

Compte-rendu du Projet de SIN – 1ère STI2D

Dispositif d'assistance aux personnes malvoyantes dans un musée.



Sommaire :

1/ Présentation

2/Diagrammes systèmes

3/Études de solutions

4/contraintes techniques

5/Etude de sous-systèmes

6/Problèmes rencontrés et améliorations

7/Programme et conception

8/Répartition des tâches

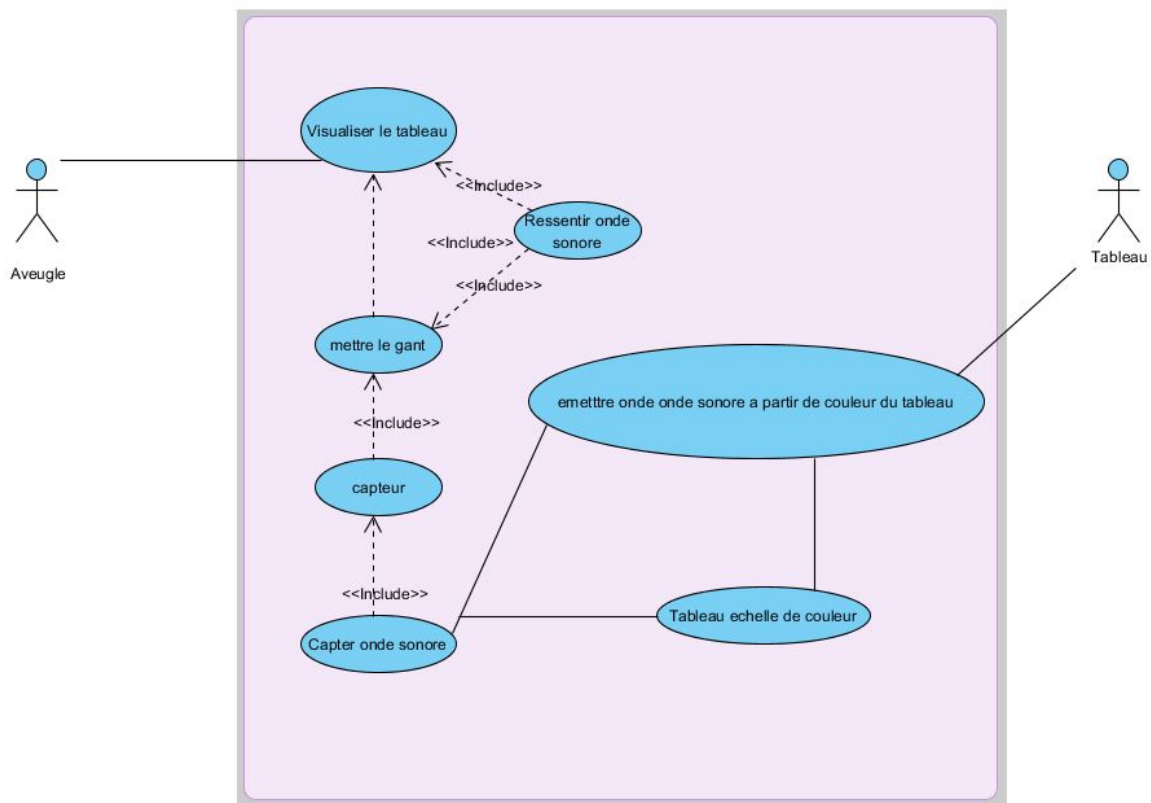
9/ Bilan et conclusion

1/Présentation

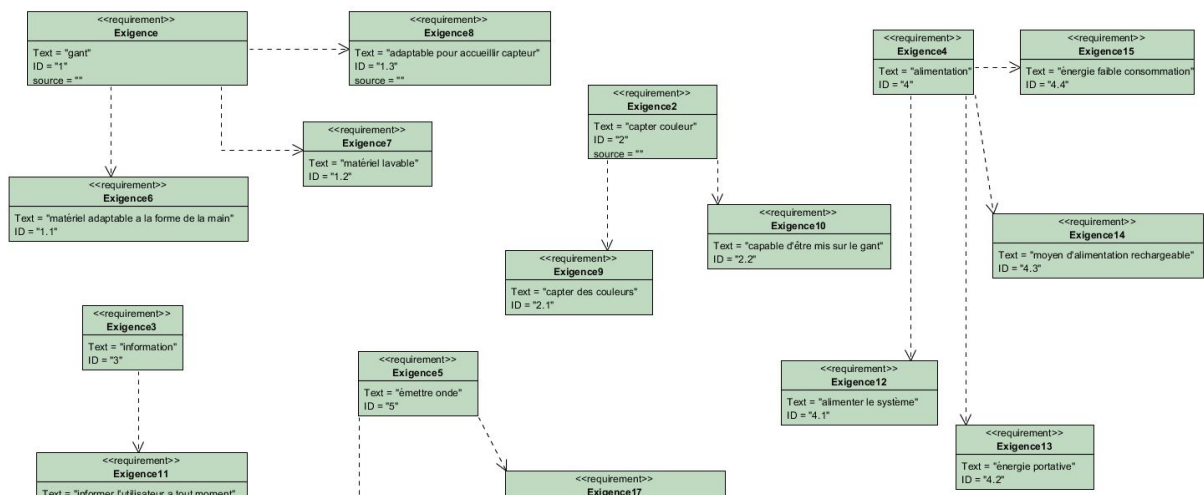
Il y a 257 millions de personnes aveugles ou malvoyantes dans le monde dont 1,7 millions en France ce qui correspond à environ 2,5% de la population française. C'est pourquoi notre objectif est de mettre en place un dispositif d'assistance aux personnes malvoyantes adapté au musée pour que même ces personnes puissent avoir leur propre ressentie et vision du tableau.

