

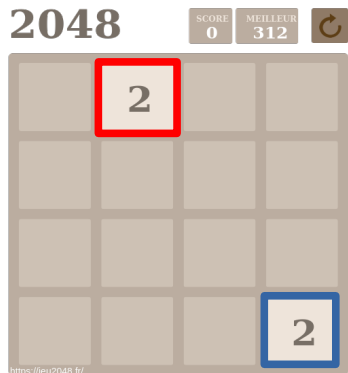




# Développement du jeu 2048 en PHP

Vous allez développer le jeu 2048 en utilisant **exclusivement** les langages *php*, *html* et *CSS*. Ce jeu a été conçu par un jeune développeur italien *Gabriele Cirulli* en 2014 .

## 1) Le jeu

Le but du jeu est de faire glisser des tuiles sur une grille de  $4 \times 4$  cases, pour combiner celles de mêmes valeurs en les additionnant et ainsi réussir à obtenir une tuile portant le nombre 2048. Le joueur peut toutefois continuer à jouer après cet objectif atteint pour faire le meilleur score possible. Après chaque mouvement, une tuile est ajoutée dans la grille, placée de manière aléatoire sur les cases vides restantes. Elle prend pour valeur 2 ou 4 (valeur choisie de manière aléatoire). Les tuiles ont donc une valeur correspondant à une puissance de 2. Le joueur a perdu quand il ne reste plus de case vide dans la grille et que le score 2048 n'est pas atteint.

Pour mieux comprendre comment fonctionne ce jeu, voici ci-dessous le début du déroulé d'une partie (les captures d'écran ont été réalisées via le site <https://jeu2048.fr>).

 <p><b>Grille en début du jeu</b></p> <p>Deux tuiles avec la valeur 2 positionnées aléatoirement.</p>	 <p><b>Mouvement vers le haut</b></p> <p>La tuile, entourée de bleu sur la première grille, se positionne en haut. La rouge ne bouge pas car le mouvement n'est pas possible. La verte est ensuite positionnée aléatoirement dans une case vide.</p>	 <p><b>Mouvement vers la gauche</b></p> <p>Les tuiles bleu et rouge précédentes ont fusionné en la tuile violette. La tuile verte se positionne à gauche sur sa ligne, la tuile orange est générée ensuite à une position aléatoire.</p>
 <p><b>Mouvement vers le haut</b></p>	 <p><b>Mouvement vers le haut</b></p>	

## 2) Structure des 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
- une vue qui affichera le résultat de la partie (perdue ou gagnée), le score obtenu, 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. Un lien permettra au joueur de rejouer une partie ou de revenir sur la vue d'authentification.
- des vues relatives aux gestions des erreurs (mauvaise authentification ...)

## 3) Les tables associées au jeu

Nous allons travailler avec *sqlite* qui est un moteur de bases de données relationnelles. Sa particularité est de ne pas reproduire le schéma habituel client/serveur. L'intégralité de la base de donnée (déclarations, tables, index et données) est stockée dans un seul fichier indépendant de la plateforme. Vous avez ci-dessous quelques commandes pour l'utiliser. Si besoin, vous avez un descriptif plus complet sur <https://doc.ubuntu-fr.org/sqlite>.

- importer un fichier sql dans une base sqlite3 nommée ici *db2048.db*  
=> **sqlite3** db2048.db < sql2048.sql
- se connecter à la base => **sqlite3** db2048.db
- affichage du contenu d'une table  
=> après connexion, au prompt (>): select \* from JOUEURS;
- quitter sqlite3 => .quit
- **il faut que le fichier db2048.db aient ses droits d'accès positionnés en 777 (chmod)**

Vous avez sur *madoc* dans *ressources.zip* deux tables à importer dans *sqlite* qui se nomment *JOUEURS* et *PARTIES*, le code SQL de création est fourni dans le fichier *sql2048.sql*.

- la table *JOUEURS* comprend 2 champs qui sont le pseudonyme *pseudo* du joueur qui est une clé primaire ainsi que le mot de passe *password* qui lui est associé. Le mot de passe a été haché. Vous utiliserez la fonction php *password\_verify()* pour tester la validité du mot de passe entré par l'utilisateur. Dans cette table, les valeurs du pseudonyme et du mot de passe sont identiques (pour faciliter les tests).

- la table *PARTIES* comprend 4 champs qui sont l'identifiant *id* (entier qui s'auto incrémente), le pseudonyme *pseudo* qui est une clé étrangère (correspond au pseudo dans la table *JOUEURS*), le score *score* de la partie qui est un entier, et *gagne* qui est un entier qui renseigne si la partie a été gagnée ou non.

## 4) Le développement

- il faudra mettre en place une architecture MVC (Modèle-Vue-Contrôleur) pour réaliser ce site. L'arborescence du site vous est fournie dans l'archive.

- la technique des URL longues sera utilisée pour déterminer la direction de déplacement désiré .

- la technique des sessions sera utilisée pour stocker, durant une partie d'un joueur, les données relatives au jeu.

**Bonus :**

- revenir au coup précédent
- gérer la sauvegarde d'une partie non terminée et son rechargement
- gérer l'inscription de nouveaux joueurs.

**Modalités de rendu:**

La date limite de rendu sera fixée par votre enseignant de TD. Vous n'utiliserez que du php (pas de javascript).

Vous réaliserez ce mini-projet en binôme. Vous devez rendre dans une même archive sur madoc votre code php commenté et un rapport au format pdf explicitant en détail l'architecture des scripts ainsi que les techniques utilisées.

Bien respecter les consignes données. Votre site doit pouvoir s'installer sans autre manipulation qu'une copie de votre code sur le serveur web (bannir les URL en absolu!!!).