Projet : créer une application de flashcards, associée à 5 jeux vintage, qui fonctionne en ligne en tant que progressive web application, vendu sur l’apple store et l’app store d’android.

Nous commencerons par construire une base flashcards-tetris dans une architecture conçue pour ensuite donner le choix au joueur de jouer avec divers jeux vintage recrutant la même base de flashcards. Les jeux seront : Tetris, Space invaders, Pac-Man, Pong ainsi qu’un quiz (blindtest) pour jouer à plusieurs.

**Here are the main features:**

* The question is written but the answer is not shown at first so the player can retrieve the information (retrieval) and then click on : « I know » or « I dont’ know » buttons, then a multiple choice appear and the player has to click on the good one.
* The game has to entirely responsive following the guidelines of Progressive web applications with no need to scroll to see the entire page whatever is the device used from smartphones to big screens.
* An algorithm is used for spacing distribution into the game : fsrs\_algorithm.js
* All elements of the flashcard library from must be taken
* When the player loose, the system keep a recording of success and failures to apply properly spacing repetitions with fsrs\_algorithm (and do not show cards that were successfully answered in the next party)
* At the end of each game, there’s a button : “play again” (with spacing memory of previous games). After each answer, the status a the card is updated in firebase in order to always respect long term spacing
* When all the flashcards library is considered as known, up to date, then, a question appear : “your memory is up to date !” Do you want to keep playing for fun ?
* Important : the game remember our success and failures so that when we play to the game on the next day, it considers the spacing effect.
* Les cartes avec lesquelles le joueur rencontre le plus de difficultés (fréquentes mauvaises réponses) sont présentées après 7 bonnes réponses successives. Il apparaît alors une mention : Attention question de boss ! écrit sur fond rouge. Les "7 bonnes réponses consécutives" sont-elles comptées au cours d'une même session de jeu (quelque soit le deck).
* Il est mentionné le meilleur score du joueur en haut à droite pour chaque jeu
* En dessous du meilleur score est indiqué le nombre de carte à réviser aujourd’hui pour être à jour
* Au fur et à mesure que le nombre de cartes mémorisées augmente : le joueur voit son statut grimper. À la base il y a un petit icône d’enfant. À chaque changement de statut, une boîte s’affiche indiquant : Félicitation vous avez en mémoire 10 cartes, vous avez une mémoire d’Éléphant ! L’icône devient alors celle d’un éléphant. Le statut est toujours affiché sur la page d’accueil, en plus d’être affiché en petit dans l’écran de jeu en haut à droite avec l’icône correspondant. On se base sur l'intervalle FSRS actuel de la carte (≥ 3 jours). Si une carte retombe sous les 3 jours, elle sort du décompte total pour le calcul du niveau de mémoire. Les badges sont attribués tout au long de la progression :
  + Quand 5 cartes ont été mémorisées pendant au moins 3 jours : vers de terre
  + Quand 10 cartes ont été mémorisées pendant au moins 3 jours : fourmi
  + Quand 15 cartes ont été mémorisées pendant au moins 3 jours : Dory
  + Quand 20 cartes ont été mémorisées pendant au moins 3 jours : poisson rouge
  + Quand 30 cartes ont été mémorisées pendant au moins 3 jours : Homer –- Chat – Dauphin – Cachalot – Éléphant – Yoda – Marie Curie - Sherlock Holmes – Socrates - Albert Einstein
  + Quand 40 cartes ont été mémorisées pendant au moins 3 jours : Obélix
  + Quand 50 cartes ont été mémorisées pendant au moins 3 jours : Astérix
  + Quand 70 cartes ont été mémorisées pendant au moins 3 jours : Chat
  + Quand 100 cartes ont été mémorisées pendant au moins 3 jours : Dauphin
  + Quand 150 cartes ont été mémorisées pendant au moins 3 jours : Cachalot
  + Quand 200 cartes ont été mémorisées pendant au moins 3 jours : Éléphant
  + Quand 300 cartes ont été mémorisées pendant au moins 3 jours : Jedaï : Yoda
  + Quand 400 cartes ont été mémorisées pendant au moins 3 jours : Sherlock Holmes
  + Quand 500 cartes ont été mémorisées pendant au moins 3 jours : Marie Curie
  + Quand 600 cartes ont été mémorisées pendant au moins 3 jours : Wall-E
  + Quand 700 cartes ont été mémorisées pendant au moins 3 jours : Albert Einstein
  + Quand 800 cartes ont été mémorisées pendant au moins 3 jours : HAL 9000 (*2001, l’Odyssée de l’espace*)
  + Quand 900 cartes ont été mémorisées pendant au moins 3 jours : **C-3PO**
  + Quand 1000 cartes ont été mémorisées pendant au moins 3 jours : Chat GPT
  + Quand 1000 cartes ont été mémorisées pendant au moins 3 jours **: Deep Blue** (IA d’échecs d’IBM qui a battu Garry Kasparov en 1997)
* Chaque jour, le joueur peut gagner un petit ajout à son personnage en fonction du nombre de nouvelles cartes mémorisées ce jour-là. Par exemple : Félicitation vous avez mémorisé 5 nouvelles cartes aujourd’hui, vous passez en mode éclair. Un éclair est ajouté à côté du personnage. Pour obtenir Galaxie, le joueur a forcément obtenu tous les précédents (inférieurs). Le reset se fait à 2h du matin. Un message s’affiche à chaque récompense : Bravo, x (nombre) nouvelles cartes mémorisées aujourd’hui, votre cerveau s’illumine, vous passez en mode x (bougie, feu, etc)
  + 5 nouvelles cartes mémorisées aujourd’hui : bougie
  + 10 nouvelles cartes mémorisées aujourd’hui : feu
  + 15 nouvelles cartes mémorisées : ampoule
  + 20 nouvelles cartes : Flash (d’appareil photo)
  + 25 nouvelles cartes mémorisées : éclair !
  + 30 nouvelles cartes mémorisées : comète !
  + 35 nouvelles cartes mémorisées : Étoile !
  + 40 nouvelles cartes mémorisées : Galaxie !
* Le nombre de jour de pratique d’affilée est indiqué quand on débute une session :
  + Bravo, c’est ton 4e jour consécutif
  + Ce nombre de jour est indiqué pendant le jeu, en haut à droite

