Super Simon

Martin Novo et Alexis Lallemand

Sommaire

Pourquoi un Simon?

- Permet d'améliorer sa mémoire
- Jeu que nous apprécions tous les deux
- Allie code et montage

Objectif et Problématique

- 4 couleurs
- Effet visuel lumineux
- Effet sonore
- Un ajout de une couleur par niveau
- Le jeu peut-être relancé après une victoire comme une défaite

Le matériel

- 4 boutons de couleur avec LED intégrées
- Résistances
- Haut-parleur
- Potentiomètre
- Carte Arduino Uno

- Imprimante 3D
- Fer à souder

Fonctions et montage

- Méthode sequence_hasard()
- Conditions de démarrage d'une partie
- Méthode suite_couleur()

```
void loop() {
  if (LVL==1){
     sequence_hasard();

  }
  /*Condition de démarrage du jeu qui sera dans ce cas : appuyer sur un des 4 boutons*/
  if (digitalRead(bouton_bleu)==LOW || digitalRead(bouton_rouge)==LOW || digitalRead(bouton_jaune)==LOW || digitalRead(bouton_vert)==LOW){
    melodie_depart();
    suite_couleur();
    partie_simon();
  }
}

/*Cette méthode va générer au hasard MAX_LVL valeurs comprise entre 2 et 5*/
void sequence_hasard(){
 randomSeed(analogRead(0));
 for (int i=0; i<MAX_LVL; i++){
   ordre[i] = random(2,6);
 }
}



/*Cette méthode va reproduire la séquence de couleur au faire et à mesure*/
void suite_couleur(){
   for (int i=0; i<LVL; i++){
      digitalWrite(ordre[i],HIGH);
      delay(600);
      digitalWrite(ordre[i],LOW);
      son_led(ordre[i]);
      delay(600);
      digitalWrite(ordre[i],HIGH);
      delay(600);
    }

  }
```

Fonctions et montage

- Méthode partie_simon()

```
/*Cette méthode va distinguer chaque possibilité en fonction du bouton auquel on appuie et si c'est
une bonne ou mauvaise couleur*/
void partie_simon(){
for(int j=0; j<MAX_LVL; j++){
  for (int i=0; i<LVL; i++) {
    drapeau = 0;
    while (drapeau == 0) {
      if (digitalRead(bouton_bleu) == LOW) {
        digitalWrite (led_bleue, LOW);
        tone(speaker,son[0],200);
        delay(200);
        tone(speaker,son2[0],200);
        own_follow[i] = 2;
        drapeau = 1;
        delay(600);
        if (own_follow[i] != ordre[i]) {
          mauvaise_sequence();
          exit(0);
        }
        digitalWrite (led_bleue,HIGH);
      }
      if (digitalRead(bouton_rouge) == LOW) {
        digitalWrite (led_rouge, LOW);
        tone(speaker,son[1],200);
        delay(200);
        tone(speaker,son2[1],200);
        own_follow[i] = 3;
        drapeau = 1;
        delay(600);
        if (own_follow[i] != ordre[i]) {
          mauvaise_sequence();
          exit(0);
        }
        digitalWrite (led_rouge,HIGH);
      }
      if (digitalRead(bouton_jaune) == LOW) {
        digitalWrite (led_jaune, LOW);
        tone(speaker,son[2],200);
        delay(200);
        tone(speaker,son2[2],200);
        own_follow[i] = 4;
        drapeau = 1;
        delay(600);
        if (own_follow[i] != ordre[i]) {
          mauvaise_sequence();
          exit(0);
        }
        digitalWrite (led_jaune,HIGH);
      }
      if (digitalRead(bouton_vert) == LOW) {
        digitalWrite (led_verte, LOW);
        tone(speaker,son[3],200);
        delay(200);
        tone(speaker,son2[3],200);
        own_follow[i] = 5;
        drapeau = 1;
        delay(600);
        if (own_follow[i] != ordre[i]) {
          mauvaise_sequence();
          exit(0);
        }
        digitalWrite (led_verte,HIGH);
      }
    }
  }
    bonne_sequence();
  }
}
```

Fonctions et montage

- Méthode bonne_sequence()
- Méthode mauvaise_sequence()

