

Deliverable Session 04 - Information and Communication Technology

Rafael Antonio Echevarria Silva

En esta sesión hemos observado como tenemos que tratar los datos para representarlos en la matriz led, empezando con ver como establecer pixeles individuales y luego con una matriz 8x8 a traves de la función **sense.set_pixels()**.

Ex.1: Writing pixels

El enunciado nos pedía que dibujáramos el laberinto que se veía en la presentación, por lo que aprovechando el ejemplo anterior procedimos a utilizar una matriz 8x8 con las componentes de color definidas como g = (0, 0, 255) y bg = (0, 0, 0).

Y para la segunda parte establecimos el cursor en con la componente r = (255, 0, 0) y la posición de la matriz la (1, 0).

Ex.2: Displaying a simple character

Para este ejercicio nos enseñan que la función **sense.show_letter()**, respondiendo a la pregunta que se plantea en la explicación si se puede personalizar el color de la letra y del texto siempre que se defina dentro de la función.

```
sense.show_letter("z", text_colour=[255, 0, 0], back_colour=[0, 0, 255])
```

Para el ejercicio 2, nos pregunta si podemos enseñar un mensaje utilizando la misma función que nos han enseñado, y así es utilizando un bucle **for letter in range message** y añadiendo una pequeña pausa para una mejor comprensión y lectura del mensaje.

Ex.3: Selection of a random colour

Para poder representar un color random por el display, hemos tenido que importar la librería **random** para poder randomizar números enteros entre 0 y 255 para las componentes RGB. Con esto definimos una función **random_colour()** que será llamada desde una variable y se pinta en toda la matriz 8x8.

Ex.4: Use of the joystick

Con la función **stick.get_events()** detectamos las entradas de la dirección y acción del joystick. Para el ejercicio utilizamos dos bucles un **While True:** para mantener en ejecución el programa constantemente (mientras no se defina un break() o exit()), y definimos un bucle **for event in sense.stick.get_events():** para leer constantemente las acciones que se hacen en el joystick y definir que movimiento es mediante if/elif/else e imprimirlos por pantalla.

```
if event.direction == 'up': print(sense.show_letter("U"))
elif event.direction == 'down': print(sense.show_letter("D"))
elif event.direction == 'left': print(sense.show_letter("L"))
elif event.direction == 'right': print(sense.show_letter("R"))
else : print(sense.show_letter("M"))
```

Ex.5: Use of the joystick

En este ultimo apartado aplicamos todas las funciones previamente aprendidas, una vez tenemos definido el mapa, y la posición inicial del player, convertimos la matriz de 8x8, en una posición 1D dentro de una lista con la función **def maze_at(x, y):**. Para luego dibujar el laberinto con la posición del jugador, creando una copia del laberinto para no modificar el original **display = maze [:]**, luego pintamos la posición del jugador sobre el laberinto **display[player_y * 8 + player_x]** y actualizamos la matriz led. Y de igual manera que en el ejercicio anterior hacemos un doble bucle para leer y actualizar los movimientos del joystick, y con una condición **if** diferenciamos que los movimientos solo se puedan realizar en el espacio del laberinto pintado con la componente b(0, 0, 0).