

Universidad Politecnica Salesiana

Nombre: Erika Morocho

Asignatura: Simulación

Para realizar esta simulacion se utilizara la libreria pygame para ello se debe instalar pygame

conda install -c condaforge pygame.

Introduccion

En una epidemia, el parámetro fundamental, del que todo depende, es R0. Este símbolo se refiere al número de personas que, mede cada infectado contagia antes de convertirse en inofensivo (bien porque está en aislamiento, hospitalizado o ha muerto).

El valor R0 es fundamental, porque si es grande, el contagio se alarga más rápidamente. Si R0 es 2, y si el tiempo medio en el que se permanece contagiados es una semana, y hay 1.000 infectados, entonces después de una semana los infectados será 3.000 (los 1.000 del inicio más 2.000 nuevos contagiados).

Si R0 es 5, después de una semana los infectados será 6.000 (los 1.000 de partida más 5.000 nuevos contagiados). A este punto, el ciclo vuelve a empezar, con más o menos retraso, dependiendo de cuánto tiempo un nuevo infectado pmlta en convertirse a sí mismo en contagiado.

```
In [ ] : from random import randrange
import pygame
import os
from time import time

#Parametros de inicio
PROBA_MUERTE = 8.4 # Probabilidad de que la gente muera COVID
CONTAGION_RATE = 5 # Factor R0 para la simulacion COVID probabili
dad
PROBA_VACU = CONTAGION_RATE * 10
PROBA_VACU = 0 # Probabilidad de que exista una vacuna, COVID = 0
SIMULACION_SPEED = 25 # Tiempo de un dia en milisegundos (Cada 25
es un dia)
nb_rows = 100 #Numero de filas
nb_cols = 100 #Numero de columnas

global display, myfont, states, states_temp #Declaracion de variab
les globales

#Declaro colores en formato RGB
WHITE = (255, 255, 255)
BLUE = (0, 0, 255)
GREEN = (0, 247, 0)
BLACK = (0, 0, 0)

# Tiempo de ejecución
tiempo_Inicial = time()
tiempo_Actual = 0

#Obtiene los vecinos dado un punto x, y
def get_vecinos(x, y):
    incx = randrange(3)
    incy = randrange(3)
    incx = (incx * 1) - 1
    incy = (incy * 1) - 1
    x2 = x + incx
    y2 = y + incy
    if x2 < 0:
        x2 = 0
    if x2 >= nb_cols:
        x2 = nb_cols - 1
    if y2 < 0:
        y2 = 0
    if y2 >= nb_rows:
        y2 = nb_rows - 1
    return [x2, y2]

#Genero las personas que cuentan con inmunidad o vacuna
def vacunar():
    for x in range(nb_cols):
        for y in range(nb_rows):
            if randrange(99) < PROBA_VACU:
                states[x][y] = 1

#Funcion que permite contar el numero de muertos
def contar_muertes():
    contador = 0
    for x in range(nb_cols):
        for y in range(nb_rows):
            if states[x][y] == -1:
                contador += 1
    return contador

def contar_recuperados():
    contador = 0
    for x in range(nb_cols):
        for y in range(nb_rows):
            if states[x][y] == 1:
                contador += 1
    return contador

#Definimos datos de inicio
states = [[0] * nb_cols for i in range(nb_rows)]
states_temp = states.copy()
states[50][50] = 10 # Estado inicial de la simulacion
it = 0 # Variable para contar las Iteraciones
total_muerte = 0 # Contabiliza el numero de muertos
recuperados = 0 # Contabiliza el numero de muertos
vacunar() #Llamar a la funcion vacunar
dias = 0

pygame.init() #Incializo el motor de juegos pygame
pygame.font.init() #Inicializo el tipo de letra
display=pygame.display.set_mode((800,750),0,32) #Tamaño de la ven
tana
pygame.display.set_caption("Simulacion de Epidemia Covid-19 Ecuado
r")# Titulo
pygame.font.SysFont('Calibri', 40) # Tipo de letra
display.fill(WHITE) # Color de fondo

# Coordenadas de x & y para la asignación de los estados
px = 105
py = 105

while True:
    pygame.time.delay(SIMULACION_SPEED) # Sleep o pausa
    it += 1
    if it <= 10000 and it >= 1:
        dias = int((tiempo_Actual-0.025)
        states_temp = states.copy()
        for x in range(nb_cols):
            for y in range(nb_rows):
                state = states[x][y]
                if state == -1:
                    pass
                if state >= 10:
                    states_temp[x][y] = state + 1
                if state >= 20:
                    if randrange(99) < PROBA_MUERTE:
                        states_temp[x][y] = -1 # Muere
                    else:
                        states_temp[x][y] = 1 # Cura
                if state >= 10 and state <= 20:
                    if randrange(99) < PROBA_INFECT: # Infecto a l
as personas cercanas entre 10 y 20
                        neighbour = get_vecinos(x, y)
                        x2 = neighbour[0]
                        y2 = neighbour[1]
                        neigh_state = states[x2][y2]
                        if neigh_state == 0:
                            states_temp[x2][y2] = 10
                        states = states_temp.copy()
                        total_muerte = contar_muertes() # contar el numero de muer
tos
                        recuperados = contar_recuperados()

                    pygame.draw.rect(display, WHITE, (105, 20, 800, 90)) # Grafico
la fondo
                    textsurface = font.render("Muertes: "+ str(total_muerte), Fals
e, (255,160,122)) #El numero de muertos
                    display.blit(textsurface, (285, 20))
                    dias_Transcurridos = font.render("Días: "+ str(dias), False, (
255,160,122)) #El numero de muertos
                    display.blit(dias_Transcurridos, (105, 60))
                    recuperados = font.render("Recuperados: "+ str(recuperados), F
alse, (255,160,122)) #El numero de muertos
                    display.blit(recuperados, (400, 60))
                    #Graficar el estado del paciente matriz
                    for x in range(nb_cols):
                        for y in range(nb_rows):
                            if states[x][y] == 0:
                                color = BLUE # No infectado
                            if states[x][y] == 1:
                                color = GREEN # Recupero
                            if states[x][y] >= 10:
                                color = (states[x][y] * 12, 50, 50) # Infectado -
Rojo
                            if states[x][y] == -1:
                                color = BLACK # Muerto
                            pygame.draw.circle(display, color, (px, py), 2)
                            py += 6
                            px += 6
                        px = 105
                    #Escuchar los eventos del teclado
                    for event in pygame.event.get():
                        if event.type == pygame.KEYDOWN and event.key == pygame.K_
ESCAPE: #Presiona y Escape
                            pygame.quit() #Termino simulacion
                        if event.type == pygame.KEYDOWN and event.key == pygame.K_
SPACE: #Presiona y espacio
                            #Reiniciamos valores
                            tiempo_Inicial = time()
                            states = [[0] * nb_cols for i in range(nb_rows)]
                            states_temp = states.copy()
                            states[50][50] = 10
                            it = 0
                            total_muerte = 0
                            dias = 0
                            vacunar()

                            tiempo_Actual = time() - tiempo_Inicial
                            if tiempo_Actual > 1.225:
                                it = 10000
                            pygame.display.update()# Mandar actualizar la ventana

pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.
html
```

R0 investigar el valor de varianza del R0 dentro del Ecuador

Hasta el momento, el Instituto Nacional de Investigación en Salud Pública (INSPI) ha tomado 662.183 muestras para RT-PCR COVID-19 de las cuales 195.884 son casos confirmados con pruebas PCR.

Cabe indicar que puede existir más de una muestra por persona durante el proceso diagnóstico.

