

Βασίλης Βλάχος

Up1100832

2022-2023

Conways's Game of life

Περίληψη

Το παρών κείμενο αποτελεί την έκθεση της ατομικής εργασίας στα πλαίσια του μαθήματος της εισαγωγής στην επιστήμη του ηλεκτρολόγου μηχανικού. Το θέμα της ατομικής μου εργασίας είναι το πρώτο θέμα (Conway's Game of Life) από τα προτεινόμενα του καθηγητή Κυριάκου Σγάρμπα.

I. ΕΙΣΑΓΩΓΗ

Συγκεκριμένα δημιούργησα ένα πρόγραμμα που αναπαριστά γραφικά μέσω ενός παιχνιδιού τον αλγόριθμο Conway's Game of Life [1] στην γλώσσα προγραμματισμού Python. Για την δημιουργία του προγράμματος μου χρησιμοποίησα σαν template τον κώδικα [2] και το βίντεο του δημιουργού που αναλύει τον κώδικά του [3].

Αυτοί οι δύο σύνδεσμοι με βοήθησαν τόσο στην περεταίρω κατανόηση του αλγορίθμου Conway's Game of Life αλλά και στο πώς να προσεγγίσω προγραμματιστικά στο πρόβλημα.

Θα ήθελα μετά το τέλος της έκθεσης να δείτε τον template κώδικα και να τον συγκρίνετε με τον δικό μου ώστε να διαπιστώσετε πόσο πιο καθαρογραμμένος είναι ο δικός μου κώδικας αλλά και πόσες παραπάνω λειτουργίες έχει.

Ξεκίνησα γράφοντας το δικό μου πρόγραμμα με βάση το template αλλά με πολλές παραπάνω λειτουργίες καθώς και βελτιώσεις στον κώδικα. Αναλυτικότερα παρέμειναν μόνο 2 συναρτήσεις απαραίλλακτες (create_grid(), draw_grid()). Αποφάσισα να μετασχηματίσω τον τρόπο που λειτουργεί το πρόγραμμα δημιουργώντας τις λειτουργίες Pause, Change configuration, Wipe, Population count, Birth και Death count και μετασχηματίζοντας τις Reset, Generation και Stop. Επίσης, υλοποιήθηκε η προσθήκη menu ώστε ο χρήστης να μπορεί να έχει πολλά παράθυρα με παιχνίδια που τρέχουν ταυτόχρονα και να τα συγκρίνει. Επιπλέον, επιτεύχθηκε η διαδραστικότητα του χρήστη με το πρόγραμμα επιτρέποντάς του να παρέμβει στη λειτουργία του μέσω του Change configuration. Τέλος, μπορεί να γίνει προσαρμογή των διαστάσεων του grid και των κελιών, υπάρχουν οι λειτουργίες σφαιρικού κόσμου (όπου τα κελιά στα όρια του grid αλληλοεπιδρούν μεταξύ τους) και επίπεδου κόσμου καθώς και ήχος έναρξης.

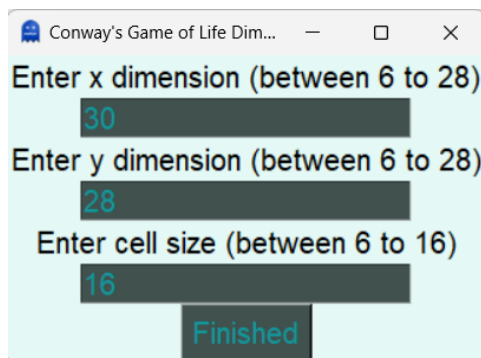
Ανοίγοντας το πρόγραμμα εμφανίζεται το παρακάτω μενού (Εικ.1) που παρέχει στον χρήστη την λειτουργία του Random generated game, του User generated game και την επιλογή exit για να κλείσει το πρόγραμμα, αυτό έχει υλοποιηθεί στην main συνάρτηση του προγράμματος.

II. ΑΝΑΠΤΥΞΗ ΠΡΟΓΡΑΜΜΑΤΟΣ

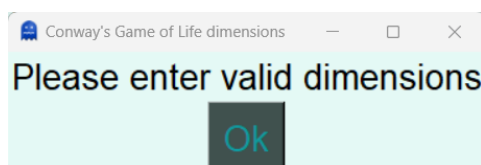


Εικόνα 1. Μενού προγράμματος.

Οποιαδήποτε από τις πρώτες 2 επιλογές επιλέξει ο χρήστης το πρόγραμμα τον ρωτάει τις διαστάσεις που θέλει να έχει το grid (Εικ.2) και το μέγεθος των κελιών, το οποίο έχει υλοποιηθεί με την κλάση Dimensions, στην συνέχεια όταν ο χρήστης επιλέξει τις διαστάσεις τότε ελέγχεται με την συνάρτηση pushed_dimensions() αν αυτές είναι έγκυρες και καλείτε ανάλογα είτε το παράθυρο του error (Εικ.3) είτε το παράθυρο που ρωτάει τον χρήστη εάν θέλει σφαιρικό ή επίπεδο κόσμο μέσω της συνάρτησης world().

