

LIVRABLE G: Prototype II et rétroaction du client

Abdelli, Mohamed Fadhel

Beaudoin, Nicolas

Clarke, Daniel

El Bitar, Rania

Mckay, Gabrielle

Le 12 Novembre 2020

Introduction:

La génération de la solution finale approche à grands pas. L'étape cruciale à laquelle nous sommes maintenant est celle du prototypage itératif et la rétroaction. Le premier prototype a été effectué à l'aide de l'outil TinkerCad et suite à la rencontre avec le client, nous avons pu avoir une rétroaction, qui nous permettra de développer notre deuxième prototype et optimiser notre solution. Ce deuxième prototype a pour but de recréer le plan virtuel du premier avec des composantes réelles afin de vérifier l'adaptation du concept dans le monde réel. Parallèlement, le prototype virtuel a été modifié pour inclure un écran LCD, comme le produit final devrait. Ceci permet de préparer le code pour l'inclusion de l'affichage dans le prototype III.

Prototype II:

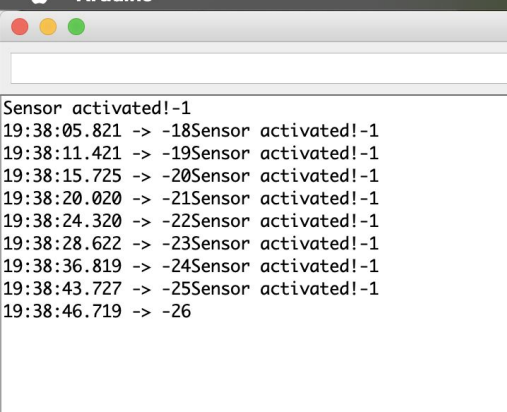
Ce deuxième prototype du projet consiste d'une carte arduino, deux DELs et deux senseurs infrarouges reliés ensemble par une plaque d'essais et des fils sans soudures. Le tout suit les plans du prototype I précédant réalisé dans Tinkercad. Le point de recréer le même prototype est d'isoler les variables qui différencient la simulation du prototype tridimensionnel. Ceci permet de constater les nouveaux problèmes qui montent à la surface, ceux qui restent les mêmes, et ceux qui disparaissent. Ceci permet aussi de se familiariser avec les caractéristiques auparavant inconnues des senseurs comme leur champ de vision, leur temps de réaction, etc. L'écran LCD ainsi que les autres méthodes d'affichage (haut-parleur, DELs) seront mis à l'épreuve plutôt dans le prototype III, car il n'y a aucun point de faire des essais sur l'affichage si la partie sensorielle du système ne peut pas être assez fiable pour isoler le comportement des autres composantes. Malgré cela, une version du prototype avec un écran d'affichage a été simulée dans Tinkercad. Le lien se trouve ci-dessous:

Lien pour simulation numérique du prototype avec écran LCD:

<https://www.tinkercad.com/things/awUTmh1eScl-copy-of-pir-motion-sensor-with-arduino-blocks/edit?sharecode=n4N10byeP66HOtCU-9KAfjQAo8GwMgCUnJX9yRMq5aA>

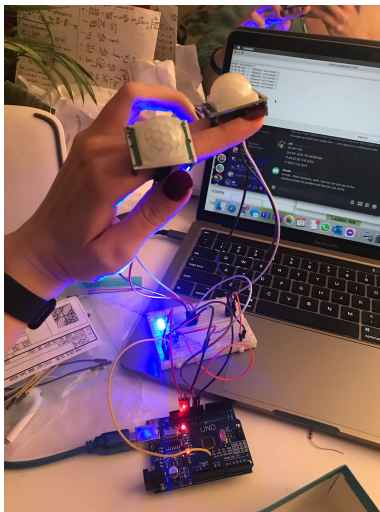
Photos du prototype:

Échantillon du Serial Monitor montrant l'activation des capteurs :



```
Sensor activated!-1
19:38:05.821 -> -18Sensor activated!-1
19:38:11.421 -> -19Sensor activated!-1
19:38:15.725 -> -20Sensor activated!-1
19:38:20.020 -> -21Sensor activated!-1
19:38:24.320 -> -22Sensor activated!-1
19:38:28.622 -> -23Sensor activated!-1
19:38:36.819 -> -24Sensor activated!-1
19:38:43.727 -> -25Sensor activated!-1
19:38:46.719 -> -26
```

Le prototype, activé :



Essais du prototype:

N° de Test	Objectif du Test (Pourquoi)	Description du prototype utilisé et de la méthode de test de base (Quoi)	Description des résultats à documenter et comment ces résultats seront utilisés (Comment)	Durée estimée du test et date prévue du début du Test (Quand)
1	Test de hauteur pour les senseurs - Pour déterminer si les senseurs peuvent efficacement détecter l'entrée d'une personne à plusieurs hauteurs.	<p>Type:...</p> <p>Les senseurs vont être placés sur la cadre d'une porte à diverses hauteurs à l'aide du "scotch tape". Nous allons commencer par 5 cm en ajoutant 10 cm à chaque fois pour déterminer la meilleure hauteur.</p> <p>Matériaux:</p> <ul style="list-style-type: none"> - Du Scotch tape 3\$ - Le prototype - Une porte 0\$ 	Les données vont être enregistrées grâce à un cahier de notes qui va contenir chaque hauteur et la performance respective des senseurs. Ces valeurs sont importantes pour notre projet pour nous permettre de déterminer la meilleure hauteur des senseurs.	Il devrait prendre 10 minutes.
2	Test de vitesse de passage des individus	<p>Type de prototype:...</p> <p>Comme il est difficile de déplacer un objet à une vitesse exacte. Nous allons passer notre main en avant des senseurs à deux vitesses approximatives. Une fois à une vitesse de marche et l'autre à une vitesse de course. Les résultats vont être enregistrés dans un cahier de notes.</p> <p>Matériaux:</p> <ul style="list-style-type: none"> - Le prototype 	Ce test nous permet d'apprendre comment le comptage des occupants pourrait être affecté par la vitesse des individus pensant par les senseurs.	Il devrait prendre 5 minutes.
3	Test de l'intervalle de temps entre le passage des individus	<p>Type de prototype:</p> <p>Les senseurs vont être montés sur la cadre d'une porte. Nous allons passé une main en avant des senseurs à plusieurs intervalles. Commenant par à chaque 5 secondes, et en enlevant une seconde à chaque essai jusqu'à une seconde.</p>	Pour but de déterminer si plusieurs individus peuvent entrer dans une salle en même temps.	Il devrait prendre 10 minutes.

4	Test de proximité au sensor	Le sensor va être attaché à une table grâce au “scotch tape” est nous allons placer une main en avant du sensor a plusieurs distances. Commenant par à 5 cm, et en ajoutant 10 cm à chaque essai jusqu'à 2 m.	Ceci va nous permettre de déterminer si notre prototype serait capable de détecter des entrées dans des portes doubles. Comme dans l'image suivant: https://bit.ly/3pkv1cl .	Il devrait prendre 20 minutes.
---	-----------------------------	---	---	--------------------------------

Rétroaction:

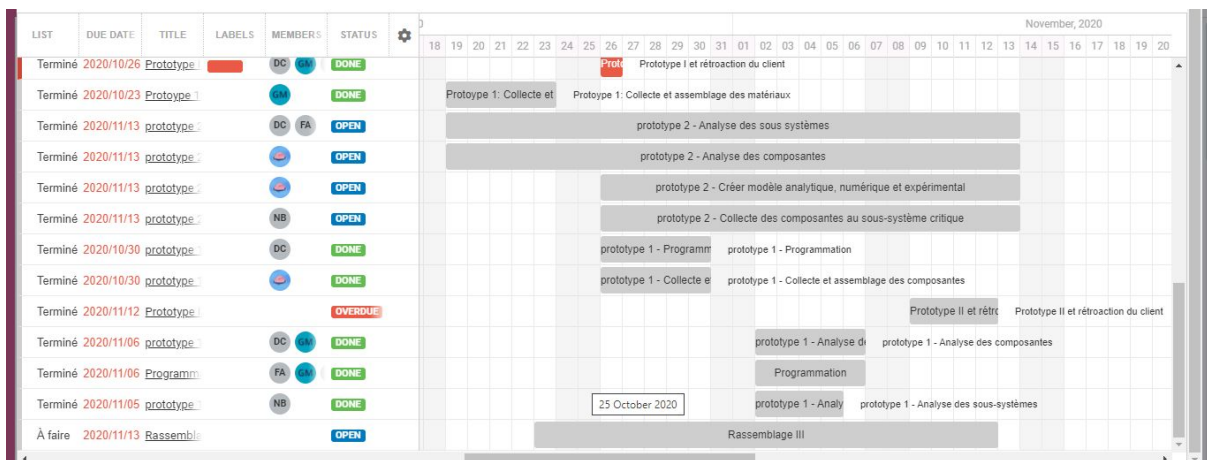
Après avoir présenté le prototype à quelques collègues étudiants, les critiques principales étaient les suivantes. D'abord, les utilisateurs test ont remarqué (tout comme l'équipe de conception) que le compte de personnes passantes donnait des erreurs lorsque deux personnes passaient à la suite de l'autre sur un trop petit intervalle de temps. Ceci indique une erreur logique dans le code du arduino (qui était connus à l'avance par l'équipe de conception mais demeurait à l'extérieur de leurs capacités de résolution). Cette source d'erreur devrait être étudiée davantage avant le prochain prototype afin de la corriger.

