

Sujet : C-ession Projetc

Introduction et objectif

Il vous est demandé par équipe de réaliser une architecture logicielle en langage C répondant aux caractéristiques ci-dessous en équipe et en temps contraint. Il est interdit d'échanger des informations entre les groupes, le cas échéant ce sera considéré comme de la triche et l'ensemble des groupes sera sanctionné.

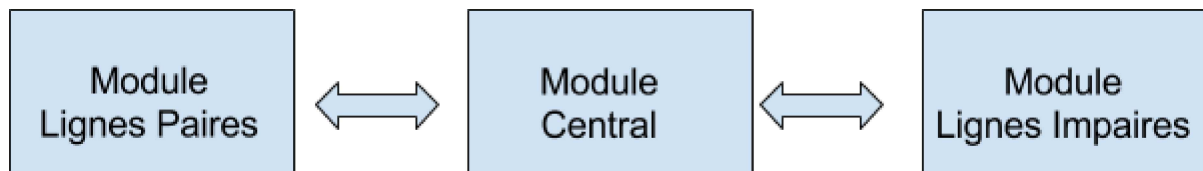
L'objectif est de réaliser trois logiciels en langage C permettant gérer un "échiquier de caractères" qui se modifiera à chaque "passe" (intervalle de temps) en fonction de règles définies par l'utilisateur.

Description fonctionnelle

Il y a deux types d'utilisateurs sur les programmes : l'administrateur et l'utilisateur de l'application (nommé utilisateur ensuite). Chacun a des droits spécifiques.

L'utilisateur fournit au logiciel un échiquier de caractères, c'est-à-dire un ensemble de lignes de taille fixe de caractères divers. Ensuite l'utilisateur intègre des règles qui modifieront cet échiquier à chaque intervalle de temps.

L'application sera construite avec l'architecture respectant le schéma suivant :



Le "Module Central" gère l'algorithme du logiciel. Les modules Lignes Paires et Lignes Impaires stockent les données de l'échiquier concernant les lignes paires ou impaires. Voici les règles régissant les modules :

1. Ces modules doivent pouvoir être déployés sur 3 machines connectées en réseau. Cependant ces modules doivent pouvoir aussi être déployés sur la même machine sans connection extérieure ;
2. Le Module Central a interdiction de stocker les données des lignes ;
3. Le Module Lignes Paires peut communiquer uniquement avec le Module Central ;
4. Le Module Lignes Impaires peut communiquer uniquement avec le Module Central ;
5. Le Module Lignes Paires ne doit jamais avoir accès à une information d'une ligne impaire ;
6. Le Module Lignes Impaires ne doit jamais avoir accès à une information d'une ligne paire ;

7. L'utilisateur ne peut communiquer qu'avec le Module Central.

Le mode de fonctionnement doit être le suivant :

1. Les modules Central, Lignes Paires, et Lignes Impaires sont activés par l'administrateur ;
2. Les modules se connectent entre eux via des données réseaux fournies par l'administrateur à ce moment-là ;
3. L'utilisateur donne au module central la situation de départ de l'échiquier, par fichier ou par insertion de caractères ;
4. Le Module Central répartit ces données sur les deux autres modules et ne garde aucune mémoire ;
5. L'utilisateur donne ensuite au Module Central les règles qui vont modifier l'échiquier pour chaque temps ;
6. Une fois toutes les règles insérées, l'utilisateur active le déroulement manuellement pour chaque temps ou lance le déroulement pour un certain nombre de temps ;
7. Le Module Central doit pouvoir fournir l'état de l'échiquier actuel et passé. Le rendu sera sous forme de fichier ou de représentation graphique.

L'administrateur doit pouvoir à tout moment vérifier les données présentes dans les Modules Lignes Paires et Lignes Impaires à l'aide d'interfaces spécifiques.

Définition des règles

L'utilisateur devra pouvoir entrer des règles du type :

Pour chaque case de type "A",

Si il y a "B1" "C1" caractère(s) de type "D1" à "E1" case(s) de distance
"F"

Si il y a "B2" "C2" caractère(s) de type "D2" à "E2" case(s) de distance
alors la case devient de type "G"

A : type de caractère d'une case

B : structure logique parmi les 4 suivantes : "au moins", "au plus", "aucun", "exactement"

C : nombre (à ne demander que si cela a du sens)

D : type de caractère d'une case

E : nombre (généralement minimum 1, si > 1 alors on englobe toutes les cases de distance inférieure ou égale au nombre)

F : structure logique parmi les 2 suivantes : "OU", "ET", non obligatoire (non présent si la deuxième ligne n'est pas présente)

