Mathieu Orhan

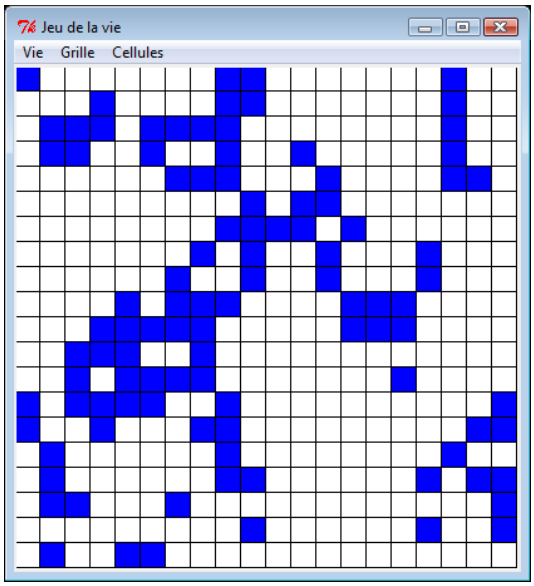
Terminale S Spécialité ISN

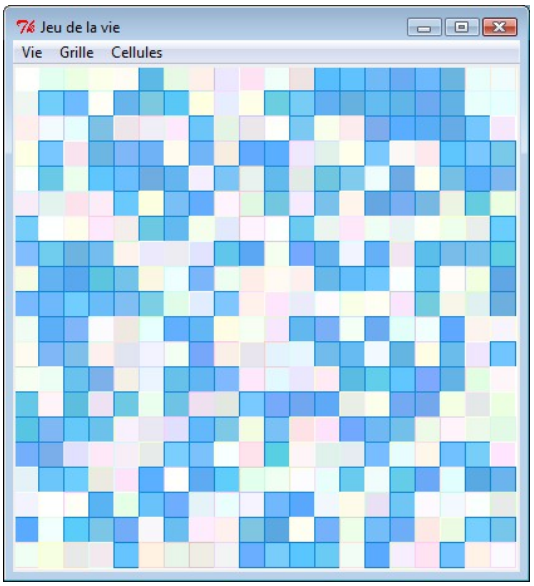
Projet d'ISN

Jeu de la vie

PyLife

Présenté pour l'obtention du Baccalauréat
soutenu en Mai





Professeur d'ISN : *Mr. Mandon*

Lycée Jean Jaurès
Saint Clément de Rivière
Académie de Montpellier

2012 — 2013

Sommaire

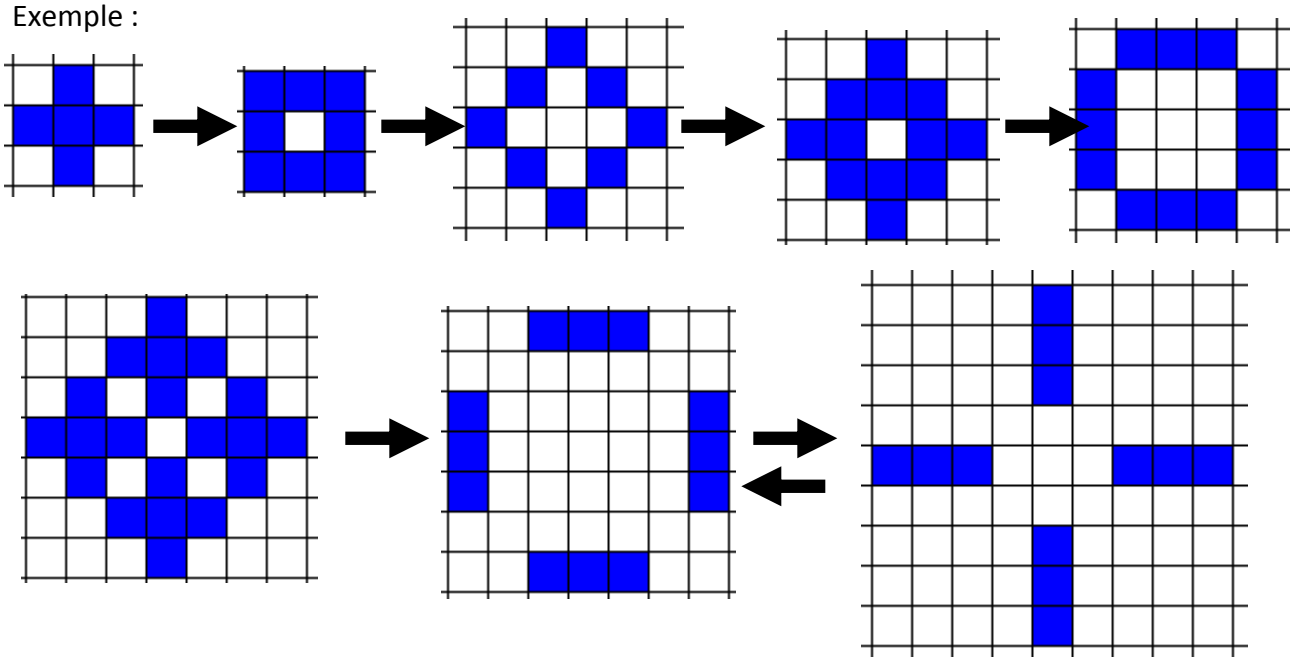
Introduction

Pour l'examen du Baccalauréat d'ISN, il est demandé de choisir un projet à réaliser.

1. A propos du Jeu de la vie

« *Le jeu de la vie, automate cellulaire imaginé par John Horton Conway en 1970, est probablement, à l'heure actuelle, le plus connu de tous les automates cellulaires. Malgré des règles très simples, le jeu de la vie permet le développement de motifs extrêmement complexes.* »
Toute cellule ayant exactement deux ou trois cellules voisines survit à la génération suivante.
Toute cellule ayant quatre cellules voisines ou plus meurt par étouffement à la génération suivante. Une cellule isolée ou n'ayant qu'un seule cellule voisine meurt d'isolement à la génération suivante.
Sur une case vide comportant exactement trois cellules voisines, il naît une cellule à la génération suivante.

Exemple :



Les cases peuvent donc avoir deux états : vivante ou morte. Une case tient compte de ses huit voisines pour déterminer son état à la génération suivante.

