



西安电子科技大学
XIDIAN UNIVERSITY

Projet C : Jeu de la Vie

KANG Jiale, 19022100087; ZHANG Liyun, 19022100093

Département Ingénierie électronique
Université de Xidian, Xi'an

Shaanxi, Chine

30 juin 2021

Résumé : Ce reportage inclus la solution de programmer le jeu de la vie de J. Conway avec la bibliothèque graphique *libsx*. D'abord, on fixe les règles de ce jeu, et le présente sur la grille de $M \times N$. Après, on prends les resultats haut et bas pour les comparer plus facile.

Mot-clé : *libsx*, jeu de la vie, Day & night, comparer, Automate cellulaire

1. Règles et le Méthode

1.1. Objet

On établit des paramètres de fenêtre d'abord.

```
1 #define winWidth 400
2 #define winHeight 400
3 #define sqarePixel 20
4 #define animateTime 800
5 #define W winWidth*2/sqarePixel
6 #define H winHeight*2/sqarePixel
```

Donc, **sqarePixel** est la longueur d'une grille, et **W** et **H** sont les grilles de graphe.

Ensuite, pour un individuel, on réfléchit 4 statuts : *survie*, *naissance*, *mort par étouffement* et *mort par isolement*.

Grâce à les statuts ne sont liés qu'à le nombre de 8 individuels entourés, on les balayage et décide les statuts.

On utilise les variables suivantes :

```
1 const int changeX[8] =
    { -1, -1, 0, 1, 1, 1, 0, -1 };
2 const int changeY[8] =
    { 0, -1, -1, -1, 0, 1, 1, 1 };
3 int lifeGame[W+2][H+2];
4 int dayAndNight[W+2][H+2];
```

changeX[8] et **changeY[8]** sont les 8 directions (dx, dy) de grilles où individuels entourés.

lifeGame[][] et **dayAndNight[][]** est le carte de ce jeu.

1.2. Méthode et présentation

On écrit 7 fichiers : **init.h**, **data.h**, **vue.h**, **init.c**, **data.c**, **vue.c** et **main.c**.

init.h et **init.c** : initialisation;

data.h et **data.c** : calculer les statues;

vue.h et **vue.c** : visualiser les grilles .

D'abord, on initialise par une fichier *StructureXXX*. Cette fichier mémorise les coordonnées d'individus survie, et XXX montre la catégorie. On la donne 3, *Périod*, *Stable*, *Vasseux*.

Ensuite, on utilise le fonction suivant pour décrire les règles.

```
1 void lifeGameData(void);
2 void dayAndNightData(void);
```

On choisit une petite grille et après on compter le nombre d'individuel qui soit survie et l'appeler *tot*.

Après on balayage toute les petites grilles avec cet règles et juge les statues, note par les tableaux **map[][]**, et ensuite on remplace le carte par le fonction **lifeGameData** et les tableaux **map[][]** et les présente.

Pour les présente, on ouvre une fenêtre et découvre 3 boutons qui montrent *QUIT*, *STOP*, *ANIMATE*(fonctionner) et 3 boutons qui montrent trois structure d'initialisation. Ensuite, on affiche deux jeux en même temps. Il représente une fois par seconde.

On utilise les fonctions de *libsx* pour dessiner.

On **ClearDrawArea** et **GetStandardColors** pour initialisation. Ensuite, on choisit le couleur noir avec **SetColor** et dessine les grilles avec **DrawLine**. Enfin, on remplace le carte à 2 étapes.

- fonction **drawAnimate** pour choisir le couleur vert qui signifie l'individuel survie or le couleur blanc qui signifie l'individuel mort;
- fonction **drawBackground** pour dessiner la background.

2. Résultat

Donc on obtient le résultat¹ ci-dessous. (On met *jeu de la vie* en vert et *Day & Night* en bleu.)

2.1. Stable

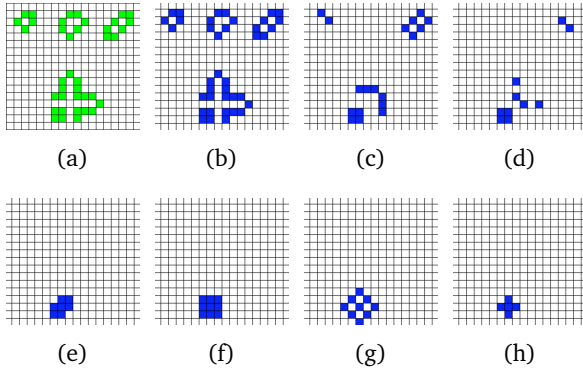


FIGURE 1: Structure Stable

Dans ce cas, on peut voir si le vert est Stable (1(a)), le bleu change à la période de 7 (1(b)-1(h)).

