

Computing

Lesson 8: Sense HAT II

Programming Part 5: Strings and Lists

Rebecca Franks

¹ Materials from the Teach Computing Curriculum created by the National Centre for Computing Education



Random pixel colour grid



Worked example 1

Code from the starter activity

```
1 from sense_hat import SenseHat
2 from random import choice
3
4 sense = SenseHat()
5
6 red = (255, 0, 0)
7 yellow = (255, 255, 0)
8 blue = (0, 0, 255)
9
10 colours = [red, yellow, blue]
11
12 colour = choice(colours)
13
14 sense.clear(colour)
```



Worked example 2

Generates a list that contains each letter of the alphabet

```
1 alphabet = []  
2 for x in range(65,91):  
3     alphabet.append(chr(x))  
4  
5 print(alphabet)
```



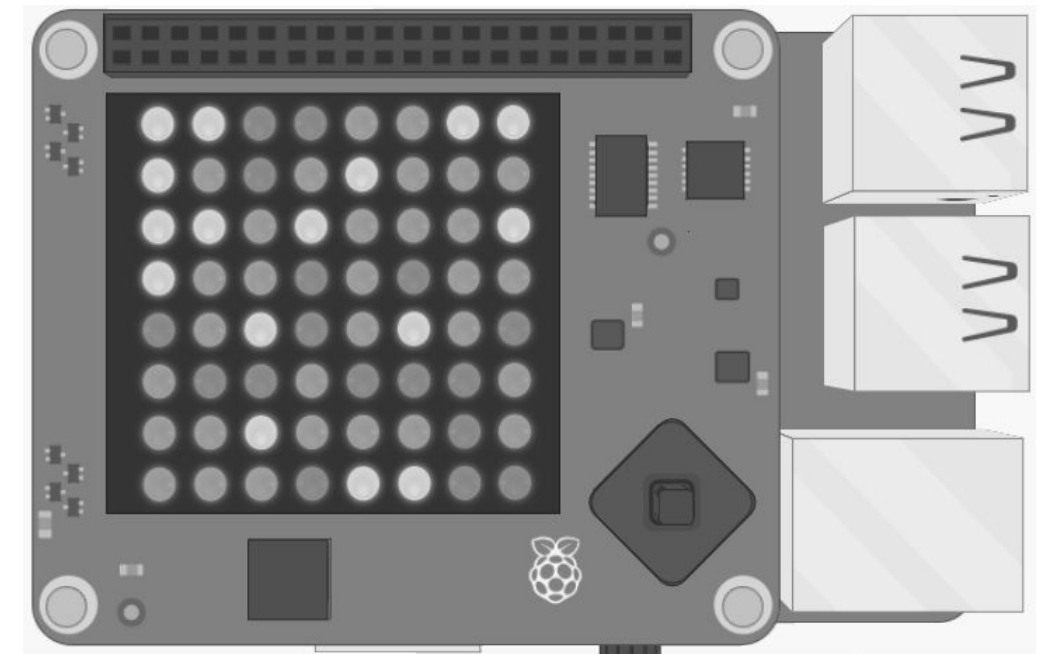
Task: Random colour pixel grid

For the **emulator** use Trinket **oaknat.uk/comp-ks4-senseHAT-emulator**

For this task you will be creating a program that creates a grid of randomly generated colours. An example output for your program would look like the image below:

Your program should:

- Create variables for each colour that you wish to use
- Create a list that will hold the variables
- Randomly pick a colour from the list
- Add the randomly picked colour to a new list
- Add enough random colours to complete the grid
- Light the LED matrix using the newly generated grid



TIP: Use the worked examples on page 1 to help you with this task



Explorer Task (Optional)

This code snippet makes use of the joystick on the Sense HAT.

```
from sense_hat import SenseHat
sense = SenseHat()

red = (255, 0, 0)
while True:
    for event in sense.stick.get_events():
        if event.action == "pressed":
            if event.direction == "middle":
                sense.clear(red)
```

Incorporate this into your program so that it generates a random pixel colour grid each time the user presses the joystick down.

In Trinket, run your code then click on the Sense HAT to test your program. Press the enter key to simulate pressing the joystick down.