- 169.804 pacientes recuperados.
- 23.748 casos con alta hospitalaria.
- 9.129 personas fallecidas (confirmados COVID-19)
- 541 hospitalizados estables.
- 350 hospitalizados con pronóstico reservado.
- 426.070 casos fueron descartados.
- 877.537 Llamadas al 171 relacionadas a COVID-19
- 350.921 Seguimiento telefónico, llamadas a pacientes con diagnóstico confirmado.
- 117.654 Teleconsultas, ciudadanos atendidos a través de APP SALUDEC y por un médico del 171

<https://www.salud.gob.ec/actualizacion-de-casos-de-coronavirus-en-ecuador/>

Valor 4

```
In [ ] : from random import randrange # Obtener un numero randomico
import pygame

#Parametros de inicio
PROBA_MUERTE = 8.4 # Probabilidad de que la gente muera COVID
CONTAGION_RATE = 5 # 0.59 - 1.10 Factor R0 para la simulacion COV
ID probabilidad
PROBA_INFECT = CONTAGION_RATE * 10
PROBA_VACU = 0 # Probabilidad de que exista una vacuna, COVID = 0
SIMULACION_SPEED = 50 # Tiempo de un dia en milisegundos (Cada 25
es un dia)
nb_rows = 50 #Numero de filas
nb_cols = 50 #Numero de columnas

global display, myfont, states, states_temp #Declaracion de variab
les globales

#Declaro colores en formato RGB
WHITE = (255, 255, 255)
BLUE = (0, 0, 255)
GREEN = (0, 247, 0)
BLACK = (0, 0, 0)

#Obtiene los vecinos dado un punto x, y
def get_vecinos(x, y):
    incx = randrange(3)
    incy = randrange(3)
    incx = (incx * 1) - 1
    incy = (incy * 1) - 1
    x2 = x + incx
    y2 = y + incy
    #Validar limites
    if x2 < 0:
        x2 = 0
    if x2 >= nb_cols:
        x2 = nb_cols - 1
    if y2 < 0:
        y2 = 0
    if y2 >= nb_rows:
        y2 = nb_rows - 1
    return [x2, y2] # Nuevos contagiados

#Genero las personas que cuentan con inmunidad o vacuna
def vacunar():
    for x in range(nb_cols):
        for y in range(nb_rows):
            if randrange(99) < PROBA_VACU:
                states[x][y] = 1

#Funcion que permite contar el numero de muertosde la matriz state
s == -1
def contar_muertes():
    contador = 0
    for x in range(nb_cols):
        for y in range(nb_rows):
            if states[x][y] == -1:
                contador += 1
    return contador

#Definimos datos de inicio
states = [[0] * nb_cols for i in range(nb_rows)]
