BoardGame Referee — Mode d'emploi

Mémo perso : comment utiliser l'app, l'admin, et dépanner les pannes utilisateur.

Admin

Admin

Consulter les feedbacks

Consulter les feedbacks

> *Dernière mise à jour : 2026-05-10*

Pourquoi

Voir où l'oracle se plante (et où il assure) pour orienter les améliorations RAG.

`/admin/feedback`

- **Tableau paginé** (20 par page)
- **Filtres** : vote (up/down), gameId
- **Stats agrégées** : total questions, count up/down, avg bestScore

Détail d'un feedback

Clic sur une ligne → panneau `AdminFeedbackDetail` se déploie en dessous :

- **Badges** : langue, intent (rules / synergy / meta / deckbuilding), ID court, date
- **Question + réponse rendue** (markdown)
- **Diagnostics** :
  - Top chunks retrouvés (snippet + page range + sectionTitle + isExtension + score reranker)
  - HyDE généré
  - Timings barre par étape (HyDE / embedding / search / reranker / LLM)

Bouton "Copier (md)"

Génère un dump markdown complet (`buildAnalysisMarkdown`) avec :

- Métadonnées
- Question
- Réponse
- Top chunks détaillés
- HyDE
- Timings

Tu peux le coller dans un éditeur ou dans une session Claude pour analyser un cas spécifique.

Export CSV

Bouton "Export CSV" en haut → POST `/api/admin/feedback/export` → fichier CSV avec toutes les questions filtrées (questionId, vote, gameId, question, answer, bestScore, intent, timestamps).

Utile pour bench RAG, dashboards externes, ou simplement archive.

Alimenter le banc d'éval

Les feedbacks votés down avec commentaire deviennent des cas-test naturels du banc `tests/rag/`. Le workflow propre serait :

1. Sélectionner les feedbacks down avec commentaire détaillé
2. Reformuler en cas-test (`question`, `expected behavior`)
3. Les ajouter dans `tests/rag/dataset.json` (à voir si le format évolue)
4. Lancer `npm run test:rag` pour mesurer la régression

(Pas encore automatisé — pour l'instant c'est manuel.)

Resync les cartes d'un TCG

Resync les cartes d'un TCG

> *Dernière mise à jour : 2026-05-11*

Pourquoi

Quand un TCG sort un nouveau set (ou que tu veux refresh la base après un fix de bug source ou une amélioration de traduction), il faut **re-synchroniser la collection Qdrant** avec la nouvelle source.

Depuis le 2026-05-11, **tout se fait depuis** `/admin` — plus besoin de SSH dans le container pour la majorité des TCG.

Où

`/admin` → section **Bases de cartes**.

Tu y trouves une ligne par TCG, avec :

- Le **nom du jeu** et le nombre de cartes actuellement indexées
- Un **badge de fraîcheur** coloré :
  - □□ **Vert** — synchronisé il y a moins de 7 jours
  - □□ **Jaune** — il y a moins de 30 jours
  - □□ **Rouge** — il y a plus de 30 jours, ou jamais synchronisé
- La **durée de la dernière exécution** (ex : "dernière exécution : 24 min")
- Un **bouton d'action** (voir ci-dessous, deux variantes)

Les deux modes de resync

□□ Mode "Live" — Riftbound uniquement

Bouton "Resync. (live)" : la source Riftbound est une API Riot interrogeable à chaud. Un clic suffit pour comparer la collection Qdrant avec ce que l'API renvoie en direct, et appliquer le diff (ajouts / mises à jour / suppressions).

La progression s'affiche **inline** sous le nom du deck : "Connexion à la source… → Diff calculé → Embedding 32/45 → Upsert Qdrant → Terminé : +3 ajouts, 12 modifs, 0 supprimé / 1 280 total".

C'est rapide (< 2 minutes) car aucune donnée n'est téléchargée localement.

