420-220-SF

ALGORITHMIQUE ET PROGRAMMATION II

TRAVAIL PRATIQUE #2

LE LABYRINTHE

PONDÉRATION : 10 POINTS

TRAVAIL EN ÉQUIPES DE DEUX OBLIGATOIREMENT

OBJECTIFS PÉDAGOGIQUES

Ce travail vise à familiariser les étudiants avec :

- l'utilisation des structures de base en C/C++
- l'utilisation des instructions d'entrées/sorties standards du C/C++
- l'utilisation des concepts de base de la POO
- l'utilisation de l'allocation dynamique
- la manipulation de structures de données dynamiques
- la manipulation de fichiers texte
- le développement d'algorithmes plus complexes
- l'application des standards de conception
- l'application des standards de programmation
- l'application des normes de documentation

MANDAT

Réaliser une application permettant de lire un fichier texte contenant une structure de labyrinthe valide et trouver un chemin vers la sortie.

Produire la documentation pertinente de l'application.

MISE EN CONTEXTE

GLaDOS (Genetic Lifeform and Disk Operating System) a été créé par la société Aperture pour faire de la science.

Vous êtes mandaté par GLaDO, ou Aperture, enfin peu importe afin de créer une nouvelle intelligence artificielle pour les robots d'Aperture!

DESCRIPTION ET TRAITEMENT – LABYRINTHE

DESCRIPTION GÉNÉRALE

Le programme doit d'abord lire un fichier texte contenant une structure de labyrinthe de taille 20 par 20. Ce fichier texte ne contient pas d'erreur et n'a pas à être validé. On considère évidemment que le labyrinthe est fermé (entouré d'un mur) et qu'il est résoluble (voir les exemples exemple1.txt, exemple2.txt et exemple3.txt).

L'application doit valider que le fichier nommé « labyrinthe.txt » est présent. S'il est présent, on considère que l'information contenue est valide et le fichier est lu. Sinon, un message d'erreur est affiché à cet effet et l'application se termine.

Dans le fichier « labyrinthe.txt », on utilise les caractères suivants pour construire le labyrinthe :

- X : Représente un mur (et donc une case non accessible)
- Espace : représente une case où il est possible de passer.
- D : Représente le point de départ du labyrinthe.
- S : Représente la sortie.

Par la suite, l'application affiche les cases à emprunter pour traverser le labyrinthe vers la sortie. L'affichage doit être de format (x, y).

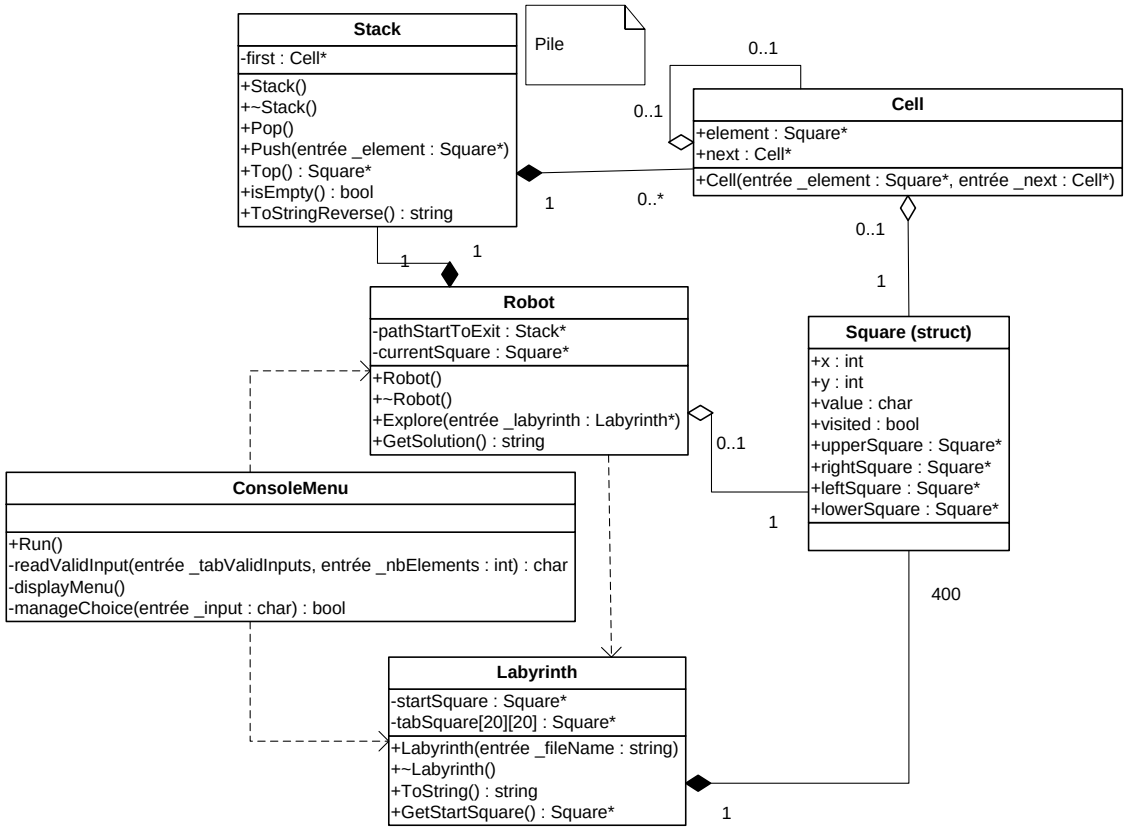
RÈGLES DE TRAITEMENT

Dans le menu principal, les choix « V » pour « Visualiser un labyrinthe », « S » pour « Solutionner un labyrinthe » ou « Q » pour « Quitter » sont acceptés en minuscules ou majuscules. Toute autre entrée générera une erreur avec un message approprié.

