

Ethan E.

Luc V.

Paul A.

Partie 1 : Cahier des charges

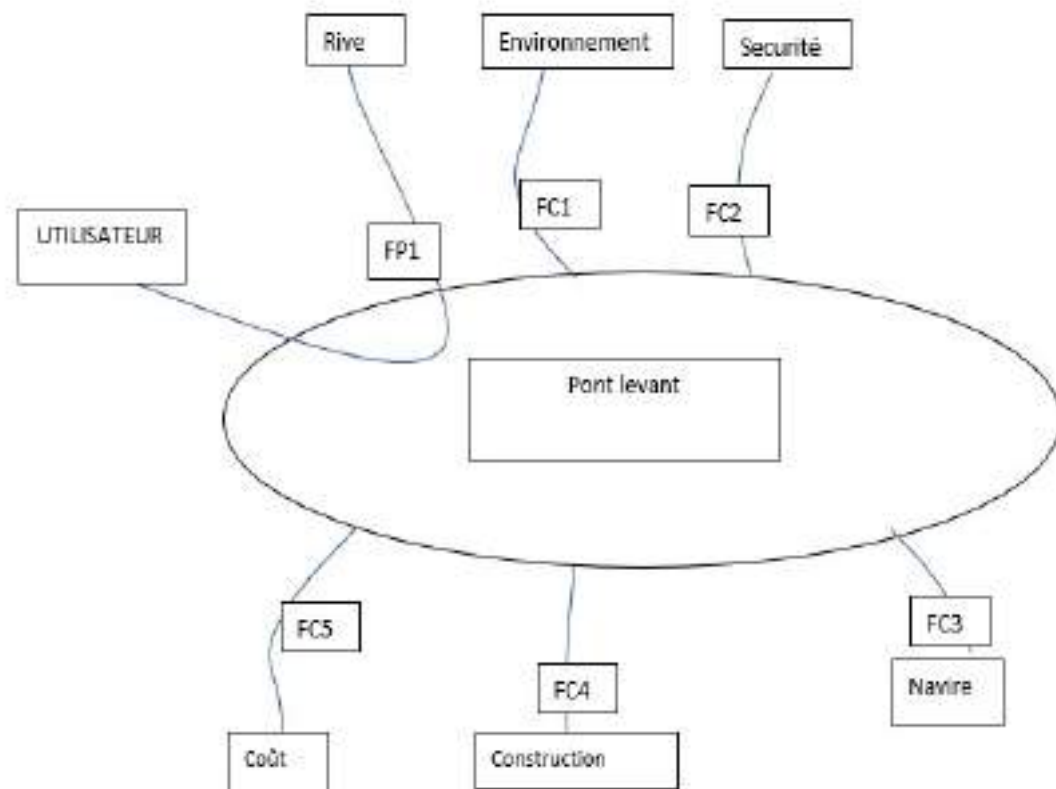
- A qui rends service cette innovation et dans quel but
- Ecologie
- Matériaux