2. Pourquoi ce projet ?

Je me suis intéressé au Jeu de la vie après avoir vu les possibilités qu'offrait ce projet en terme de programmation, l'implémentation de l'algorithme de base étant simple mais le développement intéressant. Les automates cellulaires sont aussi amusant à regarder évoluer, même si n'est pas un « jeu ».
De plus, de nombreuses ressources sont disponibles sur ce sujet : des formes répétitives, immobiles, se déplaçant…

Cahier des charges

1. Compatibilité

Compatible avec au moins Windows XP et supérieur, les principales distributions Linux, et Mac OS.
Le système devra être équipé d'une configuration moyenne pour faire tourner le programme dans de bonnes conditions.

2. Contexte

Projet à présenter devant un jury pour l'obtention du baccalauréat 2013. Réalisé seul, dans un délai de quelques mois dans le cadre des cours d'ISN avec pour professeur Mr Mandon.

3. Sujet

Le Jeu de la vie est un modèle d'automate cellulaire proposant dans la version originale les règles suivantes:

- *Une cellule a plus de 3 cellules voisines vivantes : elle meurt*
- *Une cellule a moins de 2 cellules voisines vivantes : elle meurt*
- *Si une cellule est entourée par exactement 3 cellules vivantes, elle nait*

La grille est théoriquement infinie, mais elle sera finie dans la réalité du programme.

