

Projet ITII-3

Langage C – Techniques de programmation en C

Le projet a pour objectif de produire une application (un exécutable testé et opérationnel) en situation de projet. Ce travail sera réalisé par une équipe de 2 personnes en mode projet. Le sujet est présenté dans la suite du document. Le code à produire doit être du C ANSI. La présentation de l'application se fait sous environnement Linux.

Objectifs du projet dans le cursus ITII

Approche « Programmation » :

- ✓ Il s'agit de mettre en œuvre et d'approfondir les connaissances acquises dans le cours d'algorithmique-programmation par la réalisation d'un programme en langage C.
- ✓ Le projet vous permet de faire un travail d'analyse-conception-programmation autrement qu'en réalisant de petits exercices de validation des chapitres du cours.
- ✓ Le but est de produire :
 - un exécutable testé et opérationnel en environnement Linux ou MS-Windows,
 - un rapport écrit de la réalisation du projet (analyse, conception, réalisation, choix, difficultés, etc...)
 - ainsi qu'une documentation d'installation et d'utilisation.
- ✓ L'attention doit être mise sur la qualité du code C:
 - Présentation du code : indentation, nommage des variables, commentaires, ...
 - L'architecture du code : les fonctions (dédiées à un usage, d'environ 20 lignes maximum avec variables locales et passage de paramètres) : rôle, entrées, sorties, ...

Approche « Ingénieur » :

- ✓ Techniquement vous allez aborder tous les aspects de la spécification jusqu'à la validation / livraison.
- ✓ Il faudra gérer le temps, l'organisation du travail et la communication dans le groupe.
- ✓ Enfin il s'agit de livrer et présenter un produit fini, testé et opérationnel.

Exigences Organisationnelles

Les étudiants travaillent en binôme, à titre exceptionnel (dérogation) seuls.

Rôles au sein de l'équipe

Rôles techniques

1. Rôle analyste-concepteur (expression des besoins, définition des fonctionnalités, algorithmes)
2. Rôle architecte technique (architecture du programme, spécification des fonctions en langage C, assemblage des parties, tests d'intégration)
3. Rôle développeur (production du code en C, développement suivant spécifications, tests unitaires)

Rôles de gestion de projet

1. Rôle communication : échanges d'information dans l'équipe.
2. Rôle qualité (définition des objectifs, validation des objectifs, plan de tests, validation des tests)
3. Rôle suivi du projet (gestion de l'avancement du projet)
4. Rôle de présentation du produit : présentation orale et démonstration

Les deux personnes du binôme doivent autant que possible participer à toutes les tâches du projet.

Remarque :

Les différences de niveaux de connaissances dans les différents domaines de compétence nécessaires entre coéquipier de binômes doivent permettre de progresser, non à renforcer les différences ou écarts.

Travail à rendre et évaluation

Présentation orale du projet

La présentation du projet se déroule devant un jury. Le temps imparti à la présentation vous sera précisé par la suite : il s'agit de 10 à 15 minutes. Vous aurez :

- à présenter les étapes de la réalisation de votre travail,
- présenter les parties importantes de votre code source,
- et faire une démonstration de l'utilisation de votre application.

Ce qu'il faut rendre :

- un rapport structuré et rédigé décrivant l'ensemble des étapes de votre travail et de la répartition des tâches (qui a fait quoi ?). Vous mentionnerez par exemple les difficultés rencontrées et les solutions apportées. Vous indiquerez les dates approximatives et le temps passé sur les étapes. Vous détaillerez et justifierez vos choix techniques, de structures de données en particulier. Vous indiquerez les sources exactes de vos informations de recherche (liens complets) pour la réalisation de votre projet, afin de pouvoir les retrouver
- dans le code C, vous indiquerez les sources (liens complets) où vous avez pris des parties de codes C ou d'algorithme : vous indiquerez les parties « inspirées » ou copiées dans votre code, en indiquant les sources précises
- le code source C compilable, bien documenté
- l'exécutable testé et opérationnel avec sa documentation d'utilisation.

A rendre sur clé USB (à envoyer sur la messagerie) : le fichier zippé doit porter les noms des auteurs.

Evaluation du projet

Chaque étudiant peut être interrogé individuellement sur tous les aspects de réalisation du projet lors de la présentation, mais aussi ultérieurement.

La note attribuée sera composée de l'évaluation :

- de la clarté de la présentation orale, des supports et des réponses aux questions posées
- de la qualité (lisibilité) et la structure (architecture) du code source, notamment les commentaires et le choix des noms des objets de programmation
- du respect du cahier des charges et contraintes
- de la qualité des documents écrits (informations contenues permettant l'évaluation du travail personnel réalisé, qualité rédactionnelle, structure, et lisibilité)
- de la difficulté du projet réalisé (fonctionnalités) et des objectifs atteints

L'ajout de fonctionnalités supplémentaires et originales sera particulièrement apprécié.

