

TP « Jeu de la vie »

N. Delestre

Règles ¹

En préambule, il faut préciser que le jeu de la vie n'est pas vraiment un jeu au sens ludique, puisqu'il ne nécessite aucun joueur; il s'agit d'un automate cellulaire, un modèle où chaque état conduit mécaniquement à l'état suivant à partir de règles pré-établies.

Le jeu se déroule sur une grille à deux dimensions, théoriquement infinie (mais de longueur et de largeur finies et plus ou moins grandes dans la pratique), dont les cases, qu'on appelle des « cellules », par analogie avec les cellules vivantes, peuvent prendre deux états distincts : « vivantes » ou « mortes ».

À chaque étape, l'évolution d'une cellule est entièrement déterminée par l'état de ses huit voisines de la façon suivante :

- Une cellule morte possédant exactement trois voisines vivantes devient vivante (elle naît).
- Une cellule vivante possédant deux ou trois voisines vivantes le reste, sinon elle meurt.

Objectif général

Écrire un programme qui affiche les n premières générations d'une grille initialisée aléatoirement (avec un seuil de remplissage) et dont la taille est donnée par l'utilisateur. On suppose pour cela que l'on possède la fonction suivante :

- **fonction** obtenirNaturelAléatoire (borneMax : **NaturelNonNul**) : **Naturel**

Analyse

On suppose posséder le TAD Grille suivant :

Nom:	Grille
Utilise:	Booleen, NaturelNonNul
Opérations:	grille: NaturelNonNul × NaturelNonNul → Grille
	obtenirLargeur: grille → NaturelNonNul
	obtenirHauteur: grille → NaturelNonNul
	estUneCelluleVivante: Grille × NaturelNonNul × NaturelNonNul → Booleen
	tuerUneCellule: Grille × NaturelNonNul × NaturelNonNul → Grille
	faireNaitreUneCellule: Grille × NaturelNonNul × NaturelNonNul → Grille
Préconditions:	estUneCelluleVivante(g,x,y): $x \leq$ obtenirLargeur(g) et $y \leq$ obtenirHauteur(g)

1. Wikipédia

tuerUneCellule(g,x,y): $x \leq \text{obtenirLargeur}(g)$ et $y \leq \text{obtenirHauteur}(g)$ et $\text{estUneCelluleVivante}(g,x,y)$

faireNaitreUneCellule(g,x,y): $x \leq \text{obtenirLargeur}(g)$ et $y \leq \text{obtenirHauteur}(g)$ et $\text{non estUneCelluleVivante}(g,x,y)$

- Axiomes:**
- obtenirLargeur(grille(l,h))=l
 - obtenirHauteur(grille(l,h))=h
 - non estUneCelluleVivante(grille(g,l,h),x,y)
 - non estUneCelluleVivante(tuerUneCellule(g,x,y),x,y)
 - estUneCelluleVivante(faireNaitreUneCellule(g,x,y),x,y)

Voici l'analyse descendante du problème *simulerJeuDeLaVie* :

