Mini-projet 2018-2019 DUT 2ème année

Développement du jeu *Bridges* en php exclusivement

1) Description du jeu

Vous développerez un jeu **en php uniquement,** nommé Bridges dont les règles sont données ici : https://www.chiark.greenend.org.uk/~sgtatham/puzzles/js/bridges.html
Profitez en pour vous entraîner !
L'objectif du jeu est de relier des villes, symbolisées par des cercles contenant un nombre, par des ponts qui sont soit verticaux ou soit horizontaux. Le nombre de ponts qui touchent une ville doit être strictement celui qui est mentionné dans le cercle (les villes dans votre version du jeu peuvent être autre chose que des cercles bien sûr). Il ne peut pas y avoir plus de deux ponts qui relient 2 villes. En plus, dans les régles du jeu, les ponts ne peuvent pas se croiser et toutes les villes à la fin du jeu doivent être connectées. Nous levons ces deux contraintes pour faciliter le développement.
La partie est gagnée lorsque tous les ponts ont été découverts (en respectant bien sûr les contraintes énoncées). La partie est perdue dans le cas contraire (arrêt du jeu avant la fin, incohérence détectée...)

2) Architecture MVC à mettre en place

Vous devez mettre en place une architecture MVC (Modèle Vue Contrôleur):
un répertoire *modele* qui contiendra les fichiers du modèle, un répertoire *controleur* qui contiendra les contrôleurs et le "routeur" et un répertoire *vue* qui contiendra les vues ainsi que les feuilles de style et les images éventuelles. Il est demandé aussi un fichier *config.php* contenant des constantes valuées avec les identifiants de connexion à *mysql* et celles représentant les chemins relatifs à l'architecture MVC.

3) Les différentes vues du site

Le site comprendra les vues suivantes:
- une vue qui permettra l'authentification du joueur
- une vue qui permettra au joueur de jouer. Si le joueur sélectionne tour à tour deux villes, un pont se construit entre les deux, si cela est possible (donc respecte les contraintes liées au nombre de ponts maximum affectés à ces villes et tracé du pont entre les villes horizontal ou vertical). Dans l'autre cas, un message indique l'impossibilité d'établir le pont à l'utilisateur et le plateau de jeu est ré-affiché sans changement. On pourra, à partir de cette vue, réinitialiser la partie, ou revenir à la vue d'authentification.

- une vue qui affichera le résultat de la partie (perdue ou gagnée), des statistiques concernant le joueur (le nombre de parties gagnées sur le nombre de parties jouées), le classement des 3 meilleurs joueurs avec les statistiques liées à leurs parties. Une partie sera gagnée si tous les ponts ont été découverts. Elle sera perdue si une incohérence par rapport au nombre de ponts souhaités liés aux villes a été détectée. Idem si arrêt en cours de jeu, réinitialisation du jeu…, la partie est perdue. On pourra réinitialiser le jeu ou revenir à la vue d'authentification.
- des vues relatives aux gestions des erreurs (mauvaise authentification ...)

4) Le modèle

Vous avez sur *madoc* dans *ressources.tar.gz* deux tables à importer dans *mysql* qui se nomment *joueurs* et *parties.*
- la table *joueurs* comprend 2 champs qui sont le pseudonyme *pseudo* du joueur qui est une clé primaire ainsi que le mot de passe *motDePasse* qui lui est associé. Le mot de passe a été crypté avec la fonction php *crypt()* (voir documentation php pour des explications). Vous avez le script *crypt.php* dans *ressources.tar.gz* qui vous donne un exemple d'utilisation. En fait, dans cette table les valeurs du pseudonyme et du mot de passe sont identiques (pour faciliter les tests).
- la table *parties* comprend 3 champs qui sont l'identifiant *id* (entier qui s'auto incrémente), le pseudonyme *pseudo* qui est une clé étrangère (correspond au pseudonyme dans la table joueurs) et *partieGagnée* qui est de type booléen.

Vous développerez dans le répertoire *modele* le script php pour utiliser les données des tables. Deux scripts sont aussi fournis. Il faudra les utiliser mais sans modifier le constructeur de la classe *Villes.* Ainsi, lors de la correction, l'enseignant pourra simplement remplacer le code du constructeur de cette classe par le sien pour évaluer votre développement sur un autre tableau de jeu.

5) Bonus possible
- le joueur pourra revenir en arrière, donc annuler des coups joués (donc retrouver l'état du plateau antérieur)
- prise en compte des contraintes de non croisement des ponts et de navigabilité entre les villes du graphe ainsi formé.

Modalités de rendu:
La date limite de rendu est fixée au **9 décembre 2018 à 23 h sous madoc**.
Vous réaliserez ce mini-projet en binôme. Vous rendrez dans une même archive votre code php commenté et un rapport au format pdf explicitant en détail l'architecture des scripts ainsi que les techniques utilisées en les justifiant. Vous donnerez aussi un état d'avancement de votre rendu, ce qui fonctionne et ce qu'il reste à réaliser