

Le but de ce qui suit est **d'étudier les instructions "innerHTML", "Array" et les boucles en javascript**, en construisant un jeu mélangeant agilité et connaissances.

Voici une suite exercices.

Visualisez le code source de la page Web :

http://www.juggling.ch/zgisin/a2015_oc3/page260a_questions_agilite.html

Sauvez-le dans votre dossier "website"

Ouvrez-le fichier que vous venez de sauver en local, dans Firefox et dans Netbeans.

Cette page permet de faire bouger une image horizontalement, à gauche ou à droite.

Elle permet de faire bouger au clavier une image grise, avec les flèches.

- 1) Sauvez cette page sous le nom "page260b_questions_agilite.html". "a" remplacé par "b".
- 2) Changez le nom de l'auteur de la page, en ligne 5. *name="author"* reste, *content="..."* change. Changez également l'information en bas de page, concernant la date et le nom du créateur.
- 3) Pour l'instant, il y a une ligne noire et une boule bleue, positionnées en absolu. Il n'y a aucun code javascript, tout restera à écrire.
- 4) Pour le terrain de jeu, ajoutez :
pour un bord haut, une ligne noire horizontale en *left:5px; top:0px; width:880px; height:10px;*
pour un bord gauche, une ligne noire verticale en *left:5px; top:0px; width:10px; height:630px;*
pour un bord droit, une ligne noire verticale en *left:880px; top:0px; width:10px; height:630px;*
- 5) Ajoutez les 4 variables globales suivantes, qui indiqueront la position et la direction de déplacement de la boule bleue "idBB1001" : *glvPosX=100; glvPosY=80; glvDirX=0.95; glvDirY=0.83;*
Ajoutez la variable *glnTimerID=0;* pour mémoriser l'état du Timer.
- 6) En vous basant sur ce qui a été fait dans les "pages250..._ping_pong.html", mettez la boule bleue en mouvement en utilisant un "Timer", qui déplace la boule toutes les 10 [ms] dans la direction de déplacement. Écrivez donc une **fonction** *Bouge()*.
Écrivez une **fonction** *Start()* et une fonction *Stop()* pour lancer et arrêter le Timer.
Ajoutez un **bouton** "Start" et un bouton "Stop" qui appellera les deux fonctions précédentes.
- 7) Sauvez cette page sous le nom "page260c_questions_agilite.html". "b" remplacé par "c".
- 8) Ajoutez en début de la fonction *Bouge()* l'instruction :
if ((glvPosY > 610) && (glvDirY > 0)) glvDirY = -glvDirY;
Donc si la boule touche le bord bas et si sa direction est vers le bas, change sa direction Y.
- 9) Ajoutez trois lignes similaires pour que la boule rebondisse sur les trois autres bords.
- 10) Pour de l'aide durant le développement, il est souvent utile d'afficher de l'information en cours d'exécution du programme. Une méthode est d'ajouter l'instruction *alert(string);* où le *string* est une chaîne de caractère donne l'information. C'est très souvent utile, mais cela met en pause l'exécution du programme et demande une confirmation à l'utilisateur, ce qui est parfois non souhaitable.
Voici donc une autre manière de faire, qui introduit également la propriété **innerHTML** qui permet de modifier le code HTML d'une balise, généralement <div> ou .
Ajoutez après fermeture de balise </div> principale, et avant la balise <div ...> servant à l'affichage de la date de mise à jour et de l'information sur le développeur le code suivant :

```
<span id="idInfo"
  style="position:absolute; left:0px; top:0px;" >
```

Pour des informations.

```
</span>
```

À la fin de la fonction *bouge()*, ajoutez le code suivant.

```
document.getElementById('idInfo').innerHTML = "X=" + glvPosX.toFixed(1) + " Y=" + glvPosY.toFixed(1);
```

En cours de mouvement de la boule, de l'information utile au développement s'affiche.

11) Sauvez cette page sous le nom "page260d_questions_agilite.html". "c" remplacé par "d".

