



Test technique Développeur·se Node/React

Contexte et objectifs

Il est maintenant temps de démontrer ta compétence technique en t'amusant à développer un mini-jeu ! Cette étape nous permettra bien entendu de valider ton niveau de technicité, mais surtout ta capacité d'adaptation.

Ce test technique a deux objectifs :

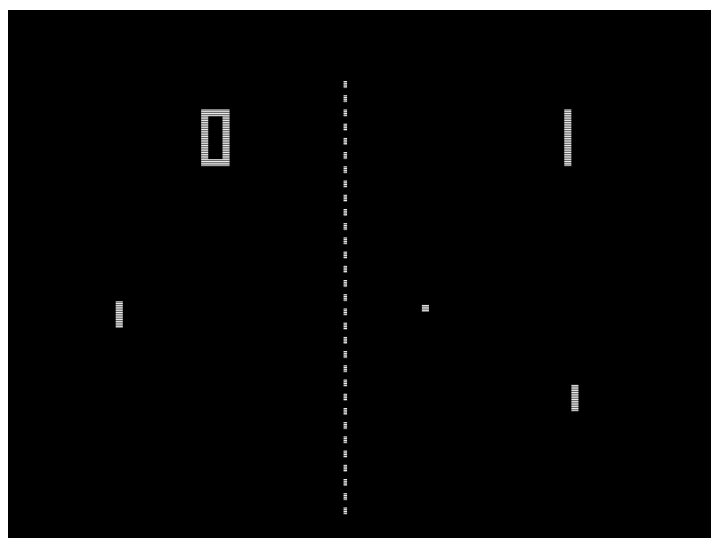
- Vérifier que tu sais t'adapter à notre stack actuelle. L'intérêt de ce test ne réside pas dans la difficulté technique mais dans la variété de technos à utiliser.
- Nous assurer qu'en plus de maîtriser ces outils, tu es à l'aise à documenter ton travail, afin de pouvoir déléguer le travail à l'équipe de dev et les faire progresser.

Une fois réalisé avec succès, ton chemin vers O'clock se conclura par un entretien en visio avec Guillaume, notre directeur technique, et Simon, notre lead dev.

Finalement le plus important, ça sera le feeling. Pour nous mais surtout pour toi, pour que tu puisses t'épanouir dans cette nouvelle aventure !

Spécifications

Le but est de créer un mini-jeu Pong.





Pour rappel, voici les règles :

Le jeu est inspiré du tennis de table en vue de dessus, et chaque joueur s'affronte en déplaçant la raquette virtuelle de haut en bas, de façon à garder la balle dans le terrain de jeu. Le joueur peut changer la direction de la balle en fonction de l'endroit où celle-ci tape sur la raquette, alors que sa vitesse augmente graduellement au cours de la manche. Un score est affiché pour la partie en cours.

Parce qu'une vidéo vaut mille mots, tu peux trouver [un exemple de jeu ici](#).

Résultat attendu

- Une application web construite avec React qui permette à deux joueurs de jouer en local : le joueur 1 utilise les touches **A** et **Q** du clavier ; le joueur 2 utilise les touches **↑** et **↓** du même clavier.
- L'application est capable d'afficher les scores des matchs joués à travers une API.
- L'application doit être accessible, autant que possible.
- L'API, construite avec Node.JS, doit être hébergée en ligne (tu peux utiliser le *free tier* de AWS si tu n'as pas d'hébergement).

Bonus

En fonction du temps dont tu disposes pour aller plus loin, tu peux essayer de rajouter :

- un système de tri des scores, par joueur ou par score
- [des sons](#)
- l'API GamePad, pour jouer avec une manette !

Livraison

Le jeu doit être disponible et jouable depuis une URL. Tout le code front et back doit être pushé sur un dépôt sur GitHub (dépôt public, ou dépôt privé en nous donnant un accès admin).

Ce dépôt doit comporter tout ce qu'il faut pour permettre à une équipe de développeurs de comprendre, de reprendre et maintenir le code. A toi de voir la meilleure façon d'y parvenir : README.md, commentaires dans le code, des tests unitaires...

Tu as **7 jours** pour réaliser ce test. Bon courage !