# TP : le jeu du plus ou du moins

L’idée est de réaliser le classique jeu du plus et du moins …  
Je vous rappelle les règles. L’ordinateur nous calcule un nombre aléatoire et nous devons le deviner. À chaque saisie, il nous indique si le nombre saisi est plus grand ou plus petit que le nombre à trouver. Une fois trouvé, il nous indique en combien de coups nous avons réussi à trouver le nombre secret.  
  
Pour ce TP, vous savez presque tout faire. Il ne vous manque que l’instruction pour obtenir un nombre aléatoire. La voici, cette instruction permet de renvoyer un nombre compris entre 0 et 100 (exclu). Ne vous attardez pas trop sur sa syntaxe, nous aurons l’occasion de comprendre exactement de quoi il s’agit dans le chapitre suivant :  
  
**Code : C# -**[**Sélectionner**](http://www.siteduzero.com/tutoriel-3-545066-tp-le-jeu-du-plus-ou-du-moins.html)

|  |  |
| --- | --- |
| 1 | int valeurATrouver = **new** Random().Next(0, 100); |

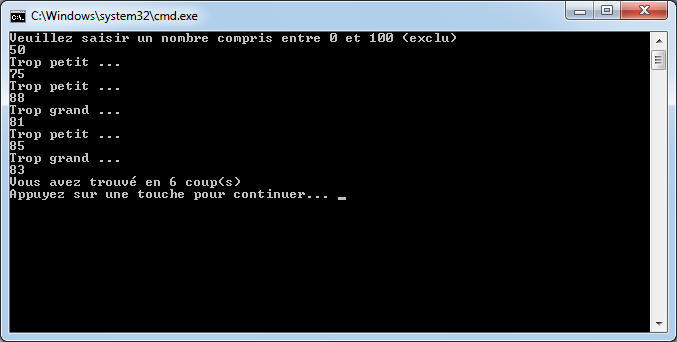
Le principe est grosso modo le suivant : tant qu’on n’a pas trouvé la bonne valeur, nous devons en saisir une nouvelle. Dans ce cas, la console nous indique si la valeur est trop grande ou trop petite. Il faudra bien sur incrémenter un compteur de coups à chaque essai.  
  
N’oubliez pas de gérer le cas où l’utilisateur saisit n’importe quoi. Nous ne voudrions pas que notre premier jeu ait un bug qui fasse planter l’application !  
  
Allez, je vous en ai trop dit. C’est à vous de jouer. Bon courage.

**Correction**

Voici ma correction de ce TP.   
  
Bien sûr, il existe beaucoup de façon de réaliser ce petit jeu. S’il fonctionne, c’est que votre solution est bonne. Ma solution fonctionne, la voici :  
  
**Code : C# -**[**Sélectionner**](http://www.siteduzero.com/tutoriel-3-545066-tp-le-jeu-du-plus-ou-du-moins.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | **static** **void** Main(string[] args)  {  int valeurATrouver = **new** Random().Next(0, 100);  int nombreDeCoups = 0;  bool trouve = **false**;  Console.WriteLine("Veuillez saisir un nombre compris entre 0 et 100 (exclu)");  **while** (!trouve)  {  string saisie = Console.ReadLine();  int valeurSaisie;  **if** (int.TryParse(saisie, **out** valeurSaisie))  {  **if** (valeurSaisie == valeurATrouver)  trouve = **true**;  **else**  {  **if** (valeurSaisie < valeurATrouver)  Console.WriteLine("Trop petit ...");  **else**  Console.WriteLine("Trop grand ...");  }  nombreDeCoups++;  }  **else**  Console.WriteLine("La valeur saisie est incorrecte, veuillez recommencer ...");  }  Console.WriteLine("Vous avez trouvé en " + nombreDeCoups + " coup(s)");  } |

On commence par obtenir un nombre aléatoire avec l’instruction que j’ai fournie dans l’énoncé. Nous avons ensuite les initialisations de variables. L’entier nombreDeCoupsva permettre de stocker le nombre d’essai et le booléen « trouve » va permettre d’avoir une condition de sortie de boucle.  
  
