



# Test technique Full Stack

## Contexte et objectifs

Il est maintenant temps de démontrer ta compétence technique en s'amusant à développer un petit jeu :)

Cette étape nous permettra de valider ton niveau de technicité. Une fois réalisé, ton chemin vers l'École O'clock sera déjà bien avancé. Le Baptême de l'Espace et l'entretien en visio, viendront alors ponctuer notre processus de recrutement.

Ce test technique a deux objectifs :

- vérifier que tu as une connaissance suffisante en développement web pour pouvoir animer la formation délivrée par l'École O'clock et intervenir sur l'évolution et la maintenance de nos outils pédagogiques.
- nous assurer que, même en codant, tu gardes bien en tête la finalité pédagogique (coder proprement, de manière facile à comprendre, en commentant, etc.). Pour t'aider à atteindre cet objectif, dis-toi que ton mini-projet pourra être fourni aux étudiants au cours de la formation afin qu'il leur serve de base pédagogique pour d'autres réalisations.

## Spécifications

Le but est de créer le jeu Memory.

## Fonctionnalités

- Au commencement du jeu, des cartes sont disposées face cachée à l'écran.
- Le joueur doit cliquer sur deux cartes. Si celles-ci sont identiques, la paire est validée. Sinon, les cartes sont retournées face cachée, et le joueur doit sélectionner une nouvelle paire de cartes.
- Un compteur de temps, avec une barre de progression, s'affiche en dessous du plateau.
- Le joueur gagne s'il arrive à découvrir toutes les paires avant la fin du temps imparti.
- Chaque temps de partie effectuée doit être sauvegardée en base de données. Avant le début du jeu, les meilleurs temps s'affichent à l'écran.



## Résultat attendu

- **Créer le jeu en HTML / CSS / JS.**

La répartition des fruits doit être aléatoire à chaque jeu.

précisions **CSS** : SASS ou autre préprocesseur encouragé.

précisions **JS** : L'utilisation d'une librairie (au hasard, jQuery) est acceptée, pour faciliter la gestion d'événements et les modifications du DOM.

- **Faire la persistance des données côté back : PHP ou Node.js**

Choisis le langage *qui te permettra le mieux de répondre à l'exercice*.

Ton application doit être codée en *objet*.

N'hésite pas à commenter ton code, et en faire un support de cours pour les étudiants, notamment pour la découverte des bases de données.

- **Cartes à jouer**

<https://static.oclock.io/challenges/tests-techniques/cards.png>

- **Charte graphique**

Pas de charte graphique imposée. Tu peux partir du design des exemples ci-dessus, ou improviser si tu sens l'inspiration pointer le bout de son nez. Quoi qu'il en soit, le code CSS doit être compréhensible et abordable.

- **Déploiement *optionnel***

Si tu connais un outil de déploiement (au hasard, Docker), et que tu es inspiré, n'hésites pas à proposer aussi cet aspect dans le projet, ce sera très intéressant pour nos étudiants plus avancés ;)

## Livraison

Le code doit être pushé sur un repo public sur GitHub.

## Exemples

Les captures d'écran représentent les situations suivantes :

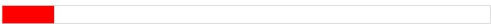
1. Le plateau de départ
2. Le plateau pendant le jeu
3. On a gagné !
4. On a perdu...



Jeu de mémoire


Jeu de mémoire

		🍓				
			🍓		🍌	
	🍊					
				🍊		



Cette page indique :  
Vous avez gagnééééééééééééé !

OK

🍊	🍏	🍊	🍋	🍋	🍇	🍉
🍊	🍏	🍇	🍎	🍊	🍓	🍏
🍇	🍏	🍓	🍉	🍎	🍏	🍏
🍌	🍇	🍊	🍏	🍊	🍏	🍌

Cette page indique :  
Vous avez perduuuuuu !

OK

			🍇			
				🍇		
			🍏			