states_temp = states.copy()
states[randrange(50)][randrange(50)] = 10 # Estado inicial de la s
imulacion Posicion del Infectado
it = 0 # Variable para contar las Iteraciones
total_muerte = 0 # Contabiliza el numero de muertos
vacunar() #Llamar a la funcion vacunar

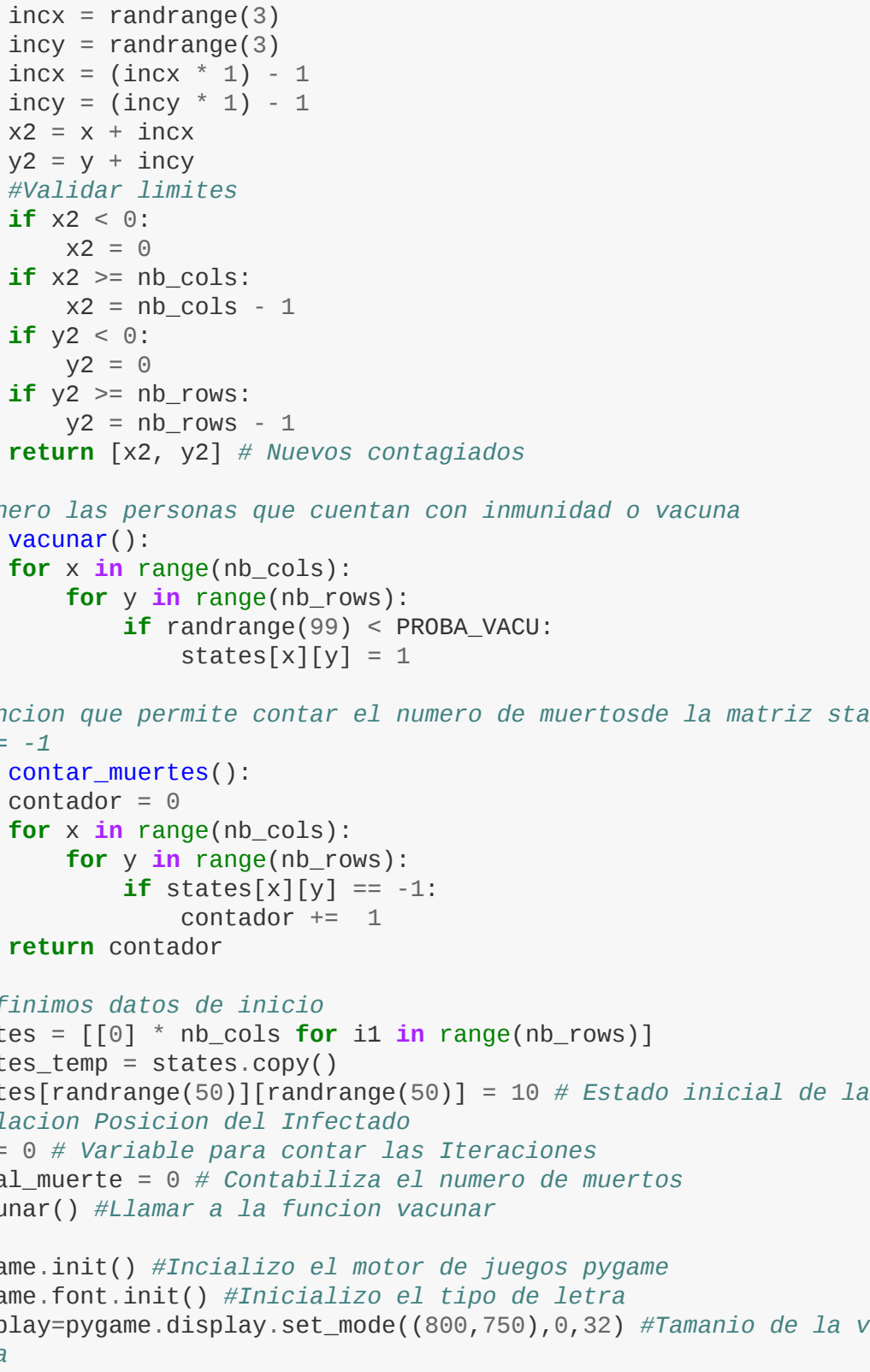
pygame.init() #Incializo el motor de juegos pygame
pygame.font.init() #Inicializo el tipo de letra
display=pygame.display.set_mode((800,750),0,32) #Tamaño de la ven
tana
pygame.display.set_caption("Simulacion de Epidemia Covid-19 Ecuado
r")# Titulo
pygame.font.SysFont('Calibri', 40) # Tipo de letra
display.fill(WHITE) # Color de fondo

while True:
    pygame.time.delay(SIMULACION_SPEED) # Sleep o pausa
    it = it + 1
    if it <= 10000 and it >= 2:
        states_temp = states.copy() #Copia de la matriz
        #Recorrera la matriz
        for x in range(nb_cols):
            for y in range(nb_rows):
                state = states[x][y]
                if state == -1:
                    pass
                if state >= 10: # Numero de dias de contagio
                    states_temp[x][y] = state + 1
                if state >= 20:
                    if randrange(99) < PROBA_MUERTE: # Genero un r
andomico para verificar si fallece o se recupera
                        states_temp[x][y] = -1 # Muere
                    else:
                        states_temp[x][y] = 1 # Cura o recupera
                if state >= 10 and state <= 20: # Rango de infecta
do
                    if randrange(99) < PROBA_INFECT: # Infecto a l
as personas cercanas entre 10 y 20
                        neighbour = get_vecinos(x, y) #Obtenemos l
os vecinos a contagiar
                        x2 = neighbour[0]
                        y2 = neighbour[1]
                        neigh_state = states[x2][y2]
                        if neigh_state == 0: #Verifico que este sa
no
                            states = states_temp.copy()
                            total_muerte = contar_muertes() # contar el numero de muer
tos
                            pygame.draw.rect(display, WHITE, (250, 30, 260, 50)) # Grafico
el fondo
                            textsurface = font.render("Total muertes: "+ str(total_muerte
