

Les projets de la période 1

Table des matières

1 Objectif :	1
2 Les documents ressources.....	2
3 Les sujets.....	2
3.1 Sujet 1 : verbes irréguliers.....	2
3.2 Sujet 3 : un jeu du type "Mastermind" (***).....	3
3.3 Sujet 3 : un jeu du type "le pendu"	3
3.4 Sujet 4 : jeu du keno.....	4
3.5 Sujet 5 : jeux de la vie.....	4
3.6 Sujet 6 : feu de forêt.....	5
3.7 Sujet 7 : déplacement de population 1 - Entonnoir.....	6
3.8 Sujet 8 : déplacement de population 2 - Croisement.....	6
3.9 Sujet 9 : déplacement de fourmis 1 - L'automate de Langton.....	7
3.10 Sujet 10 : déplacement de fourmis 2 - phéromones (***).....	8
3.11 Sujet 11 : propagation d'une épidémie.....	8
3.12 Sujet 12 : jeu snake (***).....	9
3.13 Sujet 13 : jeu de chute d'objets (***).....	9

1 Objectif :

Travailler sur :

- Les tableaux à double entrées :
- Tableaux 2D ;
- Listes de listes ;
- Les structures avec des boucles for imbriquées ;
- La structuration du code avec des fonctions.

2 Les documents ressources

Les fichiers :

- automate_console_v1.py
 - Code non structuré, sans utilisation de fonctions
 - Exemple fonctionnel
- automate_console_v2_eleve.py
 - Code structuré, organisation avec des fonctions
 - Exemple non fonctionnel :
 - Code à compléter
- automate_graphique_v1.py
 - Code non structuré :
 - Création des tableaux sans boucle
 - Création de la structure graphique sans boucle
 - Exemple fonctionnel
- automate_graphique_v2_eleve.py
 - Code structuré :
 - Création des tableaux avec des boucles
 - Création de la structure graphique avec des boucles
 - Exemple non fonctionnel :
 - Code à compléter

Les liens :

- <https://pysimplegui.readthedocs.io/en/latest/>

3 Les sujets

3.1 Sujet 1 : *verbes irréguliers*

Vous allez créer une application pour travailler sur les verbes irréguliers anglais.

Voici le début du tableau des verbes irréguliers.

Base verbale	Prétérit	Participe passé	Traduction
abide	abode	abode	respecter / se conformer à
arise	arose	arisen	survenir
awake	awoke	awoken	se réveiller
bear	bore	borne / born	porter / supporter / naître
beat	beat	beaten	battre
become	became	become	devenir
...

On peut coder le tableau des verbes sous la forme d'un tableau du type liste de liste (voir le cours).

Pour chaque étape, on demande une version en console puis une version graphique avec soit la bibliothèque PySimpleGUI soit la bibliothèque TkInter.

Version 1 :

- L'ordinateur affiche l'infinitif en anglais ;
- L'utilisateur doit répondre le prétérit.

Version 2 :

- L'ordinateur affiche l'infinitif en anglais ;
- L'utilisateur doit répondre par le prétérit, le participe passé et la traduction en français.

Version 3 :

- L'ordinateur affiche le score de l'utilisateur.

Version 4 :

- L'ordinateur affiche un tableau avec les scores de différents utilisateurs (score et durée)

3.2 *Sujet 3 : un jeu du type "Mastermind" (***)*

Voir les règles sur : <https://fr.wikipedia.org/wiki/Mastermind>

- Jeu en mode texte
- Jeu en mode graphique (PySimpleGUI)
- Joueur1 contre joueur2 sur le même ordinateur
- Joueur contre IA.

3.3 *Sujet 3 : un jeu du type "le pendu"*

Voir les règles sur : [https://fr.wikipedia.org/wiki/Le_Pendu_\(jeu\)](https://fr.wikipedia.org/wiki/Le_Pendu_(jeu))

- Jeu en mode texte
- Jeu en mode graphique (PySimpleGUI)
- Joueur1 contre joueur2 sur le même ordinateur
- Joueur contre IA.

