

Projet

Loto National

Rédigé par

David ROUMANET
Professeur BTS SIO



Changement

Date	Révision

Sommaire

A Introduction.....	1
A.1 Présentation.....	1
Projet.....	2
A.2 Cahier des charges.....	2
A.2.1 Apparence.....	2
A.2.2 Sécurité.....	2
A.2.3 Simulation longue.....	2
A.2.4 Simulation courte.....	3
A.2.5 Gains.....	3
A.2.6 Ordre des grilles.....	4
B Recommandations et aides.....	5
B.1 Règles.....	5
B.2 Recommandations.....	5
B.3 Tirage automatique d'une grille.....	6
B.3.1 Création d'un nombre aléatoire.....	6
B.3.2 Création d'une table et remplissage d'une grille de loto.....	7
B.4 Comparaison des grilles.....	8

Nomenclature :

- **Assimiler** : cours pur. Explication théorique et détaillée (globalement supérieur à 4 pages).
- **Décoder** : fiche de cours, généralement inférieure à 5 pages.
- **Découvrir** : Travaux dirigés. Faisable sans matériel.
- **Explorer** : Travaux pratiques. Nécessite du matériel ou des logiciels.
- **Mission** : Projet encadré ou partie d'un projet.
- **Voyager** : Projet en autonomie totale. Environnement ouvert : Vous êtes le capitaine !

A Introduction

La Française des jeux est une société autorisée à proposer des jeux d'argents aux citoyens français. En effet, les jeux d'argents sont fortement règlementés par l'ANJ ([Autorité Nationale des Jeux](#)). Voici ce que l'on trouve sur son site :

Seules quatre catégories de jeux d'argent et paris sportifs sont autorisées en France. Ces quatre catégories sont :

- Les jeux de loteries exploités en ligne et en point de vente sous le régime de monopole confié à la FDJ
- Les paris sportifs proposés en ligne par les opérateurs titulaires d'un agrément ANJ et en point de vente par la FDJ
- Les paris hippiques proposés en ligne par les opérateurs titulaires d'un agrément ANJ et en point de vente par le PMU ;
- Le poker en tournoi ou en "cash game" proposé en ligne par les opérateurs titulaires d'un agrément ANJ.

La loterie nationale est l'un des jeux le plus connu, avec des enjeux allant jusqu'à plusieurs dizaines de millions d'euros.

Tirage du mercredi 11 décembre 2024

Jackpot non remporté

Résultats Loto



A.1 Présentation

Le fonctionnement du Loto s'appuie sur la sélection aléatoire de cinq numéros compris entre 1 et 49, dans une grande sphère où les boules sont brassées.

Ainsi, lorsqu'un numéro sort, il ne pourra plus être sélectionné.

D'autre part, il existe un numéro chance (dans une autre sphère, souvent d'une couleur différente) entre 1 et 10. Il est indépendant de l'autre sphère, ainsi le numéro chance 6 peut sortir alors que le 6 est déjà présent dans l'autre grille.

Projet

Dans un cadre événementiel, l'association [SOS Joueurs](#) veut démontrer que les chances de gagner à un tel jeu, sont extrêmement faibles et que la dépense totale dans une vie est énorme.

Elle vous donne son cahier des charges pour créer une application permettant de le démontrer de manière ludique.

A.2 Cahier des charges

L'application doit permettre de miser sur une grille de loto et d'en simuler le tirage hebdomadaire (une fois par semaine). Les gains et les pertes sont simulés avec une base de ticket à 3 €.

Ainsi, le joueur place 3 € par semaine et peut gagner un total de 3 000 000 €.

A.2.1 Apparence

Dans l'idéal, l'utilisateur doit pouvoir saisir ses numéros dans des cases vierges (5 cases pour les numéros de 1 à 49) et une case vierge (numéro chance de 1 à 10).

Une autre possibilité est un ensemble de deux grilles que l'utilisateur coche :

- Une grille ayant des numéros de 1 à 49
- Une grille ayant des numéros de 1 à 10

Le programme présente un tirage simple ou une simulation de tirage. Dans les deux cas, le programme vérifie la grille de l'utilisateur et la grille tirée automatiquement. Il y aura donc deux boutons.

L'application aura des couleurs et des formes proches des boules du Loto

A.2.2 Sécurité

Quelque que soit l'apparence, l'application doit gérer les erreurs de saisies :

- Saisie de nombre inférieurs à 1 ou supérieurs à 49
- Saisie de nombre déjà existant dans les cases précédentes (cas méthode de saisie)
- Nombre de cases cochées égal à cinq (erreur en cas de différence, dans la méthode de cocher les numéros dans une grille)

A.2.3 Simulation longue

La simulation doit montrer le nombre d'années nécessaire pour gagner en **rang 1 ou 2**. On considère qu'il y a un tirage par semaine et qu'un an compte 52 semaines. Par exemple 105 semaines signifie 2 ans et 1 semaine. Le programme effectue autant de tirage que nécessaire pour que le joueur finisse par gagner.

