



Cégep **André-Laurendeau**

Cours 420-245-AL
Programmation Web

Remise finale :
jeudi 18 avril @ 23h55

TP #2
Création d'un jeu de carte en JavaScript

Jeu de carte CAL Jack!

LES OBJECTIFS FINAUX	3
LES OBJECTIFS INTERMÉDIAIRES	3
DIRECTIVES	3
LA LOGIQUE DE L'APPLICATION (BUT DU JEU)	3
À L'OUVERTURE DE LA PAGE D'ACCUEIL	4
LORS DE LA SAISIE DE CARACTÈRES DANS LES CHAMPS TEXTE ET PASSWORD	4
LORS DE L'ENVOI DES DONNÉES À LA PAGE DU JEU	5
À L'OUVERTURE DE LA PAGE DU JEU	5
QUELQUES STRUCTURES DE DONNÉES	6
CRITÈRES DE CORRECTION	7
ANNEXES	8
TABLEAU DES TOUCHES INVALIDES	8
CAPTURES D'ÉCRANS DE L'APPLICATION (EN GUISE DE GUIDE)	9
<i>Page d'accueil au démarrage</i>	<i>9</i>
<i>Cas d'un code utilisateur invalide lors de la saisie</i>	<i>9</i>
<i>Cas d'un mot de passe invalide lors de la saisie.....</i>	<i>10</i>
<i>Cas d'un code utilisateur invalide lors de la soumission des données (Chrome)</i>	<i>10</i>
<i>Cas d'un code utilisateur invalide lors de la soumission des données (Firefox).....</i>	<i>10</i>
<i>Cas d'un mot de passe invalide lors de la soumission des données (Chrome).....</i>	<i>11</i>
<i>Cas d'un mot de passe invalide lors de la soumission des données (Firefox).....</i>	<i>11</i>
<i>Stockage des données en cas de réussite de la validation des données</i>	<i>11</i>
<i>Page du jeu au début.....</i>	<i>12</i>
<i>Ajout d'une carte au joueur</i>	<i>13</i>
<i>Arrêt du jeu</i>	<i>13</i>
<i>Nouvelle partie</i>	<i>15</i>

Les objectifs finaux

- Créer un jeu de carte « **CAL Jack** » dont l'accès nécessite la connexion du visiteur
- Respecter les **standards** de programmation Web côté client
- Respecter l'**organisation des fichiers/dossiers** recommandée par le **W3C** (dossier pour les fichiers **.css**, dossier pour les fichiers **.js**)
 - Vous devez **OBLIGATOIREMENT** :
 - conserver l'organisation des dossiers/fichiers des **images**
 - avoir **1 fichier .js** par page
- Pour la **mise en forme**, l'utilisation de **Bootstrap 4** n'est pas obligatoire, mais fortement recommandée
 - Si vous utilisez des **feuilles de styles externes**, les fichiers devront être stockés dans un dossier

Les objectifs intermédiaires

- Utiliser les **éléments de base** du langage JavaScript (variables, constantes, boucles, conditions, tableau, fonctions nommées et fonctions anonymes, objets JavaScript)
- Utiliser l'**objet Date** et **gérer le délai d'exécution de script**
- Valider les champs de formulaire à l'aide d'**attributs HTML** et de l'**API Forms**
- Manipuler le **DOM HTML** et gérer **des événements**
- Utiliser les **API WebStorage** et **Canvas**

Directives

La logique de l'application (but du jeu)

- Pour pouvoir jouer, le visiteur devra saisir ses informations de **connexion** et choisir la **couleur** de ses cartes à l'aide d'un **formulaire**
- Les informations de connexion, ainsi que le choix de couleur des cartes seront **stockées dans le navigateur de façon persistante**
- Le jeu opposera deux **participants** : le **visiteur** (le **joueur**) du site à l'**ordinateur** (la **banque**). L'objectif est d'atteindre un **total de 21 points**, sans le dépasser, en additionnant les valeurs des cartes de chaque **participant**. **Les valeurs des cartes** sont les suivantes :
 - **AS = 1**
 - Les **cartes 2 à 10** valent la valeur nominative de la carte
 - Les **figures** (valet, dame et roi) valent **10 points chacune**
- Au début de la partie, la **banque** distribue **une carte** (face visible) au joueur et **une** à la banque
- Le joueur peut **demandeur une carte additionnelle**, en cliquant sur un **bouton**

Jeu de carte CAL Jack!