Suite à l'entrée de la touche « V », le labyrinthe contenu dans le fichier « labyrinthe.txt » est affiché.

Suite à l'entrée de la touche « S », le labyrinthe contenu dans le fichier « labyrinthe.txt » est solutionné et le chemin complet (coordonnées x et y depuis le départ jusqu'à l'arrivée) est affiché à l'écran.

Voici le diagramme de classes à implémenter :



Vous devez implémenter toutes les méthodes publiques du diagramme de classes. Aucune méthode publique (+) ne peut être ajoutée sans l'approbation de votre professeur mais vous pouvez ajouter autant de méthodes privées (-) que vous voulez (principe d'encapsulation). De plus, vous ne pouvez pas ajouter, modifier ni enlever aucune variable membre.

Pour créer les classes Stack et Cell, vous pouvez partir d'exercices formatifs faits en classe.

La classe Cell peut être une classe interne à la classe Stack pour économiser sur les accesseurs.

Pour la même raison, Square sera une structure et non une classe.

Pour vous aider à débuter, voici la définition de la fonction main et celle de la méthode

```
void main()
{
    ConsoleMenu menu;
    menu.Run();
}

void ConsoleMenu::Run()
{
    char input;
    char tabValidInputs[] = {'V', 'v', 'S', 's', 'Q', 'q'};
    const int NB_ELEMENTS = 6;
    do
    {
        system("cls");
        input = readValidInput(tabValidInputs, NB_ELEMENTS);
    }
    while(manageChoice(input));
}
```

Note1 : Vous ne pouvez pas modifier le code de la fonction main() ni celui de la méthode `ConsoleMenu::Run()`.

Note2 : La méthode `ConsoleMenu::manageChoice(char _input)` doit gérer le choix de l'utilisateur (_input) à l'aide d'un switch case.

ÉVALUATION

Le travail sera évalué selon les critères suivants :

Éléments	Pondération
Implémentation de la classe ConsoleMenu	5%
Implémentation de la classe Stack (et sa classe Cell)	15%
Implémentation de la classe Labyrinth	20%
Implémentation de la classe Robot	25%
Conception de la gamme de test et réalisation des tests	10%
Documentation du code source (interne et externe)	10%
Respect des normes de programmation	10%
Feuille de temps	5%

Conformément à la politique concernant la qualité du français écrit, le travail est sujet à une pénalité pouvant aller jusqu'à 20 % du total des points attribuables. Se référer à la section « **Évaluation** » du plan de cours pour plus de détails sur l'évaluation des travaux.

Si le professeur le juge à propos, tout étudiant(e) pourra être convoqué(e) à une rencontre d'évaluation pour vérifier son degré d'acquisition des connaissances et d'appréhension de la solution proposée.

JB/JD/H2014

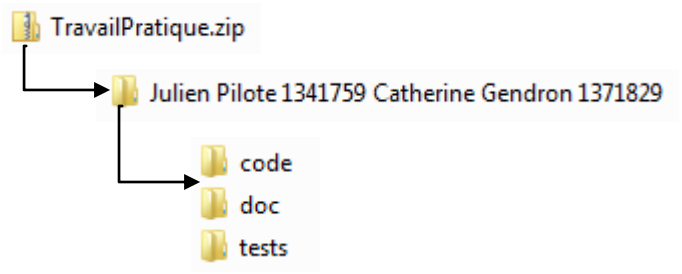
BIENS LIVRABLES

Vous devez remettre sur LEA un fichier ZIP identifié à vos deux noms et contenant les répertoires suivants :

code	Projet complet et épuré, réalisé avec Microsoft Visual Studio.
doc	Documentation externe HTML (via DOXYGEN) et feuille de temps
test	Gamme de tests et preuves d'opérationnalité

REMISES :

Exemple de la structure du fichier ZIP à remettre :



REMISES :

La conception de la gamme de tests doit être remise au plus tard vendredi le 21 mars (consulter Léa pour plus de détails).

Le travail final avec tous les éléments doit être remis au plus tard vendredi le 4 avril avant minuit (consulter Léa pour plus de détails).

SPÉCIAL BONUS

Pour ceux qui en ont envie (et ont le temps de le faire), nous donnons un bonus de 5% si vous affichez l'itinéraire en affichant le labyrinthe et en positionnant le curseur à raison de deux mouvements par seconde. Lorsque le TP original sera complété et approuvé par le professeur, celui-ci pourra fournir l'information nécessaire pour l'affichage à la console et le positionnement du curseur.

Il est très important que cette quête de bonus ne mette pas en péril vos autres cours!