

## EXERCICE 13 – ROCHE, PAPIER, CISEAUX

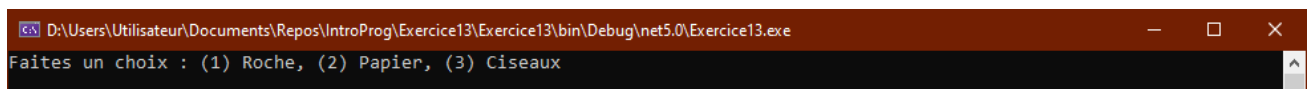
Préparé par Benjamin Lemelin et Pierre Poulin le 18 janvier 2023

## 1 Travail à effectuer

Ouvre le projet de départ fourni avec cet énoncé. Il devrait déjà contenir un fichier **Program.cs**. L'application que nous allons effectuer est un simple Roche, Papier, Ciseaux. Tout ce que vous avez à faire, c'est de compléter les fonctions déjà existantes.

### 1.1 Lire un choix

Complétez la fonction **PromptInt** permettant de lire un choix à partir de la console. Cette fonction permet de poser n'importe quelle question. Elle retourne ce que l'utilisateur a tapé sous la forme d'un entier.



```
D:\Users\Utilisateur\Documents\Repos\IntroProg\Exercice13\Exercice13\bin\Debug\net5.0\Exercice13.exe
Faites un choix : (1) Roche, (2) Papier, (3) Ciseaux
```

#### Information

Pour simplifier, vous n'avez pas à valider l'entrée de l'utilisateur. Assumez donc qu'il tape un nombre valide. Pour convertir un **string** en **int**, utilisez la fonction **int.Parse()**.

```
int value = int.Parse(Console.ReadLine());
```

Cependant, si vous avez envie d'un défi, vous êtes encouragés à effectuer la validation. Pour cela, faites usage de la méthode **TryParse** fournie avec le projet de départ. Par exemple :

```
int value;
bool success;
success = int.TryParse(Console.ReadLine(), out value);
```

### 1.2 Générer le coup de l'adversaire

Le programme doit générer de manière aléatoire un choix entre 1, 2 et 3, ce qui correspond respectivement à roche, papier et ciseaux. Complétez donc la fonction **GenerateRandomNumber**.

#### Information

Pour rappel, vous pouvez générer un nombre aléatoire avec cet extrait de code :

```
Random rnd = new Random();
int number = rnd.Next(0, 10); //Génère entre 0 et 9 inclusivement.
```

### 1.3 Déterminer un gagnant

Complétez la fonction **DetermineWinner** permettant de déterminer le gagnant d'une partie de Roche, Papier, Ciseaux. Pour rappel, la roche bat les ciseaux, les ciseaux battent le papier et le papier bat la roche.

Joueur 1	Joueur 2	Gagnant
Roche	Papier	Joueur 2
Roche	Ciseaux	Joueur 1
Roche	Roche	Partie nulle
Papier	Roche	Joueur 1
Papier	Ciseaux	Joueur 2
Papier	Papier	Partie nulle
Ciseaux	Papier	Joueur 1
Ciseaux	Roche	Joueur 2
Ciseaux	Ciseaux	Partie nulle

Cette fonction n'est pas compliquée, mais est tout de même assez longue à écrire. Ne soyez donc pas surpris si vous avez plusieurs conditions.



#### Prudence

Que devriez-vous retourner ? Un entier ? Une chaîne de caractères ? De préférence, créez des constantes et retournez la valeur de ces constantes. Pour cet exercice, utilisez un entier.

En temps normal, nous aurions retourné [une énumération](#), mais nous ne verrons cela que durant le cours *Programmation Objet I* à la session prochaine. Cela dit, rien ne vous empêche de l'essayer.

### 1.4 Convertir un choix en chaîne de caractère

Complétez la fonction **ChoiceToText** permettant de convertir le choix d'un joueur (1, 2 ou 3) sous la forme d'une chaîne de caractère (« Roche », « Papier » ou « Ciseaux »). Pour ce numéro, faites usage d'un [switch case](#).

