```
# -*- coding: utf-8 -*-
"""
Created on Thu May  6 10:55:01 2021

@author: Ng Shao Yih
"""

##Rules##
#if white turn black move right
#if black turn white move left

#added red
#if white turn black move right
#if black turn red move left
#if red turn white move right

import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import matplotlib.colors as pltcolors

#grid size
N = 64

#making grid + ant
grid = np.zeros((N,N), np.uint)
antPos = np.matrix([[N//2], [N//2]]) #ant starting location
point = np.matrix('1;0') #initial direction ant is pointing
antColMap = pltcolors.ListedColormap(['w','k','r'])
antColNum = [0,1,2]

#turning directions
left = np.matrix('0 -1; 1 0')
right = np.matrix('0 1; -1 0')


#color
red = pltcolors.to_rgb('r')
white = pltcolors.to_rgb('w')
black = pltcolors.to_rgb('k')

#move ant function
def move(grid, antPos, point):
    #check if square is white
    antPos[:] = antPos + point

    '''
    if :
        grid[antPos[0,0], antPos[1,0]] = 1
        point
        point[:] = right*point
    else:
        grid[antPos[0,0], antPos[1,0]] = 0
        point[:] = left*point

    '''
    if grid[antPos[0,0], antPos[1,0]] == 0:
        grid[antPos[0,0], antPos[1,0]] = 1
        point
        point[:] = right*point
    else:
        grid[antPos[0,0], antPos[1,0]] = 0
        point[:] = left*point


#Plotting
fig = plt.figure()
```

```
img=plt.imshow(grid,interpolation='none', vmin = 0,vmax = 1, cmap = antColMap)

def animate(i):
    move(grid, antPos, point)
    img.set_data(grid)
    return [img]


anim = animation.FuncAnimation(fig, animate,
                               frames=24, interval=100, blit=True)
plt.show()
```