Il n'est pas recommandé d'afficher toutes les valeurs calculées dans la simulation, car cela ralentit fortement l'application.

A.2.4 Simulation courte

Dans le cas de l'appui sur le bouton Tirage simple (un seul tirage), le programme doit afficher le rang du gain :

Rang(s)	Combinaison trouvée	Une chance sur
Rang 1	●●●●●○	19 068 840
Rang 2	●●●●●	2 118 760
Rang 3	●●●●○	86 677
Rang 4	●●●●	9 631
Rang 5	●●●○	2 016
Rang 6	●●●	224
Rang 7	●●○	144
Rang 8	●●	16
Rang 9	●○	28
Rang 9	○	18
Total combinaisons gagnantes		~ 5,99

Les [gains statistiques](#) pour les rangs sont les suivants :

Rang	Gain
1	3000000 € (3 millions)
2	130000 €
3	2500 €
4	485 €
5	58 €
6	22 €
7	11 €
8	5 €
9	2 €

A.2.5 Gains

Dans la simulation "courte", le gain est minimisé par le coût du ticket (3 €).

Dans la simulation "longue", il faudra soustraire 3 € par tirage et ajouter le gain associé à chaque tirage (faire le cumul). Par exemple, un gain de rang 1 après 13 ans, et 6 gains de rang 7 donne :

$$\text{Gain} = 1 \times 3000000 + 6 \times 22 - 13 \times 52 \times 3$$

A.2.6 Ordre des grilles

Pour rappel, le tirage est aléatoire, mais il faut compter le nombre de boules identiques dans les deux grilles. Exemple d'une grille gagnante de rang 2 (on ne voit pas le numéro chance).

Tirage	36	20	8	40	42
Joueur	8	20	36	40	42

B Recommandations et aides

B.1 Règles

Voici les règles à respecter :

- L'usage de l'IA est autorisé pour déboguer une partie de code, mais pas le générer entièrement
 - *Note de l'enseignant : ce paramètre est difficile à vérifier, mais pensez à l'épreuve finale qui se fera sur machine, non connectée.*
- Le programme doit être commenté correctement
- L'aspect graphique est géré dans un fichier CSS à part
- L'aspect dynamique est géré dans un fichier JavaScript à part
- Le travail en binôme est autorisé, mais doit être indiqué dans l'entête du commentaire
 - Les deux étudiants doivent comprendre l'ensemble du code
 - Il faut éviter absolument l'étudiant HTML/CSS et l'autre JS
 - Le travail doit être équitablement réparti.

B.2 Recommandations

Créer un programme peu s'avérer impressionnant, il est capital de découper ce programme en petite étape logique :

- Saisie du joueur
- Tirage de la machine
- Affichage d'un tirage
- Comparaison de deux grilles (joueur et ordinateur)
- etc.

D'autre part, les interactions avec un élément d'une page HTML se font sur la lecture d'un objet :

L'objet `document.getElementById('unElement')` contient la valeur dans `document.getElementById('unElement').value`.

Préférez l'utilisation des balises `<div>` plutôt que des balises `<table>`, `<tr>` et `<td>` qui sont moins personnalisables.

B.3 Tirage automatique d'une grille

L'algorithme implique de remplir un tableau avec 5 numéros (tous différents) allant de 1 à 49 et de tirer un nombre aléatoire de 1 à 10 pour le numéro complémentaire (une variable dédiée).

B.3.1 Création d'un nombre aléatoire

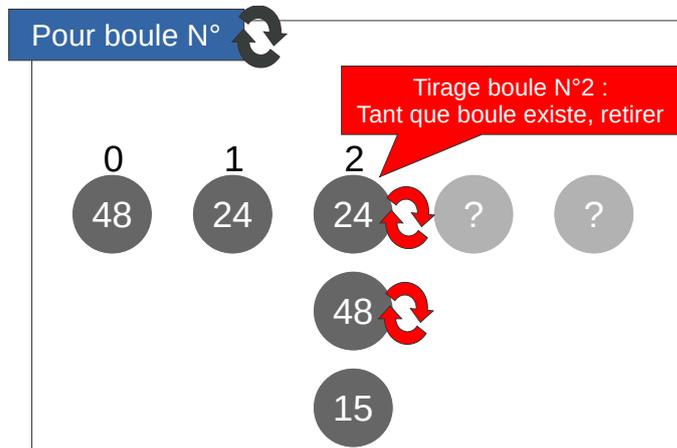
L'idéal est de créer une fonction pour laquelle on fournit deux paramètres pour déterminer une valeur comprise entre un minimum et un maximum. Par exemple, dans le cas d'un minimum à 5 et un maximum à 50, les valeurs calculées avec la fonction `Math.random()` devront être comprises entre 5 et 50.