Si le device change de fuseau horaire, on ajoute +1, c'est à dire que le streak ne sera perdu que 2 jours plus tard à 2h du matin. On peut gagner des jours de streak en plus si l'on arrive au niveau - 30 cartes : Comète (on gagne une journée de streak) - 40 cartes : Galaxie (on gagne trois journées de streak). Après 5 jours consécutifs, on gagne 1 jour de streak. Le streak n'est pas lié à la création de nouvelles cartes.

* Après chaque bonne réponse, il y a un petit son kawai japonais
* Avec chaque annonce de changement de statut ou de passage à un mode supérieur (flash, éclair, etc), un son de victoire retentit
* Les decks peuvent être publiés sur une URL partageable (ex. flash.games/deck/abc123). Toute personne qui a acheté le jeu peut l'importer.
* Les joueurs peuvent choisir de ranger leurs cartes dans un "jeu de cartes" qu'ils peuvent nommer. Mais ceci n'apparaît que lorsque l'on est en train de créer la carte. Après avoir rentrer question et réponse, il nous est proposé de créer un jeu de carte ou d'entrer cette carte dans un jeu existant mais on peut aussi refuser et ne pas ranger les cartes dans des jeux de cartes

 Sons et musique ?

 Statistiques détaillées ?

 Différents modes de jeu ?

 Système de points bonus ?

 Effets visuels ?

 Import/export de flashcards ?