On veut avoir une action si on clique sur la boule bleue

- Ajoutez le code `onclick="ClickBB(this);` à l'action de cliquer sur la boule bleue.
- Ajoutez la fonction `function ClickBB(oImage)` qui aura comme effet d'appeler la fonction `Stop()` et donc d'arrêter la boule bleue.
- Ajoutez la ligne de code suivante à la fonction `ClickBB(oImage)`
`document.getElementById('idInfo').innerHTML += " id=" + oImage.id;`
 Elle montre que `oImage.id` donne de l'information sur la boule bleu.
- Tester aussi le code suivant :
`oImage.src = "images/BouleRouge.png";`
 Remarquez que la couleur de la boule a changé. On peut ainsi changer tous les attributs que l'on désire de l'image "idBB1001".
- Quel code ajouter pour que la taille de la boule change ?
- Quel ligne (une ou deux) ajouter, pour que au "Start" la boule redevienne bleue ?

12) Sauvez cette page sous le nom "page260e_questions_agilite.html". "d" remplacé par "e".

Il faut ajouter des Informations sur le but du jeu et ses règles.

- Sous la fin de la section `<div id="idDivMain" ... </div>` ajoutez le code html suivant :

```
<div id="idInformation"
  style=" display:none; border:solid 2px black; padding:5px;
  text-align:left; width:700px; height:400px;">
Text d'information sur le jeu.<br>
<br>
<button type="button" onclick="CacheInformation();"> OK </button>
</div> <!-- fin de "idInformation" -->
```

À vous d'ajouter le texte d'information concernant ce jeu. Il sera à développer par la suite.

- Ajoutez une fonction `MontreInformation()` qui :
 arrête le Timer ;
 cache la `<div id="idDivMain"` avec `...display="none";`
 montre la `<div id="idInformation"` avec `...display="inline-block";`
- Changer le comportement de "onclick" du bouton "Information" pour qu'il appelle la fonction `MontreInformation();`
- Ajoutez une fonction `CacheInformation()` qui :
 montre la `<div id="idDivMain"` avec `...display="block";`
 cache la `<div id="idInformation"` avec `...display="none";`
- Challenge : saurez-vous faire en sorte que l'information apparaisse au centre de la fenêtre ?

13) Sauvez cette page sous le nom "page260f_questions_agilite.html". "e" remplacé par "f".

On veut que lorsqu'on clique sur la boule bleue pendant qu'elle est en mouvement, une question à choix multiple apparaisse, après avoir arrêté le jeu.

- Sous la fin de la section `<div id="idInformation" ... </div>` ajoutez le code html suivant :

```
<div id="idQuestion"
  style=" display:none; border:solid 2px black; padding:5px;
  text-align:left; width:700px; height:400px;">
Combien font 11 fois 12 ?<br>
...
</div> <!-- fin de "idInformation" -->
```

Ajoutez trois boutons, dont un contient la bonne réponse. Celui contenant la bonne réponse doit cacher la question, montrer le jeu et relancer la boule bleue.

- Ajoutez une fonction `MontreQuestion()` qui :
 arrête le Timer ;
 cache la `<div id="idDivMain"` avec `...display="none";`
 montre la `<div id="idQuestion"` avec `...display="inline-block";`
- Modifiez le code pour qu'elle soit appelée lorsqu'on clique sur la boule bleue pendant qu'elle est en mouvement.

Il serait intéressant d'avoir plusieurs questions différentes. Cela mènera à la notion de **array**, qui représente un tableau de divers éléments, tels qu'une liste de nombres ou une liste de strings.

- 14) Sauvez cette page sous le nom "page260g_questions_agilite.html". "f" remplacé par "g".
- Ajoutez la variable globale : `var glnEtape = 1; // Indique à quel niveau on se trouve dans le jeu`
 - Ajoutez une deuxième question à choix multiples, dans un `<div>`, comme fait au point précédent. À vous d'ajouter également le code pour avoir cette deuxième question. Il faut qu'après avoir répondu à la première question, `glnEtape` soit augmenté de 1, pour que la deuxième fois que l'on clique sur la boule bleue, c'est la deuxième question qui soit posée.

À discuter tous ensemble des divers implémentations qui ont été trouvées.