), False, (255,160,122)) #El numero de muertos
                            display.blit(textsurface, (250, 30)) # Graficar el texto de mu
ertes
                            #Graficar el estado del paciente matriz
                            for x in range(nb_cols):
                                for y in range(nb_rows):
                                    if states[x][y] == 0:
                                        color = BLUE # No infectado
                                    if states[x][y] == 1:
                                        color = GREEN # Recupero
                                    if states[x][y] >= 10:
                                        color = (states[x][y] * 12, 50, 50) # Infectado -
Rojo
                                    if states[x][y] == -1:
                                        color = BLACK # Muerto
                                    pygame.draw.circle(display, color, (100 + x * 12 + 5,
100 + y * 12 + 5), 1)
                                    pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 10
0 + y * 12 + 4, 1, 1))
                                    #Escuchar los eventos del teclado
                                    for event in pygame.event.get():
                                        if event.type == pygame.KEYDOWN and event.key == pygame.K_
ESCAPE: #Presiona y Escape
                                            pygame.quit() #Termino simulacion
                                        if event.type == pygame.KEYDOWN and event.key == pygame.K_
SPACE: #Presiona y espacio
                                            #Reiniciamos valores
                                            states = [[0] * nb_cols for i in range(nb_rows)]
                                            states_temp = states.copy()
                                            states[5][5] = 10
                                            it = 0
                                            total_muerte = 0
                                            vacunar() #Llamar a la funcion vacunar

                                            pygame.display.update()# Mandar actualizar la ventana

pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.
html
```

