



Master Bioinformatique  
Programmation événementielle 2017/2018  
**Travaux Pratiques**

# TP noté Javascript

**Consignes :** Le TP est à rendre au plus tard le 08 janvier à l'adresse [florent.grelard\[at\]u-bordeaux.fr](mailto:florent.grelard[at]u-bordeaux.fr). Le travail peut s'effectuer en binôme, précisez dans ce cas dans le sujet du mail les noms et prénoms des deux membres du binôme. Le code est évalué plus que le résultat : tout effort sera valorisé dans la notation.

Récupérer l'archive correspondant au TP noté. Cette archive contient du code Javascript ainsi que des images utilisées dans la suite du TP.

## Blackjack

**Préambule** Votre objectif est de développer une page Web qui permette de jouer au Blackjack. Le Blackjack est un jeu de cartes opposant un joueur contre la banque. Le but est de battre la banque sans dépasser 21 points. Dès qu'un joueur fait plus que 21, il perd la manche. La valeur des cartes est établie comme suit :

- As : 1 point
- De 2 à 10 : valeur nominale de la carte
- Figures (valet, dame, roi) : 10 points

Au début de la partie, la banque distribue une carte face visible au joueur et à la banque. Puis il demande au joueur l'option qu'il désire choisir :

1. Si le joueur veut une carte, il doit l'annoncer en disant "Carte!". Le joueur peut demander autant de cartes qu'il le souhaite pour approcher la valeur 21 sans la dépasser. Si, après le tirage d'une carte, il a dépassé 21, il perd la manche et la banque gagne.
2. Si le joueur décide de s'arrêter, il l'annonce en disant "Je reste", c'est alors à la banque de jouer.

(Source : Wikipedia)

## Blackjack



FIGURE 1

## HTML/CSS

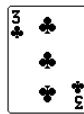
**Question 1 :** Créer une page Web statique (HTML) qui permette d'obtenir un rendu proche de celui présenté en Figure 2.

### **BlackJack**

#### **Les cartes de la banque**



#### **Vos cartes**



La banque a 10.

Vous avez 3. Carte ou reste ?

Carte

Reste

Rejouer

FIGURE 2 – Sans CSS

**Question 2 :** Créer un fichier CSS qui permette de mettre en forme votre page de façon à avoir un rendu proche de celui obtenu dans la Figure 1.

## Javascript

**Question 3 :** Au chargement de la page, donner une carte aléatoire à la banque et au joueur, en utilisant la fonction `Math.random`. Lire la documentation de cette fonction au besoin.

**Question 4 :** À chaque clic sur le bouton "Carte", ajouter une nouvelle carte à côté des cartes du joueur.

**Indice :** Le fichier Javascript présent dans l'archive contient du code permettant d'ajouter des éléments dans le DOM (arbre HTML).

**Question 5 :** Calculer le score du joueur, en faisant la somme des valeurs des cartes tirées aléatoirement (cf. préambule). **Indice :** il est possible de connaître le score d'une carte directement à partir du nom de l'image ("1.BMP", "14.BMP", "27.BMP", "40.BMP" = 1 pt) .

**Question 6 :** Actualiser le score dans la page HTML à chaque fois que le joueur appuie sur le bouton "Carte" .

**Question 7 :** Si le score dépasse 21, afficher un message dans la page HTML à l'adresse de l'utilisateur pour le notifier qu'il a perdu la manche.

**Question 8 :** Au clic sur le bouton "Reste", donner quatre nouvelles cartes aléatoires à la banque (pour un total de cinq cartes).

**Question 9 :** Actualiser le score de la banque dans la page HTML après le tirage de ces quatre cartes.

**Question 10 :** Si le score de la banque est supérieur à celui du joueur et inférieur à 21, afficher un message dans la page HTML à l'adresse de l'utilisateur pour le notifier qu'il a perdu la manche. Dans les autres cas,

annoncez-lui qu'il a gagné.

**Question 11 :** Rendez la banque plus intelligente. Par exemple, la banque peut tirer des cartes tant que son score est inférieur à celui du joueur.

**Question 12 :** Permettre au joueur de recommencer une manche avec le bouton "Rejouer".