

# **Projet final**

## **Objectifs:**

- Réaliser le montage électrique du jeu
- Compléter le corps du programme
- Améliorer le jeu (pour les plus rapide)



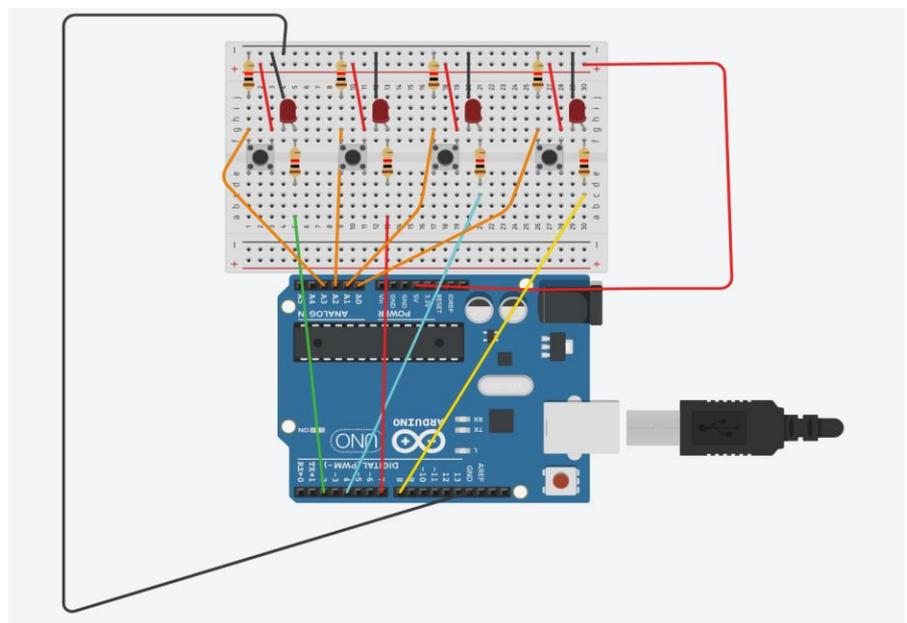
### **Règle du jeu:**

Le but du jeu est d'aller le plus loin possible dans les niveaux. Pour chaque niveau, une séquence de lumières s'allume de façon aléatoire. Le joueur doit mémoriser la séquence et appuyer sur les boutons correspondant pour retrouver la séquence. Quand un niveau est réussi, on passe au niveau suivant en ajoutant une couleur aléatoire à la séquence. Lorsqu'un joueur perd, on veut savoir son score, et remettre le jeu à 0.



## **Etape 1: Le Montage**

A l'aide des fiches précédentes, de vos connaissances et du schéma ci-dessous, réalisez le montage électrique.



**Fait le montage et vérifie avec l'encadrant !**



## Etape 2: Le Code

A l'aide de la structure de code disponible, des fiches précédentes, et de vos connaissances, complétez le code.

La structure du code donné, est la structure du jeu le plus simple possible, nous verrons dans les prochaines étapes que nous pouvons l'améliorer.



Pour compléter le code, il faut bien lire les conseils situés en haut du code. De plus à chaque endroit où il faut compléter, des indications sur ce qui est attendu est disponible.

Il est aussi conseillé dès qu'il y a des difficultés, de relire les fiches précédentes.

Chaque ligne de code est écrite dans un code simple et pas astucieux. Il ne faut donc pas aller chercher les solutions trop loin.



**Fait la programmation et lance le programme!**



## Etape 3: Améliorations (Pour les plus rapides)

Plusieurs améliorations simples et amusantes sont possibles. Vous pouvez laisser libre cours à votre imagination, ou suivre nos suggestions dans l'ordre que vous souhaitez.

