

Développement Web 1

Travail 2

Pondération : 12%

Remise : lundi 5 avril minuit

Le but de ce travail est de mettre en application les notions apprises au niveau du javascript et des sites 'responsive'.

Directives générales

- Le travail doit se faire de façon individuelle ou en équipe de 2 personnes.
- Le travail consiste à programmer un petit jeu d'images à trouver. Je vous ai donné tout le code html et css permettant de faire afficher l'interface.
Vous devez comprendre le code html et le code css fourni afin de bien programmer le fonctionnement demandé.
- Je vous ai aussi donné une partie du fonctionnement en javascript. Vous devez comprendre le code présent.
- **Vous NE devez PAS ajouter de id ou de class** dans le code html.

À faire

Rappel : Testez chaque instruction au fur et à mesure, cela rendra le débogage plus facile.

Déterminer le niveau du jeu

Trois niveaux de jeu sont possibles. Le niveau détermine le nombre d'images différentes à utiliser. Le niveau facile doit utiliser 5 images seulement, le niveau intermédiaire en utiliser 10 et le niveau difficile en utiliser 15.

- Complétez *GenererJeu()* afin de déterminer le nombre d'images à afficher (déclarée en variable globale).
Note : La fonction *Initialiser()* fait déjà le lien entre les clics sur les boutons et la fonction *GenererJeu()*.
- Faites aussi afficher le nom du niveau (facile, intermédiaire ou difficile) dans le bon paragraphe de la division 'infoTour'.

La fonction appelle aussi 5 autres fonctions, qui seront à compléter tel qu'expliqué plus bas.

- *DisparaitreNiveaux()* qui fait disparaître les boutons de niveaux.
- *GenererImageATrouver()* qui doit déterminer l'image à trouver.
- *GenererLesImages()* qui doit déterminer toutes les autres images.
- *DebuterChrono()* qui part le chronomètre.
- *VerifierToutesTrouvees()* qui vérifie s'il reste des images identiques à celle à trouver.

Générer l'image à trouver

Regardez les images contenues dans le dossier *images*. Toutes les images portent le même nom, suivi d'un chiffre.

Complétez *GenererImageATrouver()* :

- Générez un nombre aléatoire en utilisant le nombre d'images différentes à utiliser.
Ex : un chiffre entre 1 et 5 pour le niveau facile
- Formez le bon nom d'image à utiliser. Attention, aucune image ne porte le chiffre 0. De plus, il ne faut pas utiliser de if pour créer ce nom.
Ex : si le chiffre généré est de **3** alors le nom de l'image est emoji**3**.png
- Faites afficher l'image correspondante au bon endroit (id 'imageATrouver').

Générer toutes les images à afficher

Présentement, l'interface présente 30 images vues de dos. Il faut maintenant générer et faire afficher les images sur lesquelles il faudra cliquer pour trouver toutes celles identiques à l'image à trouver. Le niveau facile doit faire afficher 10 images seulement, le niveau intermédiaire en faire afficher 20 et le niveau difficile en faire afficher 30. Complétez *GenererLesImages()* de la façon qui suit :

- Il faut procéder de la même façon que pour l'image à trouver pour générer chacune des images. Les images doivent s'afficher une à la suite de l'autre dans la division 'imagesCliquables'.
- Étant donné qu'il y a plus de balises qu'il y a d'images à afficher (niveau facile et niveau intermédiaire) alors il faut faire disparaître les images en trop. Vous pouvez utiliser 'display :none' ou 'visibility :hidden'.
- Puisque le tout est généré au hasard, il est possible que le hasard fasse en sorte qu'il n'y ait aucune image identique à celle à trouver. Pour cette raison, la fonction *VerifierToutesTrouvees()* est appelée à la fin de *GenererJeu()*.

Clic sur une image

Le but du jeu est de sélectionner toutes les images identiques à celle à trouver. Il suffit de cliquer sur une image pour la sélectionner. Regardez la fonction *Initialiser()*, elle relie toutes les images à la fonction *VerifierImage()*. Il faut donc la compléter de la façon suivante :

- Pour déterminer si l'image sur laquelle on vient de cliquer est identique à celle à trouver, il suffit de comparer la source des 2 images.
- Lorsque les 2 images sont pareilles, il faut mettre la bordure de l'image en rouge afin d'avoir un aspect visuel indiquant que cette image est sélectionnée.
- Il faut aussi modifier le nombre d'images trouvées et le faire afficher au bon endroit dans la division 'infoTour'.

Attention, un clic sur une image déjà sélectionnée ne doit pas augmenter le nombre d'images trouvées.

Vérifier si toutes les images ont été trouvées

- La fonction *VerifierToutesTrouvees()* doit **compter** s'il reste des **images identiques** (même source) à celle à trouver et qui n'ont **pas** été **sélectionnées** (bordure différente de rouge).
- Le code présent permet d'arrêter le chronomètre et de faire apparaître les boutons qui permettent de choisir le niveau de difficulté lorsque toutes les images sont trouvées.

Recommencer à jouer

Lorsque toutes les images sont trouvées, l'utilisateur a la possibilité de recommencer à jouer puisque les boutons réapparaissent. Il faut alors :

- Augmenter le nombre de tours effectués et le faire afficher au bon endroit.
- Remettre le nombre d'images à trouver à 0 et le faire afficher.
- Remettre le chronomètre à 0 et le faire afficher.

Suggestion : placez ce code dans *GenererJeu()*.

Rendre le site 'responsive' de la façon suivante

Lorsqu'on redimensionne l'écran, les éléments s'adaptent déjà un peu car :

- Les boutons du menu sont flottants
- Une image est une balise 'inline', les images s'affichent donc une à côté de l'autre automatiquement
- max-width est utilisé pour le contenu donc il s'adapte à la largeur de l'écran

Il faut quand même améliorer le tout de la façon qui suit :

- Un 1^{er} breakpoint doit être ajouté à 768px (largeur standard selon le w3schools). Appliquez des règles afin que :
 - o Le header et les titres prennent moins de place.
 - o Le menu prenne moins de place.
 - o La partie des informations prenne moins de place.
- Un 2^{ème} breakpoint doit être ajouté à 600px (autre largeur recommandée par le w3schools) :
 - o Faites alors disparaître toute la partie contenant les informations.
 - o Rapetissez un peu les images.

Répartition des points

- Gérer début et fin du jeu **20 pts**
- Générer les images **30 pts**
- Vérifier le clic et si toutes les images sont trouvées **25 pts**
- Règles responsive **25 pts**