

Tetr3d

Joc de Tetris en 3D amb sincronització al núvol

Pau Abril Iranzo

Institut Puig Castellar

01 — INTRODUCCIÓ

| El punt de partida

Per què aquest projecte?

El projecte neix de **dos interessos** i **una observació tècnica**:

- Volia provar el **desenvolupament de videojocs** amb Godot, un motor 100% FOSS
- Volia experimentar amb **DocumentDB**, una alternativa lliure a MongoDB

Les implementacions de Tetris 3D existents es queden a mitges:

- No completen totes les mecàniques del Tetris clàssic
- No ofereixen cap funcionalitat distribuïda: **comptes, sincronització, rànquings**.

La proposta: Tetr3d

Un Tetris 3D fidel al clàssic + un sistema complet client-servidor.

El joc

Godot + GDScript.

Multiplataforma: Linux i Windows.

El back-end

FastAPI + Python.

Sincronització de configuració i sistema de puntuacions.

La web

Astro + Vue.

Landing, gestió de comptes, rànquings i administració.

| Tres components *independents* que es comuniquen amb una única *especificació OpenAPI*.

Objectius

Objectiu general: dissenyar i implementar un joc puzzle 3D estil Tetris amb sincronització al núvol i una web, formant un sistema complet i jugable.

Objectius específics:

- Implementar les **mecàniques del Tetris clàssic** en 3D (rotacions, hold, peça fantasma, generador sense repeticions, DAS/ARR).
- Construir un **back-end REST** amb autenticació i rànquings.
- Desenvolupar una **aplicació web** amb landing, comptes, rànquings i panell d'administració.
- Definir una **especificació OpenAPI única** com a contracte entre clients i servidor.

02 — DECISIONS TÈCNIQUES

Com s'ha construït i per què

L'stack tecnològic

Joc



Godot



GDScript

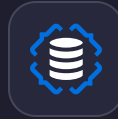
Back-end



FastAPI



Python

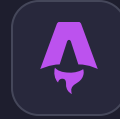


DocumentDB



Valkey

Web



Astro



Vue



TailwindCSS

Infraestructura



Docker



Caddy



OpenAPI

Tot l'stack és *programari lliure*. Una decisió de projecte, no una casualitat.

Per què aquestes tecnologies



Davant d'Unity i Unreal Engine:
FOSS, exports nadius lleugers. No calien gràfics d'alta qualitat.



Davant d'Express: ja m'era familiar i té **generació automàtica d'OpenAPI**.



Alternatives **FOSS** recents a MongoDB i Redis.



Web estàtica i ràpida, amb illes
Vue per a les parts interactives.



Contenidors per a poder tenir un desplegament **fàcil i replicable**.



Una única especificació per a generar SDKs i documentació **automàticament**.

Metodologia de treball

- Gestió amb una **instància pròpia d'OpenProject**
- Tasques i fites organitzades en un **diagrama de Gantt**
- Desenvolupament per **fases**, seguint els components:

1. Videojoc funcional

Versió jugable sense connexió.

2. Back-end

Auth, models i persistència.

3. Web

Aplicació web integrada amb l'API.

4. Integració

Connectar el joc amb el back-end.

5. Interfície

Polir la UX/UI del joc.

