

JEU DE POKER SHUFFLE

Marc Feeley

Le TP2 a pour but de vous faire pratiquer les concepts suivants : les boucles, les tableaux, les fonctions, la décomposition fonctionnelle, le traitement d'événements et la programmation web.

Le code que vous devez écrire (fichier “tp2.js”) implante un jeu qui exécute dans l'environnement du navigateur web. Bien qu'il soit possible et même judicieux d'utiliser codeBoot ou bien node.js pour développer certaines parties du jeu (fonctions de base comme trier un tableau de nombres, évaluer une main de poker, etc.), ultimement il faudra faire des tests et du déboguage avec les outils de développement du navigateur web.

Vous pouvez utiliser le code qui a été montré dans le cours mais vous ne devez pas utiliser du code provenant d'ailleurs (du web).

1 Introduction

Ce travail pratique consiste à développer un programme web pour jouer au jeu “poker shuffle”. C'est un jeu joué en solitaire (un seul joueur) avec des cartes à jouer standard (52 cartes). Le jeu consiste à piger 25 cartes une à la fois et les placer dans une grille 5 par 5 pour former des mains de poker sur les 5 rangées et 5 colonnes.

Chaque main de poker (une paire, deux paires, full house, etc.) a un nombre de points correspondants. Le joueur doit tenter de maximiser le pointage total, c'est-à-dire la somme des points pour les 5 rangées et 5 colonnes. Lorsque le joueur place la 25ième carte, le jeu se termine et le programme affiche le pointage final puis redémarre une nouvelle partie. Avant la fin de la partie le joueur peut déplacer les cartes sur la grille. Il peut aussi recommencer une nouvelle partie en cliquant un bouton.

Un modèle complètement fonctionnel du jeu est disponible à l'URL suivant :

<http://www.iro.umontreal.ca/~feeley/tp2.html>

Votre travail consiste donc à imiter son comportement le plus fidèlement possible.

2 Mains de poker et pointage

Une main de poker est un groupe de 5 cartes. Voici les différentes mains de poker et le pointage associé, par ordre décroissant de pointage :

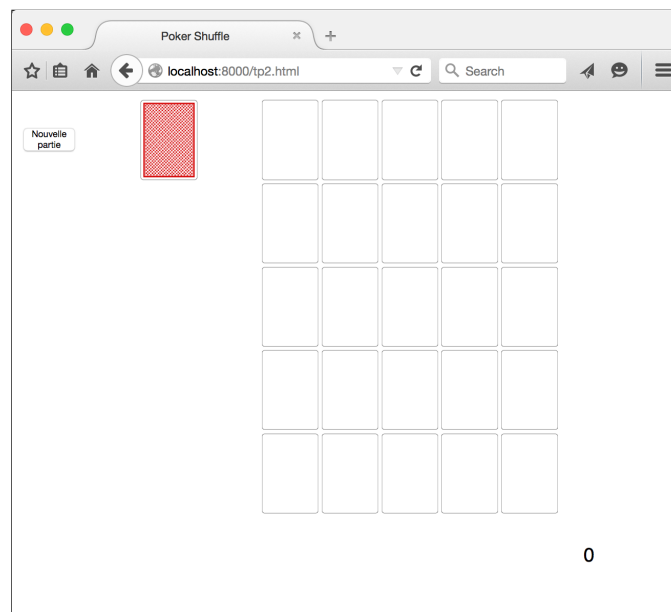
- **(100 points)** “Quinte Flush Royale” : L'as, le roi, la dame, le valet et le 10 d'une même couleur (tous treffe, tous pique, tous carreau, ou tous coeur).
- **(75 points)** “Quinte Flush” : Cinq cartes de même couleur qui se suivent (par exemple le 7, 8, 9, 10 et le valet, tous treffe, tous pique, tous carreau, ou tous coeur).

