

# **Conventions de programmation**

# Indentation

```
// First level
{
    // Second level
    {
        // Third level

    }
}
```

# Position des accolades

Votre solution est la meilleure !

```
if(...) // Approche la plus courante
{
    ...
}
else
{
    ...
}
ou
if (...) {
    ...
} else {
    ...
}
```

# Position des accolades

Les if sans accolades sont à éviter !

```
if (...)
    ...
```

# Identifiant de variable

L'identifiant d'une variable commence par une minuscule et possède une majuscule à chaque nouveau mot.

L'identifiant représente clairement le rôle de la variable.

```
int bestScore;  
std::string gameRulesText;
```

# Identifiant de variable

Les booléens sont précédés de is- ou has-.

```
bool isPlayerReady;
```

```
bool hasGameStarted;
```

# Notation hongroise

La notation hongroise consiste à précéder l'identifiant d'une variable d'un préfix correspondant à son type.

```
int iBestScore;
```

```
bool bGameStarted;
```

```
std::string sGameRulesText;
```

Cette notation est adaptée aux langages faiblement typés !

# Identifiant de variable

En C/C++, on adopte la notation hongroise pour les pointeurs :

```
int bestScore;
```

```
int *pBestScore;
```

Les pointeurs sont sensibles et sujets à erreurs graves.



# Identifiant de variable

En C/C++, on peut préfixer les identifiants de références de r- ou ref- :

```
int bestScore;
```

```
int &rBestScore = bestScore;
```

# Identifiant de fonction

Les identifiants de fonctions correspondent généralement à une action.

Ci-dessous, trois notations existantes en fonction des majuscules en début de mots :

```
int EvaluateDifficulty(...); // C#, C++
```

```
void movePlayer(...); // C, C++
```

```
string get_name(...); // C
```

# Identifiant de fonction

Les identifiants de fonctions dont le rôle est d'évaluer et dont le retour est booléen commencent souvent par is- ou has- :

```
bool IsAlive(...);
```

```
bool hasPower(...);
```

# Identifiant de fonction

Les identifiants de fonctions dont le rôle est de lire ou d'écrire une donnée commencent souvent par get- ou set- :

```
std::string getName(...);
```

```
void SetLife(...);
```

# Constante

Les identifiants de constantes sont souvent écrits en majuscules séparées par des "\_" entre chaque mot.

```
const int NB_ROUND = 5;  
const float LIFE_MAX = 100.0f;  
#define PI 3.1415f
```

# Enum

Les enums et leurs valeurs sont généralement tout en majuscules :

```
enum STATE {  
    IDLE,  
    RUNNING  
};
```

# Struct

Les structs commencent généralement par une majuscule avec une majuscule par mot.

```
struct Data {  
    ...  
};
```

# Fichier

Les identifiants de fichiers correspondent au rôle, à l'identifiant de fonction principale, à l'identifiant de struct principale ou à l'identifiant de classe principale que le fichier contient.

`main.cpp`

`Board.cpp`

`mathUtils.h`