Intro to NeoMatrix

Week 1 Exercises

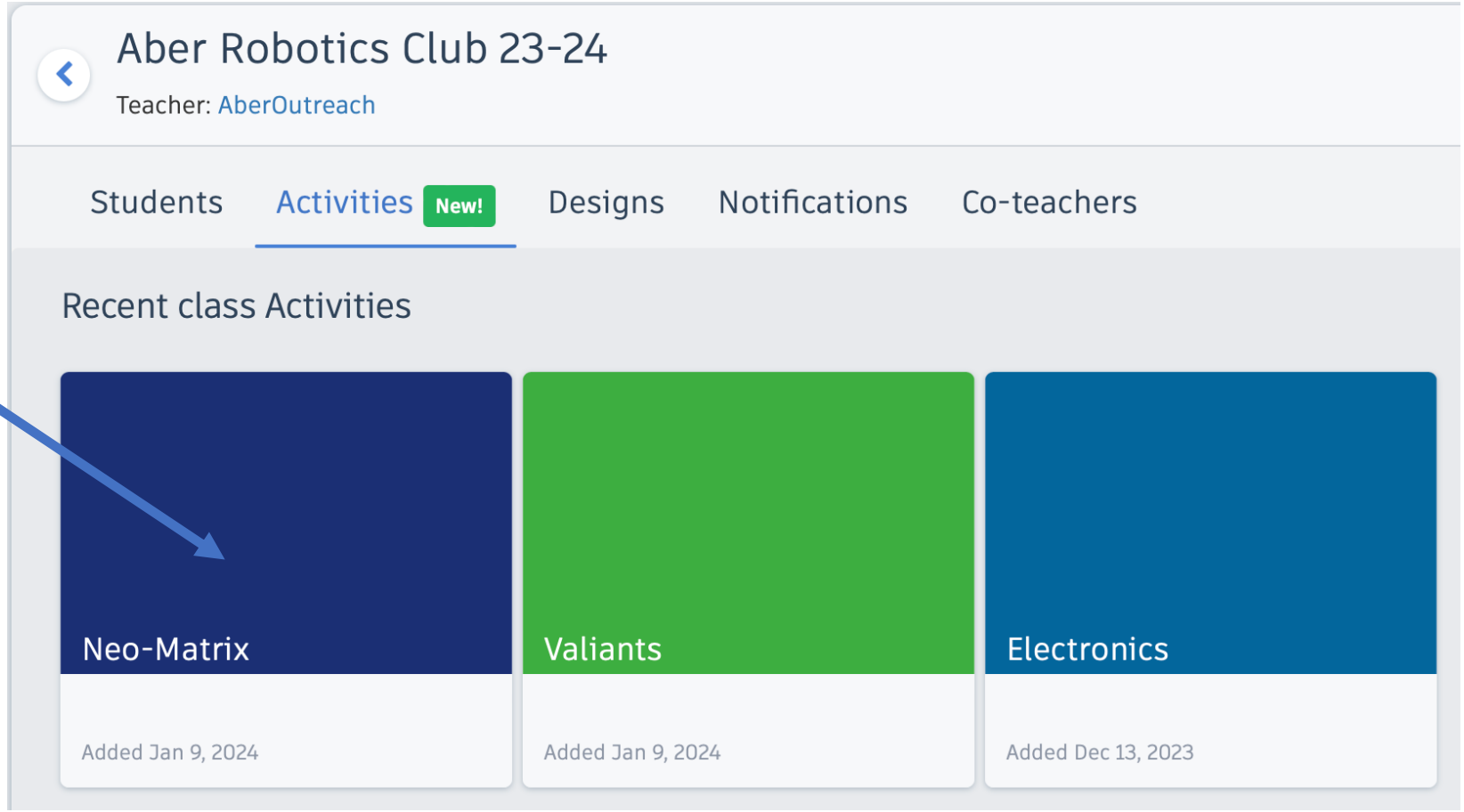
Building and Programming robots in TINKERCAD

This week you be introduced to the NeoMatrix through TinkerCAD. This workshop aims to develop your text programming skills.

