Test Technique – Développement d'un Jeu de Mémoire

1. Contexte

Vous allez développer un petit jeu de mémoire (connu également sous le nom de "Memory"/"Concentration") afin d'évaluer vos compétences en développement **full-stack**. Le but est de reproduire un jeu fonctionnel, accessible et réactif, sans maquette Figma fournie mais avec une série de captures d'écran de référence : vous devrez donc faire des choix de design tout en respectant les spécifications ci-dessous.

2. Objectifs pédagogiques

- Évaluer votre capacité à concevoir l'architecture d'une application web moderne.
- Mesurer la qualité, la propreté et la maintenabilité de votre code.
- Vérifier votre maîtrise du responsive design, de l'accessibilité et des bonnes pratiques UX/UI.
- Juger votre capacité à documenter et déployer une application.

3. Fonctionnalités attendues

1. **Modes de jeu**
    - **Solo**
    - **Multijoueur local** jusqu'à **4 joueurs** (tour par tour).
2. **Sélection du thème**
    - Tuiles contenant soit **des nombres** (1–8/18 selon la taille), soit **des icônes** (librairie libre de droits de votre choix).
3. **Taille de la grille**
    - **4 × 4** (16 cartes)
    - **6 × 6** (36 cartes)
4. **Responsive layout** : la grille et l'interface doivent s'adapter de façon optimale aux différents formats (mobile, tablette, desktop).
5. **États de survol et de focus** : toutes les actions cliquables doivent posséder un *hover* et un *focus* visibles (accessibilité clavier).
6. **Suivi de la partie** :
    - compteur de coups.
    - Tableau des scores (multijoueur) : nombre de paires trouvées par joueur.
    - Détection et annonce de fin de partie.

7. **Redémarrage / nouvelle partie** : possibilité de relancer rapidement avec les mêmes paramètres ou d'en choisir d'autres.
8. **Structure de l'application** : l'interface doit comprendre **au minimum trois pages / vues** :
    - **Page d'accueil / paramètres** : écran de lancement permettant de choisir la taille de grille, le thème (nombres ou icônes) et le nombre de joueurs.
    - **Page de jeu** : grille de cartes avec informations en temps réel (chronomètre ou compteur de coups, scores, tour actuel).
    - **Page de résultats** : récapitulatif de la partie (vainqueur, statistiques) et actions pour rejouer ou retourner à l'accueil.
    - **Page top 10**: Page affichant les TOP 10 des scores
9. **Serveur:** Sauvegarde des scores dans la base de données, endpoint pour exposer le TOP 10 scores, endpoint pour exposer le score moyen des participations ainsi que le nombre total de participations.

4. Contraintes techniques

- **Technologies imposées** :
    - **Next.js** si vous souhaitez profiter du SSR/SSG)
    - **TypeScript**
    - **TailwindCSS** pour le styling et le responsive.
    - **Python >3.12**
    - **FastAPI**
    - **Pydantic**
    - **Base de données SQL (Postgres, SQLite …)**
- Vous êtes libres d'utiliser **toute bibliothèque React open-source** supplémentaire (React Router, Zustand, Framer Motion, TanStack Query, etc.).
  Veillez simplement à justifier vos choix dans le README.
- **Aucune dépendance propriétaire ou payante**.
- **Gestion de l'état** : libre (Contexte React, Zustand, Redux, Jotai…). Justifiez votre choix.
- **Accessibilité** : conformité **WCAG 2.1 AA** (couleurs, contraste, navigation clavier, ARIA).
- **Performance** : temps de chargement raisonnables, absence de fuites mémoire.
- **Versioning Git** : historique clair (commits atomiques, messages explicites).

5. Livrables attendus

Élément	Détails
Dépôt Git	Public ou lien d'archive ; branche principale `main` ou `master`.
README.md	• Description du projet • Instructions d'installation & lancement • Décisions d'architecture

Code source	Organisé, commenté, typé si possible (TypeScript).
Docker	Docker Compose avec toutes les dépendances nécessaires

➡️ **Envoyez l'ensemble des livrables listés ci-dessus en réponse directe à cet e-mail.**

6. Critères d'évaluation

1. **Respect des fonctionnalités** (45 %)
2. **Qualité du code & architecture** (25 %)
3. **UX/UI & accessibilité** (20 %)
4. **Documentation & clarté du README** (10 %)
5. **Déploiement / CI CD** (bonus 5 %)

7. Timing indicatif

- Les livrables sont attendus **lundi (heure de Paris) avant 12 h 00** au plus tard.

Le temps de travail recommandé est d'environ **6 à 8 heures** cumulées ; organisez-vous librement sur la période.