### 1.5 Afficher le résultat

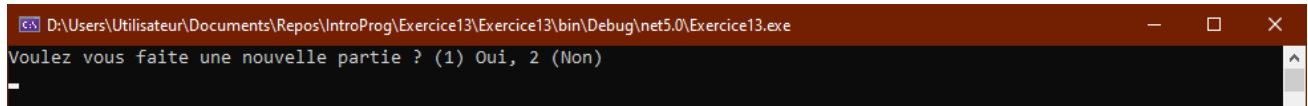
Complétez la fonction **ShowGameResults** permettant d'afficher le résultat d'une partie. Les paramètres **player1Choice** et **player2Choice** devraient contenir 1, 2 ou 3, soit le choix de chaque joueur. Le dernier paramètre **winner** devrait contenir le numéro du gagnant, soit 1 ou 2. Si c'est une partie nulle, envoyez 0 à la place.

La fonction devrait produire ce genre de résultat :

```
D:\Users\Utilisateur\Documents\Repos\IntroProg\Exercice13\Exercice13\bin\Debug\net5.0\Exercice13.exe
Choix du joueur 1 : Roche
Choix du joueur 2 : Papier
Gagnant : Joueur 1
```

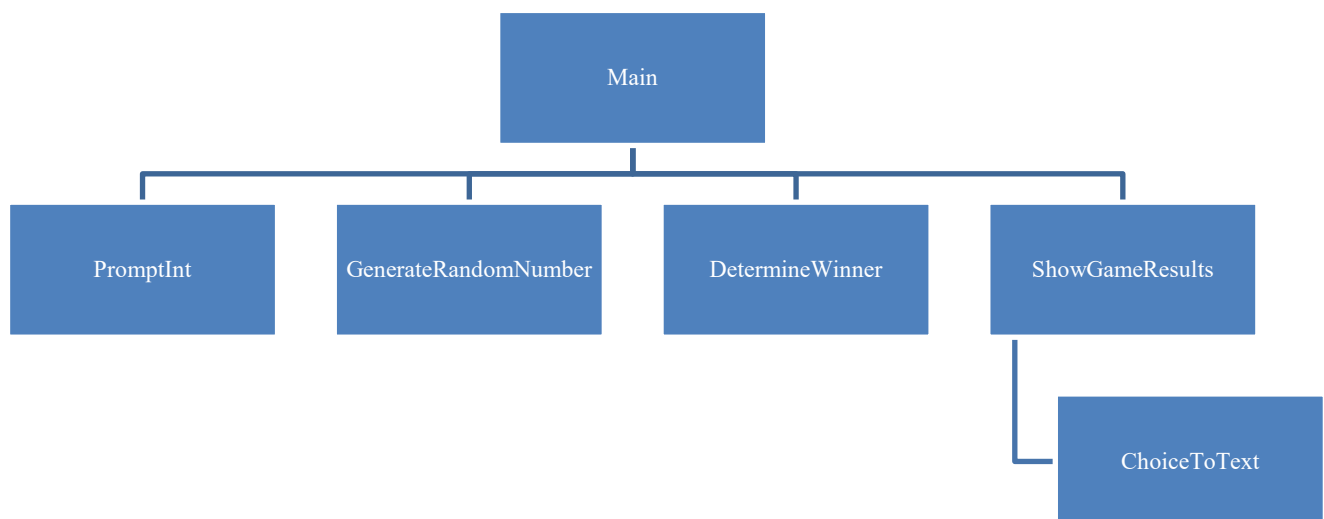
## 1.6 Boucle de jeu

Allez dans la fonction **Main** et complétez le restant de l'application en utilisant les fonctions que vous venez de compléter. Faites en sorte que le joueur puisse jouer le nombre de parties qu'il veut. Entre chaque partie, videz la console via la fonction **Console.Clear()**.



```
D:\Users\Utilisateur\Documents\Repos\IntroProg\Exercice13\Exercice13\bin\Debug\net5.0\Exercice13.exe
Voulez vous faite une nouvelle partie ? (1) Oui, 2 (Non)
_
```

## 2 Diagramme hiérarchique



## 3 Modalités de remise

Remettez votre projet Visual Studio sur LÉA, dans la section travaux, à l'intérieur d'une archive *Zip*. Supprimez tous les dossiers temporaires, à savoir les dossiers **.vs**, **bin** et **obj**.