3.4 Sujet 4 : jeu du keno

Voir les règles sur : <https://fr.wikipedia.org/wiki/Keno>

- Jeu en mode texte
- Jeu en mode graphique (PySimpleGUI)
- Simulation d'une partie
 - Le joueur choisit une grille
 - l'ordinateur simule le tirage au hasard
 - affiche le gain du joueur
 - affiche le gain du casino
- Simulation de parties successives
 - Le joueur choisi une grille
 - l'ordinateur simule le tirage au hasard
 - affiche le gain du joueur
 - affiche le gain du casino
 - puis on recommence
- L'ordinateur simule un grand nombre de parties
 - Simulation d'un joueur qui choisit une grille (fixe ou aléatoire)
 - Simule le tirage au hasard
 - Affiche le gain du joueur
 - Affiche le gain du casino

3.5 Sujet 5 : jeux de la vie

Initialement :

La tableaux est initialisé :

- soit de manière déterministe ;
- soit le contenu est aléatoirement.

Règle 1 :

Les états :

- 0 la cellule est morte
- 1 à 9 la cellule est vivante

Les transitions :

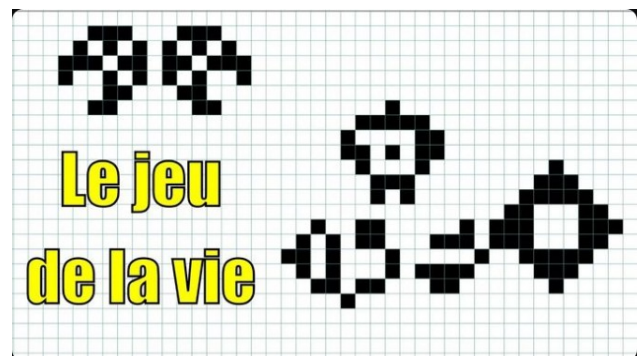
- à chaque étape l'état augmente de 1. Si $n = 10$ alors $n = 0$.
- états 1 à 4 : la cellule ne donne pas de vie aux cellules voisines
- états 5 à 9 : la cellule donne vie à une cellule voisine :
 - une cellule est choisie aléatoirement
 - la cellule prend vie avec une probabilité p

Règle 2:

Appliquer les règles de la page : https://fr.wikipedia.org/wiki/Jeu_de_la_vie

Ressources :

- <http://www.jeux-et-mathematiques.davalan.org/divers/jeuvie/index.html>



3.6 Sujet 6 : feu de forêt

Initialement :

La tableaux est initialisé :

- soit de manière déterministe ;
- soit le contenu est aléatoirement.
- les cases blanches sont sans arbre
- les cases contiennent des arbres
- quelques cases rouges représentent le feu.

Règle 1:

Les états :

- 0 : blanc - pas d'arbre présent
- 1 : vert - arbre présent
- 2 : rouge - feu, l'arbre brûle
- 3 : gris - l'arbre a brûlé, il reste des cendres

Les transitions :

- état 0 : pas de changement
- état 1 : pas de changement
- état 2 :
- les cellules voisines prennent feu
- la cellule devient grise
- état 3 : la cellule devient blanche

Règle 2:

- le rouge prend plusieurs nuances et donc plusieurs états ;
- les cellules voisines prennent feu avec une probabilité p .

Règle 3:

On suppose l'existence du vent :

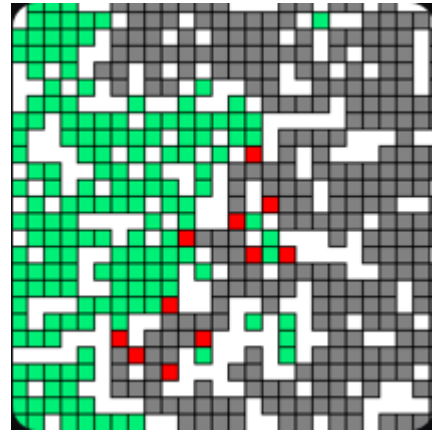
le vent est caractérisé par une direction (angle) et une vitesse (v)

le vent sera représenté par $\text{vent}[\text{vcos}(\text{angle}), \text{vsin}(\text{angle})]$

la direction et la vitesse du vent vont modifier la probabilité de propagation du feu

Pour aller plus loin :

- Créer un graphique qui montre le nombre de cellules grises en fonction en fonction du nombre de cellules avec des arbres au départ



3.7 *Sujet 7 : déplacement de population 1 - Entonnoir*

Des individus arrivent par la droite et par le haut. Ils convergent tous vers une fenêtre en bas à gauche. Un individu ne peut occuper une cellule déjà occupée par un autre individu.