Deuxièmement, les étudiants ont remarqué que le sensor peut être activé simplement en passant sa main devant. Ceci pose problème car certains utilisateurs peuvent tricher le système s'il est placé à leur portée. L'équipe de conception est à la recherche de solutions.

Finalement, les utilisateurs tests ont presque unanimement mentionné l'apparence du prototype disant que l'allure n'est pas professionnelle avec les fils qui sortent librement de tout côté. Ceci est normal pour ce prototype et devrait être corrigé pour le résultat final avec un boîtier imprimé en 3D ou découpé au laser.

Du côté de l'équipe de conception, il a été remarqué que lorsque l'objet passe à grande vitesse (comme en passant rapidement une main ou possiblement une personne qui court), les senseurs ont de la difficulté à enregistrer le mouvement. L'application réelle de cette limitation reste à voire.

Diagramme de Gantt:



Conclusion:

Pour conclure, ce prototype constituait d'une réalisation matérielle du système de senseurs utilisé dans le premier prototype virtuel. Il a permis d'évaluer certaines mesures réelles comme sa hauteur de placement et sa capacité à détecter de véritables personnes. Le résultat final est adéquat pour ce stage du processus de conception, mais il reste une bonne charge de travail pour corriger les erreurs de programmation et ajouter le système d'affichage avant d'avoir le résultat final.

ANNEXE:

1- Copie du code arduino du prototype 2

```
//TIMEOUT can be tweaked easily for better real world performance
#define TIMEOUT 800

//include lcd library
#include <LiquidCrystal.h>

LiquidCrystal lcd(7,6,5,4,3,2);

//identifying variables
unsigned long sensor1TIME=0;
unsigned long sensor2TIME=0;
unsigned long timeout;
int count=0;

void setup()
{
  pinMode(9, INPUT);
  pinMode(10, INPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  Serial.begin(9600);

  lcd.begin(12,2);
  lcd.print("Nombre de pers.");
}

void loop()
{
  lcd.setCursor(0,1);

  //two if statement to make sure only one LED is on at a time
  if (digitalRead(13) == HIGH) {
    digitalWrite(12, LOW);
  }
  if (digitalRead(12) == HIGH) {
    digitalWrite(13, LOW);
  }
  //counts the time at which sensor is activated
  if (digitalRead(9) == HIGH){
    sensor1TIME=millis();
```

```

    timeout= sensor1TIME + TIMEOUT;
    digitalWrite(9, LOW);
}
//counts the time at which sensor is activated
if (digitalRead(10) == HIGH){
    sensor2TIME = millis();
    timeout= sensor2TIME + TIMEOUT;
    digitalWrite(10, LOW);
}
//determines which sensor went off first based on the timing and prints count
if (sensor1TIME > sensor2TIME && sensor2TIME!=0) {
    digitalWrite(12, HIGH);
    Serial.println("Sensor activated!+1");
    count=count+1;
    Serial.print(count);
    lcd.print(count);
    sensor1TIME=0;
    sensor2TIME=0;
    digitalWrite(9, LOW);
    digitalWrite(10,LOW);

}
else if (sensor2TIME > sensor1TIME && sensor1TIME!=0) {
    digitalWrite(13, HIGH);
    Serial.println("Sensor activated!-1");
    count=count-1;
    Serial.print(count);
    lcd.print(count);
    sensor1TIME=0;
    sensor2TIME=0;
    digitalWrite(9, LOW);
    digitalWrite(10,LOW);

}
//if both sensors are not activated, LEDs turn off
if (millis()>timeout && (sensor1TIME ||sensor2TIME)){
    digitalWrite(13, LOW);
    digitalWrite(12, LOW);
    sensor1TIME=0;
    sensor2TIME=0;
    digitalWrite(9, LOW);
    digitalWrite(10,LOW);
}
delay(100); // to improve simulation performance
}

```


2-Diagramme de Gantt