4. Contraintes

Commandes utilisateur :

- Utiliser le programme au clavier
    - Touche <Échap> pour effacer la grille
    - Touche <e> pour éditer la grille
    - Touche <space> pour avancer automatiquement
    - Touche <space> pour stopper l'avance automatique
    - Touche <Up> pour accelerer l'avance automatique
    - Touche <Down> pour ralentir l'avanc automatique
    - Touche <Right> pour avancer d'une génération
    - Touche <r> pour générer une grille aléatoire
- Utiliser le programme avec les menus
    - Menu « Vie »
        - Evolution automatique
        - Calculer n génération
        - Prochaint état
        - Aléatoire
        - Effacer
        - Quitter
        - Edition

    - Menu « Grille »
        - Charger .LIF
        - Mode fermé
        - Mode tore
    - Menu « Cellule »
        - Mode classique
        - Mode fun
- Editer le programme à souris, même pendant l'évolution, quand le mode édition est activé.

Le programme doit pouvoir :

- Calculer une génération n + 1 à partir d'une génération n
- Calculer une génération n + x à partir d'une génération n
- Afficher un état n
- Générer des configurations aléatoires (ou pseudo-aléatoire) de grilles.
- Effacer la grille (toutes les cellules sont mortes)
- Etre édité à la souris quand le mode édition est activée à tout moment
- Supporter théoriquement toutes tailles de grilles et de cellules
- Agir de manière fermé d'une part
- Agir comme un tore d'autre part
- Posséder des variantes au niveau de l'apparence des cellules.
- Charger de manière simple des fichier Life 1.05, Life 1.06
- Calculer et afficher de manière continue de états, le temps avant chaque calcul étant modifiable en temps réel.
- Implémenter de manière naïve les règles du Jeu de la vie de Conway.
- Proposer, dès le lancement un programme, une forme connue tel qu'un oscillateur ou un glider.

5. Exemples d'utilisation

1) Edition
   Lancer le jeu de la vie.
   Appuyer sur échap : le jeu se vide.
   Appuyer sur e : lance le mode édition?
   Cliquer sur des cases pour former une un croix. Appuyer sur espace.
   Observer l'évolution en réglant de sorte à que la vitesse soit lente.

2) Fichier LIF
   Lancer le jeu de la vie.
   Aller dans grille > charger .LIF
   Sélectionner un fichier .LIF adapté à la taille de la grille.
   Appuyer sur espace, observer

3) Aléatoire
   Lancer le jeu de la vie
   Appuyer sur r
   Appuyer sur espace et régler sur une vitesse moyenne

Rapport technique

A. Choix technologiques

Pour le langage de programmation, j'ai choisi Python non pas parce que c'était le plus performant ou le plus adapté, mais parce que Python permet d'être productif et d'écrire du code vite et simplement. La dernière version (3.2) est utilisée. Aussi, c'est le langage étudié en cours, et celui connu de manière certaine du jury.
Pour l'interface graphique, mon choix s'est porté sur Tkinter, la bibliothèque fournie par défaut avec Python3.

B. Style de programmation, nomenclature et documentation

1. Une GUI avec la programmation événementielle

On parle ici de GUI et donc de programmation événementielle. Le programme tourne alors en boucle à l'infini et réagit à des événements (un clic, une pression sur une touche…).

2. La structure avec la programmation orienté objet

Plus adapté pour un programme de cette taille, bien plus simple à maintenir, mieux structuré, factorisation de code, pas de variable globales, j'ai préféré la POO
nb : *Python propose une implémentation particulière de la POO, sans notion d'encapsulation. Tout les attributs et les méthodes sont publiques.*

3. Noms de fonctions, de classes, de variables, et de fichiers

Classes : `CamelCase`
Fonctions et méthodes : `function_name`
Variables : `var_name`
Fichiers : `classname.py`
En anglais, pour des noms courts, universels et simples.

4. Commentaires et documentation

Les commentaires seront fait en dessous des nom de fonctions, sur plusieurs lignes, en utilisant une structure conventionnelle. Les commentaires seront en français uniquement. Chaque fonction est documenté de manière à ce que l'on comprenne son rôle et son fonctionnement global.
Le paramètre `self` ne sera jamais documenté : il s'agit de l'objet lui-même.
Les paramètres d'entrées sont annoté @param, de retour @return et @see désigne une référence dans le programme.

