

Grupo 3

Integrantes: Andres Yañez, Carolina Rojano, Ana Karina Pulido.

Repositorio: https://github.com/TheChieft/MACC-game_of_life_Pandemic_simulator.git

Avance en los objetivos:

1. La creación de una interfaz interactiva con el usuario, que le permite conocer el objetivo del juego, las instrucciones y algunos datos curiosos sobre que podría encontrar en la simulación.
2. Uno de los principales cambios respecto al juego de 1970, es un contador en pantalla que permite conocer en tiempo real cuántas células vivas hay.
3. Se definió una nueva regla en el juego, la cual permite que si una célula no infectada comparte cierta distancia con el virus, esta pasará a ser una célula infectada.
4. Se creó el módulo que a partir de los datos recolectados en la simulación hará las gráficas, hasta el momento se tienen tres modelos para presentar los datos al usuario.
5. Se implementó una función para conocer el tiempo que duró la simulación (en milisegundos) y en este tiempo cuántos afectados hay.
6. Es necesario mejorar la definición de las nuevas reglas y funciones ya que actualmente el comportamiento del virus en el plano no es analizable.
7. Conforme se pueda analizar mejor el comportamiento del virus podremos obtener más datos para entregar un mejor informe al usuario. Como objetivo para la entrega final, se fija la creación de gráficas animadas y con mayor información.
8. Como objetivo para la próxima entrega, se espera añadir imágenes con las instrucciones y objetivos del juego. Al igual, que agregar nuevas opciones en la interfaz para mejorar la interacción con el usuario.

Descripcion del codigo:

- I. Archivos:** `game_of_life_virus.py`, `modulo_1.py`, `Game_of_life_3.0.py`
- II. Bloque de ejecución:** `Game_of_life_3.0.py`
- III. Clases y funciones:** Para el archivo **`game_of_life_virus.py`**, Función **`show_poblacion()`**: Muestra las células vivas en tiempo real en la pantalla del juego; Función **`datos_importantes()`**: Almacena en otro archivo datos recolectados en la simulación. Para el archivo `modulo_1`, función **`dataframe(lista, lista_names)`**, se encarga de crear un serie con pandas; función **`lista(blancos, rojos, total, segundos)`**

recibe los datos de la simulación y crea una lista para manipularlos; función **letalidad(muertos, total)** estimando la letalidad del virus en la población; función **porcentajes(df)** recibe la lista creada anteriormente para crear porcentajes; función **pie_graph(porcentajes)** función que crea una grafica tipo pie para mostrar porcentaje de infectados y no infectados. Para el archivo **Game_of_life_3.0.py**, función **Rules()** donde se visualiza la introducción del juego y las reglas del mismo; **Play()** se encuentra el código del juego que será ejecutado ; **Menú()** donde se definen las opciones que tiene el usuario al ingresar al programa.

IV. Flujo del programa:

- Se importan las librerías necesarias.
- Se llama la función Menu()
- Menu() abre campo a Play(), Rules(), exit()
- En Play() se llama al archivo game_of_life_virus.py
- Dentro del archivo game_of_life_virus.py se define la ventana del juego, las reglas que cumplen las células y todo lo que pasa en la simulación.
- Igualmente, se invoca modulo_1.py el cual permite crear gráficas y manipular los datos de la simulación.
- En Rules() se utiliza la función draw.text() para ver el título en ventana, con la tecla Q se devuelve al menú.
- Al final de la ejecución (cuando el usuario cierra la ventana de la simulación) se abre una ventana con la gráfica tipo pie de lo ocurrido en el juego.