Import pygame, random, sys

Global Variables:

Window settings (width, height, etc.)

def terminate():

pygame.quit() –quit game

def drawText (text, font, surface, x, y):

create text object with given properties

show text on screen

def Tutorial:

--start screen

show background

drawText (description)

show each component of Internet of Things

(sensor, signal, machine, cloud-based app.)

drawText (explanation of each)

drawText (how to play)

def waitForPlayerToPressKey():

while loop:

check if user pressed ‘Esc’ or equivalent

terminate()

return True

return False

def signalRecieved(playerRect, receiver):

for loop going through the list of receiver sensors:

if touches:

remove the receiver sensor

return True

return False

def signalBlocked(playerRect, obstacles):

for loop check if signal touches obstacles:

return True

return False

def main ():

pygame.init()

mainClock = pygame.time.Clock()

initialize ‘windowSurface’ with global variables

import sounds

load ‘background’ image

load ‘Sensor1’ image

load ‘Sensor2’ image

load ‘Machines’ image

load ‘Signal1’ image

get\_rect –to set up the image as a manipulatable object

update display

*waitForPlayerToPressKey*()

topScore = 0

while True:

set up variables:

--list of ‘obstacles’

--list of ‘end sensors’

--current score = 0

Set the initial position of ‘signal’

Set Boolean var. for direction movements *False*

(moveRight, moveUp, moveDown, moveLeft)

Play music

While loop: (actual game loop)

Each time print new images to the window at a constant frequency

(animation)

Score+=1

For loop:

--go through a list of events in pygame.event

If quit:

*Terminate*()

If KEYDOWN:

Check which key has been pressed;

If direction keys are pressed:

Correspondent Boolean var. set true

If KEYUP:

Check if direction keys are released:

Correspondent Boolean var. set false again

Adding ‘receiver sensor’ objects to list:

Same size, setting

At a slower rate

Moving at constant speed

Adding obstacles objects to list:

Random size, get\_rect

At a constant rate

Moving obstacles:

For loop go through list:

.move\_ip()

Delet obstacle objects as they fall past the bottom (for loop)

Moving signal (player control):

According to the detected event (moveRight, moveLeft, etc.):

Move ‘signal’ at a constant rate initialized as global var.

Paint windows background

Paint textObjects (topscore, score)

Paint ‘signal’

For loops pain each obstacle in list

For loops paint each sensor in list

Update windows display

If *signalBlocked*():

Check topscore: replace with a heigher score

Break

MainClock.tick(FPS)

Stop music

DrawText (game over, instructions)

Update display

waitForPlayerToPressKey()