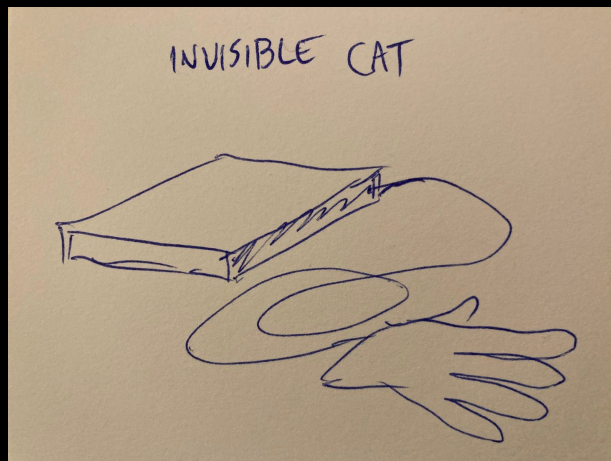


THE INVISIBLE CAT
CART360 Prototype
Martin Hanses

My original proposal was for an artifact that was intended to empower and assist the user in overcoming certain physical or mental challenges, by emitting certain types of haptic and auditory feedback to facilitate a particular response. It was meant to be worn on the shoulder, strapped tightly against the body, and be activated by touch (pressing one's hand against the shoulder).

As time passed and I mulled over the idea, I came to the realisation that there was a very high chance that despite my best intentions, an object like this would very quickly become annoying to wear - most likely even feel condescending, like an overbearing caretaker constantly checking if you're alright. Testing it out was equally disappointing - the main draw, activation by touching one's own shoulder, felt flat and boring.

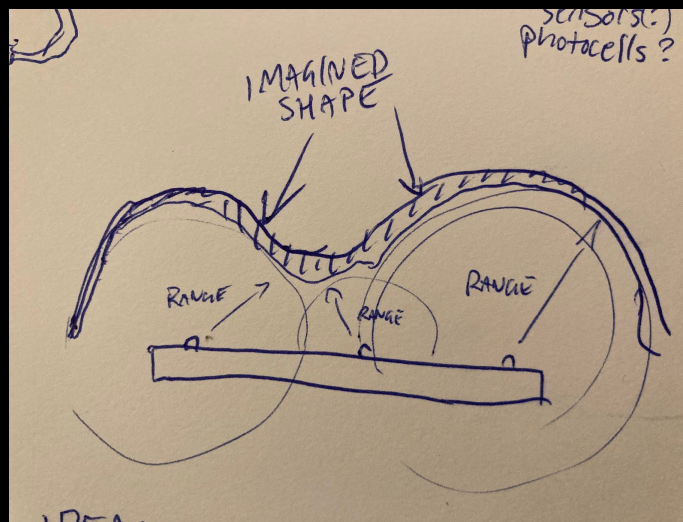
The one thing that kept resonating with me, however, was working with physical and haptic feedback. I'd already purchased small vibration motors for use in the original idea, so using them as well as all the other sensors I'd prepared, I started playing around with different combinations and eventually, I felt like I had narrowed down the scope and idea into something that felt interesting. Feeling your own shoulder isn't very interesting to feeling something that *isn't there*. I changed the name of the project from "The Jiminy Cricket" to "Invisible Cat."



The main goal for Invisible Cat was for the user to be able to feel an object that didn't exist; as if they were petting a spectral cat. Using photocells and small vibration motors, the first plan was to build an object that senses light in a small loaf/kitten-sized area, and when the user's hand was close enough, would cause the motors to vibrate and hopefully sell the illusion that the user was touching something.

The first test was built with three photocells, three vibration motors, and three LEDs - the LEDs were not intended to be used in the final artifact, but mostly worked as troubleshooters.

Whenever the hand was close enough, they would light up, and the vibration motors would begin to rumble. It was fine, I suppose - but it essentially just felt like a proximity sensor. If one pressed one's hand against the photocells, the effect was the same. The illusion of the object was ruined when I was actively avoiding it, as I intuitively was moving my hand in and out of range of the photocells. The fun part was feeling the vibrations, not trying to avoid them. That meant the first fix was to create a threshold of space where the hand was close enough, but not too close - this immediately changed the experience. Now, the motors would vibrate whenever the hand became the proverbial Goldilocks, and since they weren't attached to anything at this point, they would rattle against the table. Almost like a cat purring.



Playing around with this was honestly a lot of fun, but eventually, the novelty wore off. As I got used to the sensation, I started feeling like I wasn't touching the object, and it again felt like a proximity sensor. Like I mentioned, the first idea I proposed hinged on the use of pressure sensors, and the experience of using them was underwhelming to say the least. While testing them out and playing around with them, however, I found that they reacted to not only being pressed, but also being bent. Realising this, I placed the sensor in the palm of my hand, so it would react whenever gripping. This worked surprisingly well (after some minor tuning) and the sensor even registered the tightness of my grip.

This led to a very obvious new part of the object: a glove. On the glove, I'd attach all three vibration motors (spread across the hand - one at the base of the thumb, index finger and pinky) as well as the pressure sensor (from the base of the hand to the tip of the middle finger.)

Using the input from this MacGyvered pressure sensor and the new cyberglove, I stopped all direct proximity-based vibrations in my code and instead connected them to the opening/closing of the hand. If the user's hand is in the right place, and they close it, the motors go off, giving the impression that their hand is closing on something. Testing this out, I felt it made a world of difference. I found myself spending way longer than I should have, feeling the air and closing my eyes, imagining the shape of the object. True to the new project name, my favorite new pastime became stroking an invisible cat from left to right.