8. Bonus optionnels (non obligatoires mais fortement valorisés)

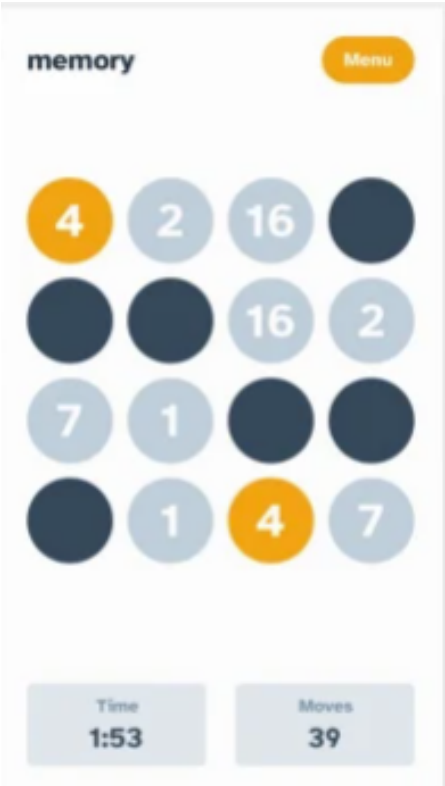
- Mode **dark / light** automatique ou au choix utilisateur.
- Animation de flip de carte fluide (CSS 3D ou WebGL).
- Sauvegarde de partie dans le *localStorage* (reprise après rafraîchissement).
- Utilisation de **TypeScript** et d'un **linter** (ESLint, Prettier).
- Pipeline **CI /CD** (lint, build) avec GitHub Actions.
- Internationalisation (i18n) FR/EN via un fichier de traductions.
- Thèmes des cartes en utilisant une API depuis le back-end (Thème Pokemon, chiens, films …)

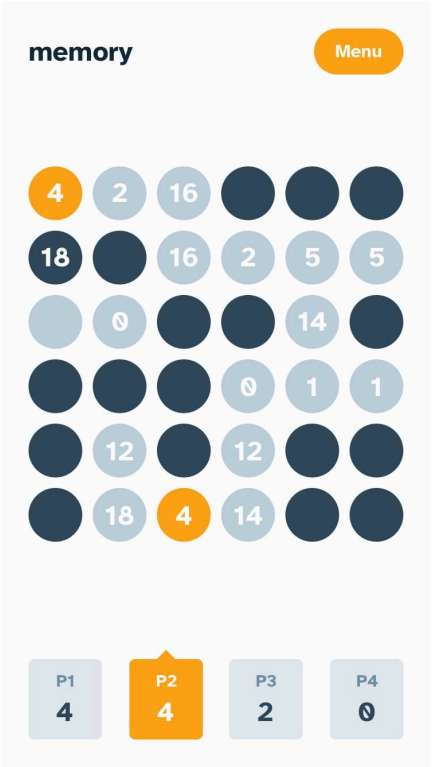
9. Questions & Support

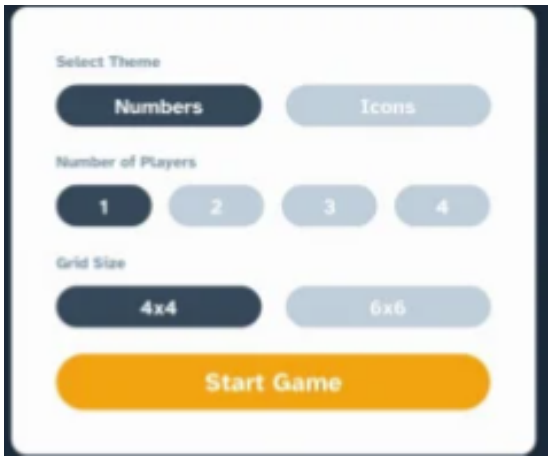
Aucune maquette ni README de référence ne sont fournis : prenez le temps de justifier vos choix de design et d'implémentation dans votre propre README.
Pour toute question technique ou besoin d'éclaircissement, vous pouvez nous contacter par e-mail ou Slack.

Bonne chance !

Pièce jointes :

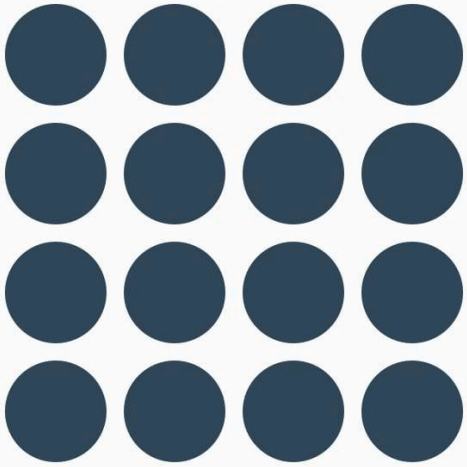

memory
Menu
4 2 16
16 2
7 1
1 4 7
Time
1:53
Moves
39


memory
Menu
4 2 16
18 16 2 5 5
0 14
0 1 1
12 12
18 4 14
P1
4
P2
4
P3
2
P4
0


Select Theme
Numbers
Icons
Number of Players
1
2
3
4
Grid Size
4x4
6x6
Start Game

memory

Restart New Game



Time 0:00

Moves 0


memory
Restart
New Game
Player 1
1
Player 2
3
Player 3
1
Player 4
0
CURRENT TURN