⚙ Mode "Pipeline" — MTG, Lorcana, FAB, Terraforming Mars, Ark Nova

Bouton "Lancer le pipeline" : ces TCG nécessitent un téléchargement préalable (bulk Scryfall pour MTG, LorcanaJSON pour Lorcana, package npm pour FAB, données locales pour TM/Ark Nova). Un clic enchaîne **automatiquement** toutes les étapes :

1. **Téléchargement** de la source (sauf FAB qui est bundlé)
2. **Extraction / parsing** (filtrage FR, dédoublonnage, etc.)
3. **Embedding + ingestion Qdrant**

Le détail des étapes est affiché dans le **modal de progression** qui s'ouvre au clic. Tu y vois :

- La liste des étapes avec leur statut (en attente / en cours / terminé / erreur) et leur durée
- Une **console scrollable** qui streame en temps réel la sortie des scripts npm (logs `[ingest:mtg] 18 234 cartes chargées…`)
- Le **chrono total** dans le footer
- Un bouton **Fermer** en haut à droite

Tu peux fermer la fenêtre

Le pipeline continue côté serveur même si tu fermes le modal ou navigues ailleurs. Si tu reviens sur `/admin` plus tard (avant la fin du job ou dans les 10 minutes qui suivent), le modal se rouvre automatiquement avec l'historique complet des logs et l'étape en cours.

C'est utile car certains pipelines sont longs :

TCG	Durée typique
FAB	~5-10 min
Lorcana	~5-10 min
Terraforming Mars	~5 min
Ark Nova	~5 min
MTG	**~25-40 min** (download bulk Scryfall + 30k embeddings)
Riftbound (mode live)	~1-2 min

Une seule resync à la fois

Pour ne pas saturer le GPU d'embedding, **un seul job tourne à la fois** (toutes collections confondues). Si MTG est en cours et que tu cliques sur FAB, le bouton FAB est désactivé et un message t'indique qu'un autre pipeline tourne déjà. Attends la fin du premier (ou ferme le modal et fais autre chose pendant que ça mouline).

Vérifier que ça a marché

À la fin du pipeline :

1. Le badge de fraîcheur passe au **vert** ("À l'instant" / "Il y a 1 min")
2. Le **compteur de cartes** est mis à jour (ex : 1 280 → 1 312)
3. La **durée** s'affiche ("dernière exécution : 24 min")

Pour valider concrètement :

- Va sur `/play` du jeu en question, ouvre une question et tape `@<nom d'une carte du nouveau set>` → la carte doit apparaître dans l'autocomplete
- Pose une question synergy : "*Quelles sont les nouvelles cartes du set X ?*" → l'oracle doit pouvoir les citer

En cas d'erreur

Si une étape échoue, le modal affiche :

- L'**étape en erreur** en rouge dans la liste
- Le **message d'erreur** dans le footer (ex : "Erreur après 3 min — `cards:mtg:download` terminé avec code 1")

- Les **logs complets** dans la console (scroll vers le bas pour voir le traceback)

Le badge de fraîcheur passe au rouge avec un tooltip qui montre les 60 premiers caractères de l'erreur.

Que faire ?

1. Lis l'erreur dans la console : la plupart du temps c'est un service externe en panne (Scryfall down, LorcanaJSON inaccessible) ou un quota dépassé (Haiku rate-limit pour MTG).
2. Attends quelques minutes, relance le pipeline.
3. Si ça persiste, ouvre les logs container côté Unraid pour voir si une dépendance backend est cassée (Qdrant, TEI embeddings).

Cas particulier FAB (toujours utile à savoir)

Le package `@flesh-and-blood/cards` est **embarqué dans l'image Docker**. Le bouton "Lancer le pipeline" ré-ingère ce que ce package contient, mais **pas plus** : si un nouveau set est sorti et que le package n'a pas été bumpé côté repo, le resync ne ramènera rien de neuf.

Workflow pour vraiment ajouter de nouvelles cartes FAB :

1. Update le package localement : `npm update @flesh-and-blood/cards @flesh-and-blood/types`
2. Commit + push → CI Gitea build l'image
3. Pull + restart le container sur Unraid
4. **Ensuite** clic sur "Lancer le pipeline" depuis `/admin`

Admin

Valider un utilisateur

Valider un utilisateur

> *Dernière mise à jour : 2026-05-10*

Pourquoi

Tous les nouveaux comptes (self-register ou créés par admin) sont en statut `pending` par défaut. Tant qu'ils sont pending, ils peuvent se connecter mais sont redirigés vers `/pending` sans accès aux features.