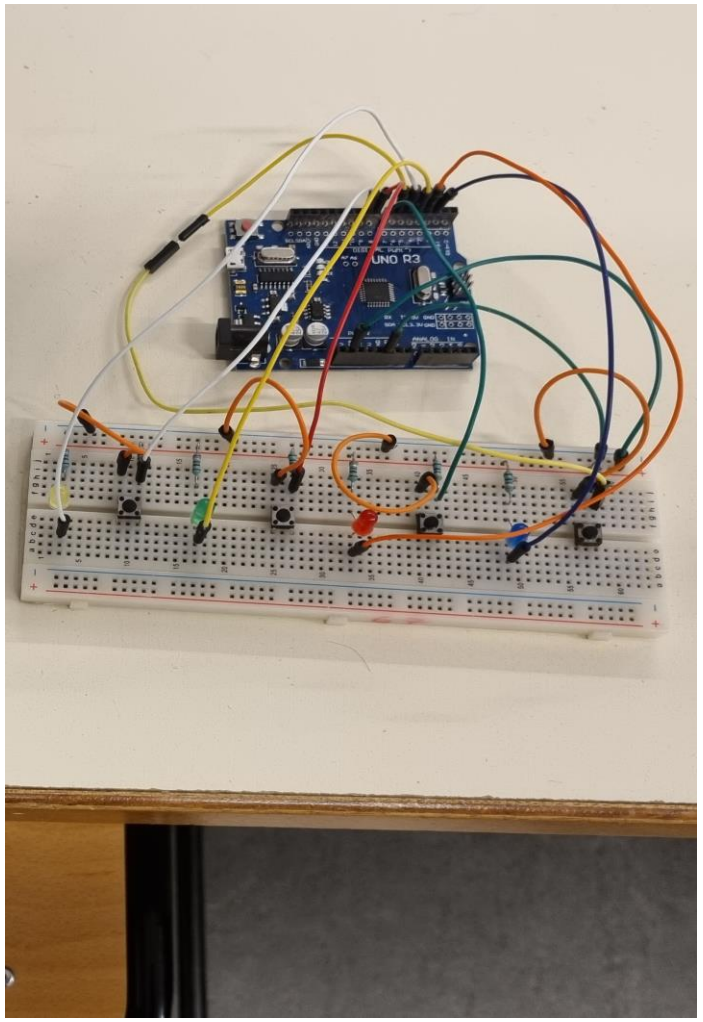
```
/*Cette méthode incrémente le LVL de 1 et va reproduire la même séquence aléatoire en y ajoutant une couleur
  parmois les 4 au hasard mais aussi allumer les 4 LED une fois que les MAX_LVL combaisons ont bien réussies*/
void bonne_sequence(){
      if (LVL < MAX_LVL+1){
         LVL ++;
      }

      if (LVL==MAX_LVL+1){
        digitalWrite(led_bleue,HIGH);
        digitalWrite(led_rouge,HIGH);
        digitalWrite(led_jaune,HIGH);
        digitalWrite(led_verte,HIGH);
        delay(1000);
        digitalWrite (led_bleue,LOW);
        digitalWrite (led_rouge,LOW);
        digitalWrite (led_jaune,LOW);
        digitalWrite (led_verte,LOW);
        tone(speaker,1318,300);
        delay(300);
        tone(speaker,1318,350);
        delay(300);
        tone(speaker,1318,350);
        delay(300);
        tone(speaker,1046,350);
        delay(300);
        tone(speaker,1318,350);
        delay(300);
        tone(speaker,1567,350);
        delay(300);
        tone(speaker,784,350);
        delay(500);
        tone(speaker,880,350);
        delay(300);
        tone(speaker,880,350);
        delay(1000);
        digitalWrite(led_bleue,HIGH);
        digitalWrite(led_rouge,HIGH);
        digitalWrite(led_jaune,HIGH);
        digitalWrite(led_verte,HIGH);
        LVL=1;

      }

      else{
        suite_couleur();
      }
}
```