- 15) Sauvez cette page sous le nom "page260h_questions_agilite.html". "g" remplacé par "h".
- Ajoutez une troisième question à choix multiples, dans un `<div>`, comme fait au point précédent. À vous d'ajouter également le code pour avoir cette troisième question. Il faut qu'après avoir répondu à la deuxième question, `glnEtape` soit augmenté de 1, pour que la troisième fois que l'on clique sur la boule bleue, c'est la troisième question qui soit posée.

On pourrait continuer ainsi avec une quatrième question, puis une cinquième, mais on constate que le code est très répétitif. C'est ici que la notion de **tableau** ou **array** en anglais simplifiera énormément le code et permettra d'ajouter simplement autant de questions que l'on désire.

Le point 16) est le plus gros point de cette série, mais il permettra une souplesse et l'adjonction de nombreuses questions très simplement.

- 16) Sauvez cette page sous le nom "page260i_questions_agilite.html". "h" remplacé par "i".
- Ajouter le code suivant sous la déclaration `"var glnEtape = 1;"` :

```
// Liste des questions
var glastrQuestion = [ 'vide'
                      , 'Combien font 11 fois 12 ?'
                      , 'À quelle couleur correspond "rgb(255, 255, 0)" ?'
                      , 'À quel nombre correspond le nombre binaire 1101 ?'
                      ];
// Liste des réponses possibles
var glastrRep1 = [ 'vide', '122', 'jaune', '11' ];
var glastrRep2 = [ 'vide', '144', 'rouge', '13' ];
var glastrRep3 = [ 'vide', '132', 'cyan', '15' ];

// Liste des numéros des réponses correctes.
var glanCorrecte = [0, 3, 1, 2];
```

Cela déclare des **tableaux (array)** de chaînes de caractères (**string**) et de nombres pour déterminer les questions.

- Modifiez le code HTML de la `<div id="idQuestion"...` comme suit :

```
<div id="idQuestion"
    style=" display:none; border:solid 2px black; padding:5px;
          text-align:left; width:700px; height:400px;">
<span id="idQ1"></span><br>
<br>
<span id="idQ2"></span><br>
<br>
<button type="button" id="idR1" onclick="TraiteReponse(this);" ></button>
<button type="button" id="idR2" onclick="TraiteReponse(this);" ></button>
<button type="button" id="idR3" onclick="TraiteReponse(this);" ></button>
</div> <!-- fin de "idQuestion" -->
```

Les textes des boutons sont vides, mais ils seront remplis au moment de poser les questions.

- c) Dans la fonction "MontreQuestion()", juste après l'appel à la fonction "Stop()", ajoutez le code suivant, qui modifiera la question et les réponses possibles :

```
// Modification de la question et des réponses possibles indiquées dans les boutons.
document.getElementById('idQ1').innerHTML = "Question n°" + glnEtape;
document.getElementById('idQ2').innerHTML = glastrQuestion[glnEtape];
document.getElementById('idR1').innerHTML = glastrRep1[glnEtape];
document.getElementById('idR2').innerHTML = glastrRep2[glnEtape];
document.getElementById('idR3').innerHTML = glastrRep3[glnEtape];
```

Les deux lignes de code :

```
glnEtape++; // Passe à la question suivante.
if (glnEtape > 3) glnEtape = 1; // Seulement 3 questions, on revient à la première, après la 3ème
    devront être déplacé dans la fonction "TraiteReponse(pBouton)" qui est ajoutée ci-dessous.
```

- d) Ajoutez une fonction "TraiteReponse(oBouton)" qui traitera la réponse.

Dans cette fonction, écrivez le code suivant :

```
function TraiteReponse(oBouton) {
//=====
// Si la réponse est correcte, cache la question,
// montre le jeu et continue le jeu.

if (oBouton.id === "idR" + glanCorrecte[glnEtape]) {
    document.getElementById('idDivMain').style.display = "block"; // Montre la partie jeu
    document.getElementById('idQuestion').style.display = "none"; // Cache la question

    glnEtape++; // Passe à la question suivante.
    if (glnEtape >= glastrQuestion.length) glnEtape = 1; // On revient à la 1ère, après la dernière.
    Start();
}
else {
    alert("Réponse incorrecte");
}
} // TraiteReponse
```

- e) Dans les fonctions "CacheInformation()" et "MontreQuestion()" effacez les codes faisant référence à "idQuestion2" et à "idQuestion3".