Comment

1. Va sur `/admin` (compte admin requis)
2. Section **Utilisateurs**
3. Filtre éventuellement par username
4. Clique **Confirmer** sur le compte concerné
5. La ligne `users.role` passe de `pending` à `user`

À partir de là, l'utilisateur peut accéder à `/`, `/play`, `/history`, etc.

Permissions complémentaires

- **`canAddGames`** : par défaut `false` pour les nouveaux users. Toggle disponible sur le compte (`/admin` → Users → bouton "Autoriser ajout de jeux"). Quand `true`, l'utilisateur a accès à `/add-game`. Les admins ont toujours `canAddGames` à `true` implicitement (bypass).

Notification mail

Au moment du register, l'app envoie automatiquement un mail de notification à toi (l'admin) si SMTP configuré (`SMTP_HOST` non vide). Le mail contient le username + lien direct vers `/admin`.

Pour rappel, la conf SMTP est optionnelle — si tu n'as pas configuré `SMTP_HOST` dans `.env`, aucun mail n'est envoyé, tu dois checker `/admin` manuellement.

Supprimer un utilisateur

`/admin` → Users → bouton Supprimer (avec ConfirmDialog).

Effets :

- Suppression de la ligne `users`
- Suppression de toutes ses lignes `questions` (cascade)
- Réassignation des jeux qu'il avait ajouté (`addedBy`) à toi (l'admin courant)

Ajouter un jeu

Ajouter un jeu

Ajouter un jeu

> *Dernière mise à jour : 2026-05-10*

Réservé aux users avec `canAddGames = true` (admins par défaut, autres comptes : à toi de toggler dans `/admin` → Users).

Wizard `/add-game` (3 étapes)

Étape 1 — BGG search

Tu cherches le jeu sur BoardGameGeek (API XML v2). Tu sélectionnes le résultat → ses metadata (image, méchaniques, catégories, type) sont rapatriées et stockées dans la ligne `games`.

Si le jeu n'existe pas sur BGG (rare), tu peux skip cette étape et entrer le nom à la main.

Étape 2 — Upload PDF + métadonnées

- **Le PDF** : drag & drop ou file picker. Stocké sur disque dans `/app/pdfs/<slug>-<timestamp>.pdf`, jamais dans git.
- **Type de contenu** :
  - `base` : règles principales du jeu
  - `extension` : à rattacher à un parent existant (détection de conflits activée)
  - `advanced_rules` : variantes/modes avancés d'un parent
  - `faq` : FAQ officielle d'un parent
- **Langue des règles** : `fr` ou `en` (impacte HyDE — le passage hypothétique sera généré dans la bonne langue + traduit côté backend).
- **Démarrer plus tard** (optionnel) : champ datetime, max +7 jours. Cf. page suivante.

Étape 3 — Ingest ritual

Si tu démarres tout de suite, tu vois un suivi SSE en temps réel des actes :

1. **Lecture** : pdfjs-dist extrait le texte
2. **Reconnaissance optique** (OCR — uniquement si PDF scanné, auto-détecté)
3. **Découpage** : chunking sémantique (paragraphes + overlap)
4. **Structuration** : Claude analyse l'arbre chapitre/section/sous-section
5. **Analyse contextuelle** : Claude génère 1-2 phrases de contexte par chunk (Contextual Retrieval B, 10 SSH parallèles, cache JSON)
6. **Compréhension** : embeddings TEI bge-m3 (batch de 32)
7. **Archivage** : push Qdrant
8. **Conflits** (extensions seulement) : compare aux chunks du parent, détecte replaces/modifies/extends
9. **PNG** : `pdftoppm` rend chaque page en image 300 DPI dans `/app/pdfs/images/<slug>/page-XX.png`

Durée typique : 5-15 min selon taille du PDF + utilisation Claude (quota).

