

INFO-H-100 : Introduction à la programmation

Projet 2 : *Jeu de lettres*

On vous demande de réaliser deux jeux de lettres à l'aide du langage C++ : le jeu du pendu¹ et le jeu de la recherche dichotomique².

Les deux sections suivantes sont basées sur Wikipedia.

Le pendu

Le pendu est un jeu consistant à trouver un mot en devinant quelles sont les lettres qui le composent. Le jeu se joue traditionnellement à deux, avec un papier et un crayon, selon un déroulement bien particulier.

Les deux joueurs dans cet exemple s'appellent Jacques et Thierry.

- Thierry pense à un mot et dessine une rangée de tirets, chacun correspondant à une lettre de ce mot.
- Jacques annonce une lettre.
- La lettre fait-elle partie du mot ?
 - Oui : Thierry l'inscrit à sa place autant de fois qu'elle se trouve dans le mot.
 - Non : Thierry dessine le premier trait du pendu.
- Le jeu se poursuit jusqu'à ce que :
 - Jacques gagne la partie en trouvant toutes les lettres du mot ou en le devinant correctement.
 - Thierry gagne la partie en complétant le dessin du pendu.

La recherche dichotomique

Le jeu de la recherche dichotomique est un jeu consistant à trouver un mot que l'adversaire a choisi. A chaque étape du jeu, le joueur propose un mot et l'adversaire indique au joueur si le mot qu'il a choisi se trouve avant ou après (dans l'ordre alphabétique) le mot proposé par le joueur.

Dans l'exemple suivant, Jacques est l'adversaire et Thierry le joueur.

- Jacques choisit le mot "programmation" et le garde secret.
- Thierry propose le mot "informatique" et Jacques lui indique que le mot à deviner se trouve après.
- Thierry propose le mot "système" et Jacques lui indique que le mot à deviner se trouve avant.
- Thierry propose alors le mot "mémoire" et Jacques lui indique que le mot à deviner se trouve après.
- Thierry propose enfin le mot "programmation". Jacques le félicite : Thierry a gagné en 4 coups.

¹[http://fr.wikipedia.org/wiki/Le_pendu_\(jeu\)](http://fr.wikipedia.org/wiki/Le_pendu_(jeu))

²<http://fr.wikipedia.org/wiki/Dichotomie#Exemple>

Programme à réaliser

On vous demande de réaliser à l'aide uniquement de la matière vue au cours et aux séances d'exercices un jeu de pendu et un jeu de recherche dichotomique en *C++*. Votre programme devra se dérouler comme suit :

Tout d'abord, votre programme chargera en mémoire le dictionnaire disponible sous forme de fichier sur la page web des projets³. Ce fichier, `dico.txt`, devra obligatoirement se trouver dans le même répertoire que l'exécutable de votre projet. Il contient une liste de mots triés alphabétiquement. Les mots sont séparés par un retour à la ligne.

Ensuite, votre programme propose à l'utilisateur de choisir entre les deux jeux et lance le jeu choisi. Pour chaque jeu, l'ordinateur choisit un mot au hasard dans le dictionnaire et l'utilisateur essaie de le deviner. Tous les mots choisis par l'ordinateur et saisis par l'utilisateur doivent faire partie du dictionnaire fourni, tant pour le pendu que pour la recherche dichotomique. Une fois une partie terminée, le programme propose à l'utilisateur soit de continuer, soit de choisir un autre jeu, soit d'arrêter.

Apports personnels

Les apports personnels seront valorisés à hauteur de 4 points sur 20. En d'autres mots, un projet parfait sans apport personnel aura une valeur de 16 points sur 20. Ces apports personnels peuvent être par exemple des règles ignorées dans la version simplifiée proposée ci-dessus.

Voici quelques exemples d'apports personnels :

- Gestion des scores (difficulté moyenne) ;
- Gestion de plusieurs joueurs (difficulté moyenne) ;
- Enregistrer une partie sur fichier et pouvoir la rejouer (difficile) ;
- Joueur "automatique" (difficulté moyenne) ;
- Affichage amélioré, par exemple grâce à l'Art ASCII⁴ (difficulté moyenne) ;
- Enregistrement des meilleurs scores (facile) ;
- Fichier de configuration (temps, difficulté, nombre de lettres d'un mot, ...) (difficulté moyenne).

