

# GANTS HAPTIQUES

Alejandro DEMPHITTE, Floriane DE OLIVEIRA  
Jian Long LI, Lucie RALUY et Martin SCHETTER



# Plan

**01** Contexte  
Etude de marché  
de l'existant

**02** Identification des outils  
Matériels, outils pour la réalisation

**03** Organisation  
temporelle

**04** Schéma-bloc et  
répartition des  
rôles

**05** Physiologie

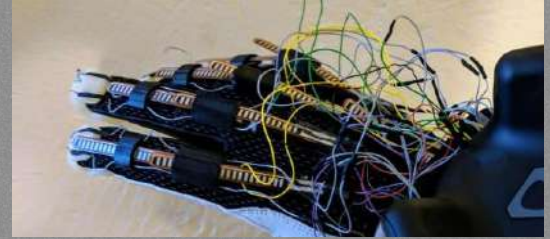
**06** Montage

**07** 1er prototype  
Reproduction de  
la sensation de  
vibration

**08** 2nd prototype  
Reproduction de  
la sensation de  
chaleur

**09** Bilan

# Contexte des gants haptiques





# Matériels Choisis

## Matériels :

Actionneur **piézoélectrique**

Création du **capteur thermique** en salles blanches : Puce PCR

Gant (**PDMS** par exemple en salles blanches)

Carte électronique (**Arduino Uno**)

Circuits électroniques & composants (fils, plaque, leds etc...)

Bouton poussoir

Capteur de position (ADXL)

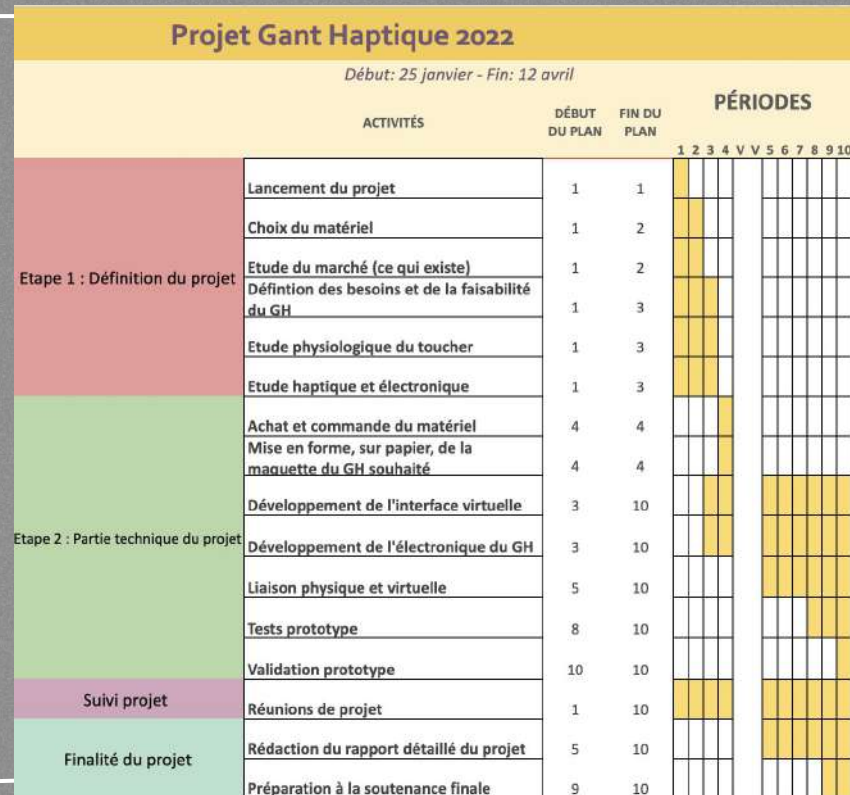
Modulateur d'intensité (Adafruit)