Pause quota Claude

Si pendant l'ingestion le CLI Claude renvoie son message « Usage limit reached, reset at HH:MM », l'app :

1. Flush les caches JSON (contextual-llm + conflict-detect)
2. Passe la ligne en `ingestStatus='scheduled'` avec `ingestScheduledAt = resetAt + 2 min`
3. Émet un event SSE `quota_pause` → l'UI affiche un panneau dédié
4. Re-déclenche `runIngestion` automatiquement à l'heure de reset
5. La reprise re-utilise les caches JSON → aucun chunk n'est retraité, zéro perte

Tu n'as rien à faire — laisse tourner.

Si l'ingestion plante

- `ingestStatus='error'` en BDD
- Logs serveur : `tail -f /app/data/logs/server.log` (`docker exec boardgame-referee tail -f /app/data/logs/server.log`)
- Possible relancer en supprimant le jeu (`/admin` → Games → bouton Retirer) puis le re-uploadant

Ingestion planifiée

Ingestion planifiée

> *Dernière mise à jour : 2026-05-10*

Pourquoi

- Lancer une grosse ingestion la nuit pour ne pas bouffer ton quota Claude pendant la journée
- Étaler 3 jeux à ingérer sans tous les lancer en parallèle
- Programmer une ingestion d'extension pour quand le jeu de base sera lui-même fini

Comment

Dans le wizard `/add-game` étape 2, coche "Démarrer plus tard" et choisis une date/heure.

Contraintes : doit être dans le futur, max +7 jours.

Que se passe-t-il en interne

1. Le PDF est uploadé sur disque immédiatement (`/app/pdfs/...`)
2. La ligne `games` est créée avec `ingestStatus='scheduled'` + `ingestScheduledAt`
3. Un `setTimeout` est armé dans `src/cron/ingest-scheduler.ts` pour déclencher `startIngestJob(gameId)` à l'heure prévue
4. Le PDF n'est PAS analysé tant que l'heure n'est pas atteinte

Si tu redémarres le container avant l'heure

Au boot, `restoreScheduledOnBoot()` relit la BDD et re-programme tous les timers en attente. Si une date est déjà passée pendant le redémarrage, le job démarre immédiatement avec une grace de 1 seconde.

Annuler une ingestion planifiée

`DELETE /api/games/:id/scheduled` (via `/admin` → Games → bouton Annuler) :

1. Clear le timer
2. Supprime le PDF du disque
3. Supprime la ligne SQLite

Renvoie 409 si le jeu est déjà passé en `running` ou `done`.

Distinction reprise après quota

Le même mécanisme (`scheduleIngestStart`) est réutilisé pour la pause auto en cas de quota Claude. Côté UI, le motif est différencié (`reason: 'quota'` vs `'user'`) pour afficher la bonne copie. Pour l'utilisateur c'est transparent : laisse tourner, ça reprendra tout seul.

Bienvenue

Bienvenue

À quoi sert cette app

À quoi sert cette app

> *Dernière mise à jour : 2026-05-10*

Boardgame Referee est ton arbitre personnel pour les règles de jeux de société et de TCG. Tu lui poses une question en français, il répond en citant le passage exact du livret (avec page cliquable).

Cas d'usage typiques

- **Pendant une partie de Gloomhaven**, savoir si une figurine peut traverser une case piège sans subir les dégâts.
- **Avant de monter un deck Magic**, lui demander quelles cartes synergisent avec un thème (« propose-moi des elfes verts mana coût ≤ 3 »).
- **Sur Flesh and Blood**, coller ton decklist Fabrary depuis le presse-papier et lui demander d'analyser les couvertures de rôle.
- **Quand tu doutes** (« est-ce que cette extension change la règle de base sur X ? ») il te montre la règle d'origine ET la modification d'extension côte-à-côte.
- **Sur Lorcana**, lever le doute sur ce que fait exactement une carte sans avoir à fouiller dans une boîte.