- f) Effacez les <div> correspondant aux questions 2 et 3, car elles ont été intégrées dans la simplification rendue possible par l'adjonction du "array".

- 17) Sauvez cette page sous le nom "page260j_questions_agilite.html". "i" remplacé par "j". Ajoutez quelques nouvelles questions. Remarquez à quel point c'est simple de le faire !

Il serait intéressant d'avoir plusieurs boules bleues qui bougent en même temps.

- 18) Sauvez cette page sous le nom "page260k_questions_agilite.html". "j" remplacé par "k".

- Ajoutez une deuxième boule bleue ayant l'id "idBB1002".
- Ajoutez les 4 variables indiquant sa position et sa direction.
- Ajoutez le code nécessaire dans la fonction "Bouge()" pour qu'elle se déplace.
- Éliminez le code qui change la taille de la boule
- Éliminez le code qui rend bleu la couleur de la boule dans la fonction "Start()", ainsi que le code qui change la taille de la boule.

- 19) Sauvez cette page sous le nom "page260l_questions_agilite.html". "k" remplacé par "l".

- Ajoutez une troisième boule bleue ayant l'id "idBB1003".
- Ajoutez les 4 variables indiquant sa position et sa direction.
- Ajoutez le code nécessaire dans la fonction "Bouge()" pour qu'elle se déplace.

On pourrait continuer ainsi avec une quatrième boule, puis une cinquième, mais on constate que le code est très répétitif. La notion de **array** simplifiera de nouveau le code.

La notion de boucle **for** sera également utile.

19) Sauvez cette page sous le nom "page260m_questions_agilite.html". "l" remplacé par "m".

a) Ajoutez le code suivant dans la déclaration des variables, pour stocker la position et la direction des boules dans des **array**.

```
var glavPosX = [0, 100, 100, 200]; // Position X des boules
var glavPosY = [0, 80, 180, 180]; // Position Y des boules
var glavDirX = [0, 0.95, 0.35, 0.45]; // Direction X de déplacement des boules
var glavDirY = [0, 0.83, 0.53, 0.23]; // Direction Y de déplacement des boules
```

b) Remplacez le code de la fonction "**Bouge()**" par le suivant :

```
function Bouge() {
//=====
// Fait se déplacer les images des boules

var nn = 0; // pour la boucle for
var imgBBID = 0; // pour référencer les différentes boules.

// Boucle sur toutes les boules
for (nn=1; nn<glavPosX.length; nn++) {
// Test si on touche les bords
if ((glavPosX[nn] > 870) && (glavDirX[nn] > 0)) glavDirX[nn] = -glavDirX[nn];
if ((glavPosX[nn] < 10) && (glavDirX[nn] < 0)) glavDirX[nn] = -glavDirX[nn];
if ((glavPosY[nn] > 610) && (glavDirY[nn] > 0)) glavDirY[nn] = -glavDirY[nn];
if ((glavPosY[nn] < 5) && (glavDirY[nn] < 0)) glavDirY[nn] = -glavDirY[nn];

// Modifie la position de la boule numéro nn.
glavPosX[nn] = glavPosX[nn] + glavDirX[nn];
glavPosY[nn] = glavPosY[nn] + glavDirY[nn];

imgBBID = document.getElementById("idBB" + (1000 + nn));
imgBBID.style.left = parseInt(glavPosX[nn]) + "px";
imgBBID.style.top = parseInt(glavPosY[nn]) + "px";
} // for

//document.getElementById('idInfo').innerHTML =
} // Bouge
```

Remarquez la ligne : **for (nn=1; nn<glavPosX.length; nn++) {**

qui va exécuter les lignes qui suivent pour nn=1, puis pour nn=2, puis pour nn=3, etc. jusqu'à ce que nn=glavPosX.length-1.

