

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. Es necesario redefinir la función **Play()** para que el juego se pueda ejecutar directamente sin que el usuario deba seguir instrucciones de redireccionamiento.
8. 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.
9. El objetivo principal para la entrega final es unir el código de la interfaz con el juego, al igual que lograr visualizar la propagación del virus en la población .

Descripcion del codigo:

- I. **Archivos:** game_of_life_virus.py, modulo_1.py
- II. **Bloque de ejecución:** game_of_life_virus.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()** donde se redirecciona al usuario al archivo del juego para que sea ejecutado; **Menú()** donde se definen las opciones que tiene el usuario al ingresar al programa.

Importante: De momento, para poder ejecutar el juego es necesario que en la función **Play()** se escriba la dirección del archivo **game_of_life_virus.py** en su computador.

IV. Flujo del programa:

- Iniciar Pygame y elaboración de ventana.
- Configuración de características de la ventana, como el color y el tamaño.
- Definición de variables globales.
- inicialización de todas las celdas en estado inicial 0 (muertas).
- Creación de virus en posición aleatoria.
- Creación de función para poner marcador en pantalla.
- Creación de función para el almacenamiento de datos importantes.
- Elaboración de vectores 2D para el virus.
- Definición de eventos (¿Que pasa cuando el estado de la celda es x número?)
- Elaboración de vectores 2D para las células.
- Definición de las reglas del juego.
- Definición de los estados del juego.
- Actualización de la ventana.
- Finalizar Pygame.