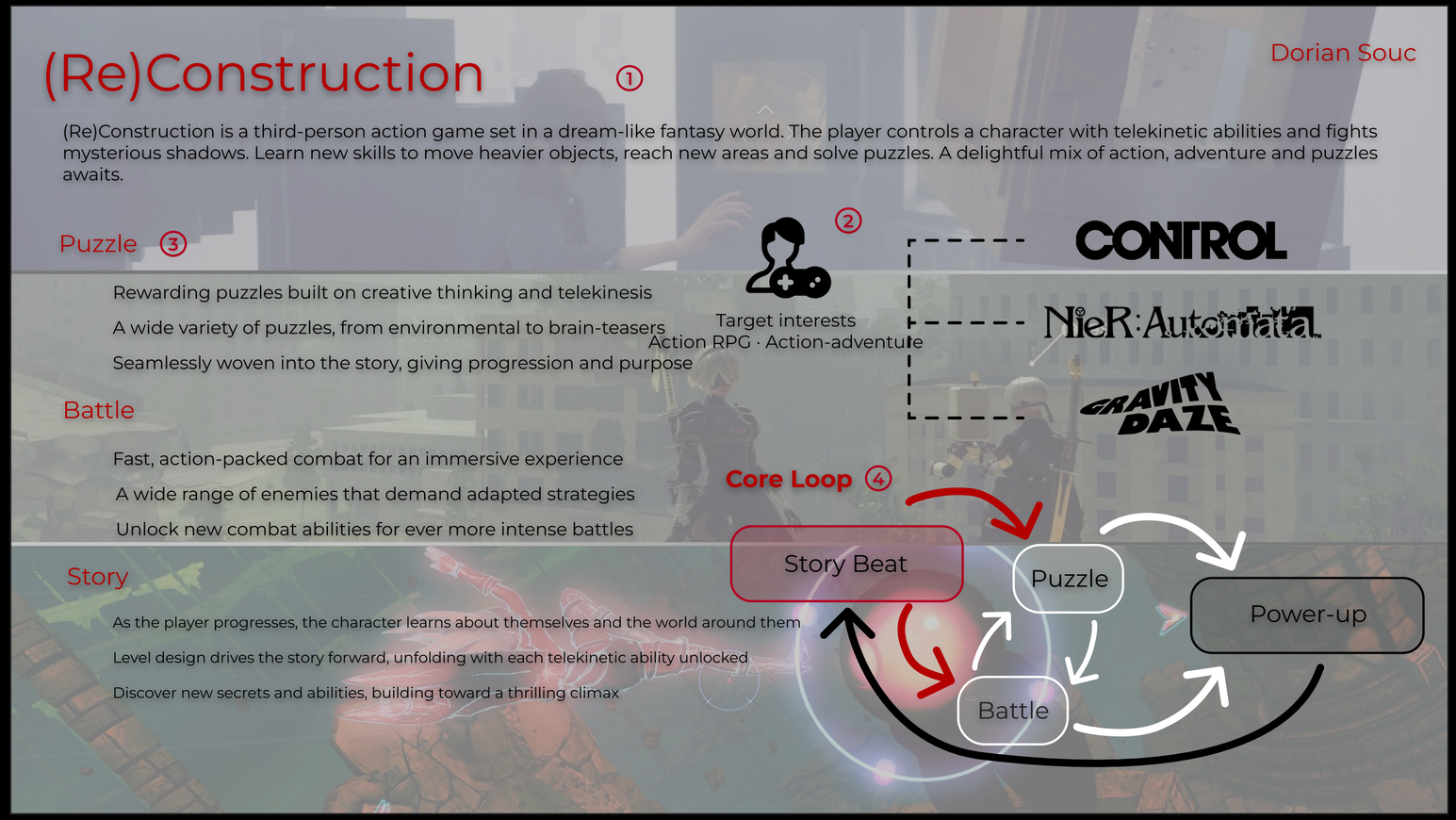
- Puzzle: reshape the environment with your power; solving pays off
- Combat: enemy types force you to change how you fight
- Story: the level design itself carries the narrative forward
- Unlocking a power advances puzzle, combat and story at once

Reference titles

Control / NieR:Automata / GRAVITY DAZE — listed to frame the target audience.

Note

This stayed at one-pager stage — no detailed specification, no implementation.



The one-pager itself. Three pillars on the left, target audience and core loop on the right.

PERSONAL STUDY

Pre-battle unit placement — UX study

A study of how tactical RPGs let you place units before a battle, and of the flow I would build

Overview

I compared how three existing titles handle pre-battle placement, then built my own screen flow from that. This is not client work, and part of it was deliberately left at exploration stage.