- **(50 points)** “Carré” : Quatre cartes de même valeur (par exemple quatre valets).
- **(25 points)** “Full House” : Trois cartes de même valeur et une paire de cartes de même valeur (par exemple trois as et deux 9).
- **(20 points)** “Couleur” ou “Flush” : Toutes les cartes de même couleur (tous treffe, tous pique, tous carreau, ou tous coeur).
- **(15 points)** “Quinte” : Cinq cartes qui se suivent (par exemple le 8, 9, 10, valet et dame). Il est à noter que l’as peut être le début ou la fin de la séquence.
- **(10 points)** “Brelan” : Trois cartes de même valeur (par exemple trois rois).
- **(5 points)** “Double Paire” : Une paire de cartes de même valeur et une autre paire de cartes de même valeur (par exemple deux rois et deux 7).
- **(2 points)** “Une Paire” : Une paire de cartes de même valeur (par exemple deux as).

3 Déroulement de la partie

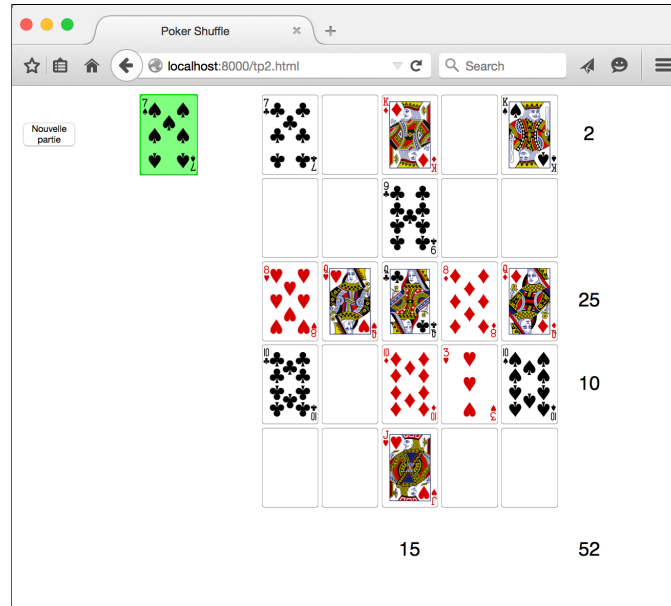
L’interface graphique du jeu contient (de gauche à droite) un bouton, la pile de donne et une grille de 5x5 cases. Au bout de chaque rangée et colonne il y a la place pour mettre un pointage, et dans le coin inférieur droit de la grille on trouve le pointage total.

Au début d’une partie la pile de donne montre le dos d’une carte, la grille contient des silhouettes de cartes vides, le pointage total est 0 et les pointages des rangées et colonnes n’affichent rien (c’est seulement lorsqu’il y a une main de poker valant des points qu’on doit afficher le pointage correspondant). Voici à quoi ressemble la fenêtre au début d’une partie :



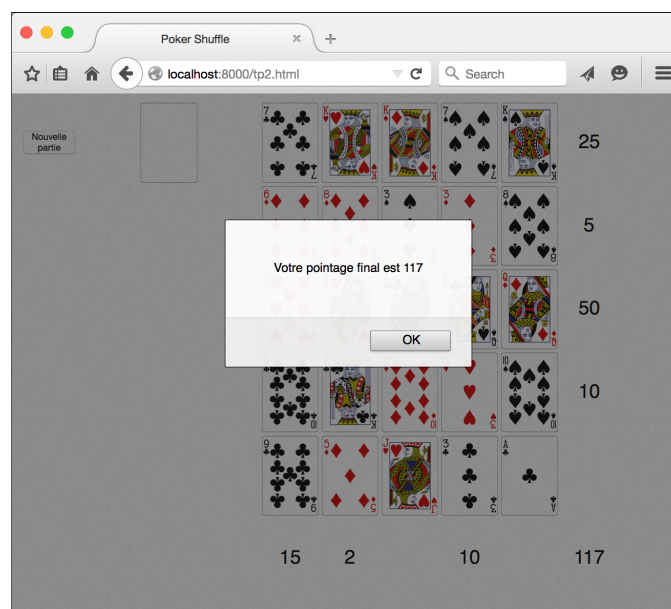
Le joueur doit cliquer sur la pile de donne pour voir la face de la carte. En cliquant sur la pile, cela sélectionne la carte (elle aura un fond de couleur vert vif). Cela permettra avec un clic sur une case vide de

