Rapport
Programme de Mini-jeux en PHP

Anthony ROBIN

11 décembre 2012

Sommaire du rapport

Introduction

Ce rapport rédigé en Latex va permettre d'expliquer mes démarches à la réalisation du programme de mini-jeux en PHP. Le but de ce projet étant de développer le mini-jeu du nombre mystère, du mastermind, ainsi que d'un jeu au choix. J'ai pour ma part choisi, en guise de jeu au choix de faire un "vrai ou faux" ainsi qu'un "questions/réponses".

Chapitre 1

Le Programme

1.1 Le contrôleur

Le controleur est le fichier de base qui sert à lancer les différents mini-jeux. Je l'ai nommé index.php.

Dans ce fichier on déclare une nouvelle instance de la classe Joueur, qui permet de récupérer le nom du joueur. Ensuite se trouve le menu principal du programme qui, grâce à un switch, récupère le choix de l'utilisateur.

Pour chacun des jeux, je déclare une variable $\$nomDuJeu$, une variable $\$regles$ qui contient les instructions propres à chaque jeu, puis j'instancie la classe du jeu auquel je souhaite jouer. Je lance ensuite la méthode "message-Bienvenue" en passant en paramètres les règles, le prénom du joueur et le nom du jeu de façon à pouvoir récupérer ces informations dans la suite du programme.

1.2 Classe Joueur

La classe Joueur contient uniquement une méthode, celle qui récupère le pseudo du joueur. Vu la simplicité de ce programme il n'est pas nécessaire d'obtenir d'avantages d'informations sur le joueur mais si le jeu avait été réalisé en HTML, il aurait pû être intéressant de demander d'avantage d'informations et de tout sauvegarder dans une base de données.

1.3 Classe Jeu

Ma classe Jeu est la classe parente à tous mes mini-jeux. Elle est définie en abstraite ce qui veut dire que l'on ne peut pas instancier directement cette classe Jeu.

Chacun des mini-jeux "extends Jeu", ce qui veut dire que je peux réutiliser les méthodes définies dans la classe mère. De plus j'y définis toutes les méthodes qui se retrouvent pour chacun des jeux dans le but déviter d'avoir à les réécrire dans chacun d'entre eux (les méthodes "messageBienvenue", "reglesDuJeu" et "replay" par exemple).

1.4 Les différents jeux

1.4.1 Méthodes communes aux différents jeux

Chaque jeu est constitué de deux méthodes :
- une nommée "niveauDifficultes" qui définit le choix du niveau de difficultés (à travers un switch)
- une deuxième nommée "debutJeu" qui contient tout le jeu à l'intérieur d'un "do... while" de façon à ce que si la personne souhaite rejouer elle reprenne au début de la boucle.

Pour chaque jeu, je déclare des variables en début de classe pour pouvoir les réutiliser dans toutes les méthodes de la classe à l'aide de $\$this->$

1.4.2 Classe Nombre Mystère

niveauDifficultes

A travers le switch, je déclare une variable $\$nombreMax$ qui servira a déterminer le nombre maximum que l'utilisateur pourra avoir à trouver. J'ai choisi de définir 100 pour le niveau facile, 500 pour l'intermédiaire et donner la possibilité au joueur de choisir ce nombre max en expert tout en testant la valeur rentrée : si elle est inférieure à 500 un message invitera le joueur à se rabattre sur le niveau intermédiaire.

debutJeu

Cette méthode contient le programme du jeu : elle demande à l'utilisateur de choisir un nombre puis le programme effectue une vérification par rapport au nombre mystère. A chaque tour de boucle, j'enregistre dans une variable $\$max$ et $\$min$ la valeur qui a été saisie par l'utilisateur pour restreindre la plage possible.

A chaque tour de boucle, j'incrémente également une variable $\$nbEssai$ qui totalisera le nombre de coups nécessaires pour trouver le nombre mystère.

J'ai défini ma propre méthode de calcul du score en fonction du nombre de coups.

Une fois que l'utilisateur a trouvé ce nombre, j'enregistre le résultat dans un fichier txt en passant en paramètres le prenom du joueur, le nom du jeu, le score obtenu ainsi que la difficulté choisie). Le fichier de score porte le même nom que le jeu.

1.4.3 Classe MasterMind

niveauDifficultes

A travers le switch, je déclare une variable $\$nombreMax$ qui servira a déterminer le nombre maximum que l'utilisateur pourra avoir à trouver. J'ai choisi de définir 4 pour le niveau facile, 7 pour l'intermédiaire et 9 pour le niveau expert.

debutJeu

A l'intérieur d'un "do...while", j'ai écrit tout le code propre au jeu. Je définis l'intervalle des valeurs à trouver en prenant en compte la valeur transmise au moment du chois de la difficulté. Puis pour chacun des 4 nombres de la combinaison, je pioche au hasard dans mon tableau une valeur. Pour éviter de tomber sur des doublons, je retire du tableau la valeur qui vient d'être choisie.
Il ne reste plus qu'à faire un test sur les valeurs proposées par l'utilisateur et la combinaison mystère.

