

EXERCICE 10 (PROJET 2)

Pondération et échéance

- ☐ Cet exercice (projet 2) compte pour 3 % de la note finale du bulletin.
- ☐ Date de remise de l'énoncé : **Jeudi, 25 octobre 2018**
- ☐ Date de remise de la solution : **Jeudi, 8 novembre 2018 en fin de journée**

Une pénalité de 10 % par jour de retard, incluant les jours de congé, sera imposée, et ce, jusqu'à concurrence de 5 jours. (Après ce délai, la note attribuée est zéro.)

Fichier à copier

Remettre le fichier compressé **nomfamille-projet02.zip** contenant le dossier **nomfamille-projet02** via LÉA.

Dossier <i>nomfamille-projet02</i>		
Dossier	Fichier	Description
\App_Code	librairie_projet02.cs	Contient les fonctions C# de librairie
\images	40 images	Images entreposées dans le dossier <i>projet02</i>
\	default.aspx	Contient la partie présentation de l'application
	default.aspx.cs	Contient les fonctions C# spécifiques à l'application
	default.js	Contient les fonctions <i>JavaScript</i> spécifiques à l'application.
	librairie.js	Contient les fonctions <i>JavaScript</i> de librairie
	test-default.htm	Application HTML permettant de tester l'application <i>default.aspx</i>

Barème de correction

Exécution de l'application (l'application fait ce qu'elle a à faire !) 100 %
 Respect des spécifications..... jusqu'à -100 %
 Interface utilisateur (présentation et navigation)..... jusqu'à -20 %
 Disposition du code source jusqu'à -20 %

Vous serez évalués sur le produit final. (Une application ne produisant pas des résultats tangibles ne vaut rien.) Mieux vaut remettre une application incomplète, mais fonctionnelle qu'une qui (semble) complète, mais qui ne produit pas les bons résultats.

Énoncé du problème

On vous demande de concevoir le prototype d'une application *ASP.NET* permettant de jouer au *bonhomme pendu*. Pour avoir une idée du travail à effectuer, exécutez l'application dans votre navigateur : (<http://424x.cgodin.qc.ca/projet2-solution>).

Présentation

Comme pour l'exercice précédent, l'écran se divise en trois grands panneaux (*en-tête, corps de la page et pied de page*).

Le **corps de la page** se divise en trois rubriques soit (1) *On recherche un mot de 99 lettres* où les lettres [, tirets et apostrophes] composant le mot recherché apparaissent à mesure que le jeu progresse, (2) *Lettres à choisir* affichant les lettres [, tiret et apostrophe] qui n'ont pas encore été sélectionnées par le joueur et (3) *Verdict* où apparaît le bonhomme attaché à sa potence (Réf.: Fig. 1).

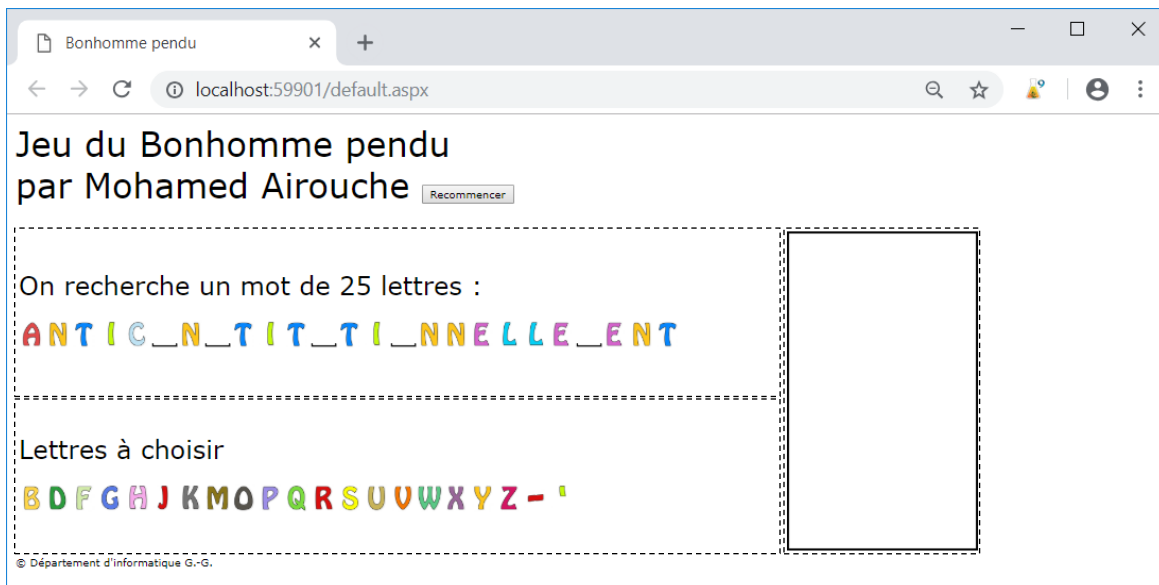


FIGURE 1 (LES POINTILLÉS APPARAISSENT À TITRE INDICATIF SEULEMENT.)

Règles du jeu

L'utilisateur clique sur la lettre [, tiret ou apostrophe] qu'il croit faire partie du mot. Deux scénarios sont alors possibles...

