import pygame

# Define some colors

black = ( 0, 0, 0)

white = ( 255, 255, 255)

green = ( 0, 255, 0)

red = ( 255, 0, 0)

# This sets the width and height of each grid location

width=10

height=10

# This sets the margin between each cell

margin=2

# Create a 2 dimensional array. A two dimesional

# array in our implementation is simply a list of lists.

grid=[]

for row in range(30):

# Add an empty array that will hold each cell

# in this row

grid.append([])

for column in range(30):

grid[row].append(0) # Append a cell

# Set row 0, cell 0 to one. (Remember rows and

# column numbers start at zero.)

grid[0][0] = 1

# Initialize pygame

pygame.init()

# Set the height and width of the screen

size=[362,362]

screen=pygame.display.set\_mode(size)

# Set title of screen

pygame.display.set\_caption("My Game")

# Loop until the user clicks the close button.

done=False

# Used to manage how fast the screen updates

clock=pygame.time.Clock()

# Create a list of live cells, initially empty

alive=[]

# Change to play mode when user clicks start position

started=False

######################################

# This is code section which you need to implement

def nextgen(gen):

return gen

######################################

# -------- Main Program Loop -----------

while done==False:

if not(started):

for event in pygame.event.get(): # User did something

if event.type == pygame.QUIT: # If user clicked close

done=True # Flag that we are done so we exit this loop

if event.type == pygame.MOUSEBUTTONDOWN:

# User clicks the mouse. Get the position

pos = pygame.mouse.get\_pos()

# Change the x/y screen coordinates to grid coordinates

column=pos[0] // (width+margin)

row=pos[1] // (height+margin)

# Set that location to one

grid[row][column]=1

# If user clicks start position

if row==0 and column==0:

started=True

grid[row][column]=0

# Set up live cell list

for row in range(30):

for column in range(30):

if grid[row][column] == 1:

alive.append((row,column))

if started:

for event in pygame.event.get(): # User did something

if event.type == pygame.QUIT: # If user clicked close

done=True # Flag that we are done so we exit this loop

if event.type == pygame.MOUSEBUTTONDOWN:

# User clicks the mouse. Get the position

pos = pygame.mouse.get\_pos()

# Change the x/y screen coordinates to grid coordinates

column=pos[0] // (width+margin)

row=pos[1] // (height+margin)

# If user clicks stop position

if row==0 and column==0:

started=False

alive=[]

# Clear the grid

for row in range(30):

for column in range(30):

grid[row][column]=0

# Set live cells

for (row,column) in alive:

grid[row][column]=1

# Set up next generation

alive=nextgen(alive)

# Set the screen background

screen.fill(black)

# Draw the grid

grid[0][0]=1

for row in range(30):

for column in range(30):

color = white

if grid[row][column] == 1:

if started:

color = green

else:

color = red

pygame.draw.rect(screen,color,[(margin+width)\*column+margin,(margin+height)\*row+margin,width,height])

# Limit to 20 frames per second

clock.tick(10)

# Go ahead and update the screen with what we've drawn.

pygame.display.flip()

# If you forget this line, the program will 'hang' on exit.

pygame.quit ()