Random pixel alien



Alien code

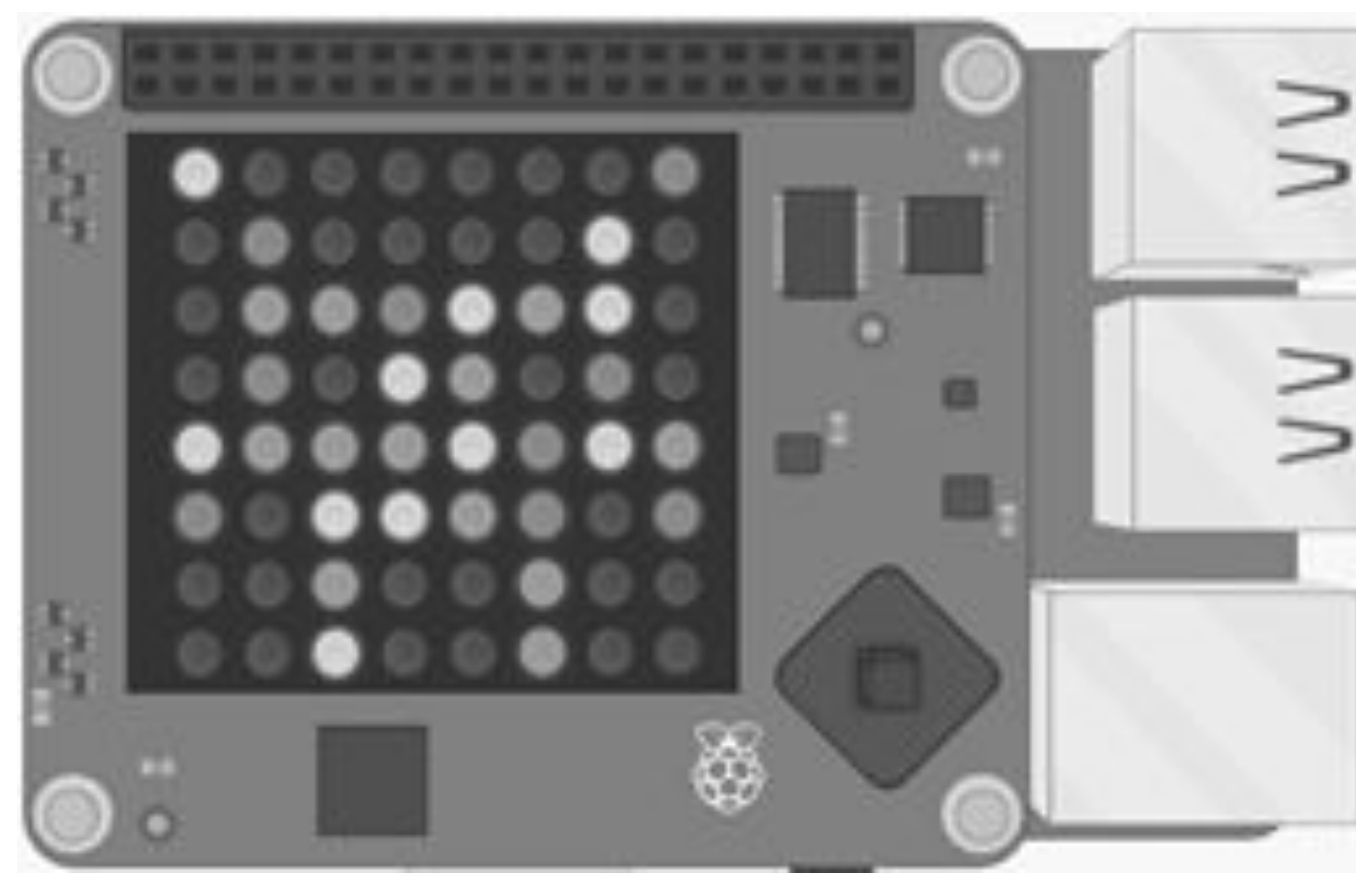
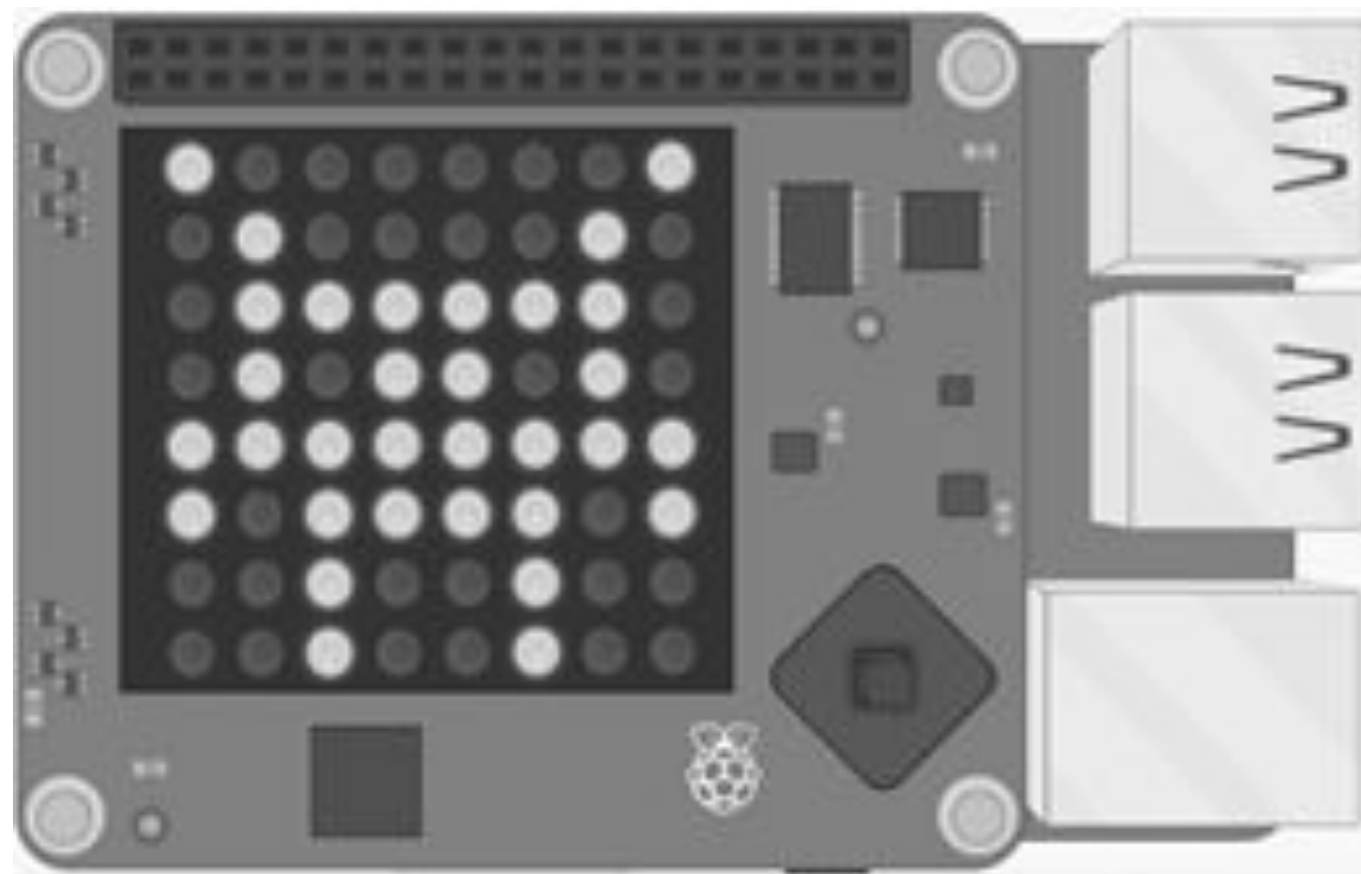
This code will create a list that contains the colours to draw an alien on the LED matrix

```
1 b = (0, 0, 0) # background
2 c = (255, 255, 0) # character
3
4 alien = [
5     c,  b, b, b, b, b, b, c,
6     b,  c, b, b, b, b, c, b,
7     b,  c, c, c, c, c, c, b,
8     b,  c, b, c, c, b, c, b,
9     c,  c, c, c, c, c, c, c,
10    c,  b, c, c, c, c, b, c,
11    b,  b, c, b, b, c, b, b,
12    b,  b, c, b, b, c, b, b,
13    ]
14
15 sense.set_pixels(alien)
```



