inventeurs du monde numérique

MANUEL D'UTILISATION

Pour le site « Spot the Lesion »

Vivian Losciale

Table des matières

Mise en place du site

Pour pouvoir développer correctement sur le site, il faudra les outils suivants :

- npm
- git

Commencez par cloner le projet avec git en utilisant la commande « git clone ».

Ensuite, à l'intérieur de ce dossier, ouvrez un terminal et tapez la commande « npm install ». Cela aura pour but de télécharger localement toutes les dépendances utilisées pour le projet.

Lorsque les dépendances auront été téléchargées, il vous suffira juste de taper « npm start » pour que le projet se lance et qui vous ouvre une page sur votre navigateur (une page localhost)

Si vous remarquez que vous n'arrivez pas à faire des requêtes avec la base de données, veuillez demander à vos supérieurs de vous donner les accès.

Il vous faudra rajouter ces variables d'environnement sur votre système. Il y en a 2. Une sous le nom de « REACT_APP_FIREBASE_API_KEY » et l'autre sous le nom « REACT_APP_SERVER_KEY »

Si vous avez suivi les instructions, tout devrait fonctionner. Ce qui veut dire… que c'est à votre tour de faire des prouesses !

Architecture du site

Le site est architecturé de la façon suivante :

src
|
|____components
|____firebase
|____hooks
|____language
|____res
|____screens
|____utils

Voici une légère description de chaque dossier :

- component : dossier qui contient des composants React qui sont récurrent dans les différentes pages du site
- firebase : dossier qui s'occupe de la liasion avec la base de donnée pour récupérer les images
- hooks : dossier qui contient des hooks personnalisés. Les hooks sont une notion de React
- language : dossier qui contient toute les traduction du site
- res : dossier qui contient toute les ressources du site, en particulier les images
- screens : dossier qui contient le code de chaque page du site
- utils : dossier qui contient des fonctions, utilisée dans les fichier des pages du site.

Dans le cas de modification de texte ou d'ajout, on modifiera les dossiers « language », « screens » et « res » si ont veut rajouter des images.

Définition du mode aventure

Le mode aventure est un parcours constitué de niveaux. Chaque niveau contient les informations nécessaires pour que celui-ci fonctionne, mode de jeu, objectif de validation de niveau et bien d'autres.

Il est aussi possible d'ajouter une partie médiation.

Vous pourrez trouver le contenu du parcours du jeu de Lésion dans : src/screens/games/Lesions/adventureItems.ts

La structure de donnée pour paramétrer ce mode aventure pour la Lésion est la suivante :

|
|____difficulty
|____gameMode

|____typeLevel
|____levelRequirement
|____requirementToStar1
|____requirementToStar2
|____requirementToStar3

|____numberOfRound
|____<?>mascot
|____theme
|____slide
|____explanation[]
|____text
|____<?>imageSrc

Cette structure correspond à un niveau, donc pour faire 4 niveaux, il faut donc faire une liste de 4 élément comportant cette structure.

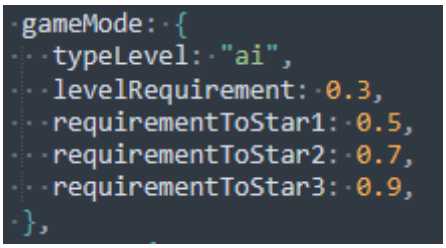
Vous pouvez apercevoir des < ? >. Cela signifie que ces éléments sont optionnels, si on n'en a pas besoin.

Regardons plus en détails chaque argument de cette structure :

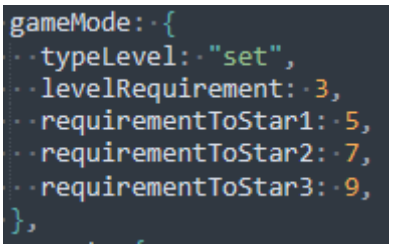
- difficulty : String ->définit le niveau de difficulté du niveau, choix entre « easy,medium,hard »
- typeLevel : permet de définir quel mode de jeu on veut, choix entre « ai » et « fastest, set » « fastest » termine le niveau dès que le score minimal est atteint, alors que « set » va attendre « numberOfRound » tours avant de finir le niveau. « ai » est un combat contre l'IA
- levelRequirement: number -> score minimum pour valider le niveau
- requirementToStar1: number -> score pour atteindre 1 étoile
- requirementToStar2: number -> score pour atteindre 2 étoiles
- requirementToStar3: number -> score pour atteindre 3 étoiles
- numberOfRound: number -> nombre de tour maximal avant la fin du niveau
- theme: number -> correspond au numéro du thème disponible dans la partie « explication »
- slide: number -> correspond au numéro de la slide de l'onglet « theme » défini plus haut
- explanation : tableau regroupant les différents slide à présenter
- text: String -> texte que l'on veut afficher
- imageSrc: String -> lien vers l'image souhaitée que l'on veut afficher