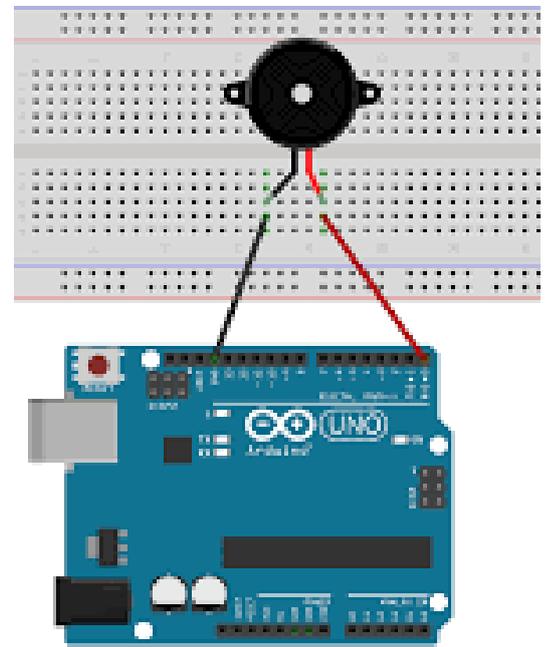
### **1<sup>er</sup> suggestion: Le Buzzer (niveau moyen)**



Un buzzer est un petit haut-parleur capable de donner plusieurs notes avec une intensité réglable.



Pour les branchement il suffit de suivre le schéma. Cependant attention à mettre le buzzer sur une sortie analogique ~ par exemple la sortie 11 convient très bien.



**Fait le montage et vérifie avec l'encadrant !**



Pour le code voici un exemple de code, à vous de le déchiffrer et de l'utiliser dans votre jeu.

Notons tout de même que « tone » permet de définir la note à laquelle est joué le note et on utilise un analogueWrite car la sortie que l'on donne est un entier entre 0 et 1024 qui définit la puissance du sons émis.

```
1 int buzzer = 11;
2
3 void setup() {
4   pinMode(buzzer;OUTPUT);
5
6 }
7
8 void loop() {
9
10  tone(buzzer;500,500); // allumer le buzzer au tone 500 ( les tones vont de 0 à 1024), pendant 500ms
11  analogueWrite(buzzer,0); // éteindre le buzzer
12  delay(500);
13 }
```



**Fait la programmation et lance le programme!**

## 2<sup>ème</sup> amélioration: le potentiomètre (niveau facile)

Il faut ici ajouter un potentiomètre qui permet de gérer la vitesse d'apparition du code couleur.

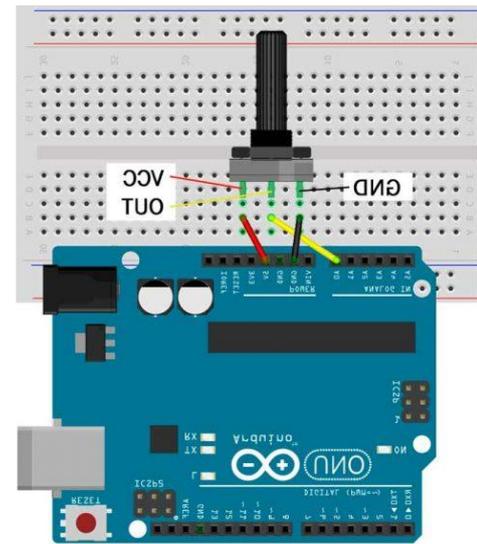
## ? Point de cours:

Pour commencer, un potentiomètre est une résistance réglable. Autrement dit, on peut régler le R de la résistance en tournant. Cela permet, en utilisant la loi d'Ohm  $U=RI$ , d'obtenir une sortie analogique qui va de 0 à 1024. En effet comme on connaît U, ici 5V, alors en fonction de comment on règle notre potentiomètre (R), alors on obtient un I. Le potentiomètre relève ce I et grâce à une amplification donne à la carte une valeur de 0 à 1024.



### Comment fait-on:

Voici le schéma de branchement du potentiomètre, on récupère l'information analogique via la broche du centre.



**Fait le montage et vérifie avec l'encadrant !**



Voici un exemple de code pour comprendre comment il fonctionne.

```
1 int pot = A0;
2
3 void setup() {
4   pinMode(pot, INPUT);
5   Serial.begin(9600);
6 }
7
8 void loop() {
9   Serial.print(analogRead(pot)); // permet ici d'afficher sur l'ordinateur (moniteur série) la valeur
10  | | | | | | | | | | | | | | | | | | // du potentiomètre
11
12 }
```



Finalement l'idée est de récupérer l'information du potentiomètre et de l'adapter pour qu'il corresponde à un temps de delay entre l'allumage des LED. Autrement dit il faut remplacer les delay de 500 ou 1000 par une variable directement lié au potentiomètre.