

## Programmation interactive client-serveur

Groupe 15612

### TP1

#### Remise par LEA le : voir LEA

Vous veillerez à structurer votre code en utilisant un maximum de fonctions (pensez à DRY). Vous penserez aussi à commenter le code et à respecter le guide de codage. Mettez des entêtes à vos pages.

A faire en équipe de **2 étudiants maximum**.

#### Exercice 1 : Formulaire

Réalisez un formulaire avec les champs suivants que l'on validera avant de l'envoyer au serveur :

- Nom et Prénom
- Nom d'utilisateur (pseudo)
- Mot de passe
- Confirmation du mot de passe
- Adresse
- Ville
- Code postal
- Adresse courriel
- Date de naissance
- Numéro d'assurance sociale (NAS)
- Type de paiement (Chèque ou Carte de crédit)

Si la carte de crédit est choisie les 2 champs de saisie suivants s'affichent :

- Type de carte de crédit (Visa, Mastercard, ...)
- Numéro de carte de crédit
- Date d'expiration

Si le paiement par chèque est choisi les champs suivants apparaissent :

- Banque
- Numéro de chèque

Pour valider le formulaire, vous devez écrire les fonctions suivantes dans un fichier **formulaire.js** et l'intégrer à votre page html:

- `estMotDePasse` : contient au moins 6 caractères et une lettre Majuscule ;
- `estPseudo` : au moins 6 lettres ou chiffres ;
- `estEmail` : est une adresse courriel ;
- `estCode postal` : au format H1H 1H1 ;
- `estDate` : est une date (jj/mm/aaaa) ;
- `estNAS` : est un NAS valide (9 chiffres séparés par 2 espaces. Le numéro doit vérifier la formule de Luhn);
- `estNumeroCC` : est un numéro valide de carte de crédit (formule de Luhn).

D'autres fonctions seront certainement nécessaires.

Le numéro de téléphone sera saisi au format xxxxxxxxxx et transformé par une ER au format (xxx)xxx-xxxx

Une fois validées, les données seront **affichées par un script PHP** pour s'assurer de leur bonne conformité.

## Exercice 2 : Un jeu de puzzle en Javascript.

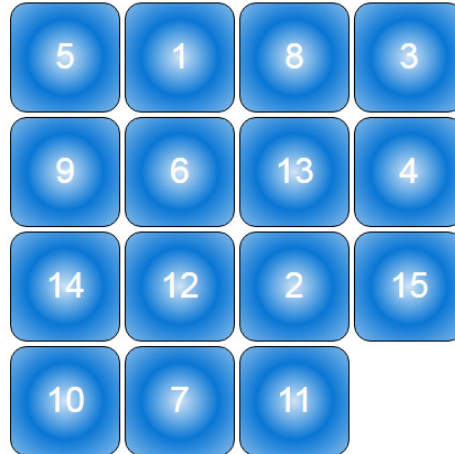
Ce jeu se joue sur une grille 4\*4 dont la dernière case est vide.

Chaque case est un bouton qui porte un numéro. Un clic sur un bouton le déplace vers la case visée, si elle est voisine.

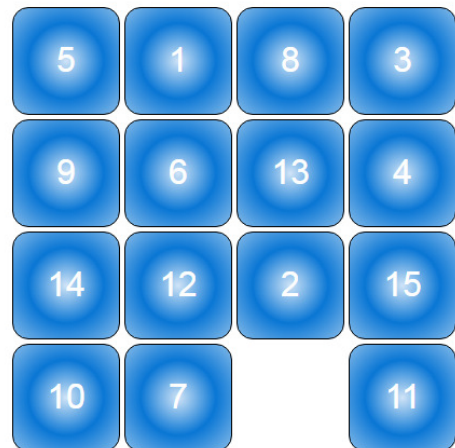
Au départ les numéros des boutons sont répartis aléatoirement sur la grille.

Le but du jeu est de remettre les jetons en ordre en un nombre minimal de coups.

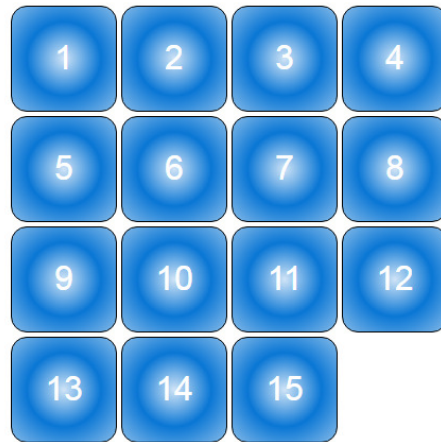
Exemple de grille de départ :



Après un coup (clic sur le 11):



## Fin de la partie



A chaque fin de partie, on affichera le nombre de coups pour résoudre le puzzle et le comparer à un record qui sera stocké dans le stockage local ainsi que le nom du joueur qui sera demandé à la fin de la partie, si celui-ci réalise un record.

On peut aussi envisager le calculer le temps que ça lui a pris (et baser le record sur le temps, plutôt que sur le nombre de coups).

Les données des meilleurs scores seront conservées en utilisant le stockage local.

### Barème de correction :

Exercice 1	
- Tous les champs sont validés par des fonctions adéquates	/4 pts
- Le DOM est bien géré	/2 pts
- Le code est bien commenté (entêtes aux scripts et aux fonctions)	/1 pt
- Les expressions régulières sont utilisées et correctes	/2 pts
Exercice 2	
- Le jeu fonctionne bien	/6 pts
- L'apparence de la grille est bien respectée	/2 pts
- Le code est bien commenté (entêtes aux scripts et aux fonctions)	/1 pt
- Le code est organisé en fonctions	/1 pt
- Les fonctions sont bien définies	/1 pt