2/Diagrammes systèmes

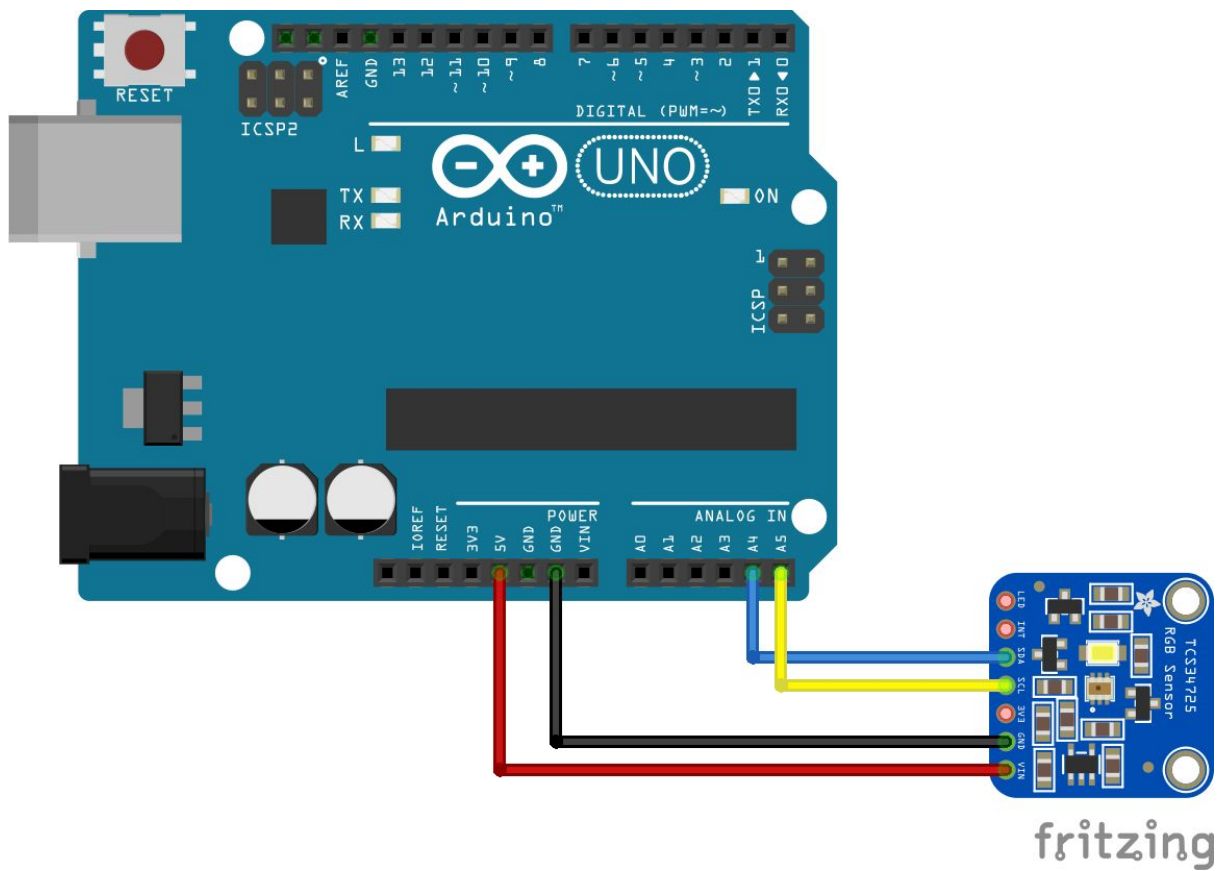
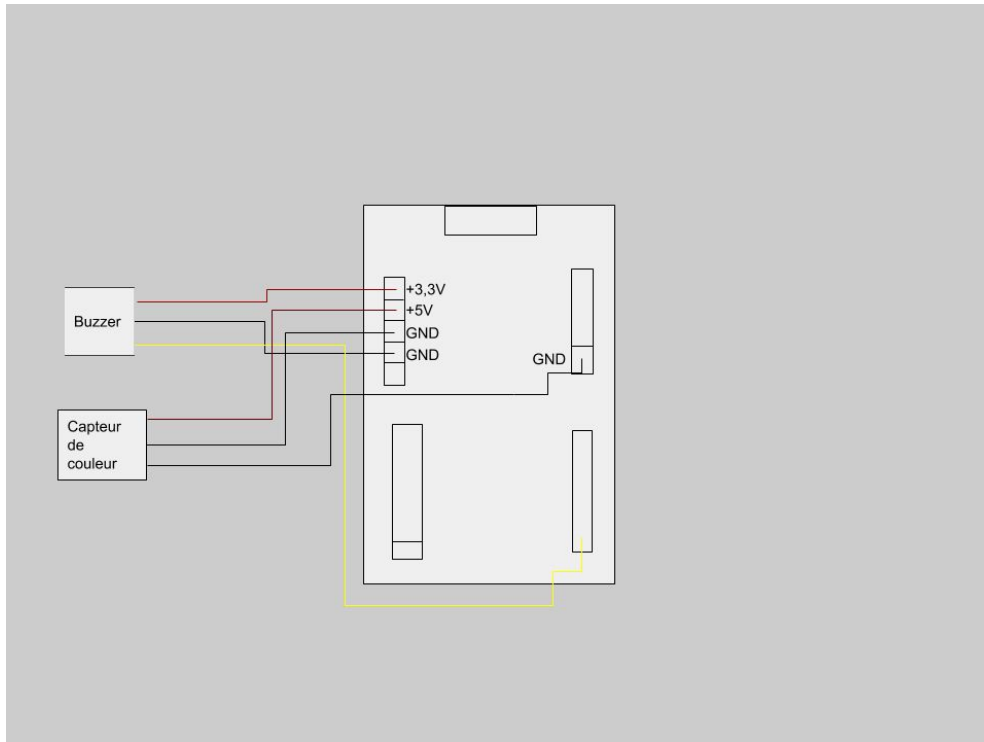
- La solution devra répondre au schéma cas d'utilisation.



- Notre système devra répondre aux exigences suivantes.







- Nous devons faire une structure électronique optimisée pour l'ajout d'un capteur couleur et d'un émetteur sonore.



3/ Étude solution

- Nous avons fait un tableau qui rassemble les différents composants qui pourraient convenir à notre système. Ce tableau compare les composants sélectionnés pour répondre au mieux à nos exigences.

Capteur de couleur		
référence	TCS230	TCS3200
alimentation	2,7 à 5,5 V	2,7 à 5,5 Vcc
dimension	24 x 30 mm	Ø 37 x 18 mm
couleur	RGB	RGB
sortie max fréquence	12/120/600 KHz	12/120/600 KHz
sortie	digital	digital
interface	analogique	analogique
Photo du composant		

Retranscription fréquence sonore		
Référence	WV14542	JBLT110BLK
Compatible avec arduino	oui	non
qualité du son	basse	haute
Photo du composant		

Energie	
référence	ALIM-BL4AA1
Type	Bloc Alimentation pour 4 piles AA
tension	6V
Photo du composant	

