

GAME OF LIFE



Εργασία στο μάθημα Εισαγωγή στην επιστήμη του Ηλεκτρολόγου
Μηχανικού 2^{ου} εξαμήνου.

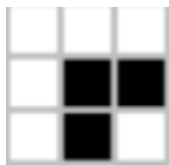
Επιβλέπον καθηγητής κ. Σγάρμπας.

Από: Κωνσταντίνος Τσάμπρας

up1083865

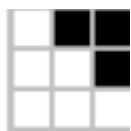
To Game of Life είναι ένα από τα πιο γνωστά κυτταρικά αυτόματα, δηλαδή υπολογιστικά συστήματα τα οποία με συγκεκριμένες αρχικές συνθήκες και (συνήθως) απλούς κανόνες μπορούν να καταλήξουν σε καταστάσεις που χαρακτηρίζονται από χαοτική πολυπλοκότητα ή σε καταστάσεις όπου επικρατεί τάξη και συχνά περιοδικότητα. Συγκεκριμένα στο Game of Life δεδομένων των αρχικών συνθηκών μπορούμε να προβλέψουμε την κατάσταση του συστήματος μετά από οποιονδήποτε αριθμό «βημάτων», αφού δεν απαιτείται ή προβλέπεται κάποια άλλη είσοδος.

Οι κανόνες: Το σύστημα αποτελείται από ένα (συνήθως) ορθογώνιο πλέγμα το οποίο περιέχει Χ κελιά (ή κύτταρα). Το κάθε κύτταρο μπορεί να έχει δύο καταστάσεις: νεκρό (λευκό) ή ζωντανό (μαύρο). Η κατάσταση του κάθε κελιού κάθε χρονική στιγμή καθορίζεται από την κατάσταση του αλλά και από την κατάσταση των γειτονικών του κελιών την προηγούμενη χρονική στιγμή. Πιο συγκεκριμένα όταν έχουμε ένα ζωντανό κελί αυτό θα παραμείνει ζωντανό αν την επόμενη χρονική στιγμή υπάρχουν είτε δύο είτε τρία ζωντανά κελιά στην «γειτονιά» του:



(εδώ το κεντρικό κελί θα παραμείνει ζωντανό την επόμενη χρονική στιγμή)

Ενώ ένα νεκρό κελί θα ζωντανέψει μόνο εάν υπάρχουν ακριβώς 3 ζωντανά κελιά στην γειτονιά του:



(εδώ το κεντρικό κελί θα ζωντανέψει την επόμενη χρονική στιγμή)

Σε κάθε άλλη περίπτωση το κελί είτε γίνεται είτε παραμένει νεκρό.

Αυτοί οι κανόνες προσομοιάζουν μια κοινωνία οργανισμών, αφού όταν σε ένα δεδομένο χώρο υπάρχουν πολλοί οργανισμοί (σε αυτή την περίπτωση πάνω από τρεις) υπάρχει υπερπληθυσμός και έτσι πληθυσμός αναγκαστικά θα μειωθεί, ενώ αν υπάρχουν πολύ λίγοι

οργανισμοί (σε αυτή την περίπτωση λιγότεροι από δύο) υπάρχει υποπληθυσμός και δεν μπορούν να επιβιώσουν οι οργανισμοί. Ενώ για να δημιουργηθούν/γεννηθούν νέοι οργανισμοί χρειάζονται ιδανικές συνθήκες (σε αυτή την περίπτωση να είναι τρεις οι ζωντανοί οργανισμοί στην γειτονιά τους).