Il y a une spécificité concernant les attributs de scores (ceux contenant le mot « Requirement »).

Lorsque le mode de jeu est « ai », on utilise des pourcentages. Le but est d'avoir X*100% de bonne réponse par rapport à l'IA ou plus, avec X étant une valeur entre 0 et 1.
Par exemple, si on met le « levelRequirement » à 0.5, et que l'IA fait un score de 7, il faut que le joueur fasse au moins un score de 3.5 pour valider le niveau.

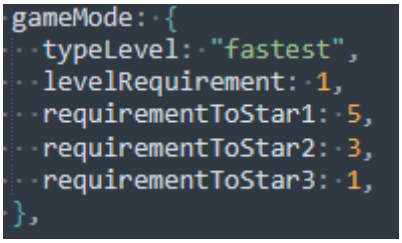


Lorsque le mode de jeu est « set », on spécifie le nombre d'image correcte pour valider le niveau.



Lorsque le mode de jeu est « fastest », on attribue en combien de manche ou doit finir le niveau. Dans ce cas précis, le « levelRequirement » sera égale au nombre de lésions trouvées qu'il faut avoir pour finir le niveau et les « requirementToStar » représentent en combien de tours on a réussi à trouver les « levelRequirement » lésions.

Ici, il faut trouver 1 lésion pour valider le niveau. On peut avoir 1 étoile si on a validé le niveau en 5 tours ou moins, 2 étoiles en 3 tours ou moins, et 3 étoiles en 1 tour (correspond a trouver la lésion dès la première image)



Ainsi, en modifiant cette structure, les modifications sur le site se feront tout seul. Si vous ajoutez un niveau dans un parcours, vous n'avez qu'a rajouter un niveau dans la structure montrée et tout se générera tout seul.

Ajouter un contenu de traduction

Dans tous les cas présentés ci-dessous, il faudra aller modifier les fichiers dans le dossier « language » pour pouvoir les traduire dans les différentes langues.

Dans le mode aventure

Dans le cas où vous voulez ajouter du contenu de médiation, il suffit d'ajouter l'attribut « mascot » dans l'un des niveaux pour le faire apparaître. Il faudra ensuite créer le nombre de slide que l'on veut mettre pour cette partie médiation.

La structure pour les explications est la suivante :

|____theme
|____slide
|____explanation[]
        |____text
        |____<?>imageSrc

Cette structure représente donc ce que la mascotte va dire sur le niveau choisi, voyons en détails les attributs :

- theme: numero du pour la référence sur le bouton « en savoir plus »
- slide : numero de slide pour la référence sur le bouton « en savoir plus »
- explanation: tableau contenant les slides que l'on veut représenter
- text : **LABEL** du texte à afficher
- imageSrc : image à afficher

Étant donné que le site contient plusieurs langues, on ne peut pas écrire dans ce fichier une phrase dans une langue spécifique, on va donc seulement spécifier un label, et ce sera dans le fichier LesionGame.json se trouvant dans src\language\FR (si FR) qu'il faudra ajouter le label en question pour y mettre la phrase. Il faudra aussi le faire dans les autres langages.

```
mascot: {
    theme: 0,
    slide: 1,
    explanation: [
        {
            text: "Test1-1",
        },
        {
            text: "Test1-2",
            imageSrc: doctor,
        },
        {
            text: "Test1-3",
        },
        {
            text: "Test1-4",
            imageSrc: lesion,
        },
        {
            text: "Test1-5",
            imageSrc: lesionfound,
        },
    ],
},
```

```
"Test1-1": "Bonjour, je suis Numera. Aujourd'hui, j'ai besoin de ton aide !",
"Test1-2": "Il faut que tu m'aides à trouver les lésions sur ces images. Tu ne sais pas ce que
"Test1-3": "Une lésion est une altération d'un organe, cela peut être causé par une multitude de
"Test1-4": "Cette photo contient une lésion, mais la retrouver requiert un certain savoir.",
"Test1-5": "Pour trouver la lésion, il faut trouver une zone qui a une couleur différente que le
```