## Logiciels :

Unity : pour un environnement virtuel en 3D

Trello : pour organiser le projet

# Organisation du Projet



# Schéma-bloc

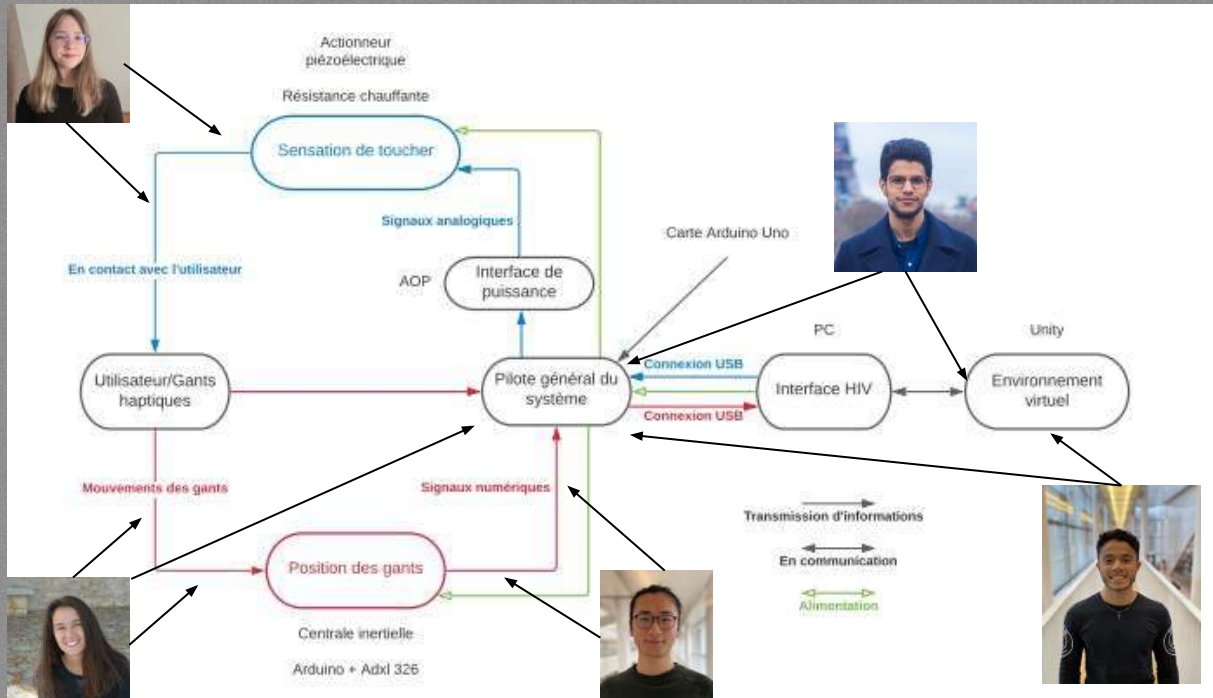


Schéma bloc du projet



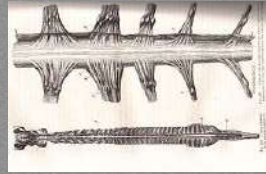
# Physiologie

## Système nerveux somatique

commande les mouvements et la position + permet la réception des sensations par la peau



