

Progetto di Recupero: LIFE

Programmazione: Modulo 1 – Laboratorio Di Programmazione
Barbara Guidi, Susanna Pelagatti

AA 2022/23

1 Introduzione

Il progetto tratta della simulazione interattiva del *Gioco della Vita* ¹

”Si tratta in realtà di un gioco senza giocatori, intendendo che la sua evoluzione è determinata dal suo stato iniziale, senza necessità di alcun input da parte di giocatori umani. Si svolge su una griglia di caselle quadrate (celle) che si estende all’infinito in tutte le direzioni; questa griglia è detta mondo. Ogni cella ha 8 vicini, che sono le celle ad essa adiacenti, includendo quelle in senso diagonale. Ogni cella può trovarsi in due stati: viva o morta (o accesa e spenta, on e off). Lo stato della griglia evolve in intervalli di tempo discreti, cioè scanditi in maniera netta. Gli stati di tutte le celle in un dato istante sono usati per calcolare lo stato delle celle all’istante successivo. *Tutte le celle del mondo vengono quindi aggiornate simultaneamente nel passaggio da un istante a quello successivo: passa così una generazione.*

Le transizioni dipendono unicamente dallo stato delle celle vicine in quella generazione:

- Qualsiasi cella viva con meno di due celle vive adiacenti muore, come per effetto d’isolamento;
- Qualsiasi cella viva con due o tre celle vive adiacenti sopravvive alla generazione successiva;
- Qualsiasi cella viva con più di tre celle vive adiacenti muore, come per effetto di sovrappopolazione;
- Qualsiasi cella morta con esattamente tre celle vive adiacenti diventa una cella viva, come per effetto di riproduzione.”

2 Cosa vogliamo realizzare

Vogliamo realizzare uno script Python attivabile attraverso una opportuna GUI progettata dallo studente e realizzata con il modulo tkinter. Le caratteristiche minime dell’interfaccia sono specificate nel seguito.

¹https://it.wikipedia.org/wiki/Gioco_della_vita

Lo script simula una griglia quadrata del mondo di lato l per t istanti. Deve essere possibile caricare la griglia da un file o iniziare una nuova simulazione da zero.

Nel primo caso l'utente deve poter specificare il nome del file da caricare oltre che il numero t di passi di simulazione da eseguire.

Nel secondo caso, l'utente deve poter specificare i valori del lato l , del numero dei passi di simulazione t da eseguire e la percentuale di riempimento iniziale della griglia $0 < \sigma < 1$. Viene quindi creata una griglia $l \times l$ con $\sigma * l * l$ celle vive.

Una volta caricata o generata la griglia di partenza, si simula il gioco visualizzando ad ogni passo la griglia utilizzando il modulo `turtle` (suggerimento: in questo caso usare il metodo `fill` per riempire delle forme e il metodo `goto` per spostarsi in un punto del piano cartesiano – leggere attentamente la documentazione online come usarli e per altri metodi utili). La visualizzazione avviene ad ogni passo con intervalli di 1 secondo fra una visualizzazione di un passo e la visualizzazione del successivo.

Alla fine della simulazione, la GUI interagisce con l'utente per sapere se se lo stato finale della griglia deve essere salvato su file, e in questo caso permette all'utente di specificare il nome del file. Se il file esiste viene richiesta conferma per sovrascrivere.

In sintesi, GUI deve permettere le seguenti azioni:

- **attivare una simulazione:** in questo caso deve chiedere all'utente se desidera avviare una nuova simulazione o caricare da file una simulazione salvata precedentemente.
 - nel caso di *nuova simulazione* deve permettere all'utente di scegliere il valore del lato l , il numero dei passi t e la percentuale di riempimento $0 < \sigma < 1$
 - nel caso di *simulazione caricata da file* deve permettere all'utente di scegliere il numero dei passi t e il nome del file. E' richiesto di gestire eventuali errori ed informare l'utente.
- **salvare su file lo stato di una simulazione terminata.** In questo caso l'utente deve poter specificare il nome del file. E' richiesto di gestire eventuali errori ed informare l'utente.

3 Cosa consegnare

Lo script deve essere realizzato in un unico file `life.py`, eventuali test automatici (opzionali) devono essere predisposti in un file separato `testLife.py`. Il codice deve essere documentato da commenti chiari secondo le convenzioni viste a lezione.

Inoltre deve essere predisposta una relazione (formato PDF) che spiega le scelte progettuali, le strutture dati e le classi utilizzate, gli eventuali test automatici e motiva le scelte fatte in modo adeguato. La relazione non deve spiegare

il codice passo passo ma fornire una visione ad alto livello, in modo da far comprendere al lettore il progetto complessivo.

La relazione deve essere completata con un Manuale Utente che spiega come attivare la GUI, come interagire con essa e come attivare gli eventuali test.

Lo script e la relazione devono contenere i nomi dei componenti del gruppo (massimo 2 persone), la loro matricola e i riferimenti per contatti via mail e opzionalmente telefono.

La consegna deve avvenire via mail a entrambi i docenti e contenere:

- il file `life.py` ed eventualmente `testLife.py`
- un file contenente una griglia di esempio (per poterlo caricare dalla GUI)
- la relazione formato PDF

4 Parti opzionali

Opzionalmente si possono aggiungere funzioni alla GUI o allo script, ad esempio si può rendere possibile la scelta dell'intervallo in secondi fra una visualizzazione o l'altra o la scelta del colore delle celle vive o morte o altro.

Ricordiamo tuttavia che tutte le funzioni richieste dal presente documento devono essere presenti e che le funzionalità opzionali devono essere spiegate e documentate nella relazione e nel Manuale Utente.

5 Materiale didattico e informazioni

Tutto il materiale didattico si trova sul sito Moodle ufficiale del corso. I docenti sono disponibili tutto l'anno per ricevimento in presenza, online o per email.