Total muertes: 188



Valor 1.4

```
In [ ] : from random import randrange # Obtener un numero randomico
import pygame

#Parametros de inicio
PROBA_MUERTE = 7.1 # Probabilidad de que la gente muera COVID
CONTAGION_RATE = 1.4 # 0.59 - 1.10 Factor R0 para la simulacion C
OVID probabilidad
PROBA_INFECT = CONTAGION_RATE * 10
PROBA_VACU = 0 # Probabilidad de que exista una vacuna, COVID = 0
SIMULACION_SPEED = 50 # Tiempo de un dia en milisegundos (Cada 25
es un dia)
nb_rows = 50 #Numero de filas
nb_cols = 50 #Numero de columnas

global display, myfont, states, states_temp #Declaracion de variab
les globales

#Declaro colores en formato RGB
WHITE = (255, 255, 255)
BLUE = (0, 0, 255)
GREEN = (0, 247, 0)
BLACK = (0, 0, 0)

#Obtiene los vecinos dado un punto x, y
def get_vecinos(x, y):
    incx = randrange(3)
    incy = randrange(3)
    incx = (incx * 1) - 1
    incy = (incy * 1) - 1
    x2 = x + incx
    y2 = y + incy
    #Validar limites
    if x2 < 0:
        x2 = 0
    if x2 >= nb_cols:
        x2 = nb_cols - 1
    if y2 < 0:
        y2 = 0
    if y2 >= nb_rows:
        y2 = nb_rows - 1
    return [x2, y2] # Nuevos contagiados

#Genero las personas que cuentan con inmunidad o vacuna
def vacunar():
    for x in range(nb_cols):
        for y in range(nb_rows):
            if randrange(99) < PROBA_VACU:
                states[x][y] = 1

#Funcion que permite contar el numero de muertosde la matriz state
s == -1
def contar_muertes():
    contador = 0
    for x in range(nb_cols):
        for y in range(nb_rows):
            if states[x][y] == -1:
                contador += 1
    return contador

#Definimos datos de inicio
states = [[0] * nb_cols for i in range(nb_rows)]
states_temp = states.copy()
states[randrange(50)][randrange(50)] = 10 # Estado inicial de la s
imulacion Posicion del Infectado
it = 0 # Variable para contar las Iteraciones
total_muerte = 0 # Contabiliza el numero de muertos
vacunar() #Llamar a la funcion vacunar

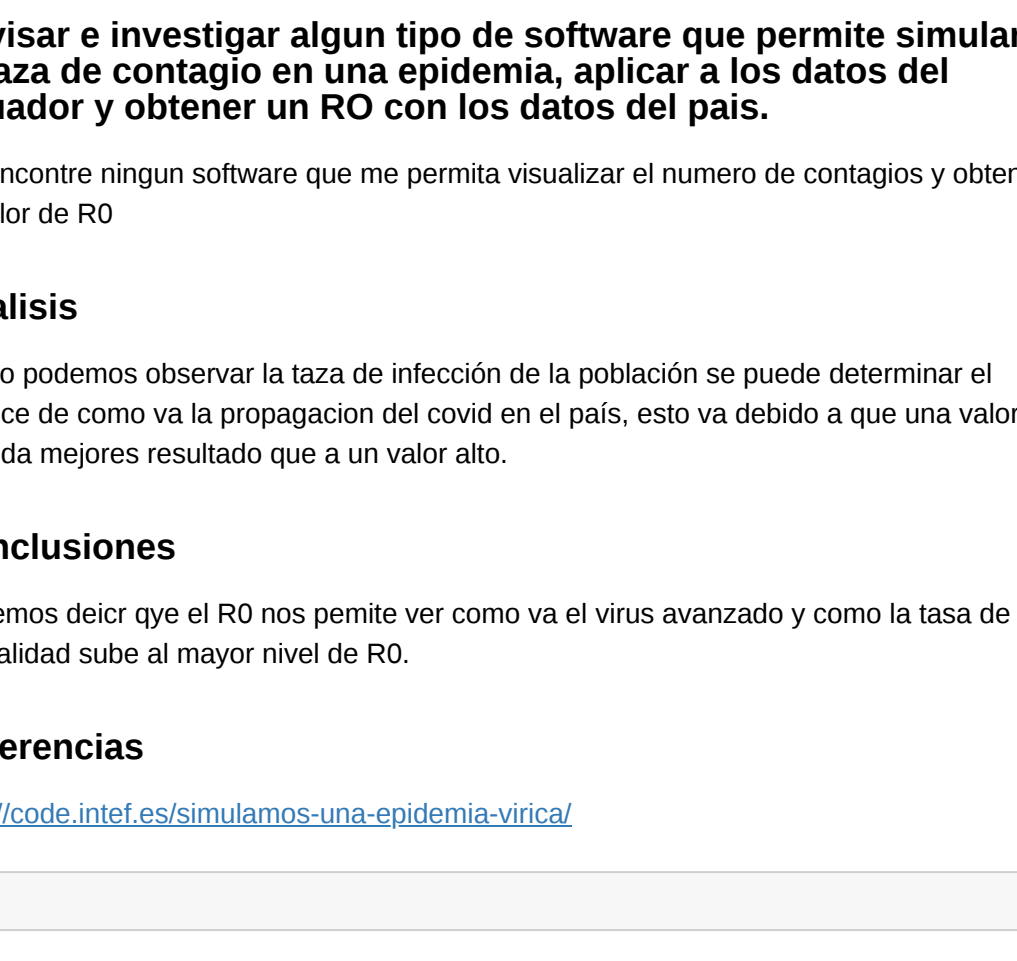
pygame.init() #Incializo el motor de juegos pygame
pygame.font.init() #Inicializo el tipo de letra
display=pygame.display.set_mode((800,750),0,32) #Tamaño de la ven
tana
pygame.display.set_caption("Simulacion de Epidemia Covid-19 Ecuado
r")# Titulo
pygame.font.SysFont('Calibri', 40) # Tipo de letra
display.fill(WHITE) # Color de fondo

while True:
    pygame.time.delay(SIMULACION_SPEED) # Sleep o pausa
    it = it + 1
    if it <= 10000 and it >= 2:
        states_temp = states.copy() #Copia de la matriz
        #Recorrera la matriz
        for x in range(nb_cols):
            for y in range(nb_rows):
                state = states[x][y]
                if state == -1:
                    pass
                if state >= 10: # Numero de dias de contagio
                    states_temp[x][y] = state + 1
                if state >= 20:
                    if randrange(99) < PROBA_MUERTE: # Genero un r
andomico para verificar si fallece o se recupera
                        states_temp[x][y] = -1 # Muere
                    else:
                        states_temp[x][y] = 1 # Cura o recupera
                if state >= 10 and state <= 20: # Rango de infecta
do
                    if randrange(99) < PROBA_INFECT: # Infecto a l
as personas cercanas entre 10 y 20
                        neighbour = get_vecinos(x, y) #Obtenemos l
os vecinos a contagiar
                        x2 = neighbour[0]
