

Cas pratique — WIZDLE

1 Présentation du projet

WIZDLE est un jeu web quotidien inspiré du concept de Wordle, transposé dans l'univers de la fantasy et de la magie.

Chaque jour, l'utilisateur doit deviner un personnage mystère à partir d'indices progressifs, dans une expérience immersive et narrative.

Ce projet est né de l'envie de créer une application ludique, accessible, mais aussi techniquement structurée, mêlant logique métier, UX soignée et univers visuel fort. WIZDLE n'est pas seulement un jeu : c'est une expérience.

2 Contexte et problématique

Contexte

Dans le cadre de ma formation en développement web full-stack, je souhaitais concevoir un projet personnel permettant de :

- pratiquer React et TypeScript
- gérer un état applicatif non trivial
- travailler la logique métier
- développer une identité visuelle cohérente

Problématique

Comment créer un jeu simple à comprendre, mais suffisamment engageant pour inciter l'utilisateur à revenir chaque jour, tout en restant maintenable techniquement

3 Objectif du projet

Objectifs utilisateurs

- Comprendre les règles en quelques secondes
- Jouer une partie courte et plaisante

- Avoir un sentiment de progression
 - Être immergé dans un univers narratif
-

Objectifs techniques

- Structurer une application React claire et évolutive
 - Mettre en place une logique de jeu quotidienne
 - Gérer l'état du jeu (essais, indices, victoire/défaite)
 - Créer une base extensible pour de futures fonctionnalités
-

4 Parcours utilisateur (UX)

- L'utilisateur arrive sur la page du jour
- Il découvre le thème et les règles
- Il effectue ses premières tentatives
- De nouveaux indices apparaissent
- La partie se termine par une réussite ou un échec
- Le jeu se réinitialise le lendemain

L'UX a été pensée pour être fluide, sans surcharge d'information et en cohérence avec l'univers magique

5 Choix techniques

Stack principale :

- Front-end : React, TypeScript
- Routing : React Router
- Styles : CSS modulaire / animations
- Outils : Git, GitHub, Vite

React pour sa gestion de l'état et sa modularité

TypeScript pour la robustesse et la lisibilité du code

Une architecture par composants pour faciliter l'évolution

6 Architecture de l'application

L'application est structurée autour :

- de composants UI réutilisables
- d'un état global de jeu
- d'une séparation entre logique métier et affichage

Exemple :

- composants d'interface (grille, indices, messages)
 - logique de validation des tentatives
 - gestion du mot / personnage du jour
-

7 Fonctionnalités clés

- Jeu quotidien (1 partie par jour)
 - Système d'indices progressifs
 - Validation des tentatives
 - Feedback visuel immédiat
 - Gestion de la victoire / défaite
 - Univers graphique immersif
-

8 Défis rencontrés

Défi 1 : logique de jeu quotidienne

Gérer un contenu qui change chaque jour sans casser l'expérience utilisateur a nécessité une réflexion sur la gestion du temps et de l'état.

Défi 2 : équilibre difficulté / plaisir

Trouver le bon niveau de difficulté pour rester accessible tout en étant engageant.

Défi 3 : UX narrative

Créer une ambiance immersive sans nuire à la lisibilité.

Ce que j'ai appris

Ce projet m'a permis de :

- structurer une application React complète
 - gérer une logique métier complexe
 - penser un projet du point de vue utilisateur
 - itérer entre design, UX et technique
-

Améliorations futures

- Ajout d'un système de score / statistiques
 - Historique des parties
 - Mode multilingue
 - Backend pour persistance des données
 - Authentification utilisateur
-

Conclusion

WIZDLE représente ma vision du développement web :
un équilibre entre technicité, créativité et expérience utilisateur.