

Algorithmique & Programmation

Cas d'étude : Jeu du pendu

yann.secq@univ-lille.fr

ABIDI Sofiene, ALMEIDA COCO Amadeu, BONEVA Iovka, CASTILLON Antoine,
DELECROIX Fabien, LEPRETRE Éric, Timothé ROUZÉ, SANTANA MAIA Deise,
SECQ Yann

```

Mot secret (nombre d'erreur = 0) : *****
Veuillez entrer une lettre minuscule de l'alphabet : a
Mot secret (nombre d'erreur = 0) : *****a*****
Veuillez entrer une lettre minuscule de l'alphabet : i
Mot secret (nombre d'erreur = 0) : i*****a*i**
Veuillez entrer une lettre minuscule de l'alphabet : e
Mot secret (nombre d'erreur = 1) : i*****a*i**
Veuillez entrer une lettre minuscule de l'alphabet : r
Mot secret (nombre d'erreur = 1) : i***r*a*i**
Veuillez entrer une lettre minuscule de l'alphabet : s
Mot secret (nombre d'erreur = 2) : i***r*a*i**
Veuillez entrer une lettre minuscule de l'alphabet : n
Mot secret (nombre d'erreur = 2) : in**r*a*i*n
Veuillez entrer une lettre minuscule de l'alphabet : o
Mot secret (nombre d'erreur = 2) : in*or*a*ion
Veuillez entrer une lettre minuscule de l'alphabet : f
Mot secret (nombre d'erreur = 2) : infor*a*ion
Veuillez entrer une lettre minuscule de l'alphabet : m
Mot secret (nombre d'erreur = 2) : informa*ion
Veuillez entrer une lettre minuscule de l'alphabet : t
Bravo, vous avez gagné !

```

Jeu du pendu

1. Quelles informations ?
2. Quels traitements ?
3. Quels types et structure de données ?

```

Mot secret (nombre d'erreur = 0) : *****
Veuillez entrer une lettre minuscule de l'alphabet : a
Mot secret (nombre d'erreur = 0) : a*****
Veuillez entrer une lettre minuscule de l'alphabet : e
Mot secret (nombre d'erreur = 0) : a*****e
Veuillez entrer une lettre minuscule de l'alphabet : i
Mot secret (nombre d'erreur = 0) : a****i****e
Veuillez entrer une lettre minuscule de l'alphabet : o
Mot secret (nombre d'erreur = 0) : a**o*i****e
Veuillez entrer une lettre minuscule de l'alphabet : t
Mot secret (nombre d'erreur = 0) : a**o*it**e
Veuillez entrer une lettre minuscule de l'alphabet : f
Mot secret (nombre d'erreur = 1) : a**o*it**e
Veuillez entrer une lettre minuscule de l'alphabet : e
Mot secret (nombre d'erreur = 2) : a**o*it**e
Veuillez entrer une lettre minuscule de l'alphabet : e
Mot secret (nombre d'erreur = 3) : a**o*it**e
Veuillez entrer une lettre minuscule de l'alphabet : e
Mot secret (nombre d'erreur = 4) : a**o*it**e
Veuillez entrer une lettre minuscule de l'alphabet : e
Désolé, vous avez perdu ...

```

Jeu du Pendu

- **Informations** : un mot secret, une visibilité associée à chaque lettre du mot secret
- **Traitements** (complexes)
 - Mettre à jour le mot en fonction de la lettre
 - Déterminer si le joueur a gagné
- **Types et structure de données**
 - **Lettre** = caractère + visible ou pas (nouveau type)
 - Un mot devient un tableau de **Lettre**

Jeu du Pendu

- **Informations** : un mot secret, une visibilité associée à chaque lettre du mot secret
- **Traitements** (complexes)
 - Mettre à jour le mot en fonction de la lettre
 - Déterminer si le joueur a gagné
- **Types et structure de données**
 - **Lettre** = caractère + visible ou pas (nouveau type)
 - **Mot** = un tableau de lettres

Type Lettre

```
class Lettre {  
    char caractere;  
    boolean visible = false;  
}
```