cerveau



moelle épinière

muscles

neurones sensitifs



motoneurones centraux



motoneurones périphériques



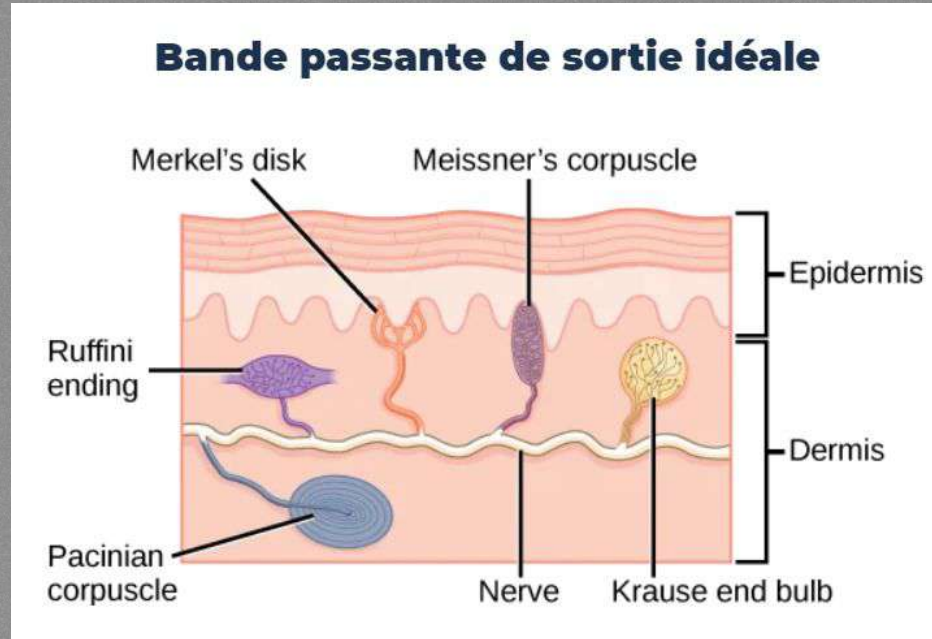
neurones sensitifs



corpuscules

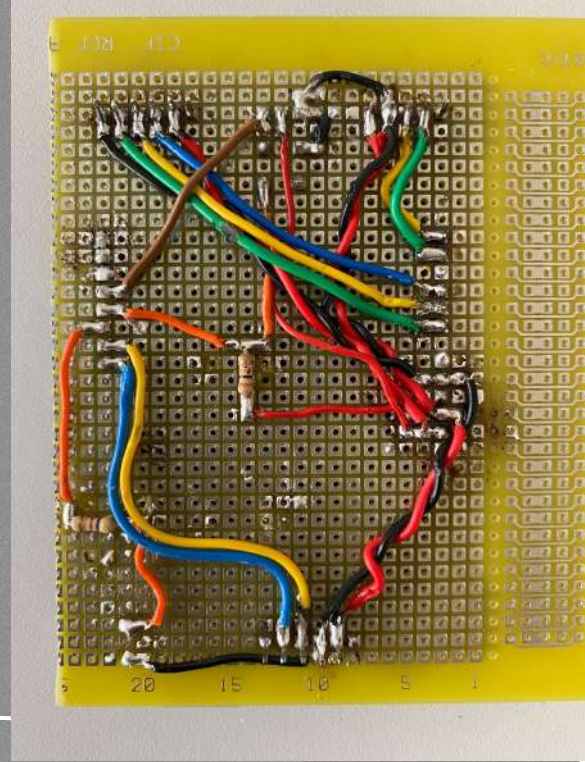
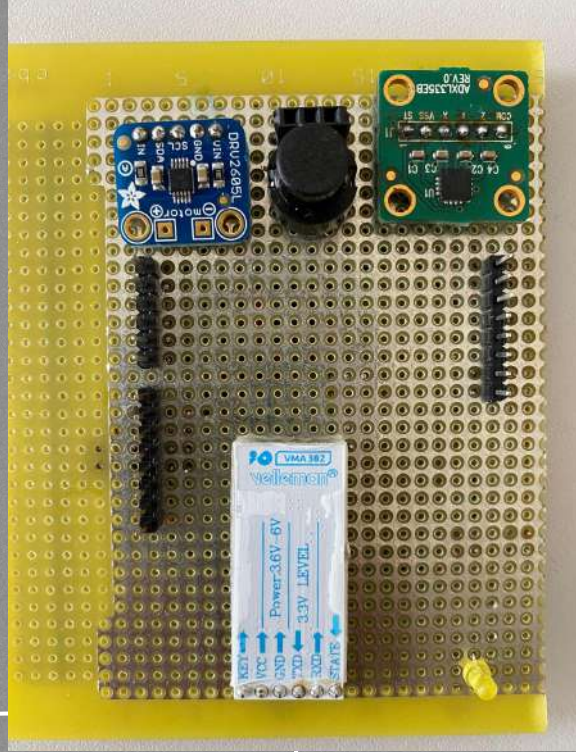
Les flèches représentent l'acheminement du message par des cellules nerveuses