                        y2 = neighbour[1]
                        neigh_state = states[x2][y2]
                        if neigh_state == 0: #Verifico que este sa
no
                            states = states_temp.copy()
                            total_muerte = contar_muertes() # contar el numero de muer
tos
                            pygame.draw.rect(display, WHITE, (250, 30, 260, 50)) # Grafico
el fondo
                            textsurface = font.render("Total muertes: "+ str(total_muerte
), False, (255,160,122)) #El numero de muertos
                            display.blit(textsurface, (250, 30)) # Graficar el texto de mu
ertes
                            #Graficar el estado del paciente matriz
                            for x in range(nb_cols):
                                for y in range(nb_rows):
                                    if states[x][y] == 0:
                                        color = BLUE # No infectado
                                    if states[x][y] == 1:
                                        color = GREEN # Recupero
                                    if states[x][y] >= 10:
                                        color = (states[x][y] * 12, 50, 50) # Infectado -
Rojo
                                    if states[x][y] == -1:
                                        color = BLACK # Muerto
                                    pygame.draw.circle(display, color, (100 + x * 12 + 5,
100 + y * 12 + 5), 5)
                                    pygame.draw.rect(display, WHITE, (100 + x * 12 + 3, 10
0 + y * 12 + 4, 1, 1))
                                    #Escuchar los eventos del teclado
                                    for event in pygame.event.get():
                                        if event.type == pygame.KEYDOWN and event.key == pygame.K_
ESCAPE: #Presiona y Escape
                                            pygame.quit() #Termino simulacion
                                        if event.type == pygame.KEYDOWN and event.key == pygame.K_
SPACE: #Presiona y espacio
                                            #Reiniciamos valores
                                            states = [[0] * nb_cols for i in range(nb_rows)]
                                            states_temp = states.copy()
                                            states[5][5] = 10
                                            it = 0
                                            total_muerte = 0
                                            vacunar() #Llamar a la funcion vacunar

                                            pygame.display.update()# Mandar actualizar la ventana

pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.
html
```

Total muertes: 0



Revisar e investigar algun tipo de software que permite simular la tasa de contagio en una epidemia, aplicar a los datos del Ecuador y obtener un R0 con los datos del pais.

No encuentre ningun software que me permita visualizar el numero de contagios y obtener el valor de R0

Analisis

Como podemos observar la tasa de infección de la población se puede determinar el avance de como va la propagación del covid en el país, esto va debido a que una valor bajo da mejores resultado que a un valor alto.

Conclusiones

Podemos decir que el R0 nos permite ver como va el virus avanzado y como la tasa de mortalidad sube al mayor nivel de R0.

Referencias

<http://code.intel.es/inulamos-una-epidemia-virica/>

```
In [ ] :
```