Ce qui le rend différent

- **Citations cliquables** : chaque réponse pointe vers la page exacte du PDF, ouverture en un clic.
- **Vision** : pour les questions visuelles (icônes, plateau, schémas), Claude lit la PNG de la page directement.
- **Multi-livret** : tu coches les extensions actives, l'oracle compose les règles base + ext + FAQ avec priorité explicite.
- **Mémoire conversationnelle** : les cartes citées au tour 1 restent dans le contexte au tour 5 (sticky mentions).

- **Self-hosted** : tout tourne sur ton Unraid, aucune donnée envoyée à un cloud tiers (sauf appels Claude via la VM oracle).

Ce dont tu as besoin

Ce dont tu as besoin

> *Dernière mise à jour : 2026-05-10*

Côté utilisateur

- **Un compte** : créé soit par toi en self-service (`/register`), soit invité par un admin. Les nouveaux comptes sont en statut `pending` jusqu'à ce qu'un admin les confirme depuis `/admin`.
- **Un navigateur récent** : Chromium / Firefox / Safari modernes. SSE + fetch streaming requis (compatible avec tous les navigateurs depuis 2023).
- **Connexion réseau** vers `https://rules.thymon.fr` (ou le LAN `192.168.10.x` si tu es sur le subnet local).

Côté admin (toi)

- Accès SSH à l'Unraid pour redémarrer le container ou consulter les logs.
- Accès à l'instance Gitea (`gitea.thymon.fr`) pour pousser les builds.
- Connaissance des PDFs des règles que tu veux ingérer (ils restent sur le serveur, pas dans git).

Côté infra (déjà en place sur ton Unraid)

- Container `boardgame-referee` (image Gitea registry)

- Container `qdrant` (vector DB)
- Containers TEI (`bge-m3` sur 8099 + reranker `bge-reranker-v2-m3` sur 8990) sur la RTX 3060
- VM SSH avec compte `oracle` qui exécute Claude Code CLI
- Reverse proxy NPM avec cert Let's Encrypt sur `rules.thymon.fr`

Dépannage

Dépannage

L'oracle est en pause quota

L'oracle est en pause quota

> *Dernière mise à jour : 2026-05-10*

Symptômes

- Une ingestion en cours s'arrête net et passe en `ingestStatus='scheduled'`
- L'UI affiche un panneau "L'oracle se repose, reprise à HH:MM"
- Les questions `/play` peuvent renvoyer une erreur ou rester bloquées en "thinking"

Pourquoi

Le compte Claude utilisé par la VM `oracle` a atteint son quota d'usage (Pro / Team plan, fenêtre 5h glissante). Le CLI `claude` renvoie alors un message du genre :

> Usage limit reached, your limit will reset at 21:00 PM

Le service `claude-quota.ts` détecte ce message (multiples patterns supportés : `reset at`, `try again at`, `available again at`, formats 12h/24h, timestamp Unix) et lève un `ClaudeQuotaError` au lieu de renvoyer la réponse vide.

Ce que fait l'app automatiquement

Pendant une ingestion

1. Les workers SSH parallèles (contextual-llm, conflict-detect) arrêtent de spawn de nouvelles tâches dès qu'une `ClaudeQuotaError` est détectée
2. Les caches JSON sont **flushés sur disque AVANT** de propager l'erreur (critique pour la reprise sans perte)
3. Le `coordinator.ts` rattrape l'erreur, bascule la ligne SQLite en `ingestStatus='scheduled'` avec `ingestScheduledAt = resetAt + 2 min`
4. Émet un event SSE `quota_pause` avec `{ resetAt, retryAt }` → le wizard UI affiche la copie dédiée
5. À l'heure de reset, le scheduler relance `runIngestion` qui re-rentre dans les stages, lit les caches JSON, et ne re-traite que les chunks manquants

Pendant une question /play

L'oracle peut hallu silencieusement quand le quota est dépassé (le CLI renvoie parfois la quota notice comme une réponse assistant normale). Le détecteur scanne aussi le contenu streamé pour ce cas. Si détecté, l'erreur est exposée à l'UI qui affiche un message clair.