Αυτοί οι κανόνες δημιουργήθηκαν το 1970 και έκτοτε έχουν δημιουργηθεί άπειροι προσομοιωτές και έχουν μελετηθεί άπειρα σχήματα που δημιουργούνται με διάφορες αρχικές συνθήκες. Αυτό όμως δεν σημαίνει ότι θα πρέπει και εμείς να χρησιμοποιήσουμε αναγκαστικά τους ίδιους κανόνες. Για παράδειγμα θα μπορούσαμε να θεωρήσουμε ιδανικές συνθήκες για την γέννηση ενός νέου οργανισμού και την ύπαρξη ακριβώς δύο ζωντανών οργανισμών στη γειτονιά του ή θα μπορούσαμε να αλλάξουμε τον ορισμό της γειτονιάς που μέχρι τώρα ήτανε ένας κύκλος με 1 ώστε να είναι ένας μεγαλύτερος κύκλος π.χ. ακτίνας 2. Έτσι σκέφτηκα στα πλαίσια της εργασίας αυτής να δημιουργήσω ένα πρόγραμμα το οποίο θα δίνει τη δυνατότητα στο χρήστη να επιλέγει τα εξής: το μέγεθος των κελιών, το πλήθος των στηλών και των γραμμών του συστήματος, και τους κανόνες δηλαδή το πόσα ζωντανά κελιά πρέπει να υπάρχουν στην περιοχή ενός ζωντανού κελιού ώστε αυτό να παραμείνει ζωντανό την επόμενη χρονική στιγμή, αλλά και πόσοι ζωντανοί οργανισμοί θεωρούνται ιδανική συνθήκη για τη γέννηση ενός καινούργιου, ενώ τέλος ο χρήστης μπορεί και να αλλάζει το μέγεθος της ακτίνας που χαρακτηρίζει τη γειτονιά του οργανισμού.

Size of cells: 25
Number of rows: 25
Number of columns: 25
Radius of effect: 1
Live cells survive if neighbors: 2,3
Dead cells come alive if neighbors: 3
Time step: 1
Insert file name of the layout: .txt

Create Board
LOAD LAYOUT
START
AUTOMATE
NEXT STEP
SAVE LAYOUT
END

Steps= 0

To interface:

Η είσοδος των μεταβλητών αυτών γίνεται μέσω entry boxes στην pythοn στις οποίες ως προεπιλογή εμφανίζονται οι Κανόνες του αρχικού Game of Life όπως τους όρισε ο Conway:

Size of cells:	25
Number of rows:	25
Number of columns:	25
Radius of effect:	1
Live cells survive if neighbors:	2,3
Dead cells come alive if neighbors:	3

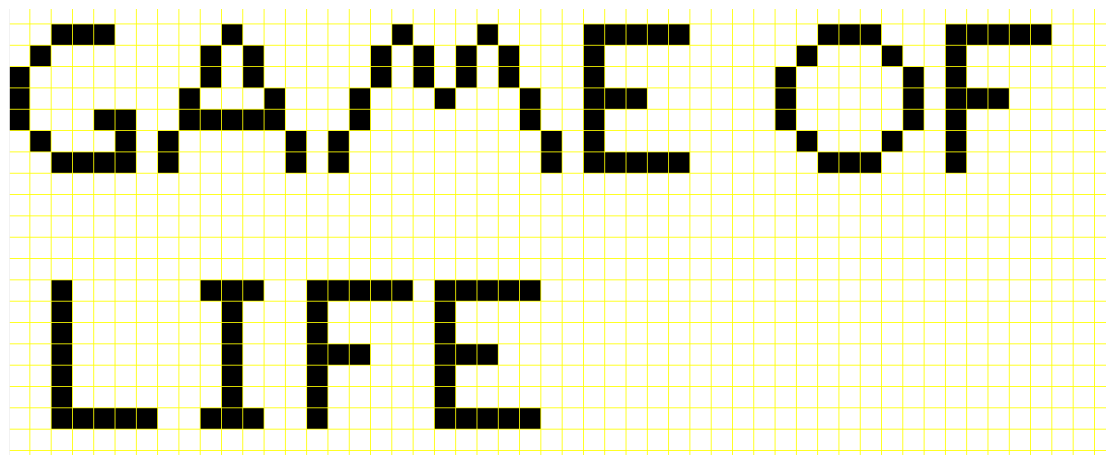
Δημιουργία αρχικών συνθηκών: Όταν ο χρήστης επιλέξει το κουμπί Create Board θα δημιουργηθεί ένας «κενός» πίνακας σύμφωνα με τις διαστάσεις που έδωσε ο χρήστης:

Size of cells: 25
Number of rows: 25
Number of columns: 25
Radius of effect: 1
Live cells survive if neighbors: 2,3
Dead cells come alive if neighbors: 3
Time step: 1
Insert file name of the layout: .txt

Create Board
LOAD LAYOUT
START
AUTOMATE
NEXT STEP
SAVE LAYOUT
END

Steps = 0

Όμως προφανώς για να υπάρξει δραστηριότητα κατά την διάρκεια της προσομοίωσης πρέπει ο χρήστης να επιλέξει τα κελιά τα οποία θα είναι ζωντανά την χρονική στιγμή 0. Αυτό γίνεται με ένα αριστερό κλικ στο εκάστοτε κουτάκι, το οποίο θα αλλάξει την κατάσταση του.