What I did

- Benchmarking three titles (Digimon / Fire Emblem / Disgaea)
- Listing where placement controls tend to go wrong
- Building a screen flow across four passes
- Working through the confirm, swap and cancel branches

Tools & date

Figma (wireframes, screen flows)

Duration: 1–2 days, before joining Bandai Namco

Full document

6-page PDF available at doriansouc.com.

Game screenshots © Bandai Namco / WITCHCRAFT — used for study purposes only.



My own placement screen: board, available units and the confirm action kept on one screen.

Digimon Survive	min. 6 inputs per unit
Unit auto-selected (turn order)	
Select "Move"	Select a tile
Select "Attack"	Select a target
Select "End turn"	
Set unit facing	

Benchmark: Digimon

Fire Emblem	min. 3 inputs per unit	
Select a unit	Select destination tile	
Select "Attack"	Select an enemy	
or		
Select a unit	Select an enemy	Confirm the attack

Benchmark: Fire Emblem

Process

Check how existing titles do it, then close the control branches one at a time.

01 Benchmark ▶ 02 First flow ▶ 03 Control branches ▶ 04 Revised flow

Disgaea	min. 4 inputs per unit
Select a unit	Select destination tile
Select "Attack"	Select an enemy
additional	
Set unit facing	

01 | Benchmark: Disgaea. Every input needed to place a unit, written out step by step.



02 | First flow — only the order of the screens at this stage.



03 | Control branches: where swapping and cancelling get handled.



04 | Revised flow, aimed at fewer steps before the battle starts.

FREELANCE CONTRACT

Herd

From taking over an existing prototype, to writing the GDD, to near-final UI wireframes

Overview

I took over a prototype the client had already built — a 1-v-1 rhythm battle RPG for mobile, with deck building, collection and crafting — and was responsible for writing the rules down and designing the screens.

My role

- Analysing the existing prototype and listing its problems
- Writing the GDD (design document)
- Greybox screen design based on that GDD
- Extending it to near-final UI wireframes (around 240 screens)

Tools & duration

Figma (wireframes, screen flows)

Duration: about 3 months

Full document

27-page PDF available at doriansouc.com. These two pages are the summary.

The project was reworked and rebooted after my involvement, and now ships on Steam. The screens here are the mobile version as it stood at the time.



Battle HUD. Round state, the five music categories and the opponent all on one screen, at near-final UI quality.



Home / ranking screen

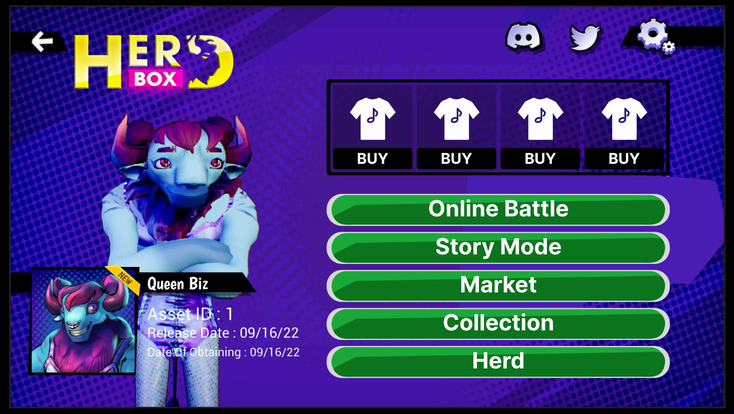


Deck building screen

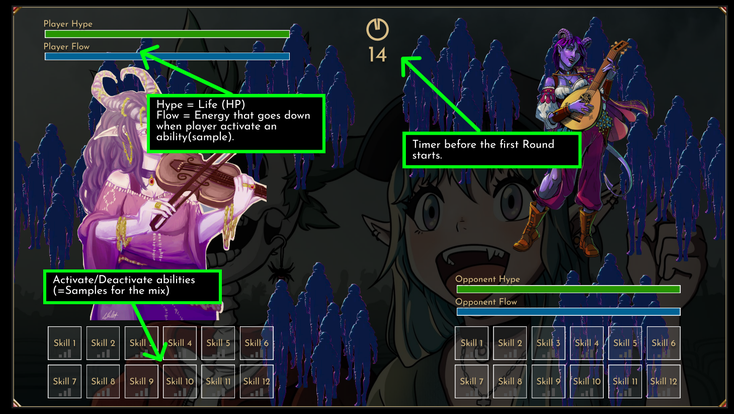
Design process

Write the existing game down without breaking it, then take it down to screens.