Un projet dont l'exécutable ne fonctionne pas correctement n'a pas la moyenne. Il est donc impératif de viser des objectifs atteignables et de le tester de manière approfondie.

Présentation du sujet

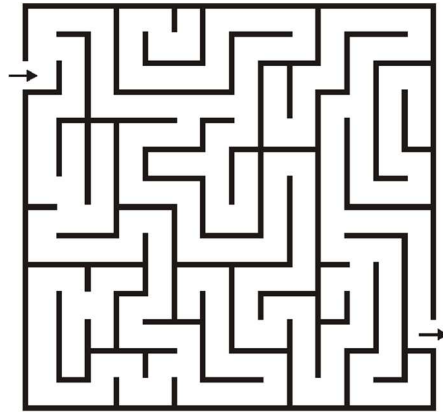
Il s'agit de réaliser un jeu de labyrinthe en 2 dimensions.

Wikipedia : https://fr.wikipedia.org/wiki/Jeu_de_labyrinthe

Le **jeu de labyrinthe** (en anglais, *maze game*) est un [type de jeu vidéo](#) dont le [gameplay](#) repose sur le motif du [labyrinthe](#). L'objectif de jeu requiert de naviguer avec succès dans un labyrinthe¹, c'est-à-dire dans un décor [sinueux](#), fait d'embranchements, d'impasses et de fausses pistes, destiné à perdre ou à ralentir le personnage. Il existe diverses variantes de ce type de jeux : la plus célèbre est sans doute celle incarnée par [Pac-Man](#).

Exigences fonctionnelles

Vous allez créer un programme informatique qui met en œuvre le jeu informatique qui invite l'utilisateur de se déplacer d'une position initiale vers une position finale dans labyrinthe, appelée entrée et sortie du labyrinthe. Une grille devra reproduire des formes du type de la grille suivante :



La position courante dans le labyrinthe devra être symbolisée graphiquement et être visible.

Vous essayerez d'implémenter les fonctionnalités **obligatoires** suivantes:

- ✓ Chargement d'une grille enregistrée dans un fichier parmi au moins 4 grilles différentes
- ✓ Détection de l'arrivée à la sortie du labyrinthe (fin du jeu).

Exemples de fonctionnalités optionnelles :

- ✓ Evaluation de la distance du parcours réalisé pour arriver à la sortie : score
- ✓ Mesure du temps pour accéder à la sortie du labyrinthe
- ✓ Affichage des 4 meilleurs scores et temps pour la grille choisie
- ✓ Tracé du parcours suivi
- ✓ Dimensions variables des grilles
- ✓ Génération automatique de labyrinthe (méthode de la fusion de cellules)
- ✓ ...

Exigences techniques

Vous mettrez en œuvre une compilation séparée en langage C.

Pour réaliser l'aspect graphique, vous allez utiliser la bibliothèque **ncurses/pdcurses**. Vous trouverez de nombreux exemples d'utilisation sur Internet et une documentation à cette adresse : <http://tldp.org/HOWTO/NCURSES-Programming-HOWTO/>

Exigences concernant le chargement et la sauvegarde dans des fichiers

Pour cette partie, vous avez le choix d'utiliser les fichiers binaires ou les fichiers textes. N'oubliez pas de bien expliquer votre manière de coder les labyrinthes dans les fichiers.

Etapes intermédiaires :

Si vous éprouvez des difficultés avec la représentation et la gestion des labyrinthes, vous pouvez commencer par la réalisation d'un jeu plus simple pour vous familiariser avec la bibliothèque.

Exemple 1 : L'objectif est d'atteindre le mur droit d'une grille en partant du bord gauche, en contournant des obstacles (pavés) placés de manière aléatoire sur l'espace de jeu. Il faut s'assurer qu'il existe un chemin possible entre le bord gauche et le bord droit.

Exemple 2 : L'objectif est d'atteindre le mur droit d'une grille en partant du bord gauche, en attrapant des chiffres (1 à 9) sur le chemin, avec des obstacles (pavés) placés de manière aléatoire sur l'espace de jeu. Il faut s'assurer qu'il existe un chemin possible entre le bord gauche et le bord droit. Le score est alors la somme des chiffres consommés.

Exemple 3 : L'objectif est d'atteindre le mur droit d'une grille en partant du bord gauche, en attrapant des chiffres (1 à 9) sur le chemin, avec des obstacles (pavés) placés de manière aléatoire sur l'espace de jeu. Les obstacles et chiffres peuvent apparaître ou disparaître au cours du temps. Le score est alors la somme des chiffres consommés.