 Thèmes visuels différents

**Here are the features for tetris :**

* Questions appear every 3 to 11 pieces by chance.
* If the player is not able to find the right answer, there's automatically a new line created at the bottom making the game more difficult.
* When the player is answering the question, the game keep going on, pieces are still coming down behind the flashcard, so that if it take too much time to answer the player can loose.
* If the player have choosen “I know” and click on the wright answer, the bottom line disappear. If he’s wrong, a new bottom line is created in the game
* If the player have choosen “I don’t know” and then click on the wright or wrong answer, a new bottom line is created in the game
* En début de jeu, on peut choisir le type de jeu (Tetris ou un des 3 autres) ou ajouter une flashcard (question/réponse) ou modifier des flashcards avec une fonction de recherche qui permet de choisir et modifier la carte de notre souhait.
* Lorsque le joueur clique sur « je ne sais pas », s’il fait ensuite un bon choix dans le qcm, cela n’est pas compté comme bonne réponse et donc on ajoute une ligne et on entre dans l’algorithme que la réponse est fausse.
* Lorsque l’on n’a pas eu le temps de répondre à une question et qu’une nouvelle question apparaît, cela est compté comme une mauvaise réponse pour l’algorithme et l’ajout d’une ligne
* Points : le fait de faire descendre les pièces n’ajoute pas de points. Chaque ligne réalisée ajoute 10 points, tout comme chaque bonne réponse. Supprime l’affichage du nombre de lignes réalisées. Le meilleur score est affiché tout en haut
* Après chaque game over, on peut choisir de recommencer ou de revenir à l’écran d’accueil

**Here are the features for Space invaders :**

* Les questions apparaissent tous les 10 aliens tués
* Chaque mauvaise réponse ajoute toute une nouvelle ligne d'aliens
* Bloquer le tir quand la question est affichée tout en laissant les aliens continuer à bouger normalement sauf si on coche “figer le jeu”
* À partir du niveau 5 ajouter des missiles qui partent sur les côtés (avec une petite flamme derrière eux)
* 5 bonnes réponses d’affilée, une vie gagnée

https://console.firebase.google.com/project/tetris-flashcards/firestore/databases/-default-/data/~2Fflashcards~2F106

Firebase pour flashcards-Tetris

Wrote **dist/index.html**

✔ Wrote configuration info to **firebase.json**

✔ Wrote project information to **.firebaserc**

✔ Wrote **.gitignore**

L’application à modifier est sur le bureau de mon mac

Project Console: https://console.firebase.google.com/project/tetris-flashcards/overview

Hosting URL: https://tetris-flashcards.web.app

Quand je modifie le code, il faut toujours relancer le transfert vers le web de Firebase (de Google) : donc coller dans le terminal :

**npm run build**

firebase deploy --only hosting (mais d’abord il faut avoir été dans le bon répertoire avec la commande : cd Desktop/tetris-flashcards npm run dev)

Mettre à jour la base de donnée à 0 : aller sur la page web > inspecter > console >

apiKey: "AIzaSyBhyyYL\_9kTU-K3hx4ehRfNn4pe9-0RaIo",

authDomain: "tetris-flashcards.firebaseapp.com",

projectId: "tetris-flashcards",

storageBucket: "tetris-flashcards.firebasestorage.app",

messagingSenderId: "171031673545",

appId: "1:171031673545:web:247b28407a1dfeecd6ab18",

measurementId: "G-2DCSFM0218"

On my desk, there's a game in a folder called "tetris Flashcards". It's connected to firebase. Can you make a total check of all the code to try to explain why the game is not responsive and impossible to play on a smartphone ?

Julliet 25

* Lien à partager avec les LLM pour travailler sur mon app : <https://github.com/macotomasi/flashgames-pwa>
* Pour toute future modification :
  + Modifie tes fichiers en local
  + git add .
  + git commit -m "décrire ici ce que j’ai ajouté, modifié dans le code"
  + git push
* Le fait de donner le nom où l’on décrit ce que l’on a change permet de créer un point de sauvegarde qui permet de revenir à une version précédente.
* Ajoute tous les fichiers lourds ou inutiles à .gitignore avant de faire git add .
  + git add .
  + git commit -m "multiples corrections"
  + git push

Mobile Test Checklist

## Performances

- [ ] Tetris 60 FPS sur iPhone 8+

- [ ] Fluidité lors des flashcards

## Safe Areas

- [ ] Aucun élément masqué par le notch (iPhone X+)

- [ ] Padding correct sur iPad Pro

## Gestures

- [ ] Swipe gauche/droite pour déplacer les pièces

- [ ] Pas de conflit avec le scroll natif

## Offline

- [ ] Création de deck/carte offline

- [ ] Jeu et révision offline

- [ ] Persistance après redémarrage

## Tailles d’écran

- [ ] UI OK sur iPhone SE

- [ ] UI OK sur iPad Pro

- [ ] Pas de scroll horizontal