Ce que tu peux faire

- **Patienter** : laisser tourner, ça reprendra tout seul à `resetAt + 2min`
- **Vérifier l'heure de reset** : event SSE `quota_pause` ou logs `/app/data/logs/server.log` (grep `quota`)
- **Si urgent** : passer sur un autre compte Claude (changer les credentials dans `/home/oracle/.claude/.credentials.json` côté VM oracle, restart container backend)

Si la détection ne marche pas

Si Claude change le wording du message de quota et que `claude-quota.ts` ne le reconnaît plus, l'erreur sera générique ("answer empty") au lieu de pause planifiée.

Étendre `QUOTA_MARKERS` dans `src/services/claude-quota.ts` :

```
const QUOTA_MARKERS = [
  'usage limit reached',
  'you've hit',
  'rate limit reached',
  // ... ajouter le nouveau marker
];
```

Et tester avec un mock de stream qui contient le nouveau message.

Dépannage

Réinitialiser un mot de passe

Réinitialiser un mot de passe

> *Dernière mise à jour : 2026-05-10*

Self-service (utilisateur)

Pas implémenté pour l'instant — pas d'envoi de mail magic-link en self-service. Si tu oublies ton mot de passe, contacte un admin.

Via admin (toi)

Deux options :

Option 1 — Bouton "Send password reset"

`/admin` → Users → ouvrir un user → bouton "Renvoyer mail de reset".

Appelle `POST /api/admin/send-password-reset/:userId` qui :

1. Génère un token de reset
2. Envoie un mail à l'adresse en BDD avec le lien `/reset-password?token=...`

⚠ Nécessite SMTP configuré (`SMTP_HOST` non vide). Sans SMTP, ça plantera silencieusement (à vérifier dans les logs).

Option 2 — Reset manuel via DB

Si SMTP indisponible :

```
docker exec -it boardgame-referee sh
sqlite3 /app/data/database.db
> UPDATE users SET password_hash = '<nouveau-hash-argon2>' WHERE username = 'foo';
```

Pour générer un hash argon2 :

```
docker exec -it boardgame-referee node -e "
import('argon2').then(a => a.hash('nouveau-mdp')).then(console.log)
"
```

Page `/reset-password`

Accepte un `?token=...` depuis l'URL. Vérifie le token, demande un nouveau mot de passe, le hashe (argon2) et update la BDD. Le token est consommé (one-shot).

Si tu te bloques toi-même (admin)

Hash argon2 directement la nouvelle valeur en CLI comme ci-dessus, ou redémarre le container avec `FIRST_ADMIN_USERNAME` + `FIRST_ADMIN_PASSWORD` mis à jour — au boot, l'app crée/réinitialise l'admin défini par ces vars.

Poser une question

Attacher un deck (FAB)

Attacher un deck (FAB)

> *Dernière mise à jour : 2026-05-10*

Disponible **uniquement sur Flesh and Blood** pour l'instant (`hasCardDatabase = 'flesh-and-blood-cards'`).

Workflow

1. Sur Fabrary, ouvre ton deck → bouton "Copy as Text"
2. Sur `/play` (jeu FAB sélectionné), clique sur le bouton **Deck** dans la pill du composer
3. Modale "DeckImportModal" s'ouvre → colle le decklist dans le textarea
4. Clique "Analyser"
5. Preview affichée : héro, weapon, equipment, mainboard avec vignettes de cartes ; les non-matchés (s'il y en a) en bas
6. Clique "Attacher au chat" → un bandeau compact apparaît au-dessus du composer (héro + N cartes + bouton détacher)
7. Toutes tes questions suivantes ont le deck en sticky mentions (cap 80)

Ce qui est extrait du texte Fabrary

- En-têtes : `Name:`, `Format:`, `Hero:`, `Weapon:`, `Equipment:`, `Deck:`/`Mainboard:`/`Pitch 1/2/3`, `Sideboard:`
- Lignes : `3 Card Name`, `(3) Card Name`, `3x Card Name`, `3 - Card Name`
- Suffixe pitch optionnel : `(red|yellow|blue)`
- Lignes vides et commentaires `//` ou `#` ignorés
- Footer Fabrary (`Voir le deck complet @ https://...`) ignoré (pas de quantité en tête)