Pour vous aider, voici les éléments de création :

<code>Math.random()</code>	Une fonction appartenant à la librairie Math, capable de fournir un nombre entre 0.000 et 0.999
<code>Math.round(valeur)</code>	Une fonction de la librairie Math permettant d'arrondir une valeur en entier.
<pre>function nom(param) { return valeur }</pre>	Création d'une nouvelle fonction JavaScript nommée nom et ayant un paramètre d'entrée param . La fonction se termine en renvoyant le contenu de valeur .
$Z = X * (\text{max} - \text{min}) + \text{min}$	Calcul mathématique pour obtenir une valeur de Z comprise entre min et max

B.3.2 Création d'une table et remplissage d'une grille de loto

Il faut créer un tableau de 5 cases, puis utiliser deux fonctions. La première (en bleus sur le schéma) tire un nombre et passe à la boule suivante... mais pour cela, elle doit être sûre que le nombre tiré n'existe pas déjà dans la table. Pour cela, il faut une condition sur une boucle de tirage (tant que le nombre existe dans la liste, recommencer le tirage du nombre aléatoire).



Pour vous aider, voici les éléments de création :

<pre>for (let i=0 ; i < nbre ; i++) { }</pre>	Un bloc d'instruction permettant de créer une boucle allant de 0 à nbre .
<pre>let grille = [] grille[i] = unNombre</pre>	Création d'un tableau nommé grille et placement de la valeur unNombre dans la case i de la table grille .
<pre>If (condition) { } else { }</pre>	Un bloc d'instruction conditionnel : si la condition est vraie, c'est le premier bloc qui est utilisé, sinon c'est le deuxième. Dans une condition, il y a une comparaison entre deux termes avec un signe (< ou == ou > ou >=... etc.)
<pre>while (condition) { }</pre>	Un bloc d'instruction permettant de créer une boucle qui reste active tant que la condition est vraie

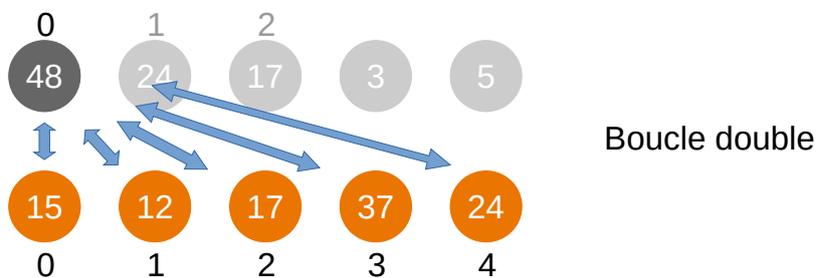
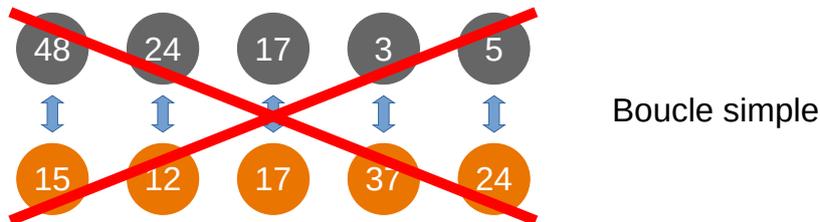
À noter : une fonction de vérification de numéro sera utile à deux étapes :

- Lorsque le programme génère une nouvelle grille
- Lorsque le joueur saisit sa grille

Il est donc important de créer une fonction dédiée à la vérification d'un nombre dans une grille, qui renvoie vrai ou faux. Le nom sera donc `estPresent(numero)` et doit retourner une valeur `true` (le numéro existe déjà dans la grille) ou `false` (le numéro n'existe pas dans la grille).

B.4 Comparaison des grilles

Une fois que le joueur a rempli sa grille et que l'ordinateur a rempli sa grille, il faut les comparer ensemble. Cependant, il faut accepter l'idée que les numéros ne sont pas forcément dans l'ordre, comme le montre la figure ci-après.



Ainsi, il faut créer une fonction capable de comparer deux grilles (que l'on pourra passer en paramètre).

L'idée est ici de faire le cumul des comptages de bonnes valeurs. Il faut une boucle pour chaque valeur de la grille A et une sous-boucle pour lire chaque valeur de la grille B et la comparer à celle de la grille A.

Pour vous aider, voici les éléments de création :

<pre>for (let i=0 ; i < nbre ; i++) { for (let j=0 ; j < nbre ; j++) { } }</pre>	Un bloc d'instruction ayant une boucle et une sous-boucle
<pre>let compteur = 0</pre>	Création d'une variable de comptage
<pre>if (condition) { compteur++ }</pre>	Bloc d'instruction pour incrémenter conditionnellement un compteur
<pre>return compteur</pre>	Une instruction de fin de fonction renvoyant le contenu de compteur .