la grille d'y transférer la carte. Il est aussi possible de cliquer sur une carte de la grille pour la sélectionner, et ensuite cliquer sur une autre case de la grille (vide ou pas) pour échanger le contenu des deux cases. Voici à quoi pourrait ressembler la fenêtre au milieu d'une partie après que le joueur ait cliqué sur la pile de donne :



Le joueur pourrait ensuite cliquer sur la quatrième case de la première rangée de la grille pour y mettre le 7 de pique, et ainsi obtenir les points pour 2 paires sur cette rangée. Il pourrait aussi cliquer sur la pile de donne à nouveau pour désélectionner cette carte ou bien cliquer sur une des cartes non vides de la grille pour sélectionner cette carte (avec l'intention de la déplacer). Vous devez expérimenter avec le programme modèle pour imiter comment les cartes sont sélectionnées.

Lorsque la dernière carte est placée sur la grille, le programme affiche le pointage final comme suit avant de redémarrer une nouvelle partie :



4 Détails techniques

Pour vous démarrer dans la bonne direction, sur Studium vous trouverez le fichier “serveur-web.js” et le fichier “documents.zip”. Le fichier “documents.zip” lorsque “dézipé” sur votre ordinateur va donner un répertoire “documents” qui pourrait être utilisé tel quel par le serveur “serveur-web.js” exécuté avec node.js pour exécuter le programme modèle. En effet le répertoire “documents” contient les fichiers “tp2.html”, “tp2.css”, “tp2.js” et “cards” (un sous-répertoire qui contient les images des cartes en format SVG, “Scalable Vector Graphics”). Vous ne devez rien changer à cette organisation de fichier et **seulement remplacer “tp2.js” par votre code**. Les autres fichiers doivent rester inchangés.

Le fichier “tp2.html” contient le nécessaire pour inclure le fichier CSS “tp2.css” et le code JavaScript “tp2.js”, et le corps du document est vide. Le corps a l’identifiant “b” et un traiteur d’événement “onload” qui fait appel à la fonction “init” définie dans “tp2.js”. Voici le contenu du fichier “tp2.html” :

```
<!DOCTYPE html>

<html>

  <head>
    <meta charset="UTF-8">
    <title>Poker Shuffle</title>
    <script src="tp2.js"></script>
    <link rel="stylesheet" href="tp2.css">
  </head>

  <body id="b" onload="init();"></body>

</html>
```

C’est donc la fonction “init” qui doit créer le contenu HTML qui sera mis dans le corps du document. En guise d’exemple, on pourrait définir la fonction “init” comme suit pour afficher une grille 2x2 de cartes avec une des cartes sélectionnée :

```
var init = function () {

  document.getElementById("b").innerHTML = '\
    <table>\
      <tr>\
        <td id="0"></td>\
        <td id="1"></td>\
      </tr>\
      <tr>\
        <td id="2"></td>\
        <td id="3"></td>\
      </tr>\
    </table>';

  document.getElementById("0").style.backgroundColor = "lime";
};
```

Évidemment, pour une grille de 5x5 il serait mieux de créer le HTML avec des boucles.

Les noms des fichiers d'image de cartes dans le sous-répertoire “cards” sont dérivés de l'anglais (“QH.svg” pour “Queen of Hearts”, “10D.svg” pour “10 of Diamonds”, “empty.svg” pour la silhouette de carte, et “back.svg” pour le dos de carte). Le format SVG a l'attrait de donner des images très nettes car c'est un format qui permet de faire un zoom infini sur l'image.

