```
# include <Servo.h>

# include <Wire.h>
# include "Adafruit_MPR121.h"

# ifndef _BV
# define _BV(bit) (1 << (bit))
# endif


Adafruit_MPR121 cap = Adafruit_MPR121();

// Keeps track of the last pins touched
// so we know when buttons are 'released'
uint16_t lasttouched = 0;
uint16_t currtouched = 0;

// créer un objet que l'on appelera myservo que l'on pourra
manipuler pour envoyer
// des informations à notre servo moteur
Servo myservo;

void setup() {

    Serial.begin(9600);

    while (!Serial) { // needed to keep leonardo/micro from starting
too fast!
        delay(10);
    }

    Serial.println("Adafruit MPR121 Capacitive Touch sensor test");

    // Default address is 0x5A, if tied to 3.3V its 0x5B
    // If tied to SDA its 0x5C and if SCL then 0x5D
    if (!cap.begin(0x5A)) {
        Serial.println("MPR121 not found, check wiring?");
        while (1);
    }
    Serial.println("MPR121 found!");
```

```
  pinMode(7, OUTPUT); // préciser que l'on veut utiliser la pin
6 comme une sortie
  myservo.attach(7);  // préciser à notre bibliothèque qu'un servo
est connecté à la pin 6
}
void loop() {


  // Get the currently touched pads
  currtouched = cap.touched();

  for (uint8_t i = 0; i < 12; i++) {
    // it if *is* touched and *wasnt* touched before, alert!
    if ((currtouched & _BV(i)) && !(lasttouched & _BV(i)) ) {
      Serial.print(i); Serial.println(" touched");
      if (i == 10) {


        // controle du mvt
        myservo.write(90); // mettre le servo en position de
repos
        myservo.write(0); // mettre le servo en buté à
gauche
        delay(1500);
        myservo.write(90);// mettre le servo en position de
repos
        myservo.write(180); // mettre le servo en buté à
droite


      }
    }
    // if it *was* touched and now *isnt*, alert!
    if (!(currtouched & _BV(i)) && (lasttouched & _BV(i)) ) {
      Serial.print(i); Serial.println(" released");
    }
  }

  // reset our state
  lasttouched = currtouched;


}
```