2.2. Vaseux

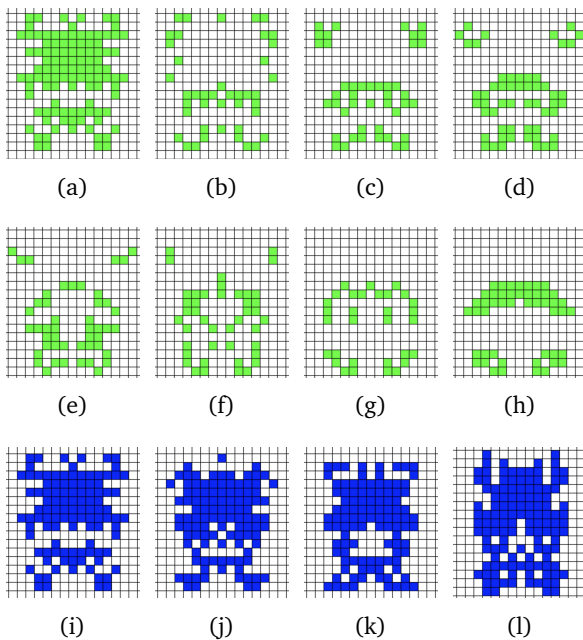


FIGURE 2: Structure Vaseux

Dans ce cas, on met le bleu qui est à la période de 5 (2(i)-2(l)), et il déplace lentement vers le bas (On capture les première 5). Mais le vert devient anormalement² (2(a)-2(h)).

1. Ces structures viennent sur <https://www.conwaylife.com/forums/>

2. On capture les première 8 changements.

2.3. Période

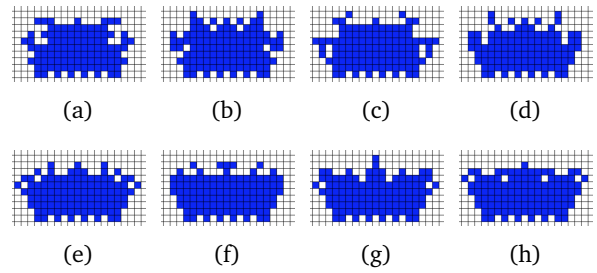


FIGURE 3: Structure Oscillateur

Dans ce cas, on met le bleu qui est à la période de plus de 20, donc on exhibe les première 8 dessins (3(a)-3(h)). On n'exhibe pas le vert car il change anormalement. Pour exposer la période complet, on donne un autre exemple ci-dessous.

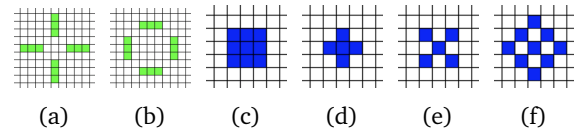


FIGURE 4: Structure Oscillateur très simple

3. Comparer

On écrit l'initialisation condition d'individu vivant et comparer les statues dans les règles différent.

On donne le condition de **Structure Période**.

On peut choisir le condition pour comparer les deux règles. D'ailleurs on crée le bouton pour PAUSE et QUIT aussi.

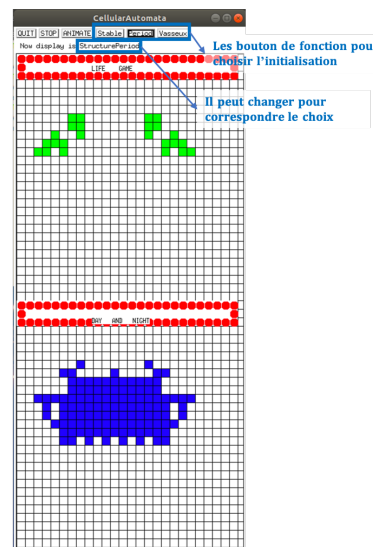


FIGURE 5: Comparer les deux conditions