Exemple :

```
class math:

   def carre(self,x):
      """
      Retourne le carré de x
      @param int x Le nombre entier x
      @return int Le nombre x au carré
      """
      return x*x
```

C. Structure

1. Découpage en fichier

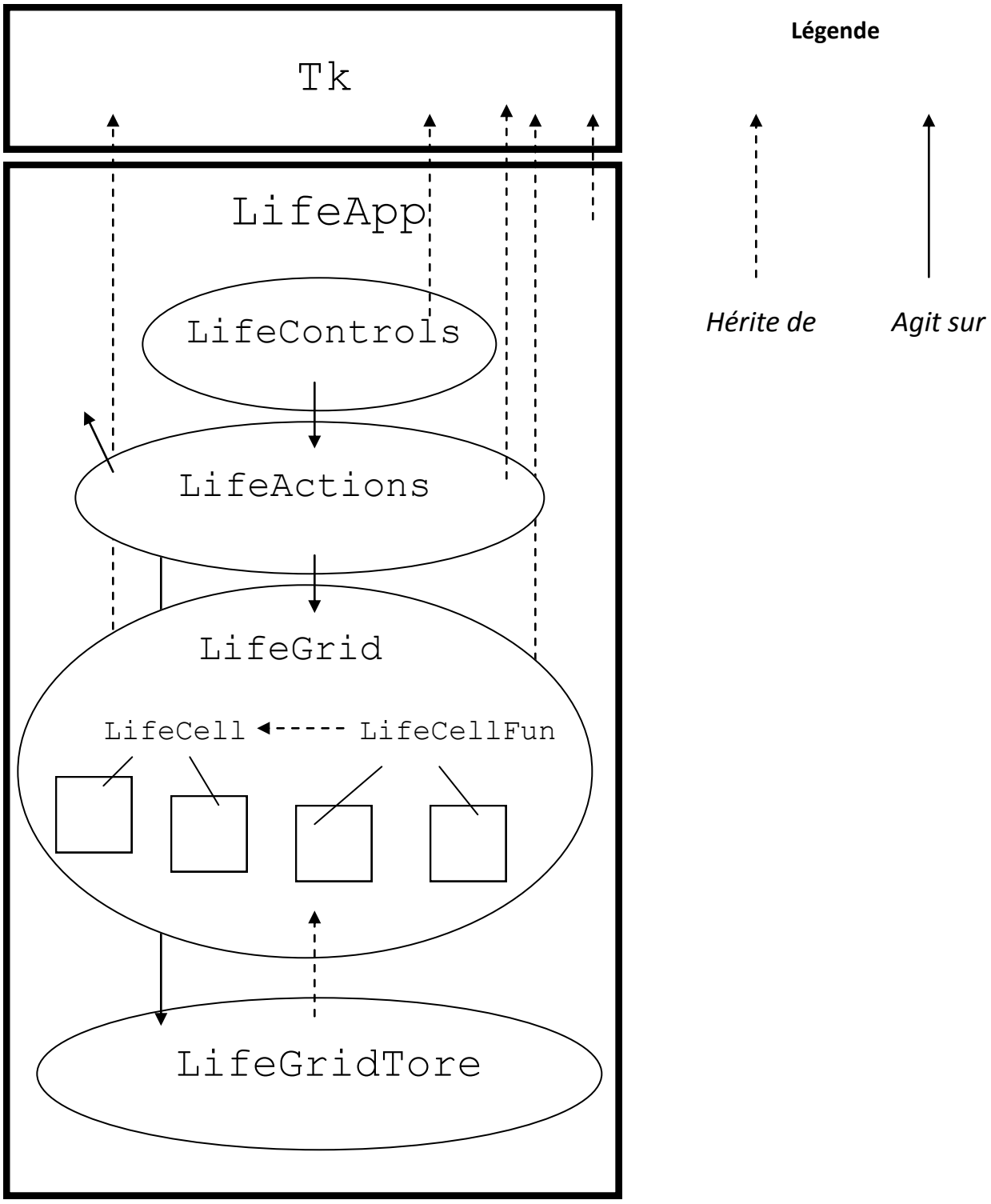
Une classe par fichier, chaque classe représente une partie du programme.

```
lifeapp.pyw
lifeaction.py
lifecell.py
lifecellfun.py
lifecontrols.py
lifegrid.py
lifegridtore.py
```