A chaque essai du joueur, j'affiche le nombre de :
- valeurs bien placées
- valeurs mal placées
- valeurs qui ne sont pas dans la combinaison.

Puis je définis un score en fonction du nombre d'essais et je passe toutes ces informations à la méthode "ecrireResultatdansTXT" qui écrira dans le fichier de score.

1.4.4 Classe Vrai Ou Faux

niveauDifficultes

A travers le switch, je déclare une variable $\$nombreMax$ qui servira a déterminer l'intervalle maximum que l'utilisateur pourra avoir à trouver. J'ai choisi de définir 10 pour le niveau facile, 50 pour l'intermédiaire et 100 pour le niveau expert. Je déclare aussi une variable $\$nombres$ qui indiquera combien de nombres seront additionnés.

debutJeu

Tout le programme est contenu dans un "do...while". Pour chacun des nombres, je génére un nombre aléatoire compris entre 1 et $\$nombreMax$ pour l'ordinateur. Puis je les additionne. Je refais la même opération avec les nombres qui seront pour le joueur.
Ensuite je demande à l'utilisateur de saisir "vrai" ou "faux" à la question posée par le programme.
A chaque réponse, le programme demande à l'utilisateur s'il souhaite rejouer. Si oui, le programme reprend au début de la boucle.

1.4.5 Classe Questions Réponses

niveauDifficultes

A travers le switch, je déclare une variable $\$fichierAOuvrir$ qui définit le nom du fichier à aller chercher. J'ai créé un fichier txt contenant les questions et les réponses différentes par niveau de difficultés. Il y a en tout 5 questions par fichier.

debutJeu

Je commence par ouvrir mon fichier contenant les questions et réponses. Je le parcours pour chacune des lignes, je récupère le premier indice (la question) et le deuxième indice (la réponse). Il ne reste plus qu'à faire une vérification sur ce que va entrer l'utilisateur et voir si la réponse correspond au deuxième indice. Les scores s'additionnent à chaque partie et dès que l'utilisateur souhaite quitter le jeu, le score final est envoyé dans la méthode "ecrireResultatdansTXT".

1.4.6 Classe Calculatrice

Cette classe fait plus office de gadget que de véritable jeu. L'utilisateur choisi au début quelle opération il souhaite effectuer (addition, soustraction, ...) puis indique les 2 nombres qu'il veut manipuler. Le programme s'occupe alors de faire l'opération.
Aucun score n'est géré dans cette mini-application.

1.5 Classe Score

4 méthodes :

- "menuScore" qui affiche un menu pour sélectionner le jeu dont on veut afficher le score.
- "lireLesScores" qui lit les fichiers de score en prenant en paramètre le nom du jeu et ouvre le fichier correspondant au jeu.
- "ecrireResultatDansTXT" qui écrit dans un fichier TXT les résultats du jeu.
- "supprimerResultatDansTXT" qui permet de supprimer et réinitialiser tous les scores.

Menu général du site :

L'élément "Afficher les scores" n'est affiché qu'à partir du moment où un score a été généré. De la même façon, seuls seront accessibles en lecture les scores qui auront déjà été créés par le programme.

Affichage des scores :

Chaque score affiche le nom du joueur, le niveau de difficulté, le score obtenu, le jeu auquel il a joué ainsi que la date (en français).

Suppression des scores :

Il est également possible de supprimer tous les scores qui ont déjà été enregistrés par le programme. Cela se passe en 2 étapes : d'abord je supprime tous les fichiers qui sont à l'intérieur du dossier puis enfin je supprime le dossier des scores.

Chapitre 2

Difficultés

2.1 L'encodage en UTF-8

J'ai développé ce programme de mini-jeux sur un ordinateur avec Windows. Ayant des conflits avec les accents dans la console windows, j'ai décidé de tous les retirer, ce qui explique qu'il n'y en ait aucun dans le programme.

2.2 Les scores

Je n'ai pas réussi à gérer les 10 meilleurs scores de chaque jeu à faire en sorte d'écraser le moins bon à partir de la $11^{ème}$ partie.

2.3 Lecture du fichier des Questions-Réponses

Dans le jeu du Question Réponses, je lit le fichier txt contenant les questions puis pour chacune des lignes je pose la question. Malheureusement, une fois le fichier lu en totalité, la question retourne à la première. J'ai n'ai pas réussi à résoudre ce problème.