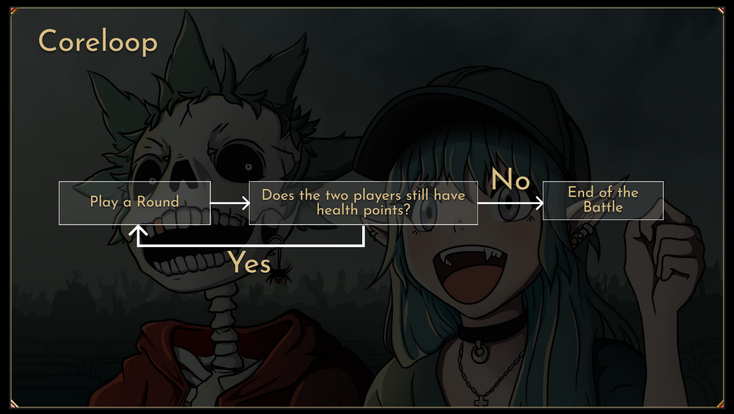
01 Analyse prototype ▶ 02 Write the GDD ▶ 03 Greybox screens ▶ 04 Near-final UI



01 | The prototype I took over — the starting point for mapping the structure and the problems.



02 | Battle screen layout defined in the GDD, and what information belongs on screen.



03 | Round flow and state transitions from the GDD, used to align with the developers.



04 | Near-final UI screen built from the GDD (crafting).

TWO-PERSON TEAM · RELEASED

Kanji Rush

Screen design for a Japanese-learning game covering kana, radicals and kanji (formerly Projet K)

Overview

A full set of screens for a Japanese-learning game built on radicals and components. I handled how the material is split, what the quiz screen shows, the free-selection mode and the progress record. Made by a team of two and released on itch.io in March 2025.

My role

- Structuring the material (2,136 jōyō kanji across 810 decks, with review gates)
- Mode selection screen (five modes plus free selection)
- Quiz screen information design (mistakes, timer, successes, difficulty)
- Progress screen (completion, level, total time)
- Light and dark theme support

Tools & duration

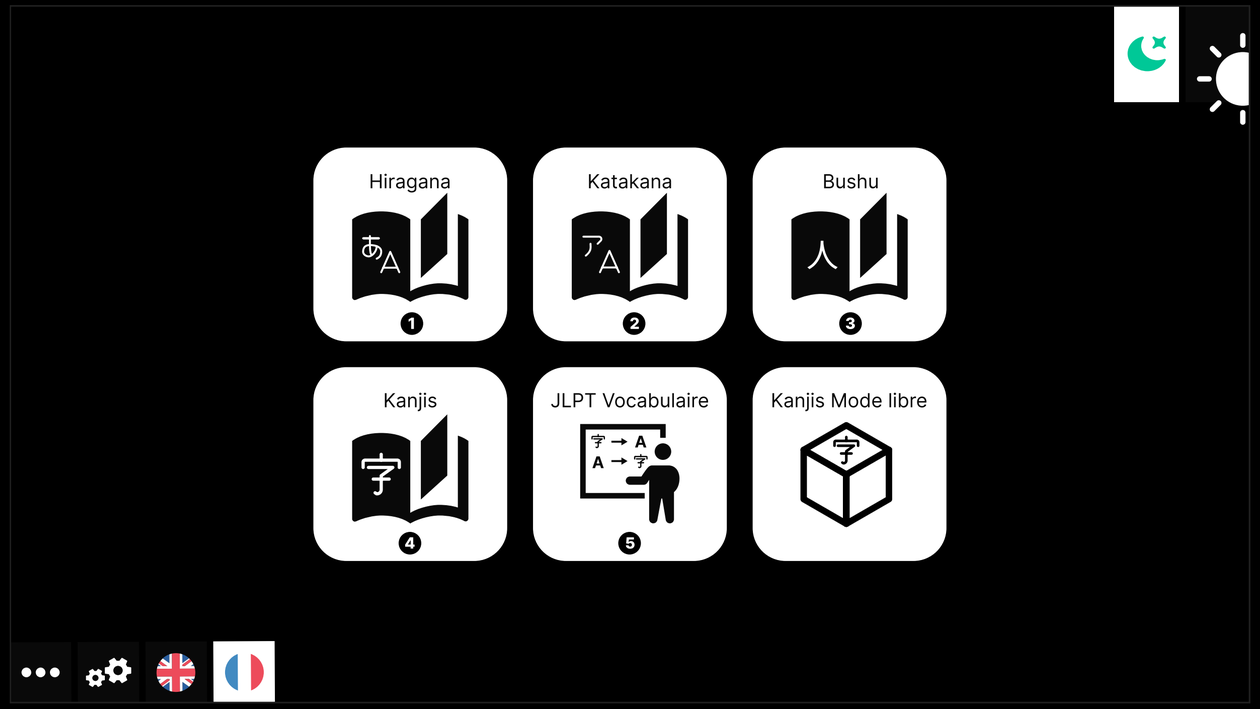
Figma (wireframes, screen flows)

Duration: about 6 months

Full document

10-page PDF available at doriansouc.com.