Petting my new non-existent companion, I drifted off into Blofeld-esque contemplation about what the artifact exactly was, and how I could improve it. My original proposal, the Jiminy Cricket, was supposed to be a self-help tool, and the Invisible Cat seemed, at first glance, to be a distraction. The more I thought, and as I realised I was still petting the damn thing, I started considering the Invisible Cat as a sculpture. Due to the usage of photocells, and the algorithm that calculated the “size” of the object, it was never exactly the same. Every time the Arduino was reset, I’d have to start over, exploring and *feeling the air*. Is it this high? Is it this far left/right? Where is it? The Invisible Cat was an invisible piece of art, and it existed only through the exploration of the user’s hand and the user’s mental image of the object. The Invisible Cat might become a counterintuitive name - especially if the sculpture is abstract - as it gives the user/viewer/feeler(?) a preconceived idea of what the object is. Maybe there is more value in letting them imagine it for themselves?

In the end, however, it was still just a prototype. I was excited about what I had built - even though both code and circuit were fairly simple - and I figured the most sensible thing to do was to have other people test the sculpture and/or give their thoughts on how it would work. While it was clear that I was now building a sculpture, the exact specifics of the sculpture were still unclear. A few things stuck with me, the first of which was the sensors.



The sensors I used in the glove and object were intended for something else, and could most likely be improved. The pressure sensor should most definitely be swapped out with a flex sensor, which should be a very easy fix. The photocells are a little more difficult, however - although some sort of proximity sensors might be more accurate and precise, they lose some of the appeal of the volatility of the photocells and their input - my experience with the sculpture changing over the course of a day as the sun went down. If the artifact was displayed in a museum, it would most likely be under fairly controlled circumstances. Maybe the best way to go is a combination of the two, or some randomization in the code? Then again, that means missing out on the appeal of the surrounding world controlling the shape.

The second thing I thought was about the shared experience - the artifact is incredibly boring for the non-user or spectator. It's an incredibly physical and explorative experience for the one with the glove, but for anyone else, it looks like someone waving their hand around. Perhaps there should be some visual representation for anyone not wearing the glove? This isn't a high priority, in my opinion, but is worth considering.

Thirdly, transparency and the wires. Although a final, evolved version of the build as it is now would be possible, the wired glove does break the illusion somewhat. Going wireless would definitely make for a more boundlessly explorative experience. The other aspect of course is the base of the item itself - photocells and sensors don't really sell the idea of a non-existing sculpture, so in the final version, the item containing the sensors should give off an empty feel. I toyed with the idea of having the boundaries of the imaginary object only exist in the glove, but I really love the idea of an empty pedestal or diorama where it's up to the viewer/feeler to find the art.

All in all, I'm very happy about the state of the prototype, although it isn't very much to look at right now. Unfortunately, I didn't receive my tools in time so it all looks kinda haphazardly put together, but for a prototype, I'm happy to overlook aesthetics for function.

CODE USED IN THE FINAL VERSION:

```
#define PRESSURE_PIN A0
#define VIBE_PIN1 5
#define VIBE_PIN2 6
#define VIBE_PIN3 7
#define PHOTO_PIN1 A1
#define PHOTO_PIN2 A2
#define PHOTO_PIN3 A3

#define LED3 2
#define LED2 3
#define LED1 4

//define the variables for the rangefinding
int upperThreshold = 0;
int margin = 50;
int lowerThreshold = upperThreshold - margin;
//define value for noise and involuntary input on the pressure sensor
int pressureThreshold = 400;
```

```

void setup() {
  Serial.begin(57600);
  while(!Serial);
  // define the output modes on all vibration motors
  pinMode(VIBE_PIN1, OUTPUT);
  pinMode(VIBE_PIN2, OUTPUT);
  pinMode(VIBE_PIN3, OUTPUT);
  setupPhotoDist();
}

void setupPhotoDist(){
  Serial.println(analogRead(PHOTO_PIN1));
  upperThreshold = (analogRead(PHOTO_PIN1) - (analogRead(PHOTO_PIN1)/8) );

  lowerThreshold = (upperThreshold - margin);
  Serial.println(upperThreshold);

}

void loop() {

  pinCheck(PHOTO_PIN1, LED1);
  pinCheck(PHOTO_PIN2, LED2);
  pinCheck(PHOTO_PIN3, LED3);

  vibration();

}

void vibration(){

  while(analogRead(PRESSURE_PIN) > pressureThreshold){

    if(analogRead(PHOTO_PIN2) < upperThreshold && analogRead(PHOTO_PIN2) >
lowerThreshold) {digitalWrite(VIBE_PIN2, HIGH);} else {digitalWrite(VIBE_PIN2, LOW);}
    if(analogRead(PHOTO_PIN1) < upperThreshold && analogRead(PHOTO_PIN1) >
lowerThreshold) {digitalWrite(VIBE_PIN1, HIGH);} else {digitalWrite(VIBE_PIN1, LOW);}
  }
}

```

```
    if(analogRead(PHOTO_PIN3) < upperThreshold && analogRead(PHOTO_PIN3) >
lowerThreshold) {digitalWrite(VIBE_PIN3, HIGH);} else {digitalWrite(VIBE_PIN3, LOW);}
    }
```

```
    if(analogRead(PRESSURE_PIN) < pressureThreshold){
        digitalWrite(VIBE_PIN1, LOW);
        digitalWrite(VIBE_PIN2, LOW);
        digitalWrite(VIBE_PIN3, LOW);
    }

}
```

```
void pinCheck(int photoPin, int LED) {

    if(analogRead(photoPin) < upperThreshold && analogRead(photoPin) > lowerThreshold)
{
    analogWrite(LED, 255);

    } else { analogWrite(LED, 0);

    }

}
```