Si vous souhaitez ajouter tout un parcours, il faudra faire une référence du fichier ts crée dans le fichier index.ts se trouvant dans \src\screens\games

A noter qu'il faudra faire un système pour référencer le jeu en relation avec le parcours, chose qui n'a pas été implémenté. Actuellement, on peut référencer de nouveaux parcours, mais si on teste ces parcours, on ne pourra faire que le jeu de la Lésion. (voir fichier GameRoute.tsx pour modifier cela)

Dans la partie explication

Pour cela, il faut aller dans src\screens\explanation\explanationText et choisir le thème dans lequel on veut ajouter des informations.

La structure pour les explications est la suivante :

|____title
|____body[]
        |____text
        |____<?>imageSrc

Cette structure représente donc un slide du thème choisi, voyons en détails les attributs :

- title : titre du slide
- body : tableau contenant les paragraphes que l'on veut représenter
- text : **LABEL** du texte à afficher
- imageSrc : image à afficher

Étant donné que le site contient plusieurs langues, on ne peut pas écrire dans ce fichier une phrase dans une langue spécifique, on va donc seulement spécifier un label, et ce sera dans le fichier explanation.json se trouvant dans src\language\FR (si FR) qu'il faudra ajouter le label en question pour y mettre la phrase. Il faudra aussi le faire dans les autres langages.

```
{
  title: "IA1Title",
  body: [
    {
      text: "IA1Text1",
      imageSrc: ai,
    },
    {
      text: "IA1Text2",
    },
  ],
},
```

```
"IA1Title": "L'intelligence artificielle, c'est quoi ?",
"IA1Text1": "L'intelligence artificielle, ou aussi appelée IA, repose sur l'utilisation d'algor
"IA1Text2": "Utilisée partout, elle aide les humains, mais l'IA ne dit pas la vérité absolue. N
```

Dans le cas où on souhaite créer un nouveau thème (et non un slide), il faudra faire une référence du fichier crée dans le index.ts se trouvant dans src\screens\explanation

Modifier un contenu de traduction

Dans le mode aventure

Pour modifier des phrases du jeu Lésion, il faut modifier le fichier LesionGame.json se trouvant dans src\language\X avec X la langue. Dans le futur, lorsqu'il y aura plusieurs jeux, il faudra modifier le fichier correspondant.

Dans la partie explication

Pour modifier des phrases de la section explication, il faut modifier le fichier explanation.json se trouvant dans src\language\X avec X la langue.

Sur une page du site

Pour modifier des phrases de la section explication, il faut modifier le fichier website.json se trouvant dans src\language\X avec X la langue. S'il s'agit d'un mot récurrent, tel que « oui », « non », alors c'est dans le fichier common.json qu'il faudra faire les modifications.

Créer de nouvelles pages

Il faut créer un fichier dans le dossier screens et implémenter le code pour la page.

Déployer le site

Il existe deux solutions selon le serveur sur lequel vous voulez le déployer.

Avant de commencer les indications qui suivent, vous devrez changer la variable « homepage » se trouvant dans le fichier « package.json » se trouvant à la racine du projet avec le lien du site cible.

Le serveur a les outils adéquats

Si le serveur possède npm et git, alors vous n'avez qu'à faire les mêmes manipulations que pour initialiser le projet.

Il faudra juste maintenir le terminal dans lequel vous avez écrit la commande en marche constante. Il existe beaucoup de façon sur Linux pour laisser un terminal tourner même si vous vous déconnectez de votre session (screen, standalone.). Voir plus d'information sur internet.

Le serveur est obsolète

Si le serveur ne peut pas contenir les outils pour le faire tourner normalement (npm et git), alors il vous faudra déployer le site sous forme statique.

Pour cela, vous devrez taper dans un terminal la commande « npm run build ».

A la fin de celui-ci, vous aurez un dossier build qui contiendra le site en question. Il vous suffit donc de le déployer sur le serveur, soit par ligne de commande, en utilisant scp, ou bien avec un logiciel.

Lorsque vous ferez des modifications, il faudra par conséquent redéployer en refaisant les mêmes étapes décrites au-dessus.