- Si le joueur décide qu'il a assez de cartes, il clique sur un **bouton** pour **arrêter le jeu**
- Lorsque le jeu est arrêté, c'est au tour de la banque de jouer. Pour simplifier la logique, on ne fait que **distribuer 4 cartes aléatoirement à la banque**
- On détermine le **gagnant**

À l'ouverture de la page d'accueil

On affiche un **formulaire** contenant les **7 champs** suivants :

1. Un champ de type **texte** pour saisir le **code d'utilisateur**
 - ☐ C'est un champ **obligatoire**
 - ☐ Composé de **5 à 10 caractères alphanumériques**
 - ☐ Uniquement composé de **lettres** et de **chiffres**
 - ☐ Aucune lettre **accentuée**, ni de **caractères spéciaux** ne sont acceptés ([voir le tableau en annexe à ce document](#), le tableau des caractères **considérés invalides**)
2. Un champ de type **password** pour saisir le **mot de passe de l'utilisateur**
 - ☐ C'est un champ **obligatoire**
 - ☐ Composé uniquement de **chiffres** et d'exactly **8 chiffres**
3. **Quatre** champs **radio** pour que l'utilisateur choisisse la **couleur** de ses cartes
 - ☐ **Par défaut**, l'option **cœur** sera sélectionnée
4. Un bouton pour **envoyer les données via l'URL** à la **page du jeu**

Lors de la saisie de caractères dans les champs texte et password

Les 2 champs sont validés avant la soumission des données (via l'URL)

- Pour le champ **code d'utilisateur**
 - ☐ À chaque entrée du clavier (événement **keypress**), on vérifie si la touche est un caractère invalide ([voir le tableau en annexe](#)). S'il s'agit d'une **touche invalide**, on change la couleur de fond (**backgroundColor**) du champ
- Pour le champ **mot de passe**
 - ☐ À chaque entrée du clavier (événement **keypress**), on vérifie si la touche est un nombre (méthode **isNaN()**). S'il ne s'agit pas d'un nombre, on change la couleur de fond (**backgroundColor**) du champ

Lors de l'envoi des données à la page du jeu

Les 2 champs sont validés lors de la soumission des données (via l'URL)

- On vérifie que les valeurs des champs respectent les **contraintes de validité (attributs HTML5)**. Utiliser les méthodes de **L'API forms** : **checkValidity()** et **setCustomValidity()** pour **afficher un message personnalisé**
 - ☐ Si la validation échoue, on change la couleur de la **bordure** des champs invalide et on stoppe l'envoi des données (la méthode **preventDefault()** de l'objet **Event**)
 - ☐ Tester avec **Firefox**. **Chrome** est moins tolérant
- Si la validation est réussie, on stocke les données **localement**
 - ☐ Utiliser **L'API WebStorage**

À l'ouverture de la page du jeu

- On affiche la **date du jour courant** au format **Jeudi, 11 avril 2019**. Cette date doit être générée à l'aide de Javascript, pour qu'elle change automatiquement
 - ☐ Utiliser l'objet **Date**
- On affiche un **chronomètre** qui affiche les **minutes** et les **secondes** qui s'écoulent
 - ☐ Utiliser la méthode **setInterval()** de l'objet **Window**
 - ☐ Pour afficher les **minutes** et les **secondes** au format 01. Si le compteur de minutes / secondes est **inférieur à 10**, on concatène la chaîne « **0** » en avant
- On affiche la **carte du visiteur** et celle de la **banque**
 - ☐ Les cartes (**images**) sont affichées dans des **canvas**, après leur chargement
 - Utiliser les méthodes de **L'API Canvas** : **getContext()**, **drawImage()**
 - ☐ Le choix de la carte de fait **aléatoirement**, dans un paquet de **12 cartes** rangées selon leur **couleur (cœur, carreau, trèfle et pique)**
 - Utiliser les méthodes **random()** et **floor()** de l'objet **Math**
 - **Rappel** : la couleur est transmise de la page d'accueil via **L'URL**
 - Utiliser les méthodes des objets **Location**, **String** et **Array** : **search()**, **split()**, **lastIndexOf()** ou **indexOf()**,
- On affiche **3 boutons**
 - ☐ Un bouton pour **ajouter une carte** au visiteur
 - Au **clic** on appelle la fonction qui permet d'afficher une nouvelle carte dans le **canvas** du joueur
 - Cette **nouvelle carte** doit aussi être choisie de façon **aléatoire**
 - À chaque ajout d'une carte, on calcule le **nombre de points** accumulés pour le joueur et l'on l'affiche
 - ☐ Un bouton pour **arrêter le jeu**, lorsque le visiteur le décide
 - Au **clic** du bouton :
 - Le chronomètre s'arrête (**clearInterval()**)
 - On distribue **4 cartes aléatoirement** à la **banque**
 - On calcule le **nombre de points** de la banque et on l'affiche

Jeu de carte CAL Jack!

- On détermine le **gagnant**, selon **les points accumulés** par le joueur et ceux accumulés par la banque
 - Si la **banque** a plus de points que le **joueur ET** si elle a **21 points ou moins**, On affiche un message(**alert**) indiquant que la **banque** a gagné!
 - Si les deux participants ont plus de 21 points, on déclare un **match nul**
 - Dans tous les autres cas, c'est le **joueur** qui a gagné
 - On **affiche le score** de chaque participant dans la page
- ☐ Un bouton pour **commencer une nouvelle partie**
- Au **clic** du bouton :
 - Le chronomètre **redémarre à 0**
 - Les **points** des deux participants sont remis à **0**
 - On efface le contenu des **canvas** et une **nouvelle carte** est distribuée à chaque participant (la méthode **clearRect()**)
 - Le **score** des participants est **remis à 0**