Cas d'usage

- **Coverage de rôles** : « ce deck a-t-il assez de defense reactions ? »
- **Synergies internes** : « quelles cartes synergisent avec Snap Shot ? »
- **Optimisation** : « quelles cartes pourrais-je remplacer pour améliorer le matchup contre Briar ? »

Limites

- Pas de persistance : un reload de page = deck détaché. Re-coller la decklist.
- Pas d'export `.dec` ni de bouton Copier
- Pas (encore) sur Magic / Lorcana / Riftbound — c'est faisable mais demande de coder un parser par format

Citer une carte avec @

Citer une carte avec @

> *Dernière mise à jour : 2026-05-10*

Sur les jeux qui ont une base de cartes (`hasCardDatabase` non-null en BDD), tu peux citer une carte directement dans ta question avec `@`.

TCG supportés

- Magic: The Gathering (`magic-cards`)
- Disney Lorcana (`lorcana-cards`)
- Flesh and Blood (`flesh-and-blood-cards`)
- Riftbound (`riftbound-cards`)
- Terraforming Mars (`terraforming-mars-cards`)
- Ark Nova (`ark-nova-cards`)

Workflow

1. Tape `@` dans le composer → un popover apparaît
2. Tape les premières lettres de la carte → liste filtrée (debounce 150ms)
3. Navigue avec ↑↓, valide avec Entrée (ou clique)
4. La carte est insérée dans le texte et "épinglée" à la question
5. Quand tu envoies, l'oracle reçoit les détails complets de la carte (nom, type, mana, abilities, set, rareté…) en plus du retrieval normal

Différence vs retrieval vectoriel

- Le retrieval vectoriel cherche dans les chunks de règles (BASE / EXT / ADV / FAQ / FORUM)
- Le bloc `CARTES CITÉES PAR LE JOUEUR (ce tour)` injecte les cartes que TU as nommées avec `@` — déterministe, pas de score, toujours présent
- Donc même si Qdrant ne sortait pas la carte X dans son top-k, elle sera dans le contexte si tu as fait `@X`

Sticky mentions (multi-tours)

Les cartes citées au tour 1 restent disponibles aux tours suivants (cap FIFO 20). Plus besoin de les re-taper. Le bloc `CARTES ÉVOQUÉES DANS LA CONVERSATION` les rappelle à l'oracle au format abrégé. Si tu veux qu'une sticky déclenche une nouvelle expansion synergy, re-tape la avec `@`.

Citation par l'oracle

Quand l'oracle veut référencer une carte dans sa réponse, il écrit `[[card:Nom exact]]` — rendu côté UI comme un bouton avec mini-image cliquable, ouvre la modale zoom détaillée (CardZoomModal).

Cap "deck attaché"

Si tu attaches un deck (cf. page suivante), le cap sticky passe de 20 à 80 — assez pour un classic constructed FAB (60 main + 12 sideboard).

Poser une question (flux classique)

Poser une question (flux classique)

> *Dernière mise à jour : 2026-05-10*

Étapes

1. **Va sur** `/play` depuis le Home en cliquant sur un jeu.
2. **Coche les extensions actives** dans le header (les non cochées sont ignorées par le retrieval).
3. **Tape ta question** dans le composer en bas.
4. **Envoie** (Entrée, ou bouton). L'oracle commence à streamer la réponse.
5. **Lis et clique sur les citations** pour vérifier la source dans le PDF.
6. **Vote pouce ↑↓** + commentaire si la réponse t'a aidé ou pas — tout est enregistré pour `/admin/feedback`.

Ce que tu vois pendant la génération

- **Phases verbalisées** dans une bulle "thinking" : « L'Oracle pèse votre question », « plonge dans le grimoire », « rédige sa réponse »

- **Tokens streamés** dans la bulle finale au fur et à mesure
- **Fallback hors-ligne** : si la connexion SSE casse, l'animation passe à « L'Oracle finalise hors-ligne… » et l'app récupère la réponse via polling sur `GET /api/ask/:questionId` (jusqu'à 45s).