# Les récepteurs du toucher dans le doigt

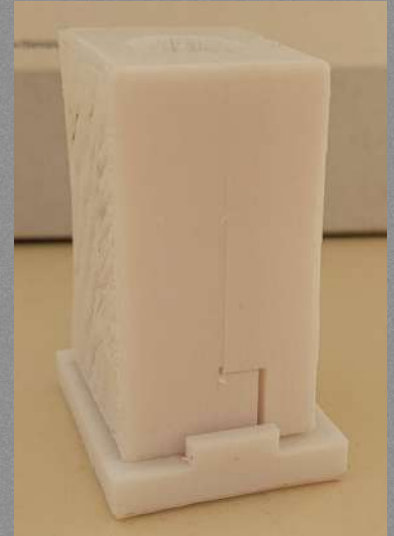
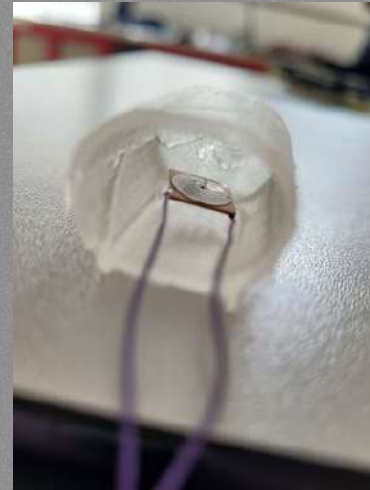




# Montage



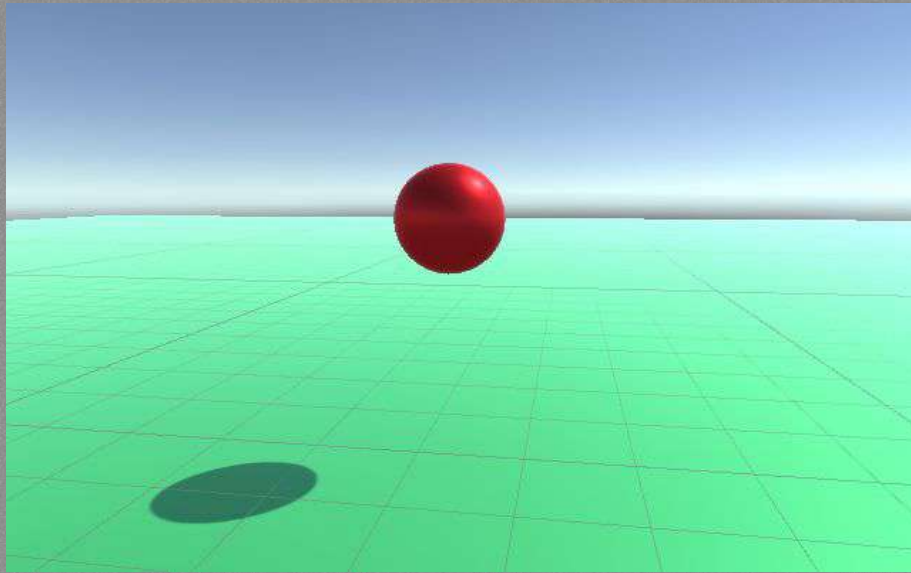
# Premier prototype : Vibration ressentie





# Partie Unity

## Reproduction d'un lâcher de balle





# Partie Unity

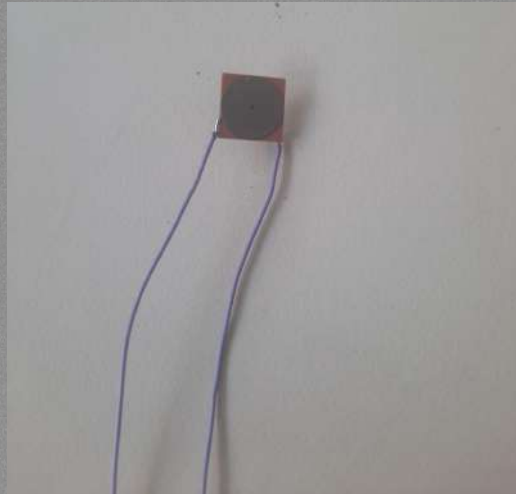
```
// Update is called once per frame
void Update()
{
    dist = Vector3.Distance(Sphere.position, Plane.position);
    if ( dist <= 0.6) {
        HapticOn();
        Debug.Log("Collision");
    }
    else {
        HapticOff();
    }

    Debug.Log(dist);
}
```

```
public void HapticOn()
{
    serial1.PortName = "COM3"; //A d
    serial1.Parity = Parity.None;
    serial1.BaudRate = 9600;
    serial1.DataBits = 8;
    serial1.StopBits = StopBits.One;
    serial1.Open();
    serial1.Write("2"); //Write(By
    serial1.Close();
}
```