1. La lettre [, tiret et apostrophe] existe bien dans le mot. Dans ce cas, elle s'inscrit au(x) bon(s) endroit(s) dans le mot s'affichant dans la rubrique *Mot recherché*, puis disparaît de la liste des lettres [, tiret et apostrophe] à choisir dans la rubrique du même nom.
2. La lettre [, tiret et apostrophe] n'existe pas dans le mot. Dans ce cas, la potence et le bonhomme pendu se dessinent progressivement. Comme pour le premier scénario, la lettre sélectionnée disparaît de la liste des lettres à choisir dans la rubrique du même nom. L'application affiche le mot recherché après 9 tentatives infructueuses (Réf.: Fig. 2).

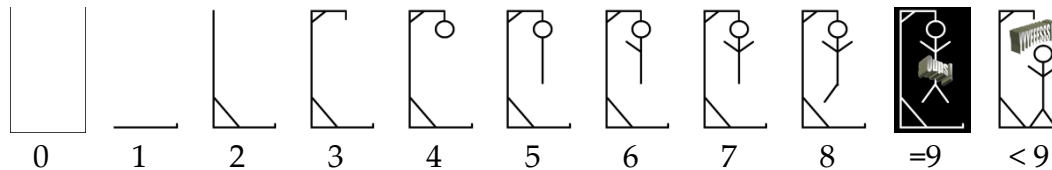


FIGURE 2

Quelques informations présentant un intérêt certain

1. Le mot recherché ne peut être constitué que des lettres de l'alphabet (aucun caractère accentué), du tiret et de l'apostrophe. Il ne doit pas dépasser 28 caractères (toutes les images sont entreposées dans le dossier *projet02/images*).

Pour aider le joueur, seule la position des lettres formant le mot doit s'afficher.

2. Pour ce prototype, il existe deux façons de choisir un mot soit (1) en assignant une valeur directement dans le code source ou (2) en saisissant le mot dans la barre d'adresse en utilisant le paramètre **Mot**. (Minuscules et majuscules doivent être traitées indifféremment.)

Ex. : `http://localhost/.../projet02/default.aspx?Mot=Recommencer`
`http://localhost/.../projet02/default.aspx?Mot=rEcOmMeNcEr`

L'exemple ci-dessous illustre comment l'application récupère le paramètre passé via la barre d'adresse (même si ce n'est pas illustré, prévoir le cas où aucun mot n'est saisi, dépasse 28 caractères ou contient des caractères autres que lettres, tirets et apostrophes) :

```
if (Request.QueryString["Mot"] == null) {
    initialiseMot("anticonstitutionnellement");
}
else {
    initialiseMot(Request.QueryString["Mot"].ToLower());
}
```

3. Le bouton **RECOMMENCER** se comporte de façon différente en fonction du moment où il est mis à contribution.

- Si l'utilisateur clique sur le bouton **RECOMMENCER** alors que la partie n'est pas terminée, une boîte de dialogue s'affiche pour lui offrir la possibilité d'annuler la partie en cours ou non (Réf.: Fig. 3).

S'il clique sur le bouton **OK**, la fonction `C# rechargePage` est exécutée; autrement le traitement se poursuit comme si de rien n'était.

- Dans le cas où la partie est terminée, la fonction `rechargePage` est exécutée automatiquement.

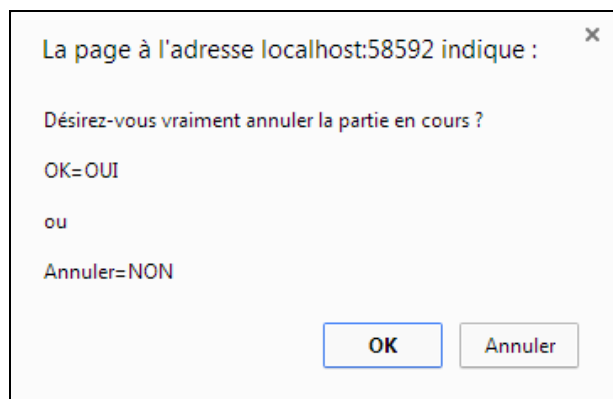


FIGURE 3

4. La page Web **test-default.htm** permet de tester les différents scénarios possibles.

Présentation de la page Web

À titre indicatif, voici les styles utilisés par l'application :

```
BODY, INPUT { font-family:Verdana; font-size:16px; }
//TD { border:dotted 1px blue; }
.sImage { width:25px; }
.sPendu { width:180px; }
.sPremiereCellule, .sTroisiemeCellule { width:720px; height:150px; }
.sDeuxiemeCellule { height:300px; }
.sReduit { font-size:10px; }
.sTitreApplication { font-size:32px; }
.sTitreSection { font-size:24px; }
.sVisible { visibility:visible; }
.sInvisible { visibility:hidden; }
```

Contraintes

1. Le contenu du formulaire doit être créé dynamiquement. Tous les contrôles doivent être ajoutés directement au formulaire dont le code apparaît ci-dessous :

```
<body>
  <form id="frmSaisie" runat="server">
    </form>
</body>
```

2. Toute la partie interactive de l'application doit être développée en *JavaScript*.

Assurez-vous que la page n'est pas rechargée à chaque clic de souris !

3. Vous devez vous servir des fonctions de librairie créées dans l'exercice précédent et mettre à jour ces dernières en fonction de vos besoins.