Lab 06 –Python / SenseHat (3)

A) SenseHat functionality - The "PixelEater"

This time you are required to write a program that uses the joystick on the SenseHat.

https://pythonhosted.org/sense-hat/api/#joystick

The actual program that you shall write will feature a **pixel eater** pixel that moves around the SenseHat matrix. As it moves, the **pixel eater** will consume the pixels on the matrix.

To get an initial idea of this functionality, you are free to try out the **PixelEater.cpython-35.pyc** program[1].

Functions to use (recommended):

- `clear(), set_pixel(), get_events()`[2]

Functionality:

- Make an 8x8 list of lists with the name `visited`. Initialize it to `False` in all positions[3].
- Start with an all white matrix. Use the `clear()` function.
- Start with the **eater** in position (x,y) . The **eater** shall be bright yellow (255,255,0)
- Mark the eater position as visited[4].
- Make an event loop
    - Save input from `get_events()` in variables (you may call them *action* and *direction*)
    - If the action is **pressed** or **held**:
        - Paint the **pixel eater** position black (0,0,0). Go one step in the indicated direction. This is done by updating the (x,y) position of the **pixel eater**. Paint the new **pixel eater** position bright yellow. Update the visited list.
        - Use the modulo operator to allow loop-around for the position updating
            - `n = (n + 1) % 8`
            - `n = (n – 1) % 8`
    - Ignore the "**released**" action
    - You may also ignore the "**middle**" direction
    - Check if all positions are visited. If so, the game is over and you can terminate the program.

---

[1] Python programs can be compiled to bytecode. A "pyc" file is a bytecode file. The bytecode is specific to the python version, and this is why the "cpython-35" is part of the filename. You can compile your own python programs too. To compile all *.py files in a folder: `$ python3 -m compileall .`

[2] See code example in the API description.

[3] This will do the trick: «`visited = [[False for i in range(0,8)] for j in range(0,8)]`».

[4] «`visited[x][y] = True`».