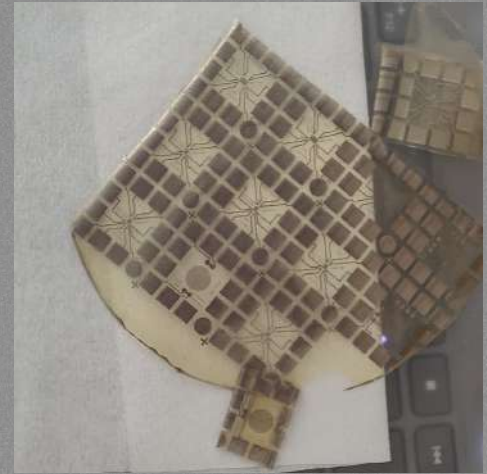
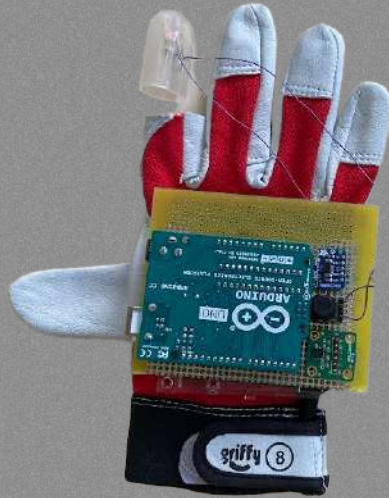
Calcul de la distance et envoi du string "2" lorsque la balle touche le sol

# Partie Arduino



```
void loop() {  
  //digitalWrite(broche2, HIGH);  
  if (Serial.available() > 0){  
    etat=Serial.read();  
    if(etat=='1'){  
      digitalWrite(broche2, LOW);  
    }  
    else if(etat=='2'){  
      digitalWrite(broche2, HIGH);  
    }  
  }  
}
```

# Deuxième prototype : Chaleur ressentie



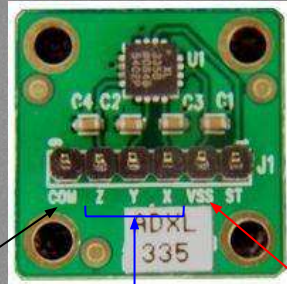


# Partie Arduino

## Le module ADXL335Z :

- Données selon 3 axes
- Accélération selon la gravitation terrestre
- Module adapté pour Arduino

- Position des gants
- Utilisation facile

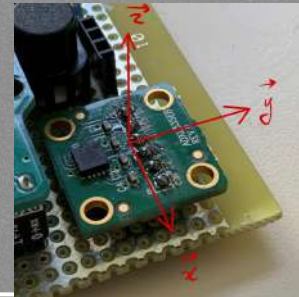


Masse

Outputs  
analogiques

Alimentation en  
5V

Schéma fonctionnel de l'ADXL335Z



# Partie Arduino



Bouton poussoir utilisé

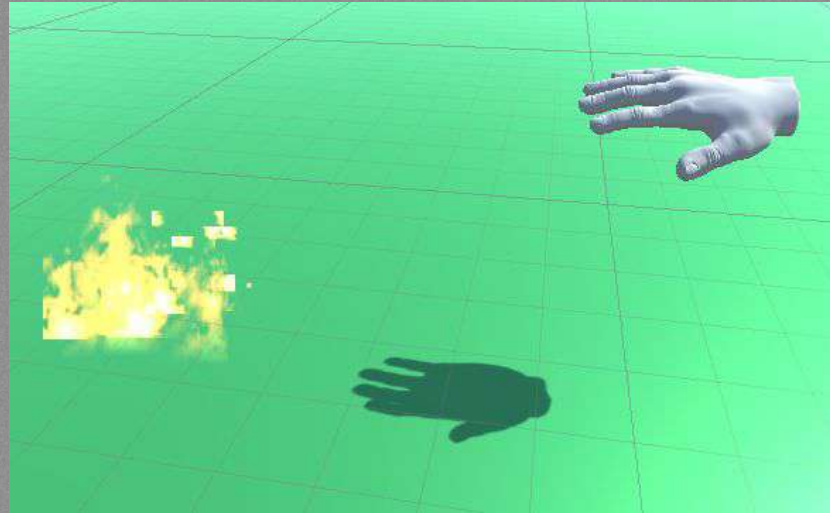
## Fonctionnalités :