Notre boucle démarre et ne se terminera qu’une fois que le booléen « trouve » sera passé à vrai (**true**).   
Dans le corps de la boucle, nous demandons à l’utilisateur de saisir une valeur que nous essayons de convertir en entier. Si la conversion échoue, nous l’indiquons à l’utilisateur et nous recommençons notre boucle. Notez ici que je n’incrémente pas le nombre de coups, jugeant qu’il n’y a pas lieu de pénaliser le joueur parce qu’il a mal saisi ou qu’il a renversé quelque chose sur son clavier juste avant de valider la saisie.  
  
Si par contre la conversion se passe bien, nous pouvons commencer à comparer la valeur saisie avec la valeur à trouver. Si la valeur est la bonne, nous passons le booléen à vrai, ce qui nous permettra de sortir de la boucle et de passer à la suite. Sinon, nous afficherons un message pour indiquer si la saisie est trop grande ou trop petite en fonction du résultat de la comparaison.  
Dans tous les cas, nous incrémenterons le nombre de coups.  
  
Enfin, en sortie de boucle, nous indiquerons sa victoire au joueur ainsi que le nombre de coups utilisés pour trouver le nombre secret.  
Une partie de jeu pourra être :



**Aller plus loin**

Il est bien sûr toujours possible d’améliorer le jeu. Nous pourrions par exemple ajouter un contrôle sur les bornes de la saisie. Ainsi, si l’utilisateur saisit un nombre supérieur ou égal à 100 ou inférieur à 0, nous pourrions lui rappeler les bornes du nombre aléatoire.  
  
De même, à la fin, plutôt que d’afficher « coup(s) », nous pourrions tester la valeur du nombre de coups. S’il est égal à 1, on affiche « coup » au singulier, sinon « coups » au pluriel.  
  
La boucle pourrait également être légèrement différente. Plutôt que de tester la condition de sortie sur un booléen, nous pourrions utiliser le mot clé **break**. De même, nous pourrions alléger l’écriture avec le mot clé **continue**.  
  
Par exemple :  
  
**Code : C# -**[**Sélectionner**](http://www.siteduzero.com/tutoriel-3-545066-tp-le-jeu-du-plus-ou-du-moins.html)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | **static** **void** Main(string[] args)  {  int valeurATrouver = **new** Random().Next(0, 100);  int nombreDeCoups = 0;  Console.WriteLine("Veuillez saisir un nombre compris entre 0 et 100 (exclu)");  **while** (**true**)  {  string saisie = Console.ReadLine();  int valeurSaisie;  **if** (!int.TryParse(saisie, **out** valeurSaisie))  {  Console.WriteLine("La valeur saisie est incorrecte, veuillez recommencer ...");  **continue**;  }  **if** (valeurSaisie < 0 || valeurSaisie >= 100)  {  Console.WriteLine("Vous devez saisir un nombre entre 0 et 100 exclu ...");  **continue**;  }  nombreDeCoups++;  **if** (valeurSaisie == valeurATrouver)  **break**;  **if** (valeurSaisie < valeurATrouver)  Console.WriteLine("Trop petit ...");  **else**  Console.WriteLine("Trop grand ...");  }  **if** (nombreDeCoups == 1)  Console.WriteLine("Vous avez trouvé en " + nombreDeCoups + " coup");  **else**  Console.WriteLine("Vous avez trouvé en " + nombreDeCoups + " coups");  } |

Tout ceci est une question de goût. Je préfère personnellement la version précédente n’aimant pas trop les **break** et les **continue**. Mais après tout, chacun fait comme il préfère, l’important est que nous amusions à écrire le programme et à y jouer. :D

Voilà, ce TP est terminé.  
  
Vous avez pu voir finalement que nous étions tout à fait capables de réaliser des petites applications récréatives. Personnellement, j'ai commencé à m'amuser à faire de la programmation en réalisant toute sorte de petits programmes de ce genre.  
  
Je vous encourage fortement à essayer de créer d'autres programmes vous-mêmes, pourquoi pas à proposer aux autres vos idées.  
Plus vous vous entraînerez à faire des petits programmes simples et plus vous réussirez à appréhender les subtilités de ce que nous avons appris.  
  
Bien sûr, plus tard, nous serons capables de réaliser des applications plus compliquées ... Cela vous tente ? Alors continuons la lecture. :)