Le programme répète ainsi plusieurs fois les mêmes instructions, avec des paramètres qui changent grâce à l'utilisation de **array**.

c) Effacez la déclaration des variables glavPosX ; glavPosY ; glavDirX et glavDirY,

20) Sauvez cette page sous le nom "page260n_questions_agilite.html". "m" remplacé par "n".

a) Ajoutez dans le code HTML une 4ème boule bleue. Donnez-lui l'id "idBB1004"

b) Ajoutez dans le tableau des positions et des directions la position et la direction de cette nouvelle boule.

Remarquez que c'est tout ce qui est nécessaire pour ajouter une nouvelle boule.

21) Sauvez cette page sous le nom "page260o_questions_agilite.html". "n" remplacé par "o".

Le but de ce qui suit est d'ajouter du **son**, chaque fois que l'on clique.

a) Ajoutez dans le code HTML suivant vers la fin de votre code, hors de tout <div>...

```
<audio id="idPlay" src="sons/tac.ogg" preload></audio>
```

Un fichier .mp3 est aussi possible comme fichier contenant un son ou de la musique.

b) Ajoutez les deux fonctions suivantes sous la fonction "Stop()"

```
function SonPlay(idPlayer) {
//=====
var player = document.querySelector('#' + idPlayer);
player.play();
} // SonPlay

function SonStop(idPlayer) {
//=====
var player = document.querySelector('#' + idPlayer);
player.currentTime = 0;
player.pause();
} // SonStop
```

c) Ajoutez la ligne de code suivante dans la fonction "ClickBB(oImage)"

```
SonPlay("idPlay"); // Joue un son lorsqu'on a cliqué sur une boule.
```

Nous avons vu l'essentiel, ce qui suit est un supplément.

Deux points sont encore désagréables dans le code actuel.

- ° Le code HTML pour ajouter des boules bleues est répétitif.
- ° Les tableaux de positions et directions des boules doit être complété à chaque adjonction de boules.

22) Sauvez cette page sous le nom "page260p_questions_agilite.html". "o" remplacé par "p".

a) Ajoutez le code suivant, juste avant le code HTML insérant la première boule bleue.

```
<!-- le code HTML pour l'adjonction de boules bleues sera placé ici par javascript -->
<span id="idBBajouter"></span>
```

b) Effacez le code HTML d'adjonction de boules

c) Écrivez la fonction suivante, qui créera les boules bleues.

```
function Initialisation() {
//=====
// Ajoute le code HTML des différentes boules.

var nn = 1;
var strS = "";

for (nn=1; nn<=4; nn++) {
  strS +=
    '\n';
} // for
```

```
document.getElementById('idBBajouter').innerHTML = strS;
Bouge(); // Pour positionner correctement les boules au départ
} // Initialisation
```

d) À la fin du code HTML, juste avant le dernier <div>, ajoutez le code :

```
<script>Initialisation();</script>
```

C'est bien, mais les tableaux des positions ne sont pas encore rempli automatiquement.

23) Sauvez cette page sous le nom "page260q_questions_agilite.html". "p" remplacé par "q".

a) Ajoutez le code suivant, au début de la boucle "for" de la fonction "Initialisation()".

```
glavPosX[nn] = Math.floor(860*Math.random()); // Position X des boules
glavPosY[nn] = Math.floor(600*Math.random()); // Position Y des boules
glavDirX[nn] = 2*Math.random() - 1.0; // Direction X de déplacement des boules
glavDirY[nn] = 2*Math.random() - 1.0; // Direction Y de déplacement des boules
```

b) Vous pouvez maintenant remplacer "nn<=4" par "nn<=10" pour avoir 10 boules bleues.

24) Sauvez cette page sous le nom "page260r_questions_agilite.html". "q" remplacé par "r".

Maintenant, **libre à vous** de développer plus le jeu. Voici des modifications - adjonctions ... :

- a) Les informations sur le but du jeu sont correctes.
- b) Une question a été ajoutée.
- c) Si une boule est rouge, le son est différent et rien ne se passe.
- d) Un chronomètre mesure le temps que l'on met pour répondre aux questions.
- e) Si une réponse est incorrecte, le chronomètre continue quand-même d'avancer.
- f) Lorsqu'on a répondu à toutes les questions, le jeu s'arrête et une indication de temps s'affiche.