



RuneBingo

Présentation du projet

Contexte et problématique

Oldschool RuneScape (*abbv. OSRS*) est un MMORPG fantastique qui compte plus de 64 millions de joueurs. Ces joueurs s'unissent afin de former des clans, où ils s'entraident dans leur progression et forment de micro-communautés. Ils organisent souvent aussi des événements, où ils compétitionnent entre eux pour gagner une cagnotte de pièces d'or, la monnaie principale du jeu.

Le type d'événement le plus répandu au sein des clans est le Bingo. Sa forme la plus simple consiste en une tuile composée de 5 colonnes et 5 rangées, où chaque case représente un objet rare à obtenir, ou un monstre ou donjon difficile à vaincre. Les joueurs s'inscrivent en payant une somme en pièces d'or à l'organisation, puis sont divisés en équipe qui compétitionnent entre elles afin d'obtenir le plus de points et de remporter la cagnotte. Chaque tuile peut valoir un nombre de points, et des points bonus peuvent être attribués pour compléter une ligne, une colonne ou un groupe de tuiles prédéfini.

Il n'existe pas vraiment de solution standard pour gérer des bingos; certains utilisent des Google Sheets, d'autres passent par l'application Discord, d'autres ont carrément créé leur propre application web pour un événement particulier, ou utilisent un mélange de toutes ces options.

Objectif du projet

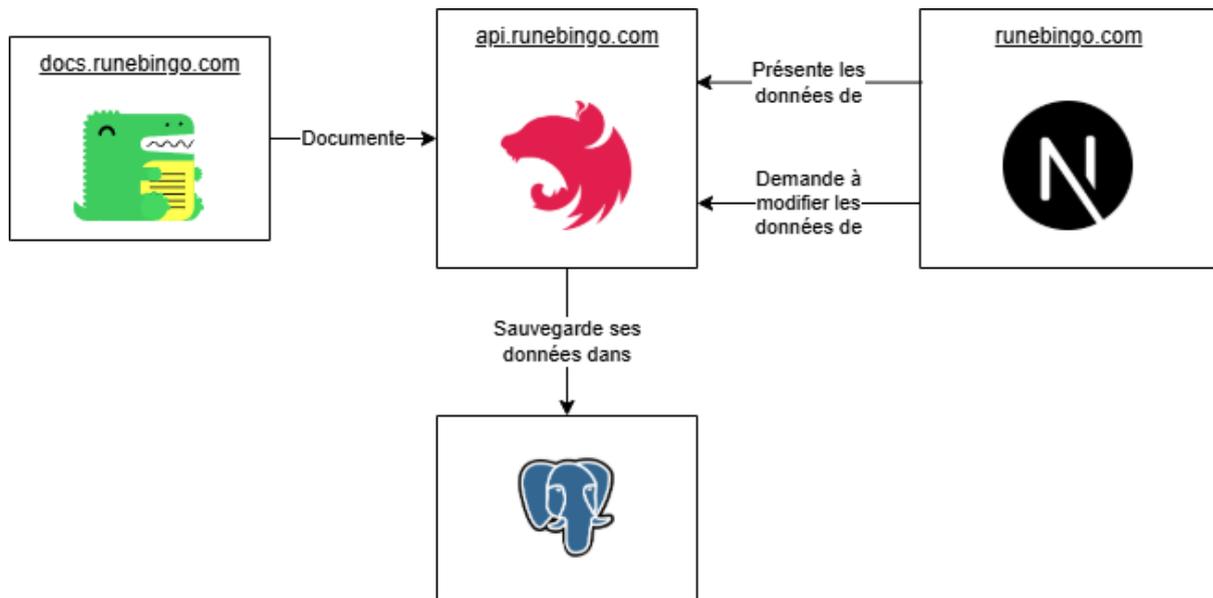
L'équipe de RuneBingo travaille actuellement à développer une plateforme Open-Source qui permet aux organisateurs de créer et gérer des événements de type Bingo, ainsi qu'aux participants de rejoindre et participer à ces événements sans qu'ils ne dépendent d'une autre application.

Planification

L'équipe souhaite lancer son produit minimum viable (*abbv. MVP*) d'ici le mois de septembre 2025. Les fonctionnalités du MVP sont décrites dans [l'analyse technique](#), qui inclut aussi une spécification complète de l'API. L'avancement du projet peut être suivi à l'aide [du projet GitHub](#), une collection de tableaux Kanbans qui organisent nos *issues*, qui se divisent en épiques, fonctionnalités et tâches.

L'équipe utilisera également un serveur Discord afin de communiquer entre nous. Des maquettes pour les vues de l'application seront aussi produites à l'aide de l'outil de design Figma.

Architecture



L'API (api.runebingo.com) est la composante centrale de notre application. Les utilisateurs s'y authentifient via un code reçu par courriel, ou via une clé d'API (dans le futur) et y font des requêtes Http. Elle est développée à l'aide de la librairie [Nest.js](https://github.com/ianstormtaylor/nest.js) et écrite exclusivement en TypeScript. Nous utilisons la librairie TypeORM, qui sert de client SQL pour les écritures et lectures vers notre base de données PostgreSQL. Nous n'avons pas encore identifié d'hébergement pour l'environnement de production, mais nous utiliserons un serveur Ubuntu physique en attendant.

Un site de documentation (docs.runebingo.com) sera éventuellement déployé avec Docusaurus et hébergé gratuitement sur GitHub pages. Il contiendra la spécification de l'API, ainsi que quelques guides utiles au besoin.

L'application web (runebingo.com) sera développée avec [Next.js](https://nextjs.org/) et contiendra les pages statiques de présentation du produit, ainsi que les vues dynamiques. Nous essaierons le plus possible de faire notre génération côté serveur pour que nos pages soient indexables sur les moteurs de recherche. Nous créerons notre *design system* à partir de [shadcn/ui](https://shadcn.com/), une librairie de composants moderne et facile à utiliser.

Équipe

Notre équipe serait composée de :

- **Raphaël Pion**, développeur lead. Gère aussi la planification des tâches, les maquettes et le design system.
- **Charles-André L'Écuyer**, étudiant à l'AEC en programmation, développeur backend. Il est en mode « découverte » et pourra éventuellement aider au développement frontend.

- **Xavier Dostie**, étudiant au bacc. en informatique, développeur frontend. Son rôle principal sera d'utiliser le design system afin de créer les interfaces de l'application correspondant aux maquettes. Il pourrait aussi être amené à créer des composantes dans le design system et faire quelques petites tâches backend, selon ses intérêts.