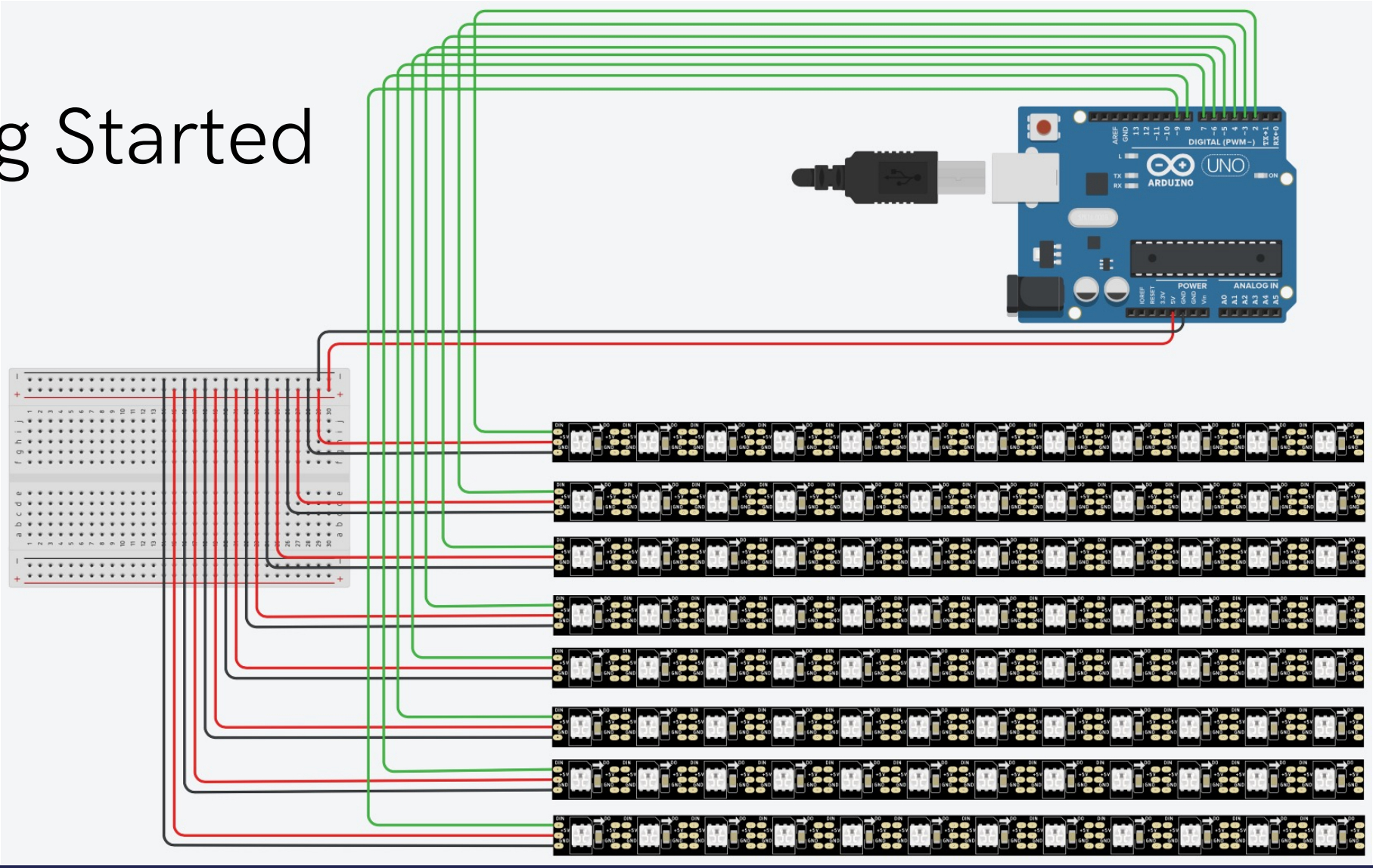
Please complete the bronze requirements before starting the silver, and complete the silver before starting the gold.

Getting started

Activity containing pre-made circuit



Getting Started



Neopixel and GFX APIs

- NeoPixel: https://adafruit.github.io/Adafruit_NeoPixel/html/class_adafruit__neo_pixel.html
- GFX API: http://adafruit.github.io/Adafruit-GFX-Library/html/class_adafruit___g_f_x.html#a59178a0e0c845a14a39b457c43567dd9

Key functions

- `matrix.fillScreen(matrix.Color(r,g,b));`
- `matrix.drawPixel(x,y,matrix.Color(r,g,b));`
- `matrix.fillRect(x, y, width, height, color);`
- `matrix.drawRect(x, y, width, height, color);`
- `matrix.fillCircle(center_x, center_y, radius, color);`
- `matrix.drawCircle(center_x, center_y, radius, color);`
- `matrix.clear();`
- `matrix.show();`

BRONZE Challenge:

Steps:

1. Fill in the function to generate a random colour
   a. The arduino function to generate a random number is:

```
random(lowerLimit, upperLimit);
```

   b. The matrix function to encode a colour is:

```
matrix.Color(r, g, b);
```

2. Use the random colour function to fill the whole screen with a different colour each time through the loop

SILVER Challenge:

Steps:

1. Write a function that will colour each pixel with a different random colour. This function itself shouldn't include the matrix.show() this will be done after the function is called

   Remember to use a nested for-loop to cycle through all the pixels in the matrix

   ```
   drawPixel (int16_t x, int16_t y, uint16_t color)
   ```

   Where the colour is generated by a call to your random colour function

2. Call your new function from your loop and call the show function to update the display

SILVER Challenge:

Steps:

1. Write a function that will colour all the pixels in a column on the grid the same colour. The full matrix should then be filled with a different colour on each column.
2. Write a function that will colour all the pixels in a row on the grid the same colour. The full matrix should then be filled with a different colour on each row.
3. Call your functions from inside your main loop

GOLD Challenge:

Steps:

1. Using either drawCircle or fillCircle, draw a selection of circles in random positions and sizes across the matrix. Each circle should be drawn with a different random colour. Add a default parameter to your function to specify how many circles to draw, and a second parameter to specify the maximum circle radius.
   Remember the function should not directly update the display

```
drawCircle (int16_t x0, int16_t y0, int16_t r, uint16_t color)
fillCircle (int16_t x0, int16_t y0, int16_t r, uint16_t color)
```

GOLD Challenge:

Steps:

2. Using either drawRect or fillRect, write a function that will draw a set of rectangles inside each other on the matrix. Each rectangle should use a random colour.
   Remember to use width and height so your function adapts to different matrix sizes.
   The function itself should not directly update the display.

```
drawRect (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)
fillRect (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)
```

Extension Challenge:

What else can you add to this program?

- Add a parameter with a default value to your functions that draw circles to switch between drawing circle outlines or filling in the circles
- Write a function that draws a circle where the outline is a different colour to the fill
- Use combinations and/or variations of the functions generated here to produce different effects. Think about whether to clear the screen or not between different function calls for different effects