Règle 1:

Les états :

- 0 : blanc - pas d'individu
- 1 : vert - présence d'un individu dans les cellules autres que celles de sortie
- 2 : bleu - présence d'un individu dans les cellules de sortie

Les transitions :

- état 0 sur la ligne du haut ou sur la colonne de droite
 - On ajoute un individu avec une probabilité p
- état 1 :
 - l'individu se déplace vers la gauche ou vers le bas
 - probabilité de 0,5 pour aller vers le bas
 - probabilité de 0,5 pour aller vers la gauche
- état 2 : l'individu sort du plateau et libère la place

Règle 2:

Les individus peuvent être du type :

- vert : vitesse normale, déplacement d'une case adjacente
- rouge : vitesse rapide, déplacement de deux cases adjacentes

3.8 *Sujet 8 : déplacement de population 2 - Croisement*

Deux populations se déplacent et se croisent :

Population 1, déplacement du haut vers le bas

Population 2, déplacement de la droite vers la gauche

Règle 1:

Les états :

- 0 : blanc - pas d'individu
- 1 : vert - présence d'un individu de la population 1
- 2 : rouge - présence d'un individu dans les cellules de sortie de la population 1
- 5 : bleu - présence d'un individu de la population 2
- 6 : gris - présence d'un individu dans les cellules de sortie de la population 2

Les transitions :

- état 0 sur la ligne du haut
 - On ajoute un individu de la population 1 avec une probabilité p_1
- état 0 sur la colonne de droite
 - On ajoute un individu de la population 2 avec une probabilité p_2
- état 1 :
 - si possible, l'individu de la population 1 se déplace vers la cellule du bas
 - sinon vers une autre cellule vers le bas
 - sinon vers une cellule sur le côté
- état 2 : l'individu sort du plateau et libère la place
- état 5 :
 - si possible, l'individu de la population 2 se déplace vers la cellule de gauche
 - sinon vers une autre cellule de gauche
 - sinon vers une cellule en haut ou en bas
- état 6 : l'individu sort du plateau et libère la place

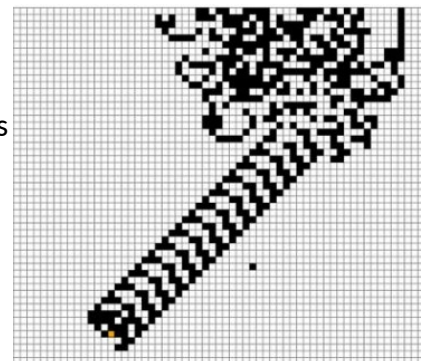
Règle 2:

Les individus peuvent être du type :

- normal : vitesse normale, déplacement d'une cellule adjacente
- rapide : vitesse rapide, déplacement de deux cellules adjacentes

3.9 Sujet 9 : déplacement de fourmis 1 - L'automate de Langton

Comme de nombreux automates cellulaires, celui de Langton a un comportement déterministe élémentaire. Sur une grille bidimensionnelle constituée de cases blanches ou noires, la fourmi se déplace d'une case dans n'importe quelle direction selon les deux règles suivantes :



Règle :

- si la fourmi est sur une case noire, elle tourne de 90° vers la droite, change la couleur de la case en blanc et avance d'une case
- si la fourmi est sur une case blanche, elle tourne de 90° vers la gauche, change la couleur de la case en noir et avance d'une case.