Quelques structures de données

- Pour valider le champ code d'utilisateur lors de la saisie des caractères, stocker les valeurs (**keyCode**) des touches invalides ([voir le tableau en annexe](#)) dans un tableau (**Array**)
- Pour représenter le **joueur** et la **banque**, créer une **classe Participants** avec 2 propriétés
 - ☐ Un **nombre de points**, initialisé à **0**
 - ☐ Un **tableau de cartes**, initialement vide. Il sera rempli pendant le jeu. On y ajoutera les **cartes du joueur** et celle de la **banque** (selon l'instance de la classe appelée)
- Pour pouvoir calculer les points, on stockera les **noms des cartes** avec leurs **valeurs associées** dans un **objet JavaScript (clé : valeur)**
 - ☐ La **clé** peut être le **nom du fichier .jpg** (ex : **as.jpg**). Dans ce cas, la **valeur** sera **1**

Jeu de carte CAL Jack!

Critères de correction

Barème de correction

Éléments	Critères	Pondération
Organisation des dossiers/fichiers	<ul style="list-style-type: none">• Respect de la structure des dossiers et des fichiers	10%
Aspect visuel professionnel	<ul style="list-style-type: none">• Disposition des éléments des pages et des champs de formulaires• Choix de couleur des pages• Uniformité de l'apparence des pages à travers le site• Flexibilité de l'affichage en lien avec le contenu des pages (RWD : Bootstrap, Media Queries)	20%
Page d'accueil	<ul style="list-style-type: none">• Respect des exigences pour les éléments à afficher• Technique et fonctionnalité de la validation des champs• Fonctionnalité de la gestion des données de l'utilisateur (validation et stockage des données avec la gestion des événements et les API Forms et WebStorage)	20%
Page du jeu	<ul style="list-style-type: none">• Respect des exigences pour les éléments à afficher• Technique de manipulation du DOM• Emploi de la programmation orientée objet et l'utilisation des structures de données JavaScript• Gestion du délai d'exécution• L'emploi de l'API canvas	50%

Annexes

Tableau des touches invalides

Caractères invalides	Code (keyCode) de la touche du clavier
\$	36
%	37
&	38
*	42
?	63
@	64
À	192
Â	194
Ç	199
È	200
É	201
Ê	202
Ë	203
Î	206
Ï	207
Ô	212
Ù	217
Û	219
à	224
â	226
è	232
ç	231
é	233
ê	234
ë	235
î	238
ï	239
ô	244
ù	249
û	251

Jeu de carte CAL Jack!

Captures d'écrans de l'application (en guise de guide)

Page d'accueil au démarrage

CAL JACK

Code utilisateur

Entre 5 et 10 caractères alphabétiques

Mot de passe

Exactement 8 chiffres



Se connecter

Cas d'un code utilisateur invalide lors de la saisie

CAL JACK

Code utilisateur

andré

Jeu de carte CAL Jack!

Cas d'un mot de passe invalide lors de la saisie

CAL JACK

Code utilisateur

andre

Mot de passe

.....

Cas d'un code utilisateur invalide lors de la soumission des données (Chrome)

CAL JACK

Code utilisateur

andr

Mot de passe

.....

Cas d'un code utilisateur invalide lors de la soumission des données (Firefox)

CAL JACK

Code utilisateur

andr

Le code utilisateur est obligatoire et doit avoir entre 5 et 10 lettres non accentuées, ni de caractères spéciaux!.

Jeu de carte CAL Jack!

Cas d'un mot de passe invalide lors de la soumission des données (Chrome)

CAL JACK

Code utilisateur

andre

Mot de passe

...

Cas d'un mot de passe invalide lors de la soumission des données (Firefox)

CAL JACK

Code utilisateur

andre

Mot de passe

...

Le mot de passe est obligatoire et doit contenir 8 chiffres!.

Stockage des données en cas de réussite de la validation des données

Application	
Key	Value
couleur	coeur
mdp	12345678
pseudo	andre

Jeu de carte CAL Jack!

Page du jeu au début

CAL JACK

Date: Mardi, 26 mars 2019

00:06

Les cartes de la banque



Vos cartes



La banque à 4 points

Vous avez 10 points

Je veux une carte

J'arrête

Rejouer

Jeu de carte CAL Jack!

Ajout d'une carte au joueur

CAL JACK

Date: Mardi, 26 mars 2019

00:53

Les cartes de la banque



Vos cartes



La banque à 4 points

Vous avez 12 points

Je veux une carte J'arrête Rejouer

Arrêt du jeu

Cette page indique

Bravo! Vous avez gagné!

OK

Jeu de carte CAL Jack!

CAL JACK

Date: Mardi, 26 mars 2019

02:03

Les cartes de la banque



Vos cartes



La banque à 26 points

Vous avez 14 points

Je veux une carte

J'arrête

Rejouer

Jeu de carte CAL Jack!

Nouvelle partie

CAL JACK

Date: Mardi, 26 mars 2019

00:19

Les cartes de la banque



Vos cartes



La banque à 2 points

Vous avez 4 points

Je veux une carte

J'arrête

Rejouer