Deliverable Session 04 - Information and Communication Technology

Rafael Antonio Echevarria Silva

In this session, we observed how we need to handle data to represent it on the LED matrix, starting with setting individual pixels and then moving on to an 8x8 matrix using the **sense.set_pixels()** function..

Ex.1: Writing pixels

The instructions asked us to draw the maze shown in the presentation, so taking advantage of the previous example, we proceeded to use an 8x8 matrix with the color components defined as *g = (0, 0, 255)* and *bg = (0, 0, 0)*.

And for the second part, we set the cursor using the component *r = (255, 0, 0)* and the position on the matrix at *(1, 0)*.

Ex.2: Displaying a simple character

For this exercise, we learned that the **sense.show_letter()** function allows us to customize the color of the letter and the background as long as it is defined within the function.

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

In Exercise 2, we are asked whether we can display a message using the same function that was demonstrated, and indeed we can by using a **for letter in message** loop and adding a small pause for better comprehension and readability of the message.

Ex.3: Selection of a random colour

To display a random color on the display, we had to import the **random** library in order to randomize integers between 0 and 255 for the RGB components. With this, we defined a **random_colour()** function that is called from a variable and displayed across the entire 8x8 matrix.

Ex.4: Use of the joystick

With the **stick.get_events()** function, we detect the direction and action inputs from the joystick. For the exercise, we used two loops: a **while True:** to keep the program running continuously (until a break() or exit() is defined), and a **for event in sense.stick.get_events():** loop to constantly read the actions performed on the joystick and determine the movement using if/elif/else statements, then print them on the screen.

```
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

In this final section, we apply all the previously learned functions. Once we have the map defined and the player's initial position, we convert the 8x8 matrix into a 1D position within a list using the function **def maze_at(x, y):**. Then, we draw the maze with the player's position, creating a copy of the maze to avoid modifying the original **display = maze[:]**, then we mark the player's position on the maze with **display[player_y * 8 + player_x]** and update the LED matrix. Just like in the previous exercise, we use a double loop to read and update the joystick movements, and with an **if** condition, we ensure that movements can only occur in the maze spaces marked with the component *b(0, 0, 0)*.