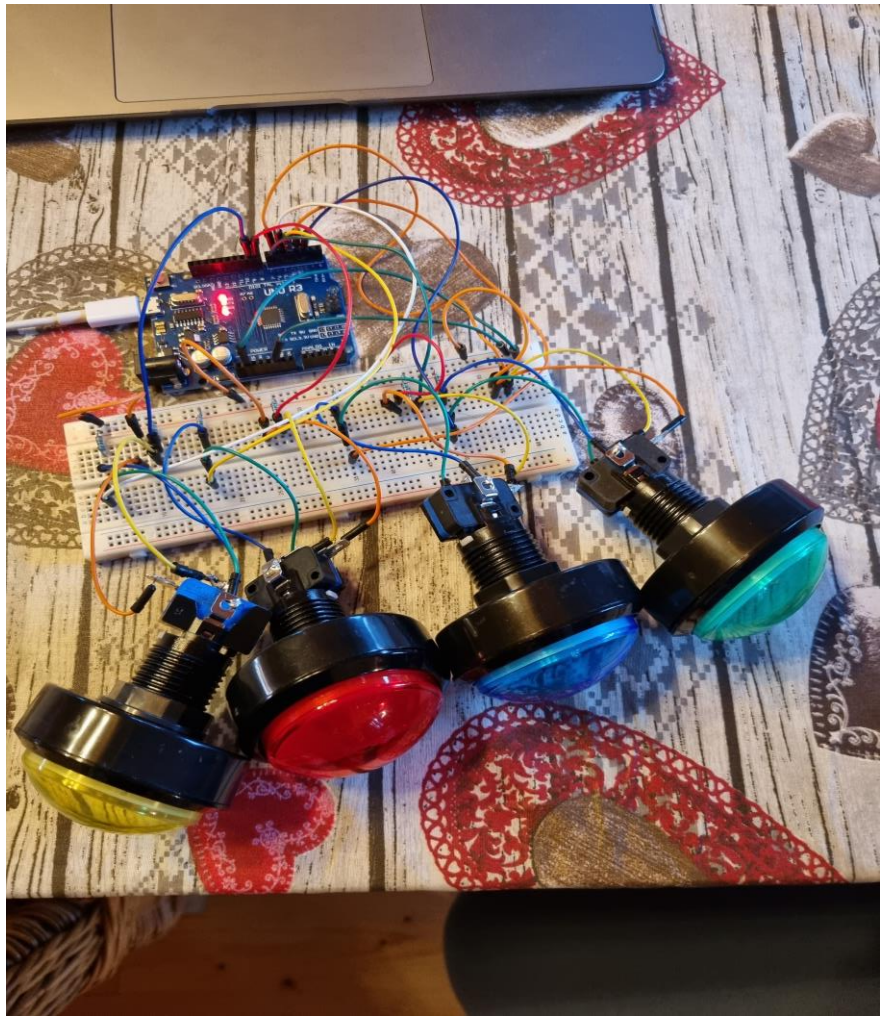
```
/*Cette méthode va éteindre la LED qui ne correspond pas à la bonne couleur et ensuite allumer d'un
coup les 4 LED puis les éteindre et enfin remettre le LVL à 1*/
void mauvaise_sequence(){
    digitalWrite(led_bleue,HIGH);
    digitalWrite(led_rouge,HIGH);
    digitalWrite(led_jaune,HIGH);
    digitalWrite(led_verte,HIGH);
    delay(1000);
    digitalWrite (led_bleue,LOW);
    digitalWrite (led_rouge,LOW);
    digitalWrite (led_jaune,LOW);
    digitalWrite (led_verte,LOW);
    tone(speaker,2217,350);
    delay(500);
    tone(speaker,1108,350);
    delay(500);
    tone(speaker,554,350);
    delay(500);
    tone(speaker,277,350);
    delay(1000);
    digitalWrite(led_bleue,HIGH);
    digitalWrite(led_rouge,HIGH);
    digitalWrite(led_jaune,HIGH);
    digitalWrite(led_verte,HIGH);
    LVL=1;

}
```

Fonctions et montage

Les montages éléctriques :