4/ Contrainte technique

- Comme contrainte, on a eu le matériel prédéfini par les professeurs, on a donc dû s'adapter. Heureusement le matériel prédéfini correspondais plus ou moins à nos solution trouvé

	contrainte	spécification	commentaire
13€	unité central utilisé	carte arduino Uno	simple d'utilisation
2.5€	émetteur	Grove buzzer	compatible avec arduino
6.5€	récepteur	TCS 34725	compatible avec arduino
Total	22€		

5/Etude de sous-systèmes

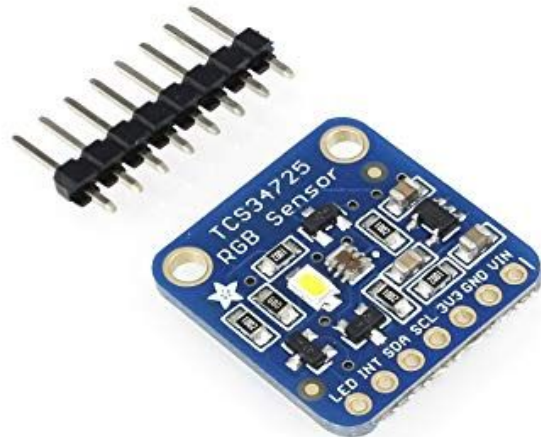
Nous avons étudié le système donné.

- Capteurs.

Capteur RGB de type numérique, comporte 4 leds blanches. Il peut détecter et mesurer une plage limitée de couleurs visibles.

Photodétecteurs rouge, bleu, vert et transparent (ce sont des filtres).

Dispose d'une interface digitale TTL et communique directement avec le microcontrôleur.



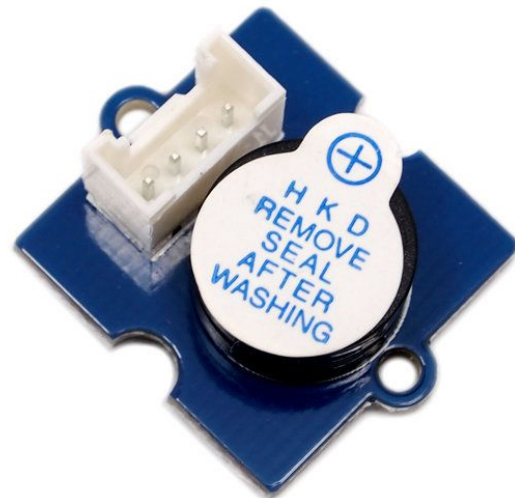
- Actionneurs.

Buzzer Grove 1 070 2000

Dispose d'une interface compatible grove, est sur base shield. Son type de commande est logique c'est-à-dire PWM.

Le voltage opéré est : 3,3V/5V.

Dimensions : 20X20 mm.



6/Problèmes rencontrés et améliorations.

Problèmes :

- Avec la sélection du buzzer, comme actionneur on a connu beaucoup de problème, comme le changement de fréquence sonore impossible qui nous a fait prendre une solution alternative, au lieu du changement de fréquence pour la couleur on a créé un nombre de bip sonore pour chaque couleur (couleur1=1 bip; couleur2 = 2 bip, etc) ce qui devient difficile appréhendé plus le nombre de couleur augmente !

Le buzzer créé un bruit fort et désagréable qui ne peut être réglé.

Puis en dernier point négatif, le son n'est pas personnel, il est audible par toute personne a proximité.

- Pour le capteur il y a un seul défaut, sa distance de détection est beaucoup trop faible, l'utilisateur serait trop proche du tableau pour que le système fonctionne.

Améliorations :

- Pour améliorer il faut commencer par avoir un meilleur capteur, avec une distance de détection plus élevé d'au moins une vingtaine de centimètre.
- Pour l'actionneur, il nous faudrait un émetteur sonore personnel comme un casque audio.

7/Programme et conception.

Le programme est coupée en trois partie.

- Initialisation du programme.

```
1 #include <Wire.h>
2 #include "Adafruit_TCS34725.h"
3 int buzzer = 7;
4 Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_700MS, TCS34725_GAIN_1X);
5 void setup(void) {
6     Serial.begin(9600);
7     if (tcs.begin()) {
8         Serial.println("Found sensor");
9     } else {
10        Serial.println("No TCS34725 found ... check your connections");
11        while (1);
12    }
13 }
```

Ici `#include` permet d'importer des bibliothèques externes, pour avoir des commandes supplémentaires.