Partie 2 : La réalisation des pièces sur SolidWorks

- Modélisation et utilisation des pièces

Partie 3 : Le codage, le montage et l'assemblage

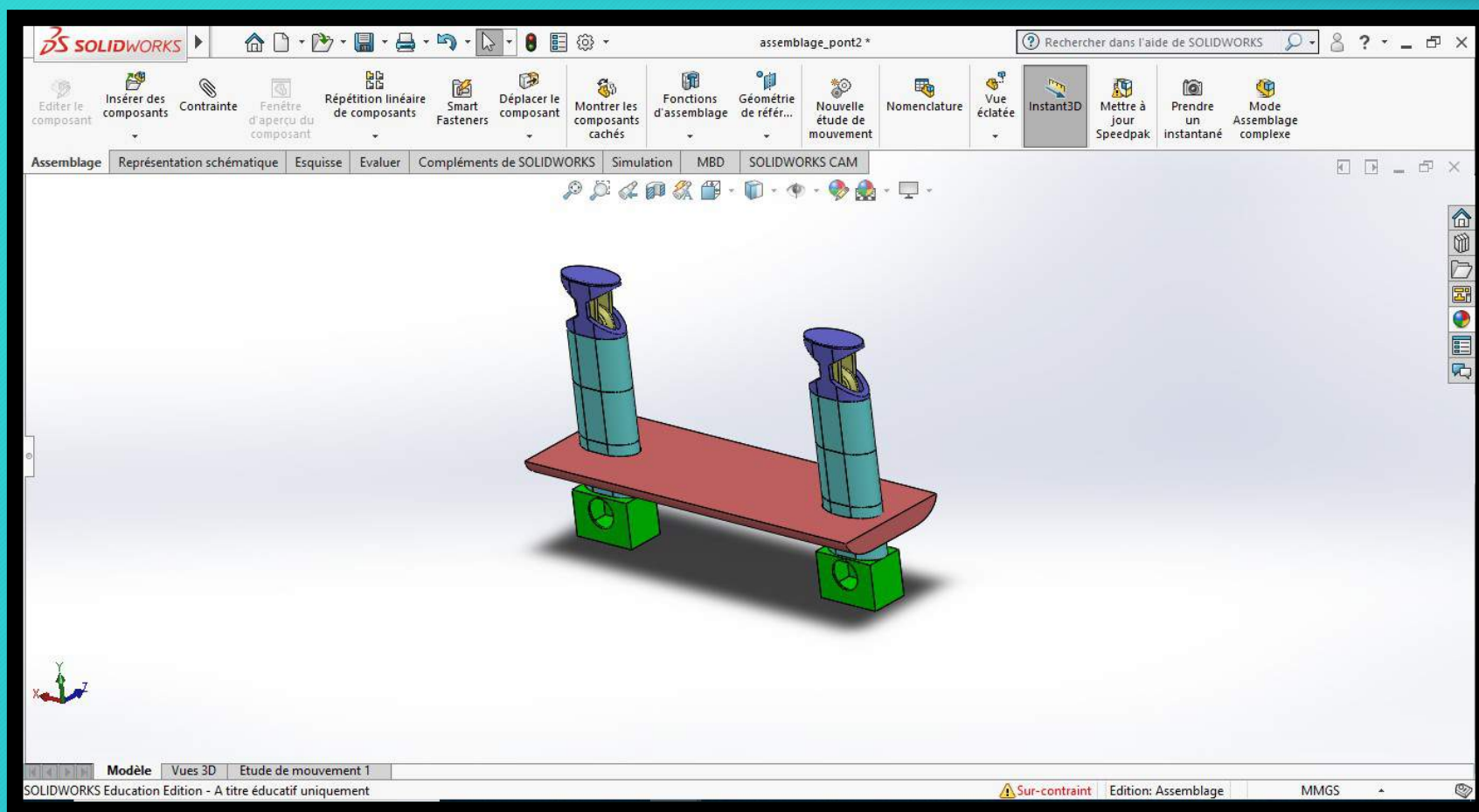
- Présentation programme
- Présentation câblage
- Présentation assemblage