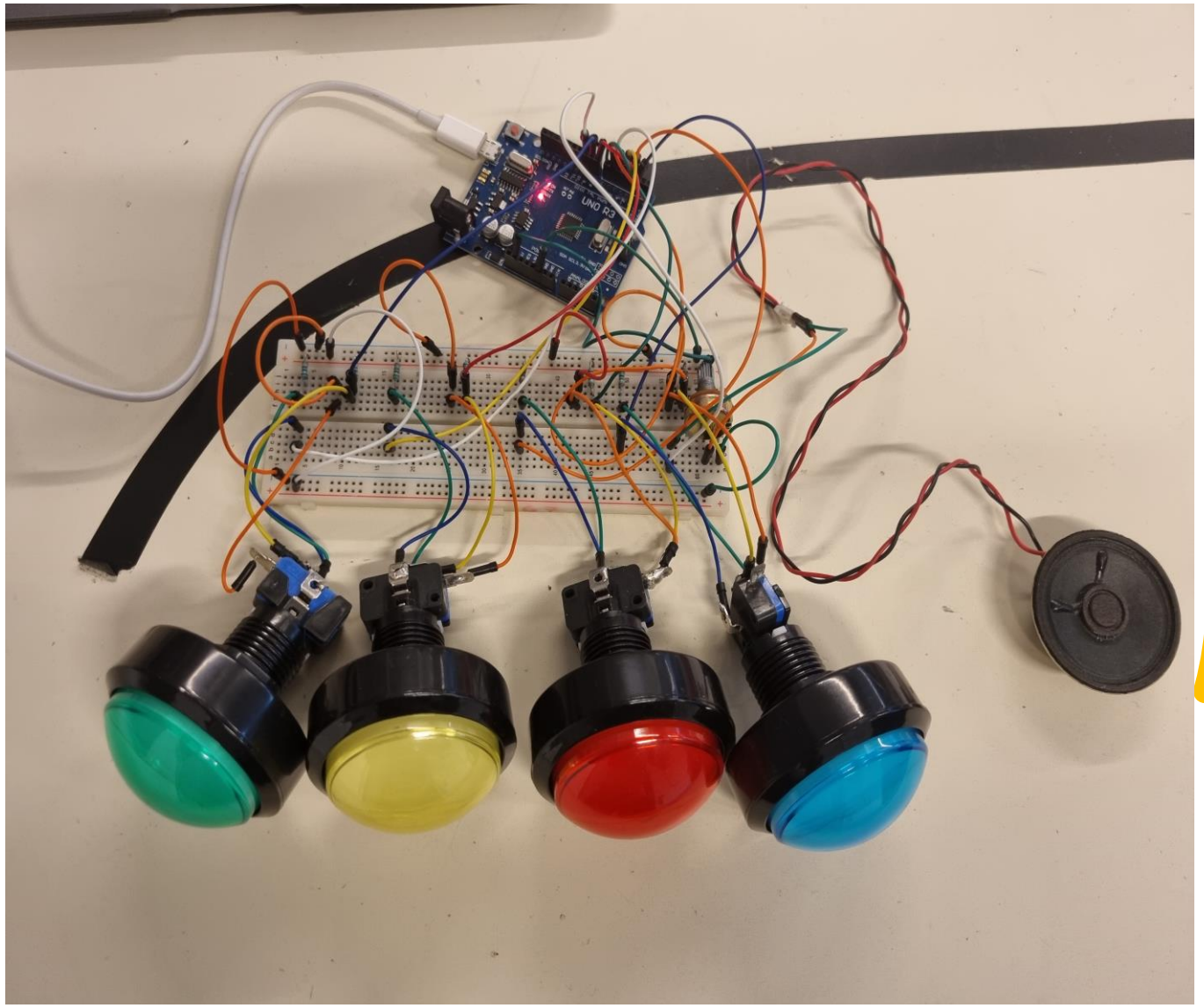
- Avec des petites LED et des boutons poussoirs
- Avec les 4 boutons



