

# Exercice 1 : Motus

## Introduction

Le jeu télévisé Motus oppose des candidats qui doivent trouver des mots d'un certain nombre de lettres. A l'origine, il était question de mots de 5 lettres mais cela a ensuite été augmenté.

Pour trouver ce mot, ils doivent proposer des mots de la taille en question. Les mots proposés doivent exister. L'affichage leur indiquera quelles sont les lettres correctes et si elles sont à la bonne position ou pas.

[https://fr.wikipedia.org/wiki/Motus\\_\(jeu\\_t%C3%A9l%C3%A9vis%C3%A9\)](https://fr.wikipedia.org/wiki/Motus_(jeu_t%C3%A9l%C3%A9vis%C3%A9))

## Le Programme

Le programme commence par tirer un mot au hasard et initialiser le tableau des mots proposés à un tableau vide.

L'écran de jeu vous affiche les mots déjà proposés, en indiquant les lettres correctes et à leur place ou correctes mais mal placées. Un champ texte permettra de proposer un mot.

Il faudra alors vérifier si le mot existe. Si le mot est trouvé, on ira sur l'écran "gagné". Il faudra ensuite déterminer les lettres correctes et à leur place et puis, vérifier pour les lettres restant si elles sont présentes On sauvera les résultats dans un tableau à part avant de revenir à l'affichage des propositions

## Détails

### Le dictionnaire

Un fichier fr-motus.php contient une liste de 4571 mots français de 5 lettres<sup>1</sup>. Quand on l'inclus, il défini une variable \$fr qui contient le tableau de ces mots. On peut déterminer le nombre de mots à l'aide de la fonction count()<sup>2</sup>

Pour tirer un mot au hasard, on utilisera la fonction rand()<sup>3</sup>.

### Afficher les propositions

Dans le jeu d'origine, le nombre de propositions était limité. Ici, on continue la partie jusqu'à ce que le mot soit trouvé. De plus, dans le jeu d'origine, la première lettre était fournie. Ici, elle ne le sera pas.

On disposera de deux tableaux. Le premier contiendra les mots et le second contiendra le statut des lettres (correct,mauvaise place,incorrect).

---

1 Cette liste est extraite du vocabulaire français lié au projet Gutenberg

2 <https://www.php.net/manual/en/function.count.php>

3 <https://www.php.net/manual/fr/function.rand.php>

Les propositions seront affichées dans un tableau, lettre par lettre. On disposera de 3 classes CSS pour les TD selon le statut de la lettre.

Pour accéder à la  $i^{\text{ème}}$  lettre d'une chaîne, on utilise la même notation que pour les tableaux :  `$\$s[\$i]$` .

## Déterminer les lettres correctes

Il faudra comparer le mot entré avec le mot tiré au hasard en début de partie (dans le start.php). Pour cela, on prend des copies des deux mots et on va faire deux passages.

Lors du premier passage, si la lettre est la même dans les deux mots, on la remplace dans les chaînes (rappel : la copie des chaînes) par un caractère "neutre" différent dans les deux chaînes (par exemple un # et un \*). Cela permettra de ne plus les prendre en compte lors du 2eme passage.

Le deuxième passage comparera chaque lettre du mot proposé avec toutes les lettres du mot à trouver. Les lettres déjà repérées dans la première étape auront été transformées et la comparaison avec ces lettres échouera. Cela permettra d'éviter de compter 3 fois le E de EPEES.