³<http://cs.ulb.ac.be/public/teaching/infoh100/projets>

⁴http://fr.wikipedia.org/wiki/Art_ASCII

Rapport

Un rapport dactylographié vous est demandé. Celui-ci devra comprendre :

- Une page de garde indiquant clairement vos noms, numéros de matricules et votre série.
- Une **brève** introduction au problème.
- Une documentation de vos fonctions, c’est à dire pour chaque fonction : son prototype, une description succincte de la tâche qu’elle remplit, ses paramètres et leur domaine (le type et les valeurs possibles), le type de la valeur de retour et son domaine, les erreurs possibles et un exemple d’utilisation. Par exemple :

```
int daysInMonth(int month, int year) : Calcule le nombre de jours du mois (month) de l’année (year) fourni en paramètres.
```

 - Paramètre `int month` : entier de 1 à 12 représentant le mois.
 - Paramètre `int year` : entier de 0 à 9999 représentant l’année au format AAAA.
 - Valeur de retour entière indiquant le nombre de jours du mois.
 - Erreurs : renvoie -1 si les paramètres ne respectent pas les domaines.
 - Exemple : `daysInMonth(04,2008)` renvoie l’entier 30.
- Une explication claire de vos choix algorithmiques pour :
 1. la vérification que la lettre proposée lors d’un tour de pendu se trouve dans le mot à deviner et à quelle(s) position(s) elle s’y trouve.
 2. la recherche dans le dictionnaire.
- Une (ou plusieurs) capture d’écran.
- Le code source en police de caractère à empattement fixe (par exemple `courier`).

Consignes

Le projet se fera obligatoirement par groupes de deux étudiants de la même série.

Le rapport au format papier (uniquement des feuilles et des agrafes, **pas de plastique** ni de reliure) devra être rendu au bureau UB4.131. Le code source de votre application sera envoyé par mail à l’adresse `ulb.code+infoh100@gmail.com`. Ce mail devra avoir comme sujet “INFO-H-100 - Projet 2 - Nom1, Prénom1 et Nom2, Prénom2”. Tout cela devra être rendu pour le **9 mai à 12h30**. Les groupes remettant leur projet après cette limite seront pénalisés.

Vous devrez défendre votre projet devant les assistants à un horaire fixé ultérieurement.

Votre code devra compiler sans avertissement à l’aide de la commande `g++ -Wall` installée dans les salles machines. Les seules bibliothèques autorisées sont `iostream`, `ctime`, `cstdlib` et `fstream`.

L’évaluation de ce projet prendra en compte les points suivants :

- La résolution du problème énoncé.
- Le respect scrupuleux de l’énoncé.
- Le respect scrupuleux des conventions et des règles de bonne pratique publiés sur le site web des projets.
- La qualité du rapport : intérêt, présentation, soin, concision, grammaire et orthographe.
- Votre défense en salle machine devant les assistants.
- L’efficacité de vos algorithmes et structures de données, en restant bien sûr dans le cadre de la matière vue aux cours et aux séances d’exercices.

Les élèves assistants sont disponibles tous les midis dans la salle Socrate afin de répondre à vos questions.

Bon travail !

Lecture et écriture sur fichier

La librairie `fstream`⁵ est une librairie standard permettant de lire et écrire dans un fichier. Cette librairie offre deux nouveaux types de données : `ifstream` pour la lecture de fichier et `ofstream` pour l'écriture. Leur fonctionnement est similaire à celui de `cin` et `cout`.

Voici un exemple de lecture d'un fichier :

```
#include <fstream>
#include <iostream>
using namespace std;

int main()
{
    char chaine[255];
    ifstream entree("test.txt");    //ouvre le fichier test.txt en lecture
                                    //ce fichier se trouve dans le meme dossier
                                    //que le programme

    entree >> chaine;
    while (!entree.eof())          //tant que le fichier n'est pas terminé
    {

        cout << chaine << endl;
        entree >> chaine;          //copie une portion dans le vecteur chaine
    }                               //(chaque portion est séparée par un espace,
                                    // une tabulation ou un retour à la ligne)

    entree.close();                //ferme le fichier

    return 0;
}
```

Voici un exemple d'écriture d'un fichier :

```
#include <fstream>
#include <iostream>
using namespace std;

int main()
{
    int valeur;
    ofstream sortie("truc.txt");    //ouvre le fichier truc.txt en écriture

    cin>>valeur;
    while (valeur != -1)
    {
        sortie << valeur << ',' ;    //écrit dans le fichier les valeurs lues
        cin >> valeur;                //au clavier
    }

    sortie.close();                //ferme le fichier

    return 0;
}
```

⁵<http://www.cplusplus.com/reference/iostream/fstream/>