Le jeu peux fonctionner de manière minimale avec lifegrid.py, et lifecell.py.

2. Découpage en classes

Le diagramme ci-dessous représente le fonctionnement des classes dans le programme. On peux observer plusieurs dépendances et des héritages nombreux.
La classe Tk est la classe de base des application Tkinter. Elle sert à créer la fenêtre, mais aussi à gérer les événements, les widgets, etc…
La classe LifeApp n'as que peu d'utilité sinon d'englober le programme.
On remarque l'utilisation de l'héritage de LifeCell et LifeGrid pour créer respectivement LifeCell-Fun et LifeGridTore.



D. Algorithmes

1. Implémentation des règles

On s'intéresses aux fonction « get_voisins » des classes « LifeGrid » et LifeGridTore. La première est une implémentation dite fermée du jeu, la seconde fonctionne comme un tore.
Nb : on remarquera que « voisins » n'est pas en anglais contrairement au reste du programme, c'est plus court et parlant cette fois-ci.

Algorithme « fermé » de décomptage

```
n_voisins = 0
    pour x_voisin allant de –1 à 1
        pour y_voisin allant de –1 à 1
            Si x_voisin et y_voisin sont dans la grille et que x_voisin et y_voisin ne sont pas nuls
                ajoute la valeur de la vie de la cellule vosine à n_voisins
renvoyer n_voisins
```

Algorithme « tore » de décomptage

x_down, x_up contiennent respectivement x-1 et x+1 si les limites du tableau ne sont pas atteintes, sinon l'autre extrémité du tableau. De même pour y_down et y_up.
On renvoi l'addition de l'état des 8 cellules voisines.

Règles du jeu :

vivant si voisins = 3 ou (voisins = 2 et cellule morte), sinon mort, pour chaque cellule

2. Fichiers LIF

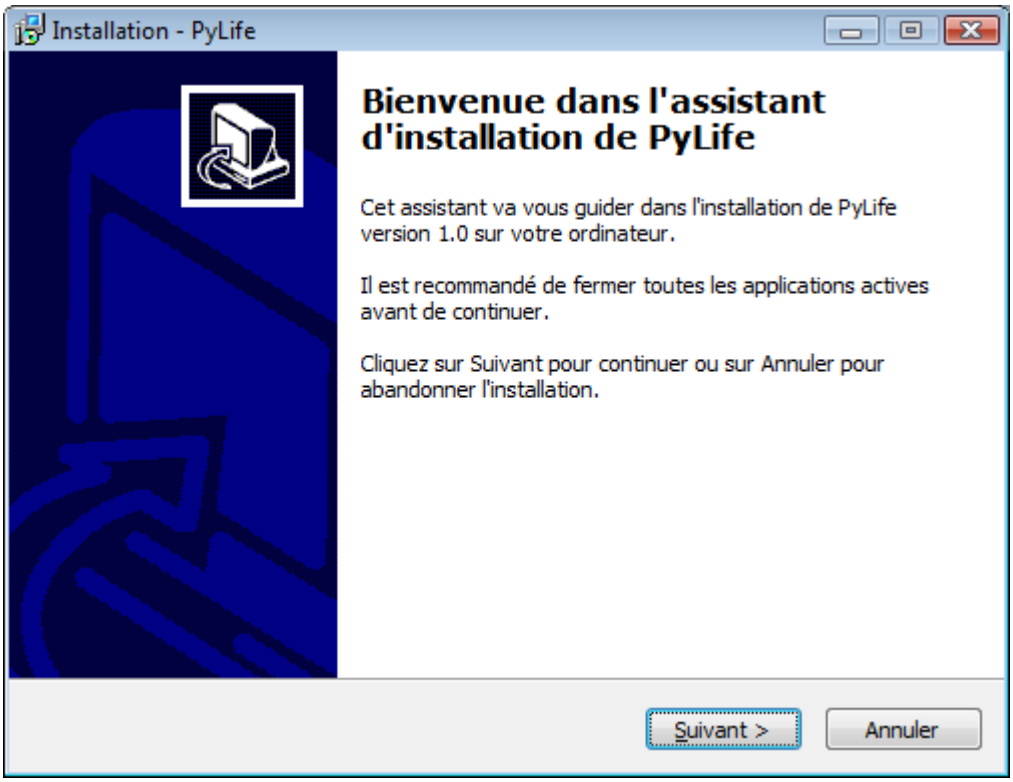
Chargement

```
Demande à l'utilisateur de choisir un fichier et l'ouvre en lecture seule
Pour chaque ligne du fichier
    Passe les lignes de commentaires du fichiers
    Si la ligne commence par « #P »
        Isole les paramètres dans settings
        Remet logic à zéro
        Passe à la ligne suivante
        Tant que la ligne n'est pas un commentaire ou vide
                Remplace « . » par « 0 » et « * » par 1 et ajoute à logic
                Passe à la ligne suivante
                Ajoute les settings et logic dans inserts
Efface la grille
Pour chaque insert de inserts
    Réalise l'insertion dans la grille à partir d'un point central
Ferme le fichier
```

