

## Projet C : Jeu de la Vie

---

*Avant de commencer : la structuration de votre programme, la qualité des commentaires, avec notamment la présence des antécédents, des conséquents, des invariants de boucle, les rôles de chacune des fonctions, ainsi que les noms donnés aux variables, l'emploi à bon escient des majuscules et la bonne indentation rentrent pour une part importante dans l'appréciation du travail. Ce projet doit permettre de montrer votre autonomie et votre compréhension tant dans la conception du programme que dans sa réalisation. Enfin, si les codes de plusieurs projets se trouvent être identiques, ou être copiés depuis le web, tous les projets concernés seront immédiatement sanctionnés par un zéro.*

---

### 1 Sujet

L'objectif de ce projet est de programmer le *jeu de la vie* de J. Conway. Ce *jeu* simule l'évolution d'organismes en société. Vous trouverez une description de ce jeu à l'adresse [http://fr.wikipedia.org/wiki/Jeu\\_de\\_la\\_vie](http://fr.wikipedia.org/wiki/Jeu_de_la_vie).

### 2 Travail à réaliser

#### 2.1 Partie 1

Dans un premier temps, vous programmerez les règles du jeu telles que J. Conway les a formulées. Au départ, une population d'individus est répartie sur une grille (théoriquement infinie). La population se transforme, étape par étape, selon des règles d'évolution. À chaque étape, les individus apparaissent et/ou disparaissent sur la grille, formant une nouvelle génération d'individus, selon les règles suivantes :

- survie : tout individu voisin de deux ou trois individus survivra ;
- naissance : toute case vide voisine de trois individus exactement donnera naissance à un individu ;
- mort par étouffement : tout individu voisin de quatre autres individus ou plus disparaîtra ;
- mort par isolement : tout individu isolé ou voisin d'un seul individu disparaîtra.

Ce sont les individus présents à chaque génération qui déterminent les naissances et les morts de la génération suivante.

La grille aura une dimension fixe  $M \times N$ . Votre programme devra permettre une présentation graphique de la grille à l'aide de la bibliothèque graphique *libsx*. L'utilisa-

tion de cette bibliothèque est **obligatoire** pour votre projet. Dans un premier temps, vous pourrez développer une interface textuelle ; toutefois le rendu final de votre projet ne devra comporter que l'interface graphique.

Vous visualiserez la grille avec les générations d'individus qui s'affichent à intervalles de temps réguliers (*e.g.* toutes les secondes), sans intervention de l'utilisateur.

#### 2.2 Partie 2

Vous mettrez en œuvre la variante "*Day and night*" décrite sur le site wikipédia à l'adresse donnée plus haut. Un individu naît d'une case vide à l'étape suivante si elle est entourée de 3, 6, 7 ou 8 individus vivants, un individu vivant reste en vie à l'étape suivante s'il est entouré de 3, 4, 6, 7 ou 8 individus vivants.

#### 2.3 Partie 3

Dans cette 3ème partie, vous mettrez en évidence les *structures classiques* (stables, périodiques, vaisseaux, ...) et les comparerez avec les deux règles de génération de votre programme demandées dans les parties 1 et 2.

Vous placerez dans des fichiers les populations initiales d'individus qui correspondent aux différentes *structures* dont l'évolution est à comparer. Les contenus de ces fichiers seront chargés en début de programme.

### 3 Remise du projet

Votre projet est à faire en binôme, pas de trinôme, au plus un monôme. Il est à rendre au plus tard :

**Dimanche 30 mai 2021 à 23h – aucun délai ne sera accordé –**

Vous déposerez votre projet dans l'espace *Projet n. 2* du cours de *Langage C* sur le moodle de l'université sous forme une archive (tar) comprimée (gz) :

`jeu-vie-nom1-nom2.tgz` avec `nom1` et `nom2` les noms de chaque étudiant du binôme

**cette archive devra contenir :**

- les fichiers sources (`.c` et `.h`) correctement documentés (chaque fonction doit avoir un commentaire, les invariants de boucle doivent être écrits), indenté, et codé (les noms de variables explicites, éviter les trop longues fonctions) ;
- les fichiers de population d'individus clairement identifiés ;

- un fichier **Documentation** **uniquement au format pdf** et décrivant le fonctionnement général du programme, les algorithmes, vos choix de programmation, ainsi que les résultats de l'étude de la partie 3 du projet ;
- la compilation avec les options **-Wall** ne doit donner aucun message d'erreur ou d'avertissement.

**Le non respects de ces directives entraînera le retraits de points sur la note finale.**

D'autre part, il ne devrait pas être nécessaire de rappeler que le travail doit être personnel et que toute ressemblance entre des projets sera très sévèrement sanctionnée. Mieux vaut donc un projet modeste personnel qu'un très beau projet copié.

Enfin, il ne devrait d'ailleurs pas être nécessaire de rappeler que le but premier d'un tel projet est de vous faire progresser en programmation en vous confrontant à une expérience de plus grande envergure qu'un simple TP.

Bon travail