Pour indiquer qu'une carte est sélectionnée, utilisez la couleur “lime” comme “backgroundColor” du style de l'élément. Pour désélectionner une carte utilisez la couleur “transparent”.

Je vous suggère d'utiliser un nombre de 0 à 51 pour identifier les cartes, et d'utiliser le code 52 pour la carte vide et 53 pour le dos de carte. Les nombres de 0 à 3 correspondent au 4 as (de chaque couleur), puis les nombres 4 à 7 correspondent au 4 “2” (de chaque couleur), et ainsi de suite. Donc si on a les cartes x et y , elles ont la même couleur si $x \& 3 = y \& 3$, et elles ont la même valeur si $x >> 2 = y >> 2$.

Un paquet de carte peut donc être représenté par un tableau de longueur 52 contenant des nombres de 0 à 51 (dans un ordre arbitraire si le paquet est mélangé). On peut créer un paquet de carte mélangé en partant d'un tableau des nombres consécutifs de 0 à 51, puis échanger le dernier élément avec un des éléments du tableau pigé aléatoirement, puis échanger l'avant dernier élément avec un des éléments du tableau (sauf le dernier) pigé aléatoirement, puis échanger l'avant avant dernier élément avec un des éléments du tableau (sauf les deux derniers) pigé aléatoirement, et ainsi de suite.

5 Évaluation

- Ce travail compte pour 20 points dans la note finale du cours (suite à l'ajustement pour les accommodements). Vous devez le faire par groupes de 2 personnes. Indiquez vos noms clairement dans les commentaires au début de votre code.
- Vous devez remettre votre fichier “tp2.js” **uniquement**. La remise doit se faire au plus tard à 23:55 mercredi le 22 avril sur le site Studium du cours.
- Chaque fonction devrait avoir un bref commentaire pour dire ce qu'elle fait, il devrait y avoir des lignes blanches pour que le code ne soit pas trop dense, les identificateurs doivent être bien choisis pour être compréhensibles et respecter le standard CamelCase.

Voici le barème pour la correction du TP2, sur 100% :

- 30%: Structure et logique du code : pas de code redondant, utilisation de paramètres, la décomposition des fonctions en sous-fonctions (pas de fonctions trop longues, il doit y avoir les fonctions `init()` et `clic(id)`, et probablement des fonctions pour créer un paquet de cartes mélangées, évaluer le pointage d'une main, trier des cartes (pour simplifier le calcul du pointage), évaluer le pointage total, générer le nom de l'image d'une carte étant donné son numéro, pour sélectionner une carte de la grille, etc.), utilisation appropriée des opérateurs JavaScript, éviter les variables globales (sauf certaines qui sont nécessaires pour que `init()` et `clic(id)` maintiennent l'état du jeu tel que le nombre de cartes jouées et le contenu de chaque case de la grille), etc.
- 30%: La lisibilité et la présence de commentaires explicatifs : chaque fonction devrait avoir un bref commentaire pour dire ce qu'elle fait, il devrait y avoir des lignes blanches pour que le code ne soit pas trop dense, les identificateurs doivent être bien choisis pour être compréhensibles et respecter le standard CamelCase.

- 30%: Le choix de tests unitaires pertinents pour le calcul du pointage d'une main (au moins pour toutes les mains de poker possibles) et pour générer le nom du fichier de l'image d'une carte. Il y a perte de points proportionnel si ces fonctions n'ont pas été testées et ont une interface compliquée et pas "fonctionnel".
- 10%: Performance du code (c'est subjectif, mais des algorithmes clairement pas performants font perdre des points en fonction du degré d'inefficacité).

Points négatifs :

- -30%: Le respect de la spécification (perte de jusqu'à 30% si la spécification n'est pas respectée, par exemple ne pas sélectionner les cartes tel que demandé, ou ne pas afficher correctement le pointage, ou ne pas démarrer une nouvelle partie à la fin, etc.).