

# 1 Tp1\_\_Le\_\_nombre\_\_secret

## 1.1 Buts:

- Utiliser les structures de contrôle (loop, if).
- Utiliser un “crate” externe pour la lecture à la console et la génération de nombres aléatoires.
- Déclarer des variables mutables et immutables.
- Paniquer!

## 1.2 Énoncé:

Écrire un programme permettant de jouer au nombre secret. Le but est de faire chercher à l'utilisateur un nombre déterminé aléatoirement par l'ordinateur. Le nombre de coups mis pour trouver la solution sera compté et affiché en fin de la partie. La borne maximum pour le choix du nombre caché sera demandée à l'utilisateur.

### 1.2.1 Nombres aléatoires

Afin d'utiliser la librairie de nombres aléatoires de Rust, il faut importer la librairie `rust_hepia_lib` (qui utilise la librairie `rand` de Rust). Pour ce faire il faut ajouter une dépendance dans le fichier `Cargo.toml`. Après la ligne

```
[dependencies]
```

il faut rajouter la ligne

```
rust_hepia_lib = { path = "$CHEMIN/rust_hepia_lib" }
```

où `$CHEMIN` est l'endroit où vous avez cloné la librairie.

Puis, il faut importer la librairie externe dans le `main.rs` avec la ligne

```
extern crate rust_hepia_lib;
```

Pour l'utiliser il faut utiliser la fonction de `rust_hepia_lib`

```
rust_hepia_lib::gen(min, max+1);
```

et stocker le résultat dans une variable.

### 1.2.2 Lecture de l'entrée au clavier

Afin de lire un entier entré au clavier, pour simplifier il faut appeler une fonction de `rust_hepia_lib`. Cela se fait avec la fonction suivante

```
rust_hepia_lib::read_int(); // lecture d'un entier quelconque
```

Puis il faut assigner la sortie de cette fonction à une variable. Il ne faut pas oublier d'annoter le type de la sortie, autrement le compilateur va râler, car il n'arrivera pas à deviner le type.

### **1.2.3 Déroulement du programme**

Au début du programme, l'utilisateur entre le nombre maximum avec lequel il veut jouer. Le programme générera un nombre pris au hasard entre 0 et ce maximum. Dans une boucle, on demandera à l'utilisateur d'entrer un nombre et on lui répondra par plus petit ou plus grand et ceci jusqu'à la découverte du bon nombre.

### **1.2.4 Fin de programme**

Chercher le moyen optimal (en moyenne) de trouver le nombre aléatoire. Afficher en fin de partie un commentaire sur le nombre de coups mis par le joueur pour déterminer le nombre secret.

### **1.2.5 Bonus**

Écrire un algorithme qui trouve le nombre secret en un nombre minimal de coups (en moyenne).