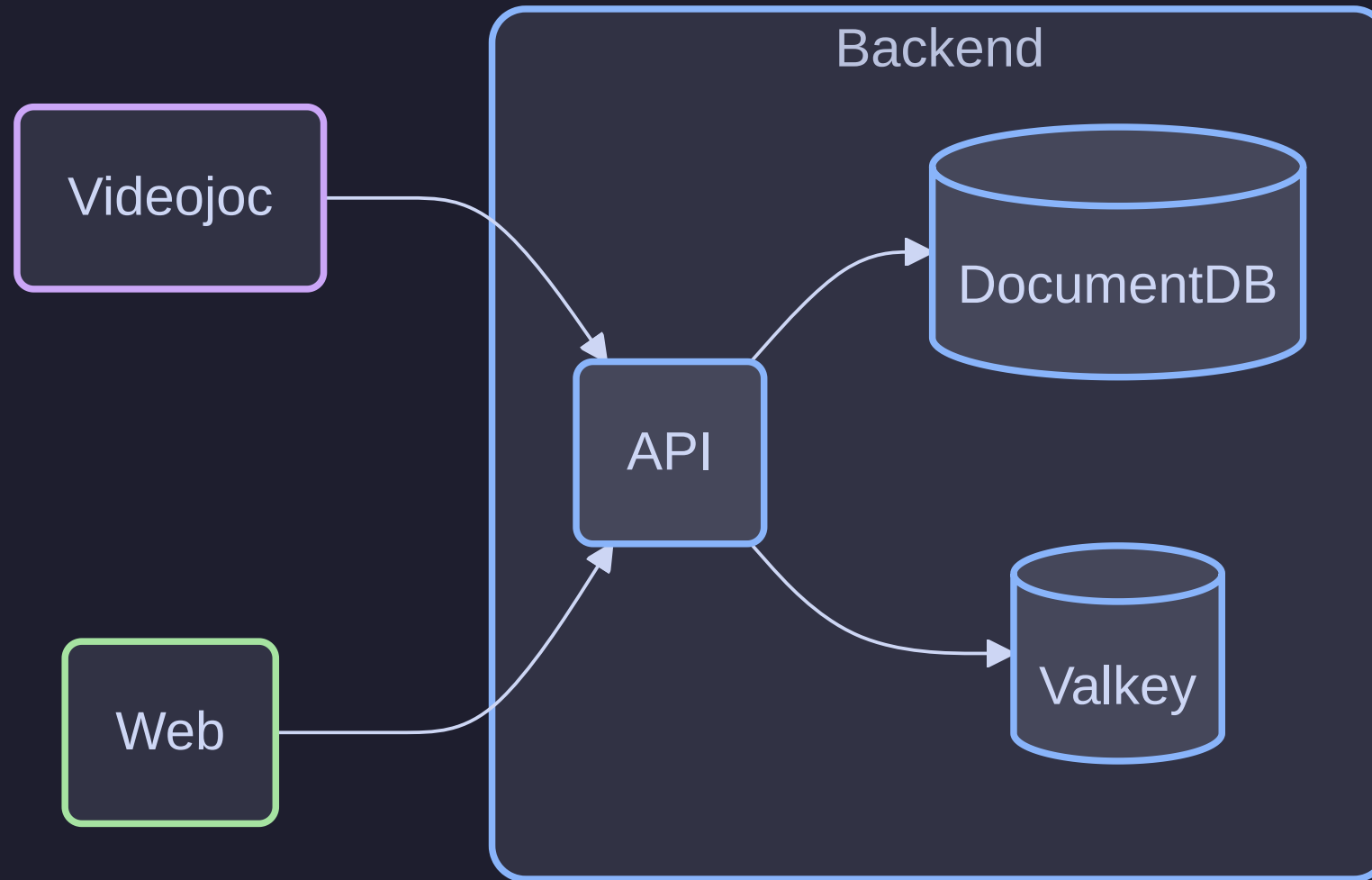
6. Correccions

Bugs finals i panell d'administració.

03 — ARQUITECTURA

Com encaixa tot

Arquitectura del sistema



L'especificació OpenAPI

El problema dels sistemes multiclient: **mantenir-los sincronitzats**.

La solució: una única especificació **OpenAPI 3.0**, generada pel back-end.

- El joc genera el seu **SDK en GDScript** (`openapi-generator`)
- La web genera el seu **SDK en TypeScript** (`@hey-api/openapi-ts`)
- Cap crida a l'API s'escriu a mà
- Els models de dades se sincronitzen **automàticament**
- La documentació de l'API és **automàtica i sempre actualitzada**

Si l'API canvia, es regeneren els SDKs i els clients s'actualitzen sols.

El joc: mecàniques

Totes les mecàniques esperades del Tetris clàssic, en 3D:

- 8 peces diferents (tetracubs)
- Moviment en X i Z
- Rotació sobre els tres eixos
- Gravetat i hard/soft drop
- Hold (peça reservada)
- Peça fantasma
- Generador en bosses de 8
- DAS / ARR configurables

Quatre modes de joc: *estàndard* (10×10×20), *petit*, *pla* i *personalitzat*.

El joc: arquitectura interna

`field_controller.gd`

El **bucle principal**. Connecta peces, camp, entrada de l'usuari i detecció de pèrdua.

`Piece`

Una **peça concreta**. Es renderitza, es mou, rota i calcula col·lisions.

`PlayField`

Les dades del **camp de joc**. Els límits, cel·les ocupades i les parets.

`PieceData`

La definició de la **forma** i **color** de cada peça.

`PieceGenerator`

El **generador aleatori** amb previsualització de les següents peces.

`theme.tres`

El **tema centralitzat** dels components de la interfície.

El back-end: API

FastAPI, rutes organitzades per recursos sota `app/routes/`.

- Autenticació JWT: tokens signats amb claus asimètriques EdDSA
- Contrasenyes amb hash bcrypt
- Validació d'entrada i sortida amb Pydantic

El back-end: Base de dades

Les dades persistents es guarden a **DocumentDB**, les dades temporals a **Valkey**.

Tres colleccions de documents a la BBDD:

users

Dades dels **usuaris**: dades d'autenticació i rol (usuari o admin).

game_configs

Configuracions de joc dels usuaris, sincronitzades entre dispositius.

games

Dades de les **partides**: estat, puntuacions i historial.

La web

Astro per a contingut estàtic + illes Vue per a la interactivitat.

Pàgines

- Landing i descàrregues
- Documentació
- Pàgines d'error

Illes interactives (Vue)

- Rànquings públics
- Login i registre
- Panell d'administració
- Estat d'auth en una store **Pinia**

| Crides a l'API amb el client TypeScript generat + **TanStack Query** per a la gestió d'estats i cache.

04 — DEMOSTRACIÓ

Vegem-ho funcionar

05 — CONCLUSIONS

Què s'ha aconseguit

Assoliment dels objectius

Assolit

- Joc 3D jugable amb les mecàniques del Tetris
- Back-end amb auth, sincronització i puntuacions
- Web completa amb rànquings i admin
- Sistema desplegable amb Docker

Pendent

- Moviment de càmera
- Registre d'usuaris des del joc
- Més operacions d'administració
- CI/CD no implementat

| *L'objectiu principal, un Tetris 3D amb sistema client-servidor complet, **s'ha assolit**.*

Reptes durant el desenvolupament

- Aprendre **desenvolupament de videojocs** des de zero amb Godot
- La **falta de documentació de DocumentDB** va causar problemes durant el desenvolupament i el desplegament
- Gestionar el **temps** com a únic desenvolupador

Valoració de la metodologia: OpenProject amb diagrama de Gantt va funcionar bé, però per a un sol desenvolupador va resultar una eina *excessiva*.

Visió de futur

Càmera

Moviment lliure o posicions predefinides.

Traduccions

Idiomes, tant al joc com a la web.

Multijugador

Partides entre jugadors.

Sessions

Renovació automàtica de l'autenticació.

Admin

Més operacions d'administració.

Modes

Nous modes de joc.

Gràcies

Tetr3d — Joc de Tetris en 3D amb sincronització al núvol

Preguntes?