Travail :

Placer une fourmi au milieu du plateau

Lancer la simulation

Observer la création de :

- figures géométriques
- des motifs qui se reproduisent à l'infini

3.10 Sujet 10 : déplacement de fourmis 2 - phéromones (***)

Les fourmis se déplacent du milieu gauche vers le milieu droit. On place un obstacle vertical au milieu. L'obstacle se matérialise par un segment vertical. Il est positionné de manière non symétrique.

Les états :

- 0 à 99 : niveau de phéromone
- 0 : pas de fourmi, pas d'obstacle, pas de phéromone
- 100 + 0 à 99 : fourmi plus niveau de phéromone
- 200 : obstacle

Les transitions :

- état 0 :
 - au centre gauche, arrivée d'une fourmi si cellule vide
 - pas de changement ailleurs
- états 1 à 99 : diminuer les phéromones d'une quantité k_1
- états 100 à 190 : présence de fourmi
 - augmente le niveau de phéromone d'une quantité k_2
 - si possible se déplacer vers une des trois cellules de droite
 - choix de la cellule avec une probabilité $p = 1 / n$. n nombre de cellules possibles
 - sinon si possible se déplacer vers le haut ou vers le bas avec une probabilité de 0,5.
- état 200 : pas de changement

Ressources :

- <http://ressources.univ-lemans.fr/AccesLibre/UM/Pedago/physique/02/recre/fourmi.html>

3.11 Sujet 11 : propagation d'une épidémie

Règle 1: modèle SIR

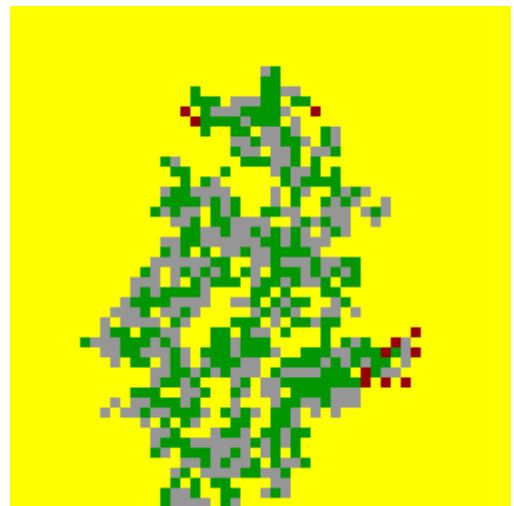
Quelques individus malades contaminent les voisins avec une probabilité p .

Les états :

- 0 : jaune - pas d'individu
- 1 : vert - individu sain, non malade
- 10 à 10 + n : rouge - Infecté. n représente le nombre de jours où l'individu est infecté
- 3 : gris - rétabli

Les transitions :

- état 0 : pas de changement
- état 1 : pas de changement
- état $e = 10$ à $e = 10+n$: infecté
 - propagation aux voisins avec une probabilité p
 - e augmente de 1
- si $e \geq 20$ alors l'individu est rétabli
- état 3 : individu rétabli - Pas de changement



Règles 2: modèle SIR étendu

Les états :

- 0 : jaune - pas d'individu
- 1 : vert - individu sain, non malade
- 10 à 10 + c : orange - l'individu est contaminé mais ne transmet pas le virus
- 20 à 20 + n : rouge - Infecté. L'individu peut transmettre la maladie
- 3 : rétabli

Les transitions :

- état 0 : pas de changement
- état 1 : pas de changement
- état e = 10 à e = 19 : contaminé
 - pas de propagation aux voisins
 - e augmente de 1
 - si e \geq 20 alors l'individu est infecté
- état e = 20 à e = 29 : infecté
 - propagation aux voisins avec une probabilité p
 - e augmente de 1
 - si e > 29 alors l'individu est rétabli
- état 3 : individu rétabli - Pas de changement

3.12 Sujet 12 : jeu snake (***)

Rechercher les règles du jeu Snake.

Utiliser les automates cellulaires pour programmer le jeu.

On peut utiliser le module PySimpleGUI.

3.13 Sujet 13 : jeu de chute d'objets (***)

Les objets sont représentés par des cellules d'une certaine couleur. Il faut éviter qu'ils arrivent en bas.

On peut utiliser le module PySimpleGUI.