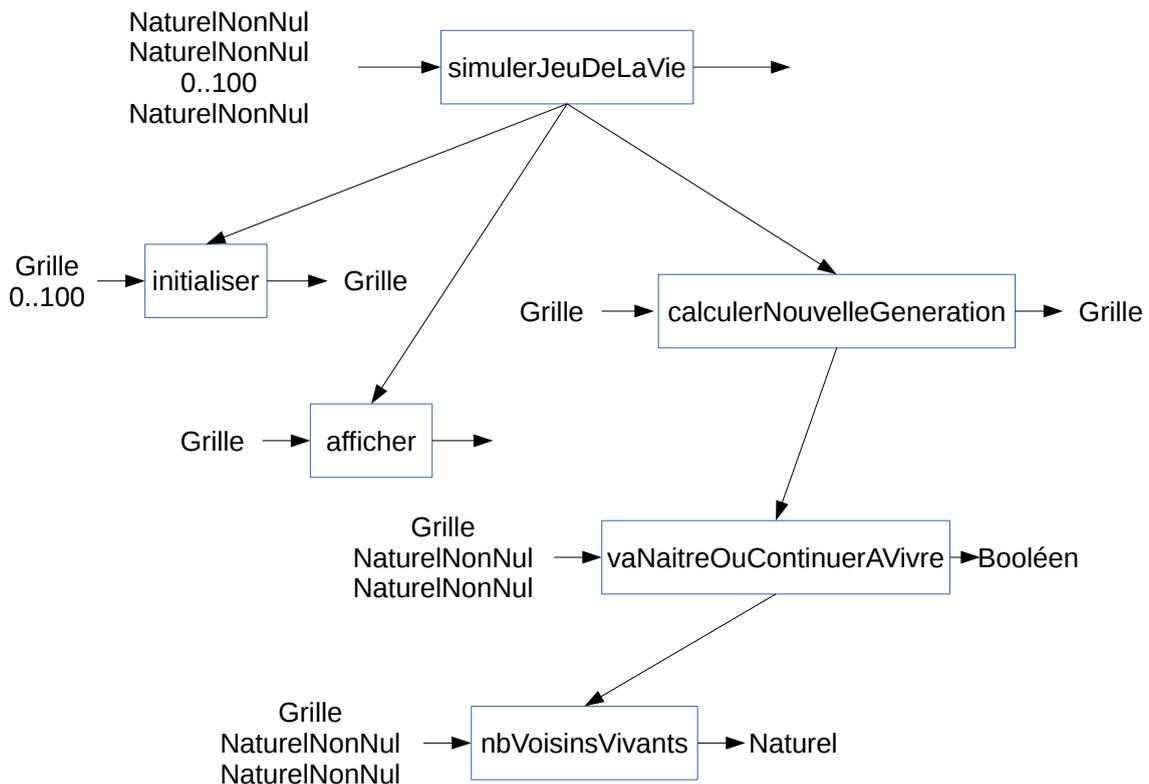


FIGURE 1 – Analyse descendante

Conception préliminaire

Voici les signatures des fonctions et procédures issues de cette analyse descendante.

- **procédure** initialiser (**E/S** g : Grille, **E** tauxDeRemplissage : 0..100)
- **procédure** calculerNouvelleGeneration (**E/S** g : Grille)
- **procédure** afficher (**E** g : Grille)
- **fonction** vaNaitreOuContinuerAVivre (g : Grille, x,y : **NaturelNonNul**) : **Booleen**

[précondition(s)] $x \leq \text{obtenirLargeur}(g)$
 $y \leq \text{obtenirHauteur}(g)$

— **fonction** nombreVoisinsVivants (g : Grille, x,y : **NaturelNonNul**) : **Naturel**

[précondition(s)] $x \leq \text{obtenirLargeur}(g)$
 $y \leq \text{obtenirHauteur}(g)$

Conception détaillée

Voici les algorithmes des précédentes fonctions et procédures.

fonction nombreVoisinsVivants (g : Grille, x,y : **NaturelNonNul**) : **Naturel**

[précondition(s)] $x \leq \text{obtenirLargeur}(g)$
 $y \leq \text{obtenirHauteur}(g)$

Déclaration i,j : **NaturelNonNul**
minx,miny,maxx,maxy, resultat : **Naturel**

debut

```
resultat ← 0
minx ← max(1,x-1)
miny ← max(1,y-1)
maxx ← min(obtenirLargeur(g),x+1)
maxy ← min(obtenirHauteur(g),y+1)
pour i ← minx à maxx faire
  pour j ← miny à maxy faire
    si estUneCelluleVivante(g,i,j) et (i ≠ x ou j ≠ y) alors
      resultat ← resultat+1
    finsi
  finpour
finpour
retourner resultat
```

fin

fonction vaNaitreOuContinuerAVivre (g : Grille, x,y : **NaturelNonNul**) : **Booleen**

[précondition(s)] $x \leq \text{obtenirLargeur}(g)$
 $y \leq \text{obtenirHauteur}(g)$

Déclaration nbVoisinsVivants : **Naturel**

debut

```
nbVoisinsVivants ← nombreVoisinsVivants(g,x,y)
si (non estUneCelluleVivante(g,x,y) et nbVoisinsVivants=3) ou (estUneCelluleVivante(g,x,y) et
(nbVoisinsVivants=3 ou nbVoisinsVivants=2)) alors
  retourner VRAI
sinon
  retourner FAUX
finsi
```

fin

procédure calculerNouvelleGeneration (E/S g : Grille)

Déclaration temp : Grille
i,j : NaturelNonNul

debut

temp ← grille(obtenirLargeur(g),obtenirHauteur(g))

pour i ← 1 à obtenirLargeur(g) **faire**

pour j ← 1 à obtenirHauteur(g) **faire**

si vaNaitreOuContinuerAVivre(g,i,j) **alors**

 faireNaitreUneCellule(temp,i,j)

finsi

finpour

finpour

g ← temp

fin

Développement

Compléter le fichier src/JeuDeLaVie.c proposé dans l'archive disponible sur moodle (un paramètre à la fonction *simulerJeuDeLaVie* a été ajouté pour séparer la logique métier de l'interface).