`int buzzer = 7;` permet de définir le buzzer sur la sortie numéro 7.

`Serial.begin(9600);` est là pour paramétrer le moniteur tandis que

`if (tcs.begin()) { }` de la ligne 7 à 11, permet d'avertir le développeur dans le moniteur si le capteur de couleur est détecté ou non par le programme.

- Partie principale du programme

```
14 void loop() {
15     uint16_t r, g, b, c, colorTemp, lux;
16     tcs.getRawData(&r, &g, &b, &c);
17     colorTemp = tcs.calculateColorTemperature(r, g, b);
18     lux = tcs.calculateLux(r, g, b);
19     Serial.print("Color Temp: "); Serial.print(colorTemp, DEC); Serial.print(" K - ");
20     Serial.print("Lux: "); Serial.print(lux, DEC); Serial.print(" - ");
21     Serial.print("R: "); Serial.print(r, DEC); Serial.print(" ");
22     Serial.print("G: "); Serial.print(g, DEC); Serial.print(" ");
23     Serial.print("B: "); Serial.print(b, DEC); Serial.print(" ");
24     Serial.print("C: "); Serial.print(c, DEC); Serial.print(" ");
25     Serial.println(" ");
26     if (r > g)
27     {
28         if (r > b)
29         {
30             sonrouge();
31             Serial.print("rouge");
32         } else if (b > r)
33         {
34             sonbleu();
35             Serial.print("bleu");
36         }
37     }
38     if (g > r)
39     {
40         if (g > b)
41         {
42             sonvert();
43             Serial.print("vert");
44         } else if (b > g)
45         {
46             sonbleu();
47             Serial.print("bleu");
48         }
49     }
50 }
```

`uint16_t`; définit les variables `r`, `g`, `b`, `c`, `colorTemp` et `lux`.

`tcs.getRawData()`; récupère les données brutes provenant du capteur et les stockes dans les variables `r`, `g`, `b` et `c`.

`tcs.calculateColorTemperature()`; calcule la température de couleur à partir des variables `r`, `g` et `b`, et stocke le résultat dans la variable `colorTemp`.

`tcs.calculateLux()`; calcule l'éclairement lumineux à partir des variables `r`, `g` et `b`, et stocke le résultat dans la variable `lux`.