Task: Random pixel alien

For this activity you will be creating a program that displays an alien with randomly changing pixels.



Task: Random pixel alien

Your program should:

- Create variables for each colour that you wish to use
- Create a list of the variables for each colour
- Create the initial alien
- Display the initial, single colour alien for 2 seconds
- Create a new list based on the original alien
 - If the item in the initial list is a c then it should generate a random colour item
 - If the item in the initial list is a b then it should use the original item
- Light the LED matrix using the newly generated grid.
- Continuously create a new list and display it until the program is stopped

TIP: Use the alien code on the first page and your code from task 1 to support you with this.



Explorer Task (optional)

This code snippet makes use of the joystick on the Sense HAT.

```
from sense_hat import SenseHat
sense = SenseHat()

red = (255, 0, 0)

while True:
    for event in sense.stick.get_events():
        if event.action == "pressed":
            if event.direction == "up":
                sense.clear(red)
```



Explorer Task (optional)

Incorporate this into your program so that it generates different aliens based on the position of the joystick:

up	Yellow alien
down	Red alien
left	Blue alien
right	Green alien
middle	Random colour pixel alien

Tip: In Trinket, run your code then click on the Sense HAT to test your program. Press the enter key to simulate pressing the joystick down. Press the arrow keys for the other directions.



Magic 8 ball



Code Snippet 1

This code performs an action based on movements from the joystick

```
1 from sense_hat import SenseHat
2 sense = SenseHat()
3
4 red = (255, 0, 0)
5
6 while True:
7     for event in sense.stick.get_events():
8         if event.action == "pressed":
9             if event.direction == "middle":
10                 sense.clear(red)
```

In Trinket, run your code then click on the Sense HAT to test your program. Press the enter key to simulate pressing the joystick down. Press the arrow keys for the other directions.



Code Snippet 2

This code displays a message that scrolls across the screen

```
1 from sense_hat import SenseHat
2 sense = SenseHat()
3
4 sense.show_message("Here is a message")
```



The Magic 8 Ball

A magic 8 ball is a toy that you ask a closed question to and it will give you a prediction back. Your magic 8 ball will have 8 different responses to a question. These are:

- Signs point to yes
- Without a doubt
- You may rely on it
- Do not count on it
- Looking good
- Cannot predict now
- It is decidedly so
- Outlook not so good



The user should be able to ask a question in their heads and press the middle button on the joystick. The middle button should reveal one of the responses from above.



Explorer Task (Optional)

Here are some ways to improve your magic 8 ball program.

- Make the user feel like they have more control over their predictions by giving them the choice to move the joystick up, down, left or right before revealing the response.
- Create a graphic for the magic 8 ball that appears at the beginning of the program and returns when the prediction has been made.