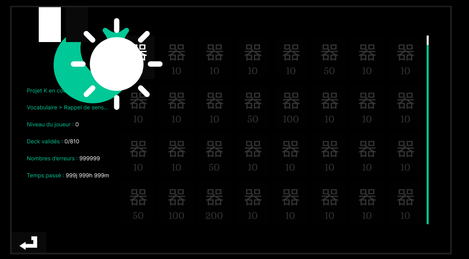
Playable: fleaurian.itch.io/kanji-rush



Mode selection. Five modes — kana, radicals, kanji, JLPT vocabulary — plus a free mode where the player builds their own range.



Quiz screen. Mistakes, time, successes, difficulty and best score always on screen.

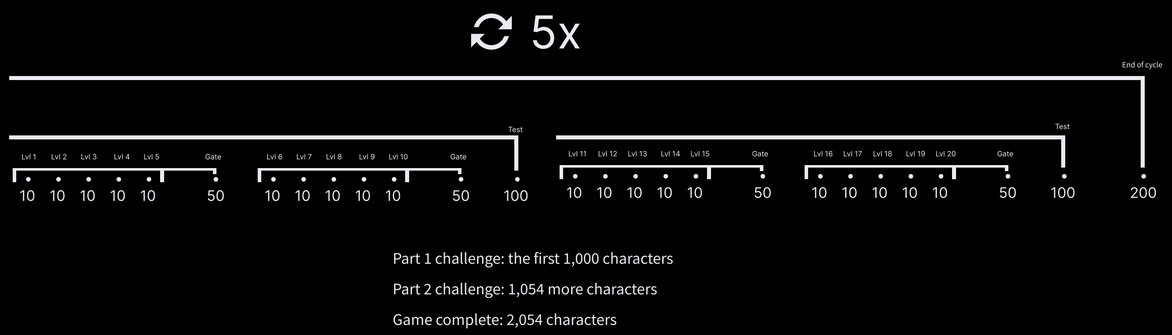


Progress screen: completion, level, mistakes, total time.

Design process

Draw the whole map first, then work through it one screen at a time.

