Álvaro Maldonado Peña

インタラクティビティとgame feelを専門とする開発者・ゲームプランナー

Web: alvaromaldonado.com/ja
メッセージ: alvaromaldonado.com/ja/mailbox
スペイン・マドリード・リモート、ハイブリッド、マドリードでの出社。転居も可

概要

インタラクティビティとgame feel（操作感）を専門とする開発者・ゲームプランナー。ゲームメカニクス、遊べるプロトタイプ、ブラウザで動くリアルタイム3D（three.js、WebGL、Godot）を設計・実装・調整し、TypeScriptとReactによるWeb・モバイルアプリも一貫して開発。自動テスト、WCAG AA準拠のアクセシビリティ、毎回の独立レビューにより、プロトタイプから完成品まで仕上げます。

主なスキル

Game design · Game feel · Interaction design · Gameplay prototyping · Animation & motion · Real-time 3D · three.js · WebGL · Godot 4 (GDScript) · TypeScript · React · Node.js · PWA · Accessibility (WCAG AA) · Automated testing (Playwright, Vitest)

職歴

ソフトウェア開発者

ソフトウェア開発会社（社名非公開）· 2026年8月〜現在
ソフトウェアチームでのプロダクト開発。

語学

スペイン語: 母語 · 英語: C2 · 日本語: C1 · ドイツ語: C1（レベルはCEFR準拠です）

プロフィール

見た目だけでなく、使ったときの感覚を大切にしています。メカニクスを設計し、実装し、考えなくても伝わるまで調整します。ジャンプの重さ、ボタンの弾み、何かが反応するその一瞬まで。

小さなスタジオのように働きます。まず何の課題を誰のために解決するのかをはっきりさせ、噛み合うパーツに分けてつくり、独立したレビューで確かめるまで完成とはしません。こうして、遊べるプロトタイプから完成品までアイデアを届けます。

スペイン語、英語、日本語、ドイツ語を話し、転居にも前向きです。どの国のチームとでも一緒に働けます。

私にできること

触れて気持ちいい、デジタル体験をつくります

一つひとつの操作の「手触り」を大切にしています。すぐに反応し、ちょうどいい重さがあり、不自然に飛びません。よくできたゲームのように、真面目なWebサイトでも。

- 見た目だけでなく、プロダクトの手触りを設計します。即座の反応、予備動作、重さ、慣性、落ち着き。瞬間移動や飛びは作りません。
- イージングと、さじ加減したjuice。操作が伝わるのに必要なぶんだけで、飾りは足しません。視覚と触覚のフィードバックも含みます。
- 自然にふるまうキャラクター：状態、注意（カーソルを目で追う）、待機、好奇心、直接操作（つかむ、投げる）への反応。
- 新しいメカニクスごとに、game feelとアニメーションの専門監査担当が、映画とゲームの基準でレビューします。
- Webでも3Dでも同じです。物理的なモノのように扱えるインターフェースを、モバイルでも、「視差効果を減らす」設定を尊重しながら実現します。
- その証拠はこのトップページにあります。私のバーチャルアシスタント、MANCAがここにさりげなく住んでいます。

ゲームを設計し、遊べる形にします

アイデアから実際に遊べるプロトタイプまで。わかりやすいメカニクスとルール、自然に反応するキャラクターを、スマホでも楽しく伝わるまで調整します。

- Godot 4（GDScript）：活動レベルに応じてふるまうキャラクターとプロシージャルアニメーション。アニメーションの監査担当がレビューします。
- リファレンスに基づくアートディレクション、組み込み前に承認するためのモデルビューア、小さな画面でのシルエットと視認性のチェック。

ブラウザで動く3D体験をつくります

リンクを開くだけ、インストール不要。リアルな物理挙動で、スマホでもなめらかに動くインタラクティブな3Dの世界です。

- three.jsとBVHによる衝突判定、Rapier（WebAssembly）による物理演算：つかむ、投げる、移動できる構造物。
- WebSocketを使った権威サーバー方式のマルチプレイヤー。クライアントとサーバーで同じシミュレーションを実行します。
- パフォーマンス：固定フレームレートを目標に、自動計測を行い、カクつきやちらつきを潰します。
- 操作感へのこだわり：イージング、重さ、落ち着き、一人称カメラ。飛びや二重表示はありません。
- ワールドはデータとして定義し（YAMLによるコンポーネント、カタログ、スタイル）、タイル単位で組み立てます。

Web・モバイルアプリを一貫して開発します

サーバーからスマホの画面まで。インストールでき、通知が届き、オフラインでも動き、端末間で同期するアプリです。

- TypeScriptで一貫開発：Node（Fastify、WebSocket、ネイティブSQLite）、React、Vite。
- Workboxによるインストール可能なPWA、Web Push通知、オフライン対応。
- GoogleサービスとのOAuth連携と、端末間のリアルタイム同期。
- Cloudflareによるエッジのインフラ（Workers、D1、Turnstile、トンネル）と、Tailscaleによるプライベートネットワーク。
- 間隔反復（FSRS）のような学習アルゴリズムを、テスト付きで実装します。

アイデアを本番まで届けます

何が大事かを決め、きちんと噛み合うパーツに分けてつくり、独立したレビューで確認できるまで完成とはしません。小さなスタジオのように働きます。

- まずルールを決めます。部品同士の共通言語（TypeScriptの型と、追加のみで既存を壊さない変更履歴）は、作業を割り振る前に確定させます。
- 並行作業：各パートを専用のブランチ（gitのworktree）で進め、共通の状態記録で連携します。
- 作り手の考えを見ない独立したレビュー（UX/UI、game feel、アニメーション、アート、アクセシビリティ、セキュリ

ティ、ライセンス）を、重大・中程度の問題がなくなるまで繰り返します。
- 生きたドキュメント：推奨案つきの判断記録、バージョン履歴、デザイン記録。

品質・アクセシビリティ・安全性を守って届けます

テスト済みで、スクリーンリーダーやキーボードだけでも誰もが使え、利用者のデータを大切に扱います。

- 自動テスト：Vitest、node:test、Playwright。複数クライアントを使ったE2Eテストも行います。
- WCAG AA準拠のアクセシビリティとUX/UI監査。モバイルファーストで考えます。
- 設計段階からのセキュリティとプライバシー：不要なものは公開しない、端末ごとの権限、多層的なスパム対策、GDPRへの対応。
- 公開前に第三者のライセンスを確認します（フォント、音楽、CC BY-SAのデータなど）。
- Windowsのデスクトップ自動化：システム起動時の自動実行、自動復旧、物理コントローラーとの連携。

使う人のことから考えます

プロトタイプで終わらせず、最初から誰が使うのか、プロダクトとしてどう続けていけるかを考えます。

- プロダクトとビジネスの視点：プロトタイプから、売れるものへ。
- 使う人と支払う人の両方を考えて、優先順位を決めます。