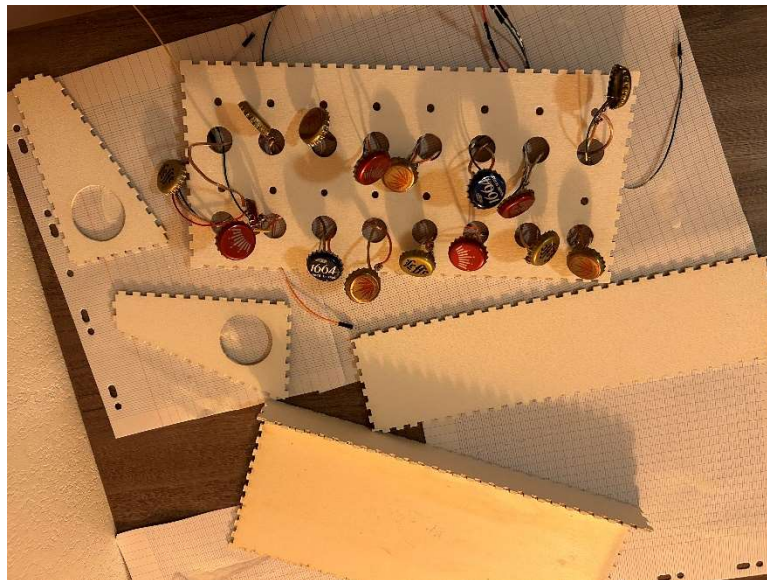


Rapport de séance :

Séance du 08/02 :

Pendant les vacances, je me suis occupée de peindre en blanc notre structure en bois.



En début de séance, comme la peinture avait été réalisée, j'ai pu coller au pistolet à colle les capsules au support.



Le reste de la séance a été consacré à commencer à construire le mode guidé.

Nous souhaitons pour ce mode qu'une led au hasard parmi les seize s'allume afin d'indiquer au joueur sur quelle capsule appuyer. Tant que le joueur n'appuie pas sur la bonne capsule la led reste allumée. Une fois que le joueur a appuyé, la led s'éteint et une autre s'allume et ainsi de suite jusqu'à ce qu'il ait fini de jouer le morceau qu'il a choisi.