Que faire si la réponse semble fausse

1. Vote pouce bas + commentaire détaillé.
2. Va sur `/admin/feedback`, filtre par ton jeu + vote down, ouvre le détail.
3. Clique "Copier (md)" pour avoir un dump markdown des diagnostics (chunks retrouvés, scores, HyDE, timings).
4. Si c'est un problème de retrieval (mauvais chunks), vérifie que le PDF est bien indexé (count chunks > 0) et que la question n'est pas trop floue.
5. Si c'est un problème de réponse Claude malgré bons chunks, l'Oracle a hallu — pas grand-chose à faire à part voter pour qu'il apprenne (le feedback alimente ton banc d'éval RAG).

Limites

- 500 caractères max par question
- 5 turns d'historique pris en compte
- 1 image PNG injectée au max (pour limiter la latence vision)
- Heartbeat SSE toutes les 8s pour ne pas se faire timeout par NPM

Premiers pas

Accéder à l'app

Accéder à l'app

> *Dernière mise à jour : 2026-05-10*

URLs

- **Prod** : https://rules.thymon.fr
- **LAN direct** (sans passer par NPM) : `http://192.168.10.100:3000` — utile pour debug si NPM tombe

Login

1. Va sur `/login`.
2. Username + password.
3. Cookie de session valide 7 jours, HTTP-only, SameSite=Lax.

Rate-limit : 5 tentatives ratées par IP avant 15 min de cooldown. Si tu te bloques, attends 15 min ou redémarre le container (la map en mémoire se reset).

Premier admin (au boot)

Au tout premier démarrage du container, l'app crée automatiquement un admin avec les credentials des env vars `FIRST_ADMIN_USERNAME` + `FIRST_ADMIN_PASSWORD`. Tu peux ensuite changer le mot de passe via `/me`.

Pas de token API

Pas d'auth header bearer, tout passe par cookie de session. Si tu veux scripter des appels (ex. depuis un cron), tu dois te login via `POST /api/auth/login` et conserver le cookie.

Si tu vois `pending` après login

Tu es en attente de validation admin. Va sur `/admin` (avec un compte admin) → `Utilisateurs` → bouton `Confirmer`. Sinon Telegram / IRL me ping.

Tour de l'interface

Tour de l'interface

> *Dernière mise à jour : 2026-05-10*

Navigation

- **Desktop** : sidebar à gauche (`NavSidebar.vue`)
- **Mobile** : tab bar en bas (`NavTabBar.vue`)

Pages principales

Route	But
`/` (Home)	Ludothèque visuelle : grille des jeux + bandeau "Reprendre la partie" si une session de moins de 6h existe
`/play`	Cœur du produit : chat avec l'oracle, citations cliquables, autocomplete `@card`, deck attaché
`/history`	Timeline de tes feedbacks groupée par jeu
`/add-game`	Wizard 3 étapes : recherche BGG → upload PDF → ingest ritual (suivi en SSE)
`/me`	Profil + change password
`/me/settings`	Préférences
`/admin`	Dashboard admin : services health, jeux, users, resync cards
`/admin/feedback`	Table des feedbacks avec diagnostics RAG complets + bouton "Copier markdown"

La table de jeu (`/play`)

- **Header** : nom du jeu + extensions actives (cases à cocher) + bouton `Deck` (si TCG supporté) + langue de règles (FR/EN)
- **Centre** : flux de messages (utilisateur en bulle parchemin, oracle avec barre ambre gauche)
- **Composer en bas** : textarea parchemin + autocomplete `@` + bouton envoyer

Citations cliquables

Quand l'oracle cite un livret (`[HEAT, p.4 : "..."]`), tu peux cliquer pour ouvrir le PDF directement à la page citée dans un viewer modal. Si plusieurs livrets sont cités (base + extension), le bon est résolu automatiquement via le nom du livret dans la citation.

Cartes citées (`[[card:Nom]]`)

Quand l'oracle cite une carte de TCG, c'est rendu comme un bouton avec mini-image. Clic → modale zoom (`CardZoomModal`) avec stats détaillées.

Mode "table" (`useTableMode`)

Toggle dans le header pour adapter l'affichage en mode tablette posée à plat (texte plus gros, contrastes renforcés).