Université d'Artois
Faculté des Sciences Jean Perrin

ALGO3 – Algorithmique et Programmation 3

Jeu de la vie

Jeu de la vie : Aperçu des objectifs

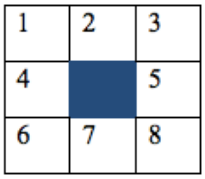
Le Jeu de la Vie, également connu sous le jeu Vie, ou tout simplement la vie, est un automate cellulaire (un système qui a des règles appliquées aux cellules et leurs voisins dans une grille) conçu par John Conway, professeur de Mathématiques discrètes à l'Université Princeton en 1970. Le jeu de La vie est un exemple de la "complexité émergente" ou "systèmes d'auto-organisation", qui étudie comment les modèles et les comportements complexes peuvent émerger de règles très simples. pour plus d'informations à ce sujet jeu, reportez-vous à :
`http://en.wikipedia.org/wiki/Conway%27s_Game_of_Life` ou
`http://www.math.com/students/wonders/life/life.html`

Comment jouer le jeu ?

Le Jeu de la Vie est en fait un jeu à zéro joueur, ce qui signifie que le jeu est déterminé par ses propres règles et le modèle initial.

Dans ce devoir, nous allons commencer le jeu sur une grille avec des cellules "$m \times n$". Chaque cellule de la grille comporte l'un des deux statuts : `vivant` ou `mort`. L'état des $m \times n$ cellules dans la grille constitue la motif initial de la grille.

Chacune des cellules a 8 voisins (sauf ceux sur les bords), comme illustré ci-dessous.



En appliquant les règles suivantes pour chaque cellule de la grille, on crée une nouvelle génération du motif. Les règles sont les suivantes :

1. Une cellule morte avec trois voisines vivantes devient une cellule vivante.
2. Une cellule vivante avec deux ou trois voisins vivants reste en vie.
3. Une cellule vivent avec moins de 2 ou supérieure à 3 voisines vivantes meurt.

Les règles sont appliquées à la grille courante, en générant une nouvelle grille de cellules de la grille. Autrement dit, les règles sont appliquer à la grille existante, mais les résultats apparaissent dans l'étape suivante de la grille, comme illustré en dessous

Les cellules de bord ont un nombre réduit de voisins.

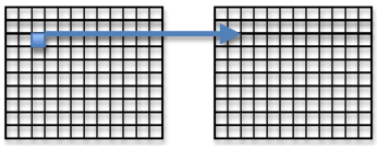


FIGURE 1 – Appliquer règles

Travail demandé

Votre tâche consiste à mettre en place le jeu en Python en utilisant des classes. Avant la mise en œuvre de la jeu, essayez de l'exemple de programme ou la version en ligne du jeu à `http://www.math.com/students/wonders/life/life.html` pour vous assurer que vous le comprenez.

Caractéristiques du Programme :

1) La taille de la grille est spécifiée par l'utilisateur au début de la partie. 2) Sur la grille, les cellules mortes sont représentés par le signe tiret (”-”), et les cellules vivantes sont représentés par le signe astérisque (”*”). Voici un exemple de la sortie :

Colonne		0	1	2	3	4
Ligne	0	-	*	-	-	-
Ligne	1	*	*	*	*	-
Ligne	2	-	-	-	*	-
Ligne	3	*	-	*	-	*
Ligne	4	-	-	-	*	-

Algorithme de haut niveau :

1. Créer une classe pour le jeu. Après le jeu commence, demander au joueur la taille (nombre de lignes et de colonnes) de la grille. Dans la classe, créer une liste de listes représenter la grille.
2. Définir une fonction membre `afficherGrille` qui affiche la grille courante à l'écran.
3. Définir une fonction de membre `getAdj` pour compter le nombre de cellules vivantes qui sont immédiatement adjacente à une cellule donnée horizontalement, verticalement ou en diagonale. La colonne et ligne le numéro de cellule donnée sont passés comme arguments à cette fonction.
4. Votre classe doit avoir une fonction de membre appelé `etapeSuivante` qui applique les règles au modèle actuel et courant pour obtenir la prochaine génération du modèle.
5. Demandez à l'utilisateur si il veut jouer le jeu avec un modèle généré de façon aléatoire ou Retrouvez tous les motifs ”Structure stable” pour la grille.
6. Si l'utilisateur choisit de jouer le jeu avec un motif aléatoire, demandez-lui le nombre de cellules vivantes de la grille.
   — Après cela, placer aléatoirement les cellules vivantes sur la grille. Pour placer les cellules vivantes, vous pourriez avoir une fonction de membre appelé `placeVCelsRandom`. Cette fonction prend le nombre initial de cellules vivantes en tant que paramètre.
   — Continuez à appeler les fonctions `etapeSuivante` et `afficheGrille` que l'utilisateur choisit de voir la prochaine génération du modèle. Créer une fonction de membre `estGrilleVide` pour vérifier si le réseau actuel est vide (pas de cellules vivantes à l'intérieur) ou pas. Si la grille est vide, le programme doit s'arrêter.

Travail demandé

Partie 1

La première partie de votre programme doit pouvoir :

— lire sur l'entrée standard la taille de la grille ;

— lire le nombre de cellules vivantes ;

— générer aléatoirement l'emplacement de ces cellules ;

— afficher nouvelle génération à chaque étape, tant que Y est tapé et s'arrêter sinon

Partie 2

Vous devez dans la deuxième partie intégré une interface graphique. L'utilisateur pourra
— Choisir d'entrer le nombre de lignes et de colonnes.

— Choisir les cellules vivantes en cliquant sur les cellules concernées.

— Cliquer sur le bouton "start" pour démarrer le jeu.
— Cliquer sur le bouton "stop" pour arrêter le jeu.
— Cliquer sur un bouton "clear" pour effacer la grille

Autres Contraintes

— Utilisez un "timer" pour afficher la génération suivante
— Si une structure stable est détecté, un message sera afficher pour le signaler à l'utilisateur.