Je me suis donc occupée de la partie code.

```

1  #include <CapacitiveSensor.h>
2  #include <SoftwareSerial.h>
3  #include <Adafruit_NeoPixel.h>
4
5  #define COMMON_PIN 53
6  #define NB_PIN 16
7
8  #define CMD_PLAY_NEXT 0x01
9  #define CMD_PLAY_PREV 0x02
10 #define CMD_PLAY_W_INDEX 0x03
11 #define CMD_SET_VOLUME 0x06
12 #define CMD_SEL_DEV 0x09
13 #define CMD_PLAY_W_VOL 0x22
14 #define CMD_PLAY 0x0D
15 #define CMD_PAUSE 0x0E
16 #define CMD_SINGLE_CYCLE 0x19
17 #define CMD_PLAY_WITH_FOLDER 0x0F
18 #define DEV_TF 0x02
19 #define SINGLE_CYCLE_ON 0x00
20 #define SINGLE_CYCLE_OFF 0x01
21
22 #define ARDUINO_RX 7 // Arduino Pin connected to the TX of the Serial MP3 Player module
23 #define ARDUINO_TX 6 // Arduino Pin connected to the RX of the Serial MP3 Player module
24 #define CS CapacitiveSensor
25
26 SoftwareSerial mp3(ARDUINO_RX, ARDUINO_TX);
27 CapacitiveSensor capList[16] = {CS(53, 51), CS(53, 50), CS(53, 49), CS(53, 48),
28 CS(53, 47), CS(53, 46), CS(53, 45), CS(53, 44), CS(53, 43), CS(53, 42),
29 CS(53, 41), CS(53, 40), CS(53, 39), CS(53, 38), CS(53, 37), CS(53, 36)};
30
31 int capacitance[16];
32 int currentKey = -1;
33 bool isHolding = false;
34
35 // Led part
36
37 #define NUMPIXELS 44
38 #define PIN 31
39 Adafruit_NeoPixel strip(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
40
41 int b = random(NB_PIN/2);
42
43 void setup()
44 {
45     Serial.begin(9600);
46     mp3.begin(9600);
47     delay(500);
48
49     mp3_command(CMD_SEL_DEV, DEV_TF); // select the TF card
50     delay(200);
51
52     mp3_command(CMD_SET_VOLUME, 100);
53
54     //mp3_command(CMD_PLAY_WITH_FOLDER, 0x0F00102); // Play mp3
55     // mp3_command(CMD_PAUSE, 0x0000); // Pause mp3
56     //mp3_command(CMD_PLAY_NEXT, 0x0000); // Play next mp3
57     //mp3_command(CMD_PLAY_PREV, 0x0000); // Play previous mp3
58
59     strip.begin();
60     turnOffLed();
61     strip.setPixelColor(b, 0, 0, 255);
62     strip.show();
63 }
64
65 void loop()
66 {
67     for(int i = 0; i < NB_PIN; i++) {
68         bool a = false;
69         capacitance[i] = capList[i].capacitiveSensor(30);
70         // Serial.print(capacitance[i]);
71         // Serial.print(" : ");
72
73         if(capacitance[i] > 250) {
74             if(!isHolding) {
75                 if(i != b) {
76                     for(int j = 0; j < 3; j++) {
77                         // turnRedLed();
78                         // delay(100);
79                         turnOffLed();
80                         delay(100);
81                     }
82                     strip.setPixelColor(b*3, 0, 0, 255);
83                     strip.show();
84                     Serial.print(b);
85                     Serial.print(" / ");
86                     Serial.println(capacitance[i]);
87                     break;
88                 }
89                 mp3_command(CMD_PLAY_WITH_FOLDER, 0x0F00100 + (i+1));
90                 turnOffLed();
91                 b = random(NB_PIN / 2);
92                 strip.setPixelColor(i*3, 0, 0, 255);
93                 strip.show();
94                 break;
95             }
96             a = true;
97         }
98     }
99     isHolding = a;
100     // long start = millis();
101     // long total2 = second.capacitiveSensor(30);
102
103     // Serial.print(millis() - start); // check on performance in milliseconds
104     // Serial.print("\t"); // tab character for debug window spacing
105
106     | | | | | // print sensor output 1
107     // Serial.print("\t");
108     // Serial.println(total2);
109
110     // if(total1 > 500 and total2 > 500) {
111     //     selected_sound = 0;
112     //     mp3_command(CMD_PAUSE, 0x0000);
113     // } else if(total1 > 500) {
114     //     if(selected_sound != 1) {
115     //         selected_sound = 1;
116     //         mp3_command(CMD_PLAY_WITH_FOLDER, 0x0F00101);
117     //     }
118     // } else if(total2 > 500) {
119     //     if(selected_sound != 2) {
120     //         selected_sound = 2;
121     //         mp3_command(CMD_PLAY_WITH_FOLDER, 0x0F00102);
122     //     }
123     // }
124     // delay(10); // arbitrary delay to limit data to serial port
125 }
126
127 void mp3_command(int8_t command, int16_t dat) {
128     int8_t frame[8] = { 0 };
129     frame[0] = 0x7e; // starting byte
130 }

```

```

128 void mp3_command(int8_t command, int16_t dat) {
129     int8_t frame[8] = { 0 };
130     frame[0] = 0x7e;           // starting byte
131     frame[1] = 0xff;           // version
132     frame[2] = 0x06;           // the number of bytes of the command without starting byte and ending byte
133     frame[3] = command;        //
134     frame[4] = 0x00;           // 0x00 = no feedback, 0x01 = feedback
135     frame[5] = (int8_t)(dat >> 8); // data high byte
136     frame[6] = (int8_t)(dat);    // data low byte
137     frame[7] = 0xef;           // ending byte
138     for (uint8_t i = 0; i < 8; i++) {
139         mp3.write(frame[i]);
140     }
141 }
142
143 void turnOffLed() {
144     for(int i = 0; i < NUMPIXELS; i++) {
145         strip.setPixelColor(i, 0, 0, 0);
146     }
147     strip.show();
148 }
149
150 void turnRedLed() {
151     for(int i = 0; i < NUMPIXELS; i++) {
152         strip.setPixelColor(i, 255, 0, 0);
153     }
154     strip.show();
155 }

```

Pour le code je me suis inspirée du code que nous avait donné Mr Masson et je l'ai combiné avec le code réalisé par Amy avant les vacances.

J'ai ensuite mis les conditions afin que les led s'allument quand on le souhaite.

Le nombre de pin est divisé par deux car pour l'instant nous avons testé qu'avec la moitié des capsules et que la bande led du haut car nous avons peur que l'arduino ne puisse pas délivrer assez pour les haut-parleur ainsi que pour les leds.

De plus, lorsque nous appuyons sur la bonne capsule pour le moment c'est une note de piano qui sort. On devra donc à la prochaine séance remplacer les notes de piano par les morceaux de musique choisie.

A la fin de la séance nous avons donc regroupé nos travaux avec Amy.

La vidéo de démonstration est disponible dans le dossier « RAPPORT DE SEANCE » → « CARPENTIER ».