

Qui Est-Ce ?

Objectif : Système de vérification d'erreurs en PHP

Nous souhaitons mettre en place un système de vérification d'erreurs en PHP sous forme d'un jeu basé sur le célèbre "Qui est-ce ?". Le jeu comporte 16 personnages, chacun ayant ou non les attributs suivants : Lunettes, Moustaches, Chapeau, Cheveux, Boucle d'oreille, Barbe et Noeud papillon. Chaque personnage est codé en binaire, avec 1 pour un attribut présent et 0 pour un attribut absent, pour un total de 7 bits par personnage.

Étapes de mise en œuvre

1. Création de la page d'accueil

La première étape consiste à créer une page d'accueil qui contient les images des personnages ainsi qu'un formulaire de questions. Le formulaire comportera 7 questions, chacune correspondant à un attribut des personnages. Les images seront placées dans un dossier appelé **Personnages**.

Nous allons créer un tableau **\$personnages** dans le fichier **variables.php** pour stocker les noms des images des personnages. Nous utiliserons ensuite une boucle **foreach** pour parcourir ce tableau et afficher chaque image avec la balise ****, en récupérant les images à partir du dossier **Personnages**.

2. Mise en place du formulaire de questions

Le formulaire sera composé de boutons radio permettant à l'utilisateur de répondre "Oui" ou "Non" à chaque question. La valeur de chaque bouton sera soit 1 (pour "Oui"), soit 0 (pour "Non"). Les réponses seront envoyées par la méthode POST vers un fichier **check.php**, qui traitera les données.

Les réponses seront stockées dans un tableau **\$persos[]**. En bas de la page, un lien sous forme de bouton permettra de rejouer et de rafraîchir la page.

3. Vérification des erreurs

La gestion des erreurs dans ce jeu consiste à détecter un "mensonge" du joueur, c'est-à-dire si l'image sélectionnée par le joueur n'existe pas dans le tableau **\$personnages**.

Dans le fichier **check.php**, nous commencerons par vérifier si le formulaire a bien été soumis et si toutes les questions ont reçu une réponse (7 réponses au total).

Pour ce faire, nous utiliserons la variable **\$_POST** et vérifierons la présence des données nécessaires.

Ensuite, pour vérifier si l'image sélectionnée existe dans le tableau **\$personnages**, nous utiliserons la fonction **in_array()**. Si l'image est trouvée, un message de succès sera affiché, et le personnage sera stocké dans **\$_SESSION["PERSO"]** afin de le mettre en surbrillance en rouge sur la page d'accueil.

Si le personnage n'est pas trouvé, cela indique un mensonge. Pour détecter et corriger l'erreur, nous utiliserons un calcul des syndromes. Il y a trois syndromes à calculer :

- **S1** : Bit à chaque position impaire (1er, 3e, 5e, 7e).
- **S2** : Bit à chaque paire de bits sans compter le premier.
- **S3** : Les 4 derniers bits.

Pour chaque syndrome, nous comptons le nombre de bits égaux à 1. Si ce nombre est pair, le syndrome est 0, sinon il est 1. Ces résultats sont comparés à un tableau de syndromes, ce qui nous permet de déterminer sur quel bit se trouve l'erreur.

	1	2	3	4	5	6	7
S1	1	0	1	0	1	0	1
S2	0	1	1	0	1	1	0
S3	0	0	0	1	1	1	1

4. Correction du mensonge

Une fois l'erreur détectée, nous récupérons l'index du bit erroné et nous inverserons la valeur de ce bit (de 0 à 1 ou de 1 à 0). Un message d'avertissement est alors ajouté dans **\$_SESSION["NOT_EXIST"]**, et le personnage corrigé est stocké dans **\$_SESSION["PERSO"]**.

5. Mise en surbrillance et messages

Sur la page d'accueil, nous appliquons du CSS pour entourer en rouge le personnage choisi par le joueur en vérifiant la présence de `$_SESSION["PERSO"]`. De même, nous affichons les messages de succès ou d'avertissement selon le cas.

Conclusion

Ce projet permet de créer un jeu interactif "Qui est-ce ?" tout en intégrant un système de vérification d'erreurs pour détecter les mensonges du joueur, grâce à un calcul basé sur les syndromes. Ce système permet de corriger automatiquement l'erreur et d'offrir une expérience utilisateur plus fluide et engageante.