G : type de caractère d'une case

Exemple : L'utilisateur doit pouvoir par exemple entrer les règles suivantes :

Règle 1 :

Pour chaque case de type R,

*Si il y a au moins 2 caractères de type Z à 1 case de distance
et
Si il y a exactement 1 caractère de type W à 2 cases de distance
alors la case devient de type X*

Règle 2 :
*Pour chaque case de type X,
Si il y a aucun caractère de type Z à 1 case de distance
alors la case devient de type R*

Précision sur l'algorithme

L'algorithme se déroule ensuite sur toutes les cases puis la modification se fait simultanément. Si plusieurs règles pour un instant donné peuvent s'appliquer pour une case, ce sera la première règle saisie par l'utilisateur qui sera appliquée.

L'algorithme doit pouvoir détecter une situation stable : c'est-à-dire que pour tout temps donné l'échiquier ne change plus. L'algorithme ne devra pas calculer les prochains temps. De même l'algorithme doit pouvoir détecter une situation cyclique : c'est-à-dire que pour tout temps donné l'échiquier permute de manière cyclique sur une situation donnée. L'algorithme ne devra pas calculer les prochains temps.

L'administrateur devra avoir accès aux numéros des règles appliquées sur chaque case. Un code couleur pourra être mis en place. Par exemple on pourra associer la règle 1 à la couleur rouge. Dans le cas où une case a appliqué la règle 1 elle devra apparaître en rouge au temps d'après.

Qualité et date de rendu

L'application devra être robuste, et pouvoir faire face à un utilisateur peu scrupuleux des consignes données par le logiciel. L'utilisateur doit pouvoir comprendre à tout moment ce qu'il fait et ce qui est en train de se passer.

Les contraintes ci-dessus décrivent la version 1 du logiciel.

Vous devrez ensuite développer les fonctionnalités suivantes pour une version 2 du logiciel :

- Possibilité par l'utilisateur de définir des règles spécifiques pour une ligne donnée. Dans ce cas ces règles ont priorité sur les règles s'appliquant sur l'ensemble de l'échiquier.

Date des rendus des version 1 et version 2 : au plus tard le jeudi 25 Janvier 12h30. Date du mail faisant foi. **Chaque soir à 18h15** un mail collectif doit être envoyé aux enseignants (en copie l'ensemble du groupe de travail), dans celui-ci : qui a travaillé sur quoi, l'avancé du travail et les éventuels soucis dans le groupe. Ce mail est dit "collectif" car c'est l'ensemble du groupe qui le rédige.

Il vous sera demandé les codes sources des 3 programmes, un fichier décrivant le lancement en réseau et en local avec les commandes permettant de tester l'intégralité du programme. Si il y a un bug et que vous en avez connaissance il faut le signaler au moment du rendu. Des fichiers de tests et de rendu sont attendus.

Pour le rendu : il est important qu'une personne non informaticienne (par exemple sous windows) puisse lancer votre application en suivant une procédure simple et exhaustive.

Nous sommes disponibles par mail pour toutes questions pendant la durée du module.

Exemple de déroulement

Exemple 1 :

Si un utilisateur entre les règles suivantes :

1. Pour une case de type X : si il y a au moins un O à coté je deviens O
2. Pour une case de type O : si il y a exactement 8 cases X à 1 case de distance je deviens X
3. Pour une case de type A : si il y a au moins un X et au moins un O autour, je deviens X

Soit l'échiquier de départ de type :

Ligne 1 : XXXX

Ligne 2 : XXOX

Ligne 3 : XXXX

Ligne 4 : XAXX

Les Lignes 1 et 3 sont stockées sur le module Impaire

Les Lignes 2 et 4 sont stockées sur le module Pair

Nous aurons en lançant l'algorithme :

Temps 0 :

XXXX

XXOX

XXXX

XAXX

Temps 1 :

XOOO

XOXO

XOOO

XAXX

Temps 2 :

Oooo

O O O O
O O O O
O X O O

Temps 3 :

O O O O
O O O O
O O O O
O O O O

Détection de stabilité.

Exemple 2 :

Règles de l'utilisateur :

1. **Rouge** : Pour une case de type O : si il y a au moins un / à côté je deviens A
2. **Bleu** : Pour une case de type A : si il y a exactement aucun O à 2 case de distance je deviens O
3. **Vert** : Pour une case de type A : si il y a au plus 2 A à côté, je deviens \$

Echiquier donné :

O/O/OS

Temps 0 :

Ligne 1 : O/O/OS

Temps 1 :

Ligne 1 : A/A/AS

Temps 2 :

Ligne 1 : O/O/OS

Détection de cycle de période 2.