Manuel d'utilisation

1. Installation Windows

Veuillez vous munir de python 3.2 et de `setup.exe` (l'installateur Windows).
Lancez setup.exe et suivez les instructions.



A la fin de l'installation, le programme se lance. Vous pourrez le lancer depuis le menu démarrer ou encore avec l'icône du bureau.

2. Installation Linux

Sur linux, décompressez le fichier zip, et lancez les commandes suivantes dans le répertoire PyLife :

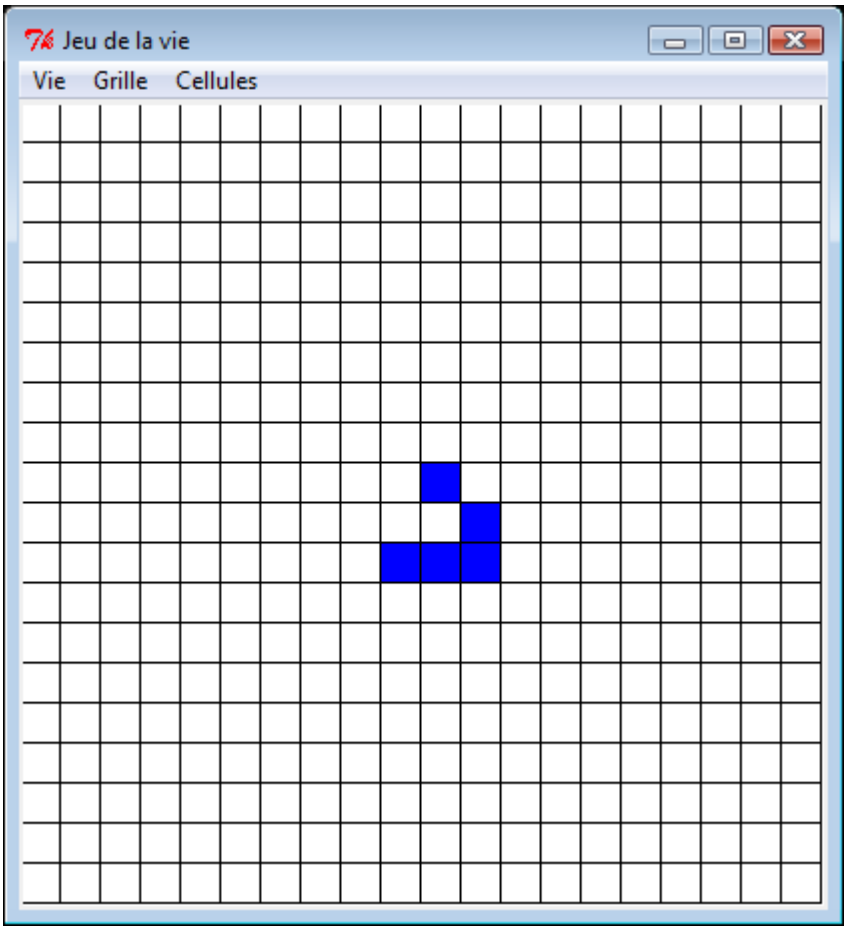
```
sudo apt-get install python3
python3 lifeapp.pyw
```