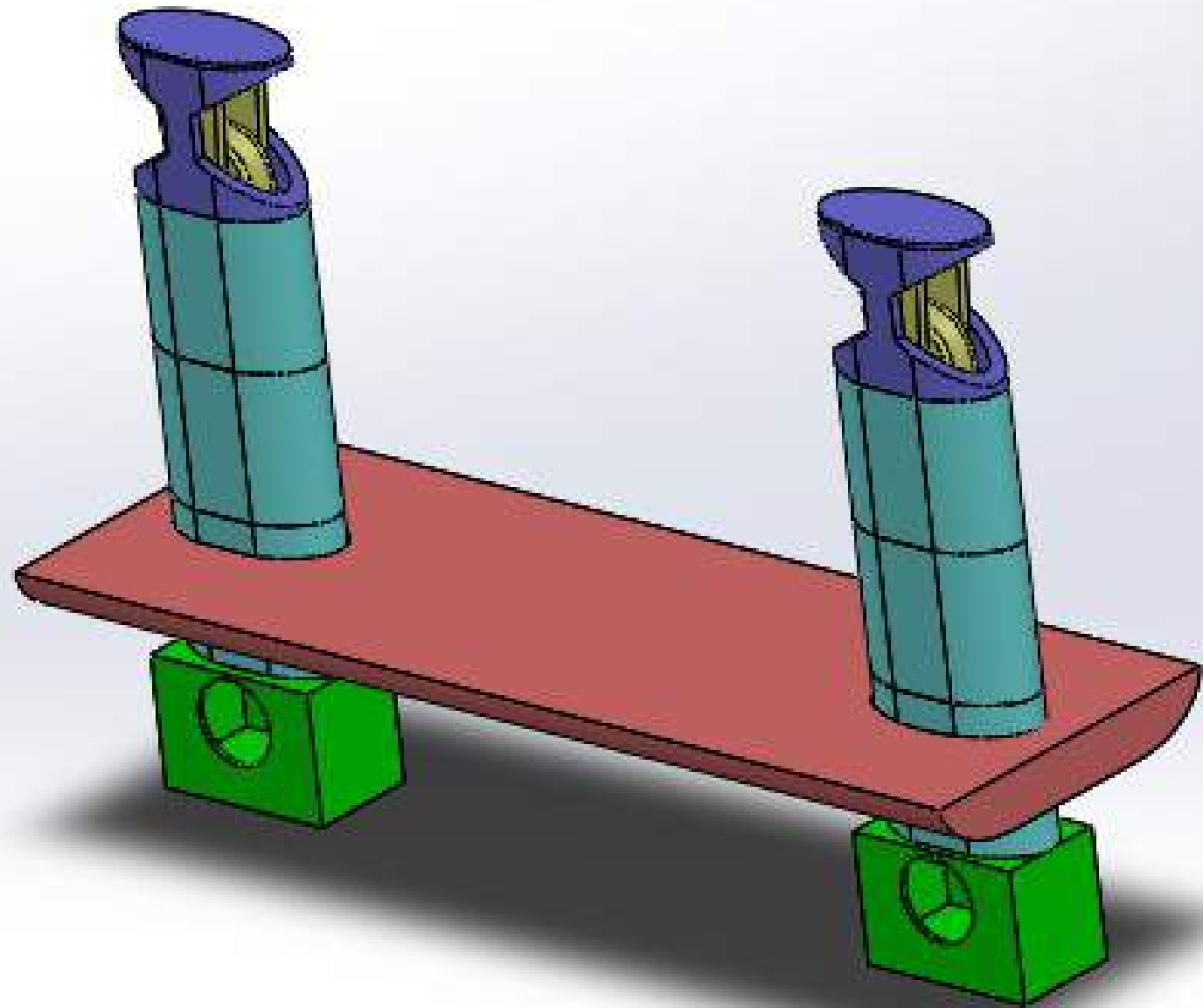


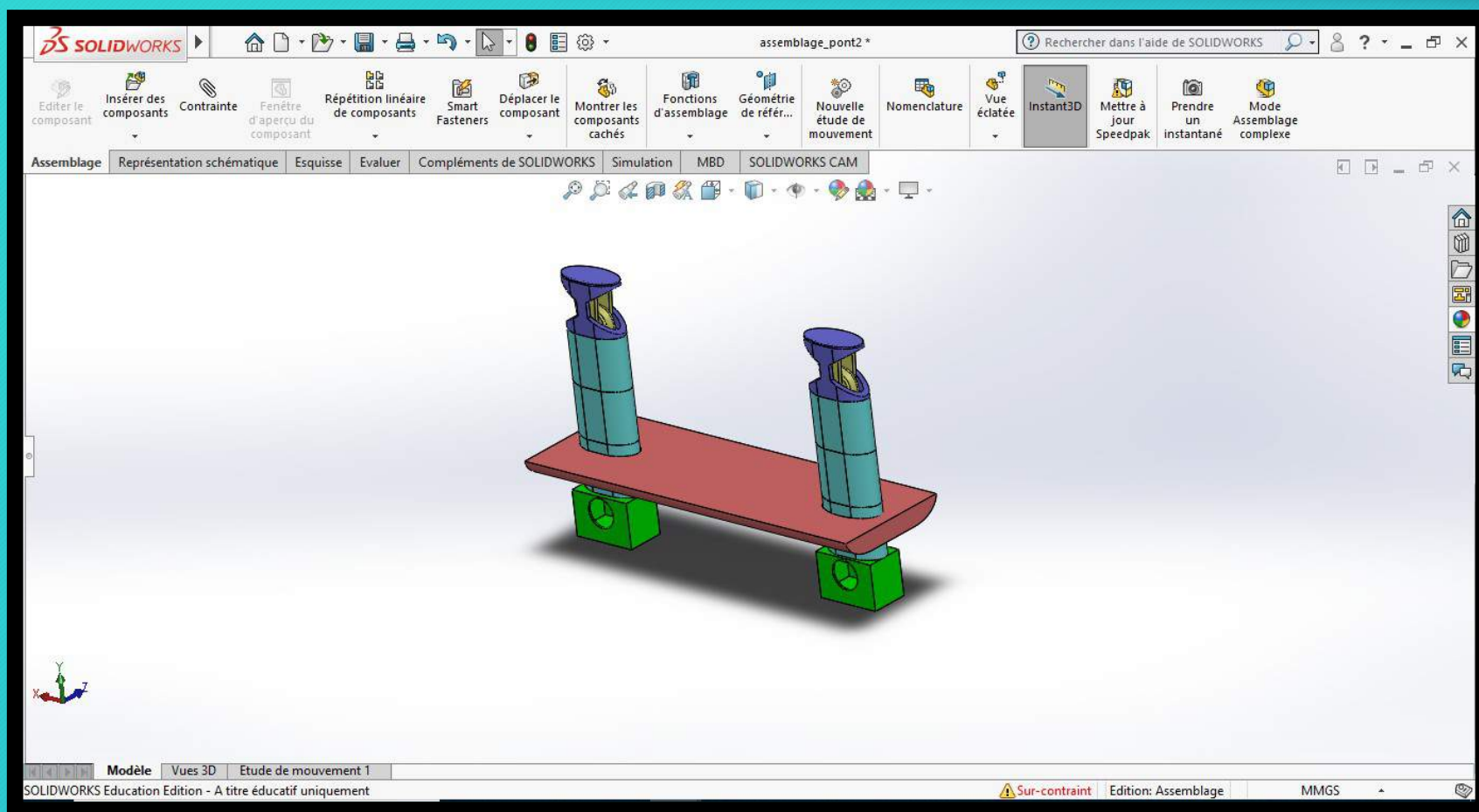
N°	Fonctions	Critères
FP1	Permettre aux usagers de passer d'une rive à l'autre	- Pente - Croisement des usagers (largeur pont) - Longueur pont
FC1	S'intégrer à l'environnement, être esthétique	Environnement de la rivière, de la ville
FC2	Permettre aux usagers d'évoluer en sécurité	- Garde-corps (hauteur) - Revêtement
FC3	Permettre aux bateaux de passer	- hauteur - distance
FC4	Résister à la circulation, aux intempéries	- matériaux - superficie du projet
FC5	Être économiquement pertinent	- cout des matériaux