- **Type** : un champs pour stocker un caractère + un booléen pour indiquer si le caractère a été découvert ou pas (initialisé à `false` car aucune lettre n'est visible au début du jeu)
- **Traitements**
 - Créer une nouvelle valeur de type `Lettre`
 - Afficher une valeur de type `Lettre`
 - Créer un tableau de `Lettre` à partir d'une chaîne
 - Afficher un tableau de `Lettre`

```
Lettre newLettre(char caractere)
```

```
String toString(Lettre l)
```

```
Lettre[] convertir(String mot)
```

```
String toString(Lettre[] mot)
```

Type Lettre

```
class Lettre {  
    char caractere;  
    boolean visible = false;  
}
```

```
Lettre newLettre(char caractere) {  
    Lettre l = new Lettre();  
    l.caractere = caractere;  
    return l;  
}
```

```
String toString(Lettre l) {  
    if (l.visible) {  
        return "" + l.caractere;  
    }  
    return "*";  
}
```

```
Lettre[] convertir(String mot) {  
    Lettre[] lettres = new Lettre[length(mot)];  
    for (int idx=0; idx < length(mot); idx++) {  
        lettres[idx] = newLettre(charAt(mot, idx));  
    }  
    return lettres;  
}
```

```
String toString(Lettre[] mot) {  
    String res = "";  
    for (int idx=0; idx < length(mot,1); idx++) {  
        res = res + toString(mot[idx]);  
    }  
    return res;  
}
```

DS2 passé !
On s'autorise ce
type de raccourci
maintenant :)

Jeu du Pendu

- **Traitements complexes**
 - **Déterminer si le joueur a gagné**
 - Si une lettre non visible encore présente, la partie n'est pas gagnée, sinon c'est le cas
 - **Mettre à jour le mot en fonction de la lettre**
 - Rendre visible les occurrences de la lettre présente et savoir si au moins une lettre a été modifiée

Jeu du Pendu

- **Déterminer si le joueur a gagné**
 - Si une lettre non visible est encore présente, la partie n'est pas gagnée, sinon c'est le cas

**DS2 passé !
On s'autorise
ce type de
raccourci
maintenant :)**

```
boolean decouvert(Lettre[] mot) {  
    for (int idx=0; idx < length(mot); idx++) {  
        if (!mot[idx].visible) {  
            return false;  
        }  
    }  
    return true;  
}
```


Jeu du Pendu

- **Mettre à jour le mot en fonction de la lettre**
 - Rendre visible les occurrences de la lettre présente et savoir si au moins une lettre a été modifiée

```
boolean miseAJour(Lettre[] mot, char lettre) {  
    boolean changement = false;  
    for (int idx=0; idx < length(mot); idx++) {  
        if (!mot[idx].visible && mot[idx].caractere == lettre) {  
            mot[idx].visible = true;  
            changement = true;  
        }  
    }  
    return changement;  
}
```

Jeu du Pendu

Algorithme principal

- Tableau de mots et choix aléatoire d'un des mots
- Conversion de la chaîne en mot (`Lettre[]`)
- Tant que mot non découvert et `nbErreurs` acceptable
 - Saisir une nouvelle lettre
 - Mettre à jour le mot secret et incrémenter le `nbErreurs` si aucune nouvelle lettre découverte
- Indiquer si la partie est gagnée ou pas

Algorithme principal

```
void algorithm() {
    final int MAX_ERREURS = 5;
    final String[] MOTS = new String[]{"algorithm", "machine",
                                         "information", "langage", "programme"};
    Lettre[] secret = convertir(motAuHasard(MOTS));
    int nbErreurs = 0;
    while (nbErreurs < MAX_ERREURS && !decouvert(secret)) {
        println("Mot secret (nb erreur = "+nbErreurs+"): "+ toString(secret));
        char lettre = readChar(); // contrôle de saisie ?
        if (!miseAJour(secret, lettre)) {
            nbErreurs++;
        }
    }
    if (nbErreurs == MAX_ERREURS) {
        println("Désolé, vous avez perdu ...");
    } else {
        println("Bravo, vous avez gagné !");
    }
}
```