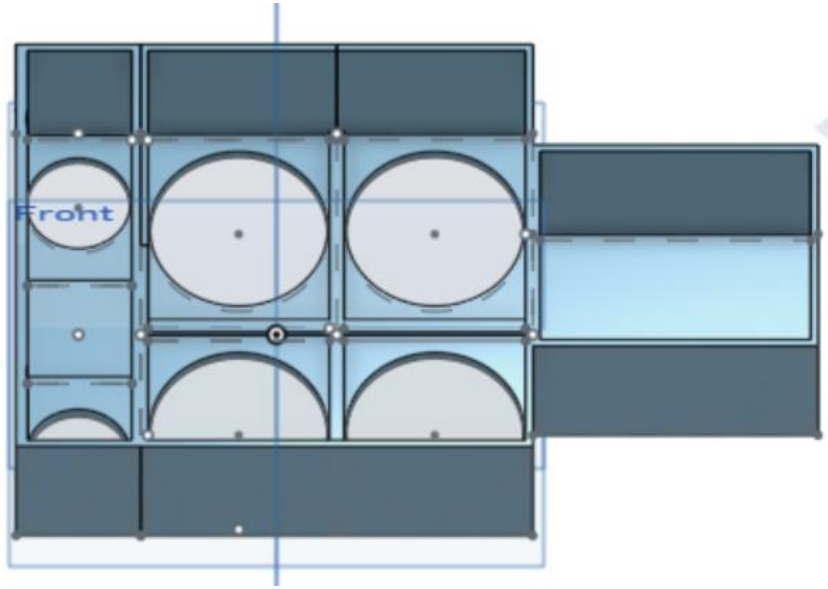


Fonctions et montage

Le montage final

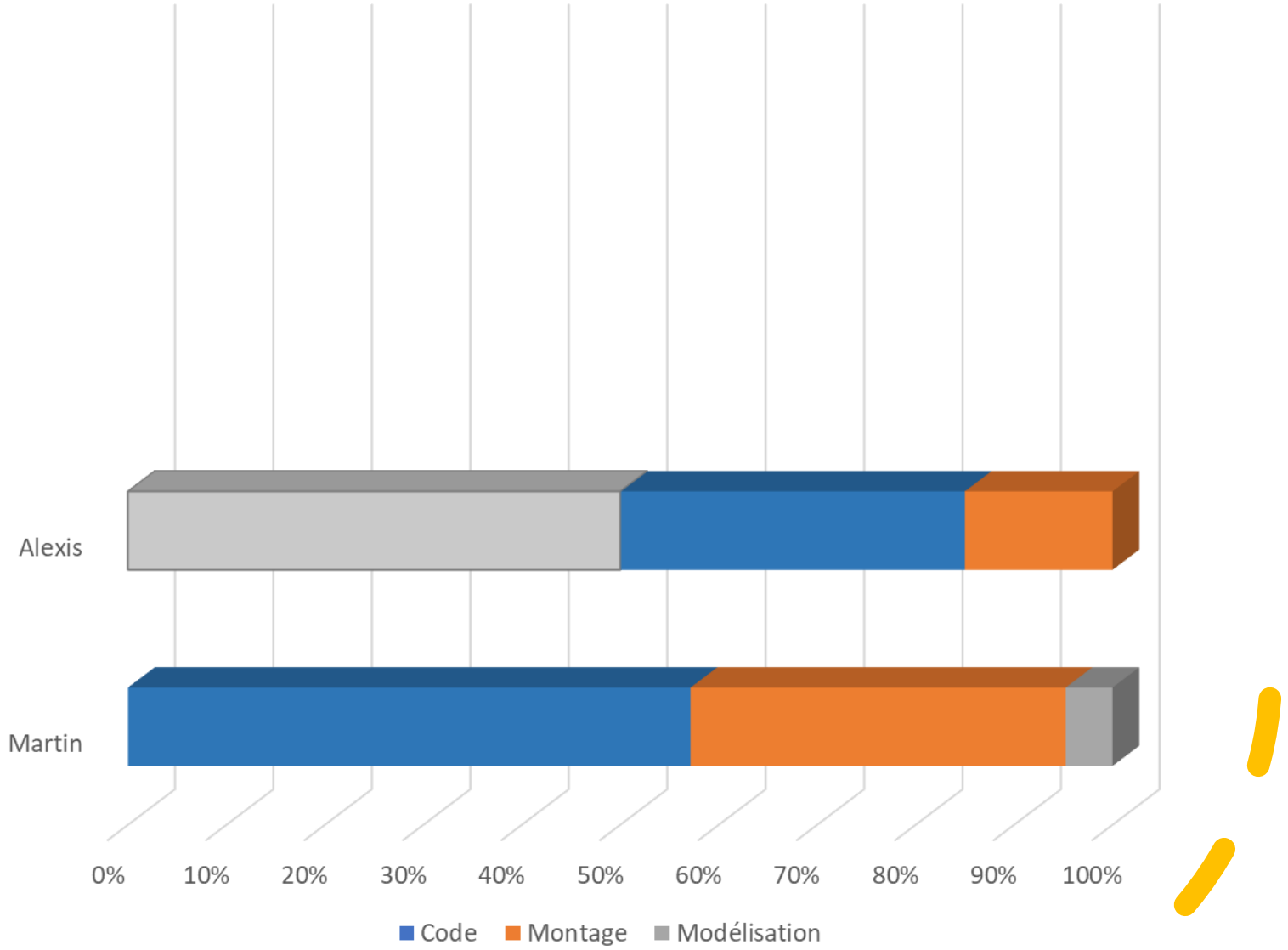


Modélisation et son




```
void partie_simon(){
for(int j=0; j<MAX_LVL; j++){
  for (int i=0; i<LVL; i++) {
    drapeau = 0;
    while (drapeau == 0) {
      if (digitalRead(bouton_bleu) == LOW) {
        digitalWrite (led_bleue, LOW);
        tone(speaker,son[0],200);
        own_follow[i] = 2;
        drapeau = 1;
        delay(1000);
        if (own_follow[i] != ordre[i]) {
          mauvaise_sequence();
        }
        digitalWrite (led_bleue,HIGH);

      }
      if (digitalRead(bouton_rouge) == LOW) {
        digitalWrite (led_rouge, LOW);
        tone(speaker,son[1],200);
        own_follow[i] = 3;
        drapeau = 1;
        delay(1000);
        if (own_follow[i] != ordre[i]) {
          mauvaise_sequence();
        }
        digitalWrite (led_rouge,HIGH);
```

Planning


Alexis
Martin
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%
Code
Montage
Modélisation

Conclusion

- Projet instructif
- Plus de complications qu’anticipées
- Confiance l’un en l’autre importante pour réussir
- Plaisir à réaliser