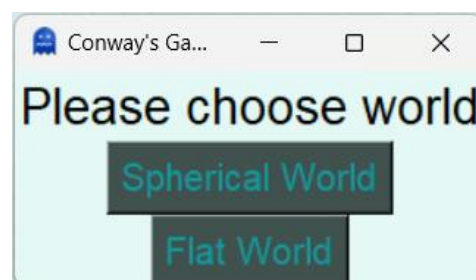


Εικόνα 2. Παράθυρο εισαγωγής διαστάσεων.



Εικόνα 3. Παράθυρο λάθος διαστάσεων.

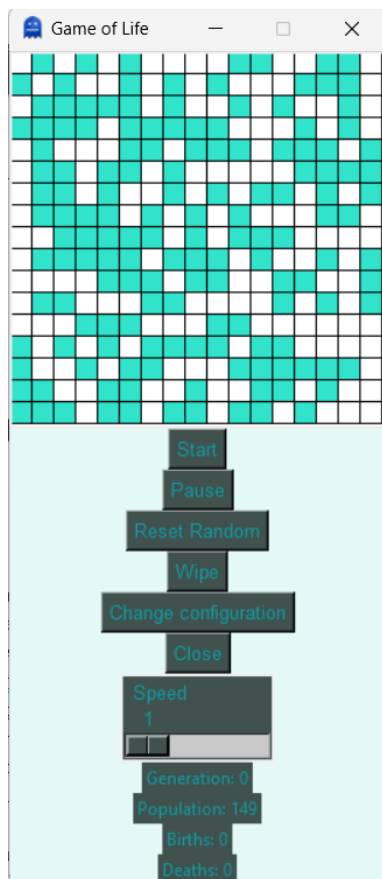
Εάν ο χρήστης πληκτρολογήσει έγκυρες διαστάσεις τότε ανοίγει το παράθυρο που ο χρήστης επιλέγει αν θέλει σφαιρικό ή επίπεδο κόσμο (Εικ.4), επιλέγοντας ένα από τα δύο ξεκινάει το παιχνίδι με τον αντίστοιχο κόσμο μέσω της συνάρτησης start(). Οι διαφορές των δύο είναι πως στον σφαιρικό κόσμο τα κελιά στα όρια του πλέγματος αλληλοεπιδρούν μεταξύ τους σαν γείτονες ενώ στον επίπεδο κόσμο τα κελιά στα όρια έχουν γείτονες μόνο τα «οπτικά γειτονικά κελιά» (δηλαδή αυτά γύρω τους στο grid) έτσι τα κελιά στην ακμή του πλέγματος αντί για 9 γείτονες έχουν 6 και αν είναι γωνιακά έχουν μόνο 3.



Εικόνα 4. Παράθυρο επιλογής κόσμου

Ας δούμε τι γίνεται εάν ο χρήστης επιλέξει το Random generated game, πληκτρολογήσει σωστά τις διαστάσεις και επιλέξει έναν από τους δύο τύπους κόσμου, τότε καλείτε η κλάση του παιχνιδιού (Εικ.5) GameOfLife με τα κατάλληλα ορίσματα (πλήθος στηλών και γραμμών, διαστάσεις κελιών, ταχύτητα του παιχνιδιού, το mode που θα τρέξει (random ή user generated) και τον κόσμο

(σφαιρικός ή επίπεδος)). Εδώ χρησιμοποιούνται διάφορες συναρτήσεις όπως η `create_grid()` που δημιουργεί το random generated grid, η `start_drawing()` που δημιουργεί το παράθυρο με τον καμβά και με την σειρά της καλεί τις `draw_grid()` που ζωγραφίζει στον καμβά με τα cells και την `create_buttons()` που δημιουργεί τα buttons και το count των generation, population, births και deaths (μέσω των συναρτήσεων `calc_population()`, `create_label()` και `update_generation_population()`).

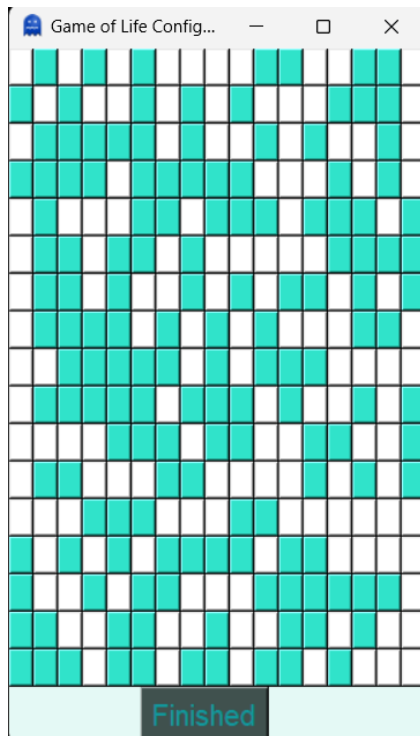


Εικόνα 5. Παράθυρο παιχνιδιού

Μπορούμε να δούμε πως ο χρήστης έχει τις δυνατότητες:

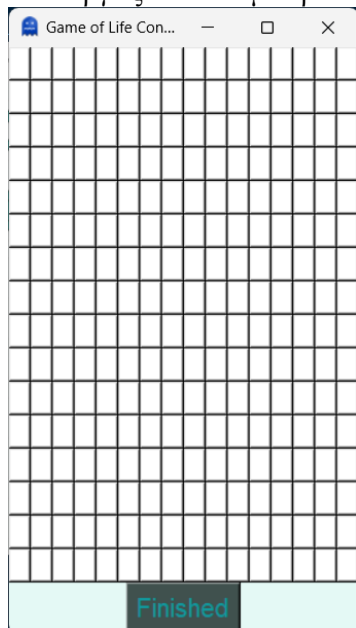
- **Start:** η οποία καλεί την συνάρτηση `run()` και ξεκινάει το παιχνίδι με την αυτό-αναφορική συνάρτηση `update()` η οποία ελέγχεται τότε καλεί ξανά τον εαυτό της μέσω της πολύ σημαντικής μεταβλητής `self.running`.

- **Pause:** η οποία αλλάζει την μεταβλητή `self.running` ώστε να σταματήσει η συνάρτηση `update` να καλεί τον εαυτό της.
- **Reset Random:** η οποία καλεί την συνάρτηση `reset()` και ξανά-ζωγραφίζει το grid τυχαία ενώ κάνει reset τα population, generation, deaths και births.
- **Wipe:** η οποία καλεί την `wipe()` που σκοτώνει όλα τα cells και κάνει reset τα population, generation, deaths και births.
- **Close:** που κλείνει το παράθυρο
- **Speed:** που ρυθμίζει την ταχύτητα του παιχνιδιού μέσω της συνάρτησης `speed()`
- **Πρόσβαση στις πληροφορίες** generation, population, births και deaths.
- **Change configuration:** η πολυπλοκότερη λειτουργία του προγράμματος η οποία καλεί τη συνάρτηση `change()` και η ίδια με τη σειρά της χρησιμοποιεί τις συναρτήσεις `clicked()`, `clicked1()`, `finished()`. Η λειτουργία αυτή κάνει pause το παιχνίδι (εάν δεν έχει γίνει ήδη από τον χρήστη), αναγνωρίζει το grid και δημιουργεί ένα καινούργιο παράθυρο (Εικ.6) με buttons και επιτρέπει στον χρήστη να αλλάξει την κατάσταση των κελιών, με αριστερό click ζωντανεύει το κελί που επέλεξε ο χρήστης και με δεξί click σκοτώνει το κελί που επέλεξε ο χρήστης. Όταν ο χρήστης τελειώσει πατάει το κουμπί `finished`, κλείνει το παράθυρο και γίνεται `update` ο καμβάς με τα κελιά καθώς και το population.



Εικόνα 6. Παράθυρο επεξεργασίας κελιών.

Τώρα εάν ο χρήστης επιλέξει το User input game στο menu και πληκτρολογήσει έγκυρες διαστάσεις τότε καλείτε η συνάρτηση change() που είδαμε πριν όμως με κενό grid αυτή τη φορά (Εικ.7). Στη συνέχεια μόλις ο χρήστης πατήσει finished κλείνει το παράθυρο και ανοίγει το παράθυρο του παιχνιδιού με όλες τις λειτουργίες που είδαμε πριν.



Εικόνα 7. Παράθυρο επεξεργασίας κελιών στο user generated mode

III. ΣΥΝΟΨΗ

Κλείνοντας, το πρόγραμμα επιχειρεί να αναπαραστήσει τον αλγόριθμο Conway's Game of Life με έναν δια-δραστικό τρόπο και να προσφέρει όσο το δυνατόν περισσότερες λειτουργίες και πληροφορίες στον χρήστη. Προσωπικά, θεωρώ πως έχω πρωτοτυπήσει σε διάφορους τομείς όπως με την αρχική λειτουργία User generated game και την in game λειτουργία change configuration που ζωντανεύει και σκοτώνει cells ο χρήστης. Επίσης με τις λειτουργίες wipe, reset random, τις πληροφορίες population, births, deaths την επιλογή στον χρήστη να διαλέξει διαστάσεις για το πλέγμα και μέγεθος για τα κελιά. Φυσικά, τις λειτουργίες flat και spherical world και τέλος το menu που παρότι φαινομενικά ασήμαντο στην πραγματικότητα δεν είναι γιατί επιτρέπει στον χρήστη να έχει πολλά παράθυρα με παιχνίδια ανοιχτά ώστε να τα παρατηρεί και να τα συγκρίνει όπως αυτός θέλει.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1]https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life

[2]<https://github.com/YTSimplePython/PythonScripts>

[3]<https://www.youtube.com/watch?v=51lN2d9N4n0>