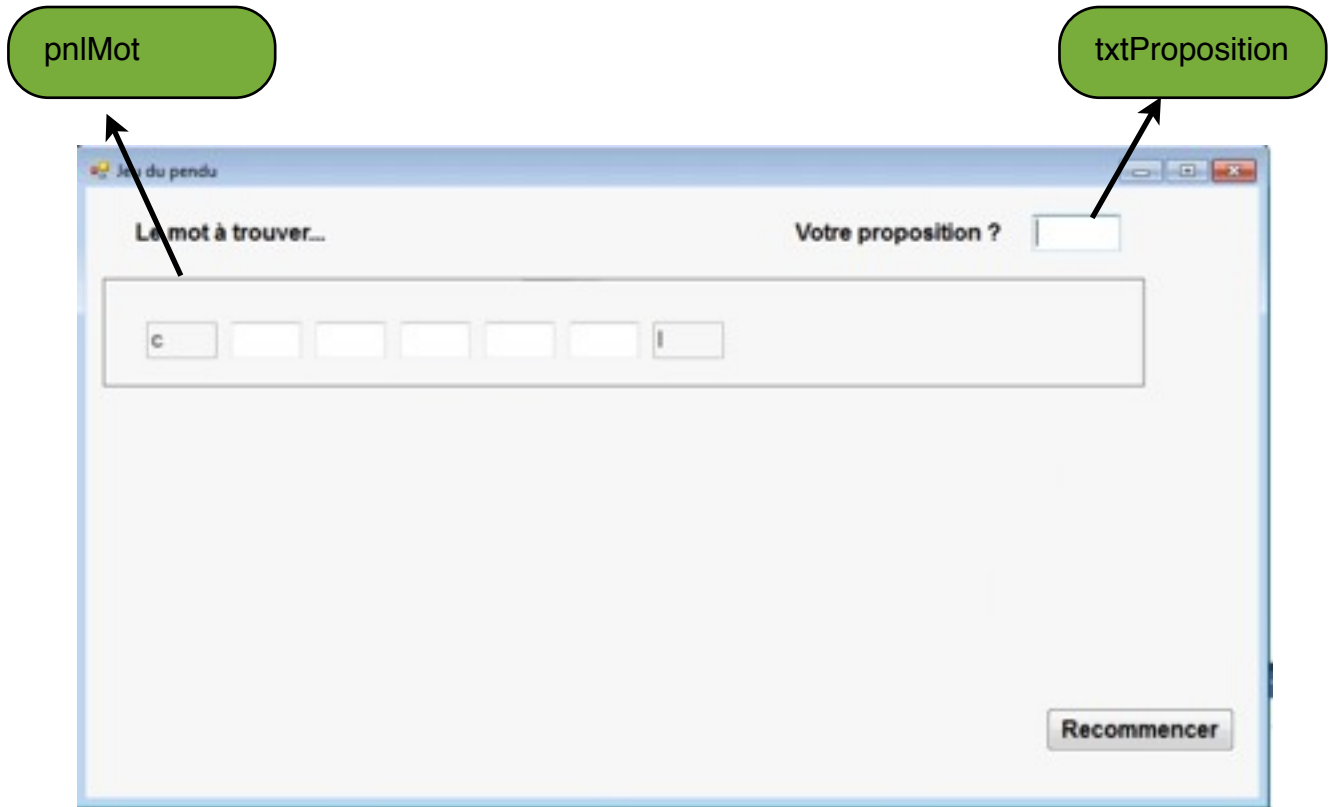


On désire concevoir une application en C# permettant de jouer au jeu du pendu, avec l'interface graphique suivante :



Principe de fonctionnement :

- Une liste de mots «mystère» est stockée dans un tableau de 50 chaînes de caractères, nomme **tabMots**.

Exemple :

```
string[] tabMots = new string[10] { "adherente", "coktail", "delation", "dahlia", "nationale",  
"rafteur", "macchabee", "schlitteur", "talquer", "twister" };
```

- Au démarrage du jeu, la position du mot mystère dans le tableau **tabMots** est générée aléatoirement par une fonction **generePlace()** et le mot mystère est stocké dans une variable globale **motMystere**.

```
private int generePlace()  
{  
    Random rnd = new Random();  
    int pos = rnd.Next(0, 7);  
    return (pos);  
}
```

- Chaque fois que le joueur propose une lettre dans la zone **txtProposition**, le programme vérifie la présence de cette lettre dans le mot. Si le verdict est positif, la lettre apparaît à la (ou les) position(s) adéquates et une image apparaît sur une PictureBox au bas du formulaire :



- Si la lettre proposée ne figure pas dans le mot, l'image d'une potence apparaît en bas du formulaire !

Travail à faire :

- 1) Concevoir une procédure **initMot()** qui génère, de manière dynamique, autant de composants Label dans un panel **pnlMot**, en prenant en compte les contraintes suivantes :

- les différentes lettres du mot doivent être espacées de 50 pixels
- seule la première et la dernière lettre doivent afficher la lettre correspondante
- si le mot contient un trait d'union ou une apostrophe, ils doivent être affichés
- le tag de chaque composant doit indiquer la position de la lettre dans le mot

- 2) Compléter à présent le code de la procédure événementielle **txtProposition_KeyPress** de manière à :

- tester si l'utilisateur vient de taper sur la touche Entrée :
- récupérer la lettre saisie dans la zone txtProposition (pas plus d'une lettre autorisée !)
- vérifier si cette lettre figure ou non dans le mot mystère
- si c'est le cas, afficher la lettre aux positions adéquates

(attention, pour cette dernière opération, vous devrez parcourir l'ensemble des labels et vérifier leur tag, pour rechercher celles correspondant aux différentes occurrences de la lettre dans le mot)

```
private void txtProposition_KeyPress(object sender, KeyPressEventArgs e)
{
    /.../
}
```

3) Compléter le code de la procédure événementielle **btnRecommencer_Click** qui permet de :

- supprimer les zones de texte présentes dans le groupbox **grpMot**
- générer un nouveau mot mystère
- procéder à la mise en place de ce nouveau mot à l'aide de la procédure **initMot** conçue à la question 1

```
private void btnRecommencer_Click(object sender, EventArgs e)
{
    /.../
}
```

4) Expliquer, dans les grandes lignes (en mentionnant les étapes par lesquelles passer, *mais sans forcément les coder*) comment procéder si l'on voulait permettre à l'utilisateur de proposer une lettre non pas en la tapant dans la zone de texte **txtProposition**, mais tout simplement en l'écrivant directement à la position où l'on suppose qu'elle se trouve, donc une de des zones de texte générées aléatoirement