3. Désinstallation

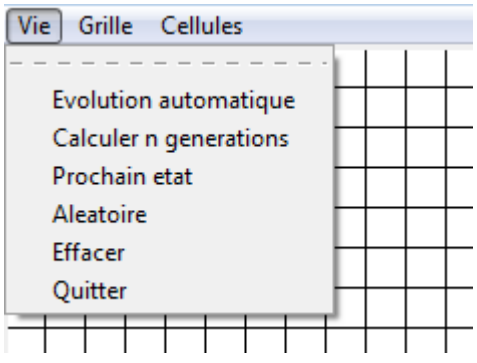
Sur Linux, supprimer simplement les fichiers du jeu.
Sur Windows, il faut aller dans Menu démarrer > Tous les programmes > PyLife > Désinstaller PyLife.

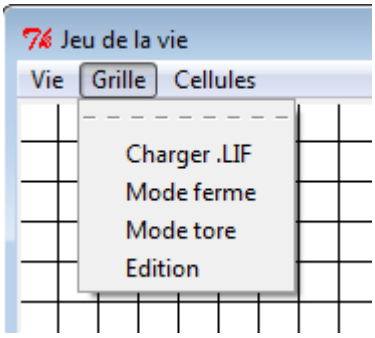
4. Interface

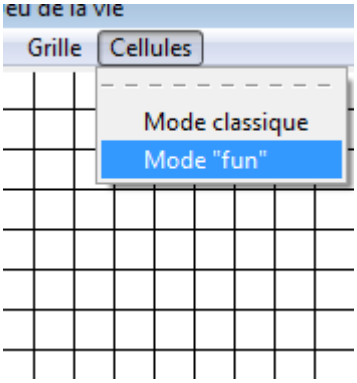
L'interface est composé d'un menu, et de la grille. Il y a 3 menu : vie, grille et cellule.



Le menu Vie permet l'évolution automatique, le calcul de n état, le prochain état, un état aléatoire, d'effacer la grille ou encore de quitter le programme.
Le menu Grille permet de changer de mode de grille, d'en charger une au format LIF, et d'éditer la grille actuelle.
Le menu Cellule permet de charger de mode de cellule.







5. Commandes

Touche	**Commande**
Espace	Arrête ou stoppe l'évolution automatique
Haut	Accélère la vitesse de l'évolution automatique
Bas	Ralentis la vitesse de l'évolution automatique
Droite	Calcule et affiche le prochain état des cellules
Echap	Efface la grille
R	Génère et affiche une grille aléatoire
E	Active ou désactive le mode édition

Conclusion

PyLife est un programme implémentant simplement le jeu de la vie. Son utilisation est simple, visuelle, et n'est limité que par l'imagination de l'utilisateur et les capacités du PC.

Bugs

Il n'existe actuellement aucun bug connu.
Le calcul de beaucoup de générations peut faire planter le programme si l'ordinateur n'est pas assez puissant. Sur le PC de test, des valeurs comme 1000000000 feront planter le programme.