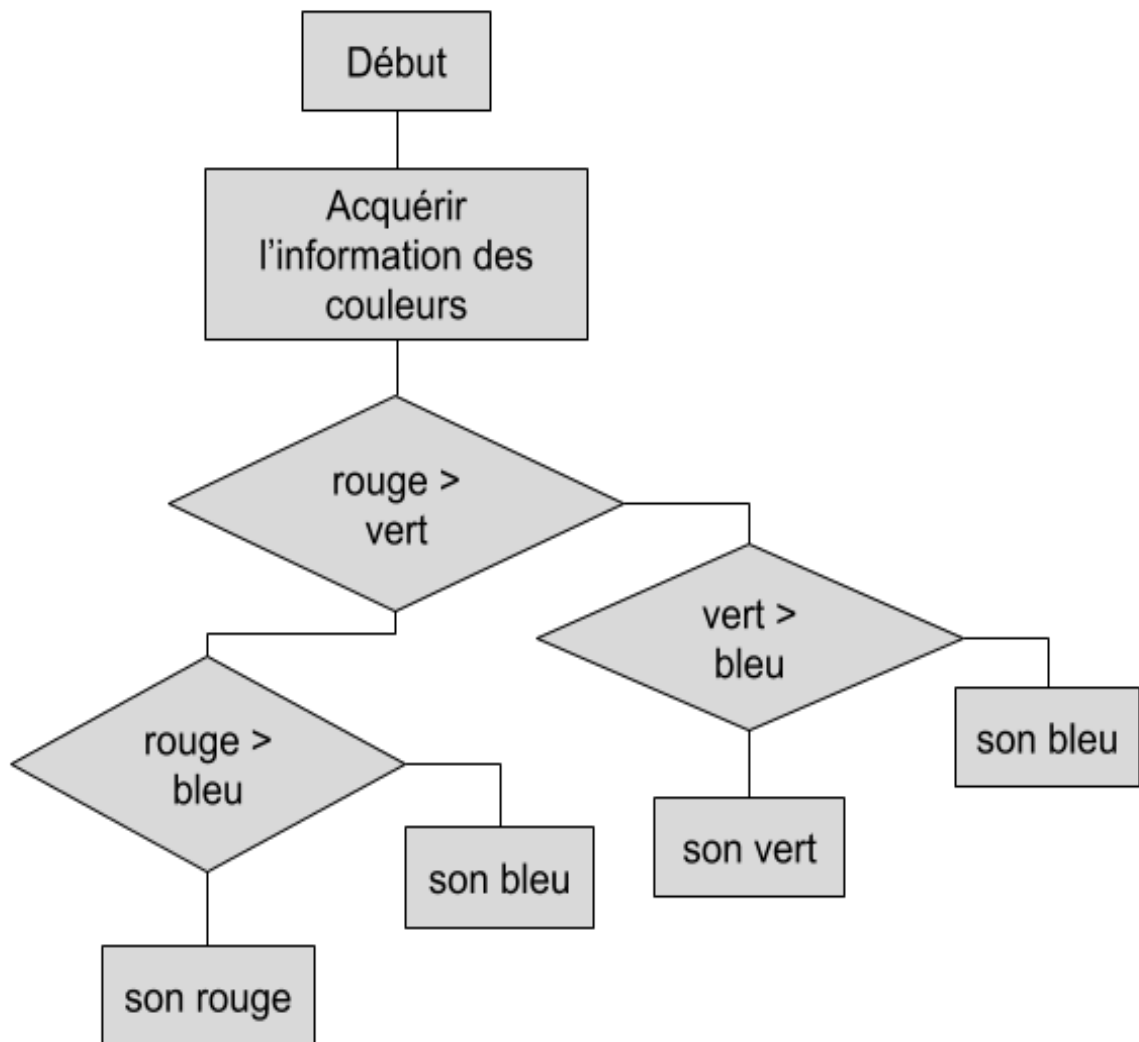
Les deux `if ()` de la ligne 26 à 49 permette de comparer les valeurs des variables `r`, `g` et `b`, et d'en déterminer celle qui a la valeur la plus élevé,

d'exécuter le sous-programme qui lui correspond et d'afficher dans le moniteur la couleur correspondante à la variable avec la plus grande valeur.

- Sous-programmes

```
51 void sonvert()  
52 {  
53     digitalWrite(buzzer, HIGH);  
54     delay (100);  
55     digitalWrite(buzzer, LOW);  
56     delay (900);  
57 }  
58 void sonbleu()  
59 {  
60     digitalWrite(buzzer, HIGH);  
61     delay (500);  
62     digitalWrite(buzzer, LOW);  
63     delay (500);  
64 }  
65 void sonrouge()  
66 {  
67     digitalWrite(buzzer, HIGH);  
68     delay (900);  
69     digitalWrite(buzzer, LOW);  
70     delay (100);  
71 }
```

Ici chaque couleur a son sous-programme, `void sonvert()` est celui de la couleur verte, lors de son exécution, `digitalWrite(buzzer, HIGH);` le buzzer bascule en état HAUT, `delay (100);` attend 100 ms puis, `digitalWrite(buzzer, LOW);` le buzzer bascule en état BAS, `delay (900);` attend 900 ms. `void sonbleu()` a le même fonctionnement mais les délais sont différents, les deux sont de 500 ms. Quant à `void sonrouge()` le premier délai sera de 900 ms et le second de 100 ms.



8/Répartition des tâches

[illegible]

9/ Bilan et conclusion

Le bilan est dans l'ensemble positif avec seulement quelques points négatif.

En effet, le projet est intéressant, il aurait fallu plus de temps et plus de matériel pour aboutir à quelque chose de parfait, malgré beaucoup de difficultés techniques comme des ordinateurs lents à démarrer ou à lancer des programmes, une cohésion d'équipe et tâche mal réparties sur le début du projet et enfin le logiciel arduino qu'on doit mettre à jour toutes les séances.

Nous avons su nous adapter et obtenir un résultat positif, grâce aussi aux professeurs qui nous ont guidé et aidé surtout sur le programme.

Pour conclure, le projet de dispositif d'assistance aux personnes malvoyantes dans un musée peut être possible.