- Ficelle
- Servomoteur
- Poulie
- Détecteur Ultrason
- Planche de bois
- Câble électronique
- Capteur fin de course
- Batterie









le_bon_code

```
const byte TRIGGER_PIN = 2; // Broche TRIGGER
const byte ECHO_PIN = 3;    // Broche ECHO
const byte FC_haut = 12;    // Broche ECHO
const byte FC_bas = 13;     // Broche ECHO
const byte moteur_D = 6;    // Broche ECHO
const byte moteur_G = 7;    // Broche ECHO

const unsigned long TIMEOUT = 25000UL; // 25ms = ~5m à 340m/s // Constantes pour le timeout
const float SOUND_SPEED = 340.0 / 1000; // Vitesse du son dans l'air en mm/us

#include <Servo.h>
Servo Servo_D;
Servo Servo_G;

void lever_pont() {
  Serial.println("lever pont");
  boolean fin=digitalRead(FC_haut); // bouton activé
  while (fin==HIGH) {
    Servo_D.writeMicroseconds(1600); // Servomoteur vitesse
    Servo_G.writeMicroseconds(1600);
    delay(50);
    fin=digitalRead(FC_haut); // Pont s'arrête
    Serial.println(fin);
  }
  Servo_D.writeMicroseconds(1500);
  Servo_G.writeMicroseconds(1500);
}

void baisser_pont() {
  Serial.println("baisser pont");
  boolean fin=digitalRead(FC_bas); // bouton activé
  while (fin==HIGH) {
    Servo_D.writeMicroseconds(1400); // Servomoteur vitesse
    Servo_G.writeMicroseconds(1400);
    delay(50);
    fin=digitalRead(FC_bas); // Pont s'arrête
    Serial.println(fin);
  }
}
```



```

}
Servo_D.writeMicroseconds(1500);
Servo_G.writeMicroseconds(1500);
}

void setup() {
  Serial.begin(115200);
  pinMode(PC_haut, INPUT_PULLUP);
  pinMode(PC_bas, INPUT_PULLUP);
  Servo_D.attach(moteur_D);
  Servo_G.attach(moteur_G);
  pinMode(ECHO_PIN, INPUT);
  pinMode(TRIGGER_PIN, OUTPUT);
  digitalWrite(TRIGGER_PIN, LOW); // La broche TRIGGER doit être à LOW au repos
}

void loop() {
  // Lance une mesure de distance en envoyant une impulsion HIGH de 10µs sur la broche TRIGGER
  digitalWrite(TRIGGER_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGGER_PIN, LOW);
  // Mesure le temps entre l'envoi de l'impulsion ultrasonique et son écho (s'il existe)
  long mesure = pulseIn(ECHO_PIN, HIGH, TIMEOUT);
  // Calcul la distance à partir du temps mesuré
  float distance_mm = mesure / 2.0 * SOUND_SPEED;
  Serial.print("Distance: ");
  Serial.print(distance_mm);
  Serial.println(" mm"); // Renvoie une distance en mm toute les 2s.
  delay(2000);

  if ( distance_mm<125 || distance_mm>25 ) {
    lever_pont();
  }
  else { baisser_pont(); } // Si la distance renvoyée est inférieur à 50mm la fonction "lever_pont" s'active
                          // et dans le cas contraire la fonction "baisser_pont" s'active.
}

```