Améliorations du programme

- Réécrire le programme en C++ avec Qt par exemple, pour une vitesse bien supérieure d'exécution.
- On peux penser à une toolbar pour contrôler le programme.
- J'avais aussi eu pour idée d'un jeu de la vie à 3 dimensions, mais Tkinter ne suffit pas pour la 3D.
- Il y aussi un algorithme bien plus puissant mais très compliqué à intégrer pour le jeu de la vie : HashLife. Ce dernier est capable de calculer de très nombreuses grilles en très peu de temps
- Mieux supporter les fichiers LIF
- Prise en charge de règles personnalisées du jeu de la vie
- Proposer plus de formes d'exemples ou par défaut.

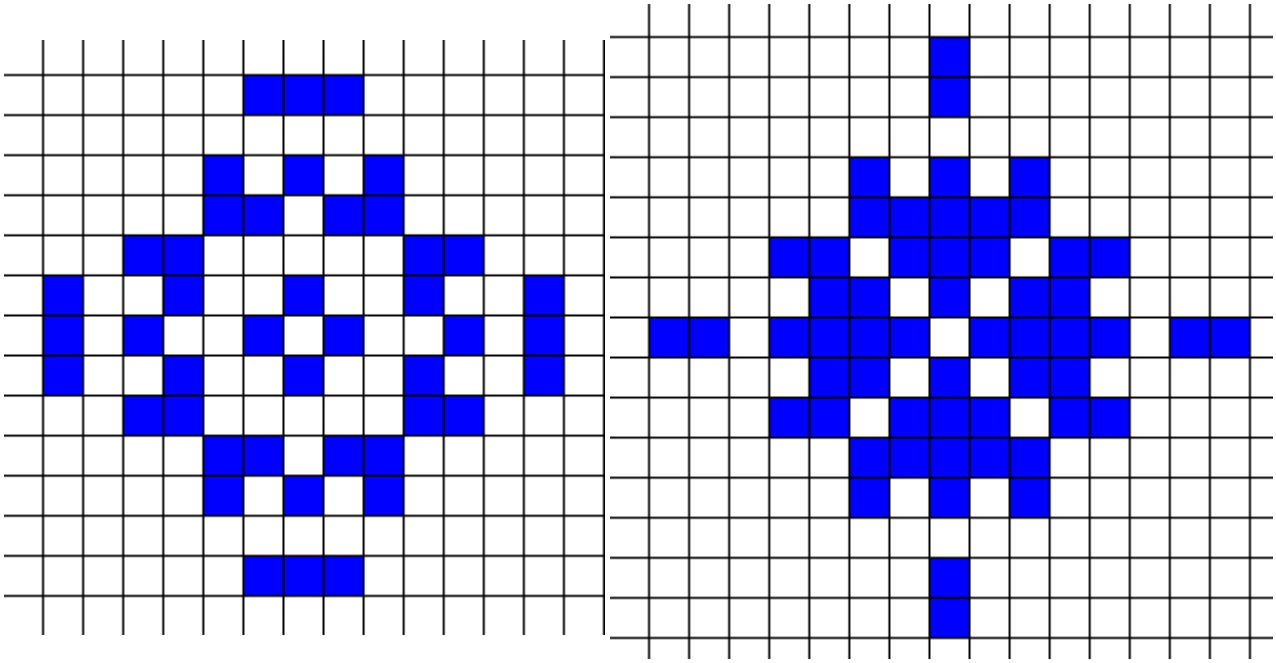
Références

http://fr.wikipedia.org/wiki/Jeu_de_la_vie
http://www.jeulin.net/automates/automate.html

PyLife

Le Jeu de la vie est créé en 1970 par un mathématicien, John Horton Conway. Depuis, il a fasciné des générations de programmateurs et de mathématiciens dans la recherche de nouvelles formes et de propriétés. Le jeu de la vie est imprévisible malgré des règles simples : c'est un automate cellulaire.

PyLife est un implémentation graphique naïve du jeu de la vie en Python avec Tkinter.



Partant de forme très simples, il est possible d'obtenir des motifs plus complexes.

Mathieu Orhan