Η προσομοίωση: Έπειτα όποτε τελειώσει με την δημιουργία των αρχικών συνθηκών πατάει στο κουμπί start simulation και η προσομοίωση είναι έτοιμη να ξεκινήσει. Πατώντας το κουμπί next step η προσομοίωση υπολογίζει και παρουσιάζει στο παράθυρο την κατάσταση την επόμενη χρονική στιγμή ανάλογα με τους κανόνες που έχει εισάγει ο χρήστης αλλά και τις αρχικές συνθήκες αλλά και τις συνθήκες της προηγούμενης χρονικής στιγμής. Επειδή πολλές προσομοιώσεις αρχίζουν να εμφανίζουν ενδιαφέροντα αποτελέσματα μετά από εκατοντάδες βήματα και επειδή ο χρήστης δεν θα ήθελε να χρειάζεται να πατήσει τόσες φορές το κουμπί nextstep υπάρχει και η επιλογή να αυτοματοποιήσει την προσομοίωση πιο συγκεκριμένα πατώντας στο κουμπί automate και πηγαίνοντας τον κέρσορα στο ειδικό πράσινο χώρο η προσομοίωση θα αρχίζει να παρουσιάζει υπολογίζει και να παρουσιάζει τις επόμενες χρονικές στιγμές ενώ ο χρόνος αναμονής για κάθε χρονική στιγμή θα είναι αυτός που εισάγει ο χρήστης στο κατάλληλο entry box:

Size of cells:	25
Number of rows:	25
Number of columns:	70
Radius of effect:	1
Live cells survive if neighbors:	2,3
Dead cells come alive if neighbors:	3
Time step:	1

Προς διευκόλυνση του χρήστη έχει προστεθεί η δυνατότητα να αποθηκεύσει σε αρχείο κειμένου με κωδικοποιημένο τρόπο το layout της παρούσας στιγμής δηλαδή το μέγεθος των κελιών το πλήθος των γραμμών και των στηλών αλλά και τις αρχικές συνθήκες που υπάρχουν εκείνη τη στιγμή στον πίνακα αλλά και η δυνατότητα να φορτώσει ένα layout από ένα ήδη αποθηκευμένο αρχείο προσθέτοντας το όνομα του αρχείου στο κατάλληλο entry box και πατώντας αντίστοιχα save ή load layout:

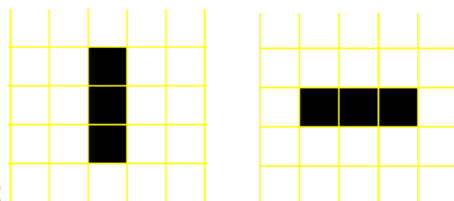
Insert file name of the layout	
Create Board	gosper_glider_gun.txt
START	LOAD LAYOUT
NEXT STEP	AUTOMATE
	SAVE LAYOUT