- Réinitialise la position de la main sur Unity
- Annule la vitesse et le déplacement sur Arduino



# Partie Unity

## Reproduction d'une flamme





# Partie Unity

```
//Affichage des valeurs du déplacement
//Serial.print("Déplacement en x = ");
Serial.print(posx);

// print a tab between values:
Serial.print(",");

//Serial.print("Déplacement en y = ");
Serial.print(posy);

// print a tab between values:
Serial.print(",");

//Serial.print("Déplacement en z = ");
Serial.print(posz);

//Serial.print(",");
//Serial.print("\n");

Serial.println();
//Deploy before next reading
delay(10);
```

```
// Update is called once per frame
void Update()
{
    if (data_stream.IsOpen)
    {
        receivedstring = data_stream.ReadLine();

        string[] datas = receivedstring.Split(',');

        Debug.Log(receivedstring);

        float x = float.Parse(datas[0],CultureInfo.InvariantCulture);
        float y = float.Parse(datas[1],CultureInfo.InvariantCulture);
        float z = float.Parse(datas[2],CultureInfo.InvariantCulture);

        // Debug.Log(new Vector3(int.Parse(datas[0]),int.Parse(datas[1])

        transform.position = new Vector3(x, y, z);
```

Réception des données et attribution des valeurs aux axes x,y et z de la position de la main

# Partie Unity

```
// Update is called once per frame
void Update()
{
    dist = int.Parse(Vector3.Distance(Hand.position, Feu.position).ToString("0"));

    if ( dist <= 5) {
        Haptic1();
        Debug.Log("1");
    }
    else if (dist <= 10) {
        Haptic2();
        Debug.Log("2");
    }
    else if ( dist <= 15) {
        Haptic3();
        Debug.Log("3");
    }
    else if ( dist > 15) {
        Haptic4();
        Debug.Log("4");
    }
    Debug.Log(dist);
}
```

```
// Update is called once per frame
public void Haptic1()
{
    //A définir
    serial1.PortName = "COM3"; //A dé
    serial1.Parity = Parity.None;
    serial1.BaudRate = 9600;
    serial1.DataBits = 8;
    serial1.StopBits = StopBits.One;
    serial1.Open();
    serial1.Write("1"); //Write(Byte
    serial1.Close();
}
```

Envoi de la distance entre la main et la flamme

# Partie Unity

```
if (Serial.available() > 0){
  dist = Serial.read();
  if(dist == '4'){
    analogWrite(broche3, 0);
  }

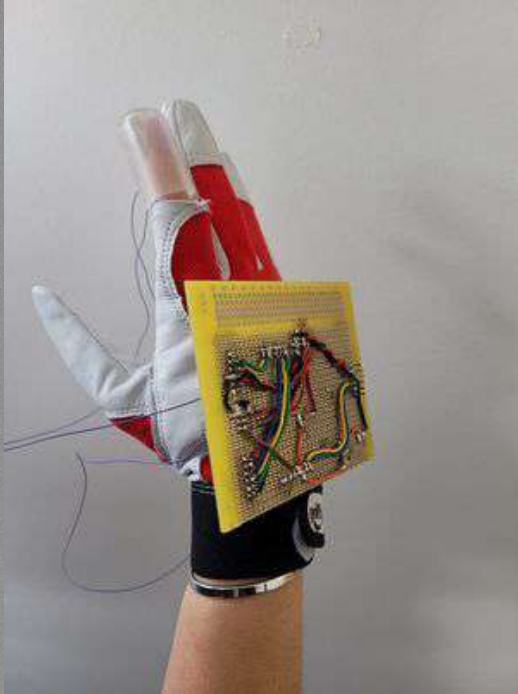
  else if(dist == '3'){
    analogWrite(broche3, 80);
  }

  else if(dist == '2'){
    analogWrite(broche3, 170);
  }
  else if(dist == '1'){
    analogWrite(broche3, 255);
  }
  //Serial.println(dist2);
}
```

Traitement de l'information par Arduino



# Bilan



→ Améliorations :

- 1) Pouvoir toucher la balle
- 2) Point de vue mobile
- 3) "Serious game"
- 4) 2 prototypes en même temps
- 5) Mobilité du gant, liaison Bluetooth

Merci !