01 Map the screens ▶ 02 Key screens ▶ 03 Per-mode flows ▶ 04 Visual pass

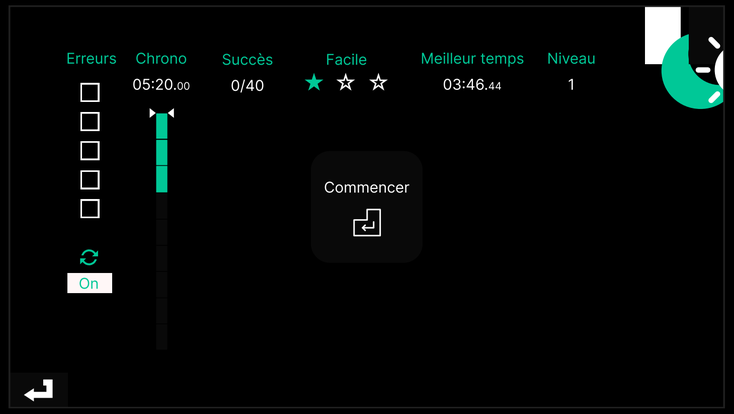


Part 1 challenge: the first 1,000 characters

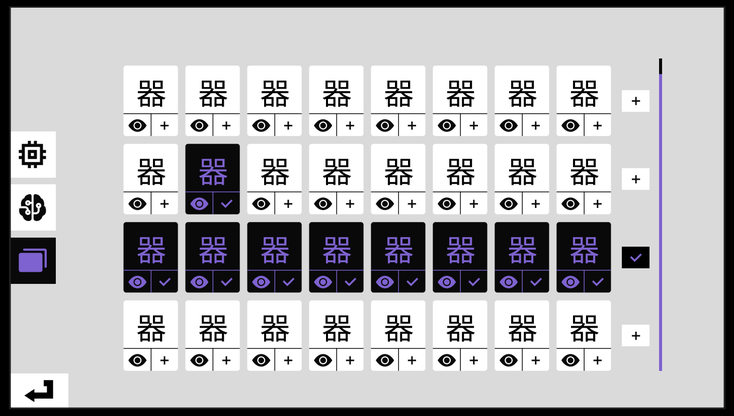
Part 2 challenge: 1,054 more characters

Game complete: 2,054 characters

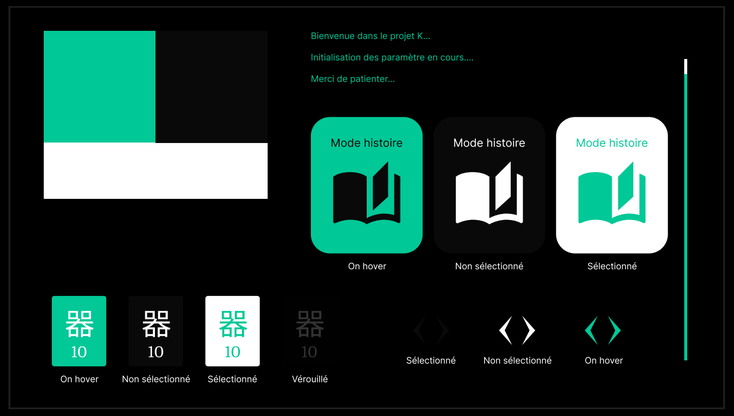
The overall screen map. Which screen leads where was fixed before anything else.



Pre-start confirmation, cut down to what the player needs right before starting.



Free selection mode, where the player assembles their own range.



Visual pass, dark theme. The light theme uses the same layout.

RECRUITMENT TEST

Ship design screen (Warden) — UX study

A screen for building a ship out of modules — delivered as a working Figma prototype, not a static mockup.

Overview

A technical test on the ship-design screen of Endless Space 2 (Amplitude Studios). Rather than stopping at static wireframes, I built the hover and selection behaviour as a Figma prototype — 54 elements carry a hover state in the first direction, 55 in the second.

My role

- Reading the brief and listing what the screen must show
- Layout and information priority
- Interaction design around handling modules
- Checking the behaviour in a Figma prototype
- Defining the UI styles

Tools & duration

Figma (wireframes + prototype)

Duration: one week (technical test)

Full document

8-page PDF available at doriansouc.com.

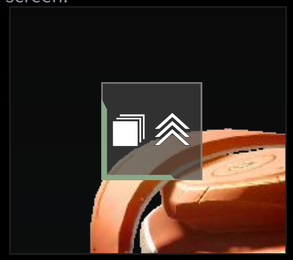
Ship model and universe © Amplitude Studios / SEGA — shown as recruitment-test work.



First version. Ship preview on the left, module list on the right, so a choice and its result stay on one screen.



Module slots on the hull



Hover state on a slot

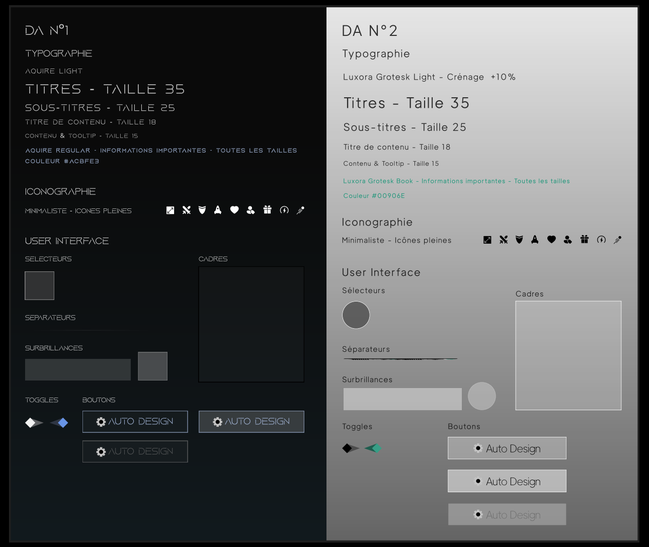
Design process

Decide what happens when you touch something first, then settle how it looks.

01 Read the brief ▶ 02 Layout ▶ 03 Interaction ▶ 04 Styling



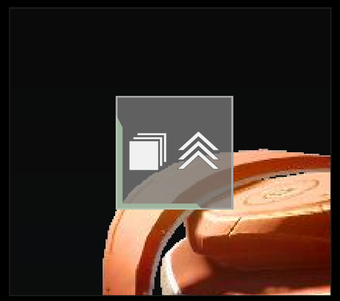
A second version with different density and placement, so the two could be compared.



UI style definition: colour, type, icons and states.



The AUTO DESIGN helper button, state 1.



The same module slot in another state (zoom). Each state was drawn separately.