Παράλληλα έχει προστεθεί και ένας μετρητής βημάτων για ευνόητους

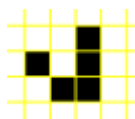
λόγους: **Steps= 420**

ΕΝΔΙΑΦΕΡΟΝΤΑΙ ΣΧΗΜΑΤΑ

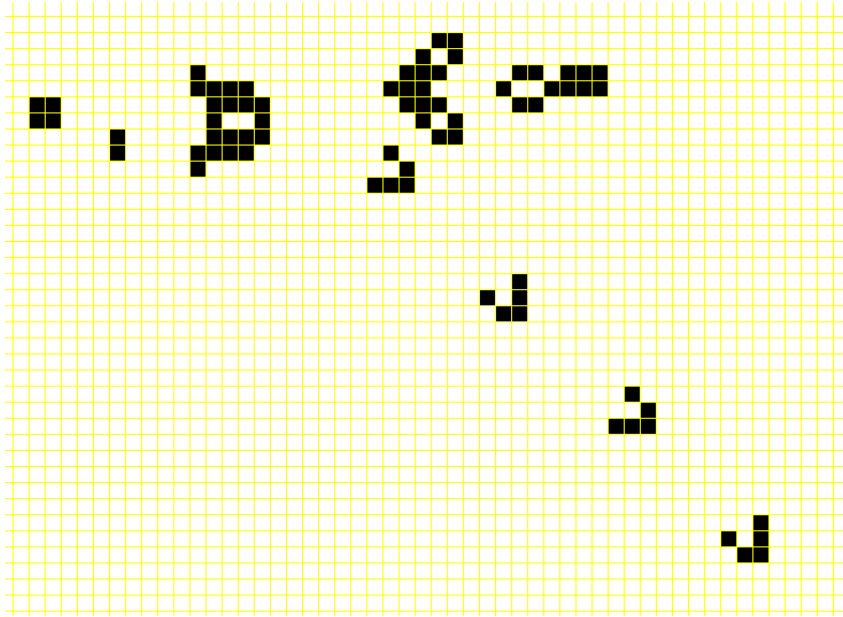
1. Απλός ταλαντωτής με περίοδο 2:



2. Glider:

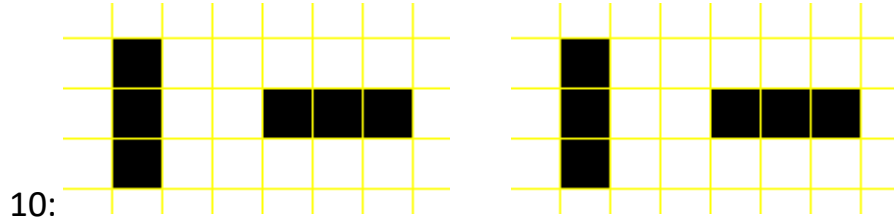


3. Gosper's glider gun:

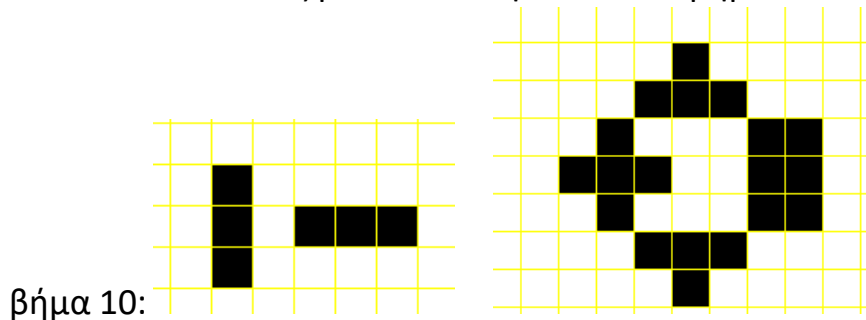


Η πολυπλοκότητα των σχημάτων που μπορούν να δημιουργηθούν είναι απεριόριστη, ενώ μία ελάχιστη διαφορά των αρχικών συνθηκών μπορεί να έχει τεράστιες διαφορές μετά από λίγα βήματα:

- 2 ταλαντωτές που απέχουν 2 κελιά, στο βήμα 0 και το βήμα



- Οι ίδιοι ταλαντωτές με απόσταση 1 κελί στο βήμα 0 και στο



ΒΙΒΛΙΟΓΡΑΦΙΑ

- https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life
[https://en.wikipedia.org/wiki/Gun_\(cellular_automaton\)](https://en.wikipedia.org/wiki/Gun_(cellular_automaton))
- Για τον προγραμματισμό:
<https://www.tutorialspoint.com/python/index.htm>
<https://stackoverflow.com/>
<https://docs.python.org/3/library/tkinter.html>
<https://pyautogui.readthedocs.io/en/latest/>

Οι βιβλιοθήκες που χρησιμοποιήθηκαν:

- Tkinter
- Numpy (απαιτεί ειδική εγκατάσταση, δεν είναι built-in!)
- Time
- Pyautogui (απαιτεί ειδική εγκατάσταση, δεν είναι built-in!)