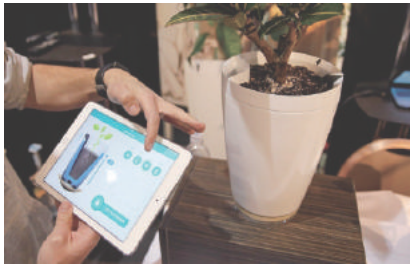
# INSPIRATION

养花现在已经成为很多人的一项爱好。但有些刚开始养花的初学者，经常会因为照顾不当而使花死亡。当我们发现这一问题时，便将课题范围聚焦到养花上，希望能做一款辅助养花的机器人。

# COMPETITIVE ANALYSIS

我们最先调研了一些其他与养花有关的智能产品，做了竞品分析，目的是希望能从中找到隐藏的问题以及可改进的点。

## Parrot Pot



Parrot Pot 可以监测植物的生长环境，包括光照、温湿度和土壤肥沃度。这些数据都会显示在手机上。用户可以通过手机应用来一键浇花。

### SHORTCOMINGS

不支持 WiFi，只要出门走远，Parrot Pot 就失联了。

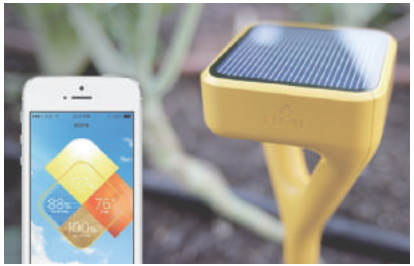
## Inbox

它内置的光感应器能发现弱光环境，并利用顶部的灯进行补光。另外，Inbox 的内置芯片能监测植物并生成一份植物日志。用户可以对比植物在不同时间段的状况。

### SHORTCOMINGS

不支持自动浇花，懒人们不得不亲自动手。

## Edyn



Edyn起检测和提醒的作用，告诉用户植物的营养情况、光照，以及温度等。如果有需要你为植物做点什么，你的手机会及时收到提醒。并支持WiFi。

### SHORTCOMINGS

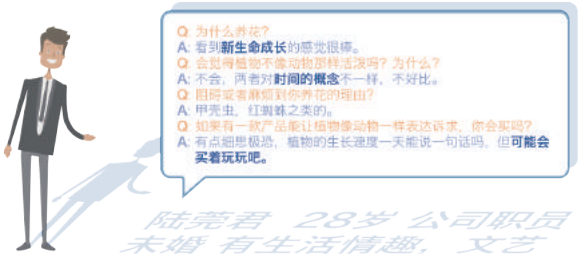
而并不会剥夺你实际操作养花的动作，也无法自动浇水或施肥。

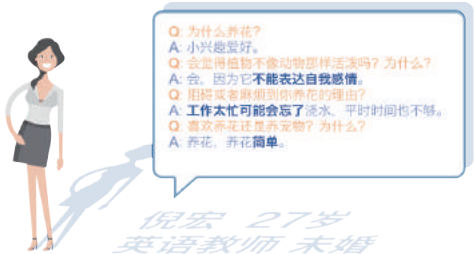
# SUPPOSTION

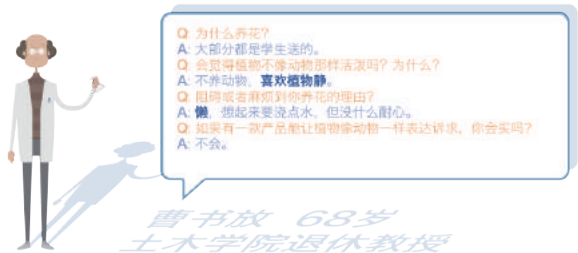
我们发现目前的所有产品都只是把养花更智能化，但我们认为人们养花更多是为了情感的需求，而现有的产品很可能反而降低了用户亲自养花所带来的成就感和喜悦。

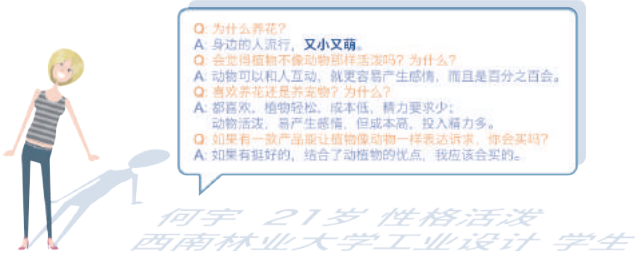
# INTERVIEW

为了验证我们的推测是否真的符合用户需求，我们对身边8位有养花经验的人做了采访。下面是其中一部分。



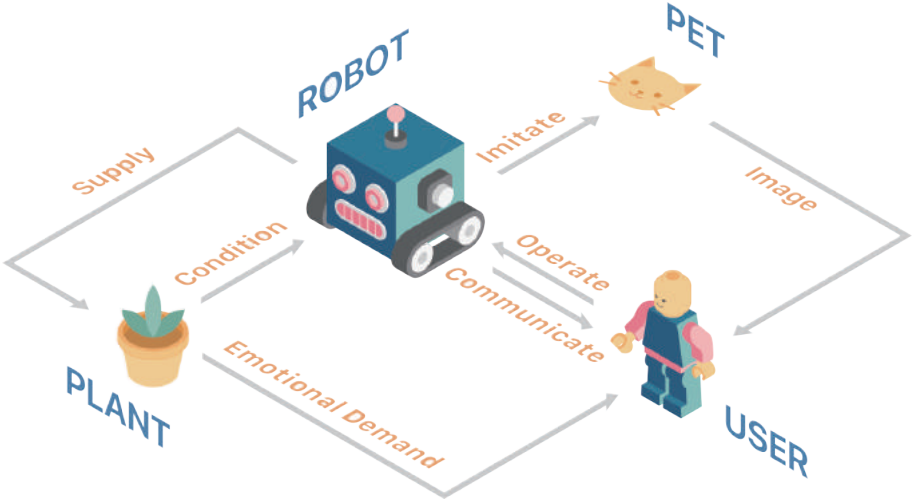
# PAIN POINT

1. 缺少相关知识或工作太忙会忘了照顾。
2. 植物太静没有及时的反馈信息，不像动物那么活泼。
3. 现有的智能花盆会降低养花本身的乐趣。

# CONCEPT

**“把植物当动物养”**

通过访谈总结的痛点，我们想到将智能花盆和电子宠物相结合，这不仅仅赋予了机器人本身生命，还能将植物无声的情感表现出来。我们尝试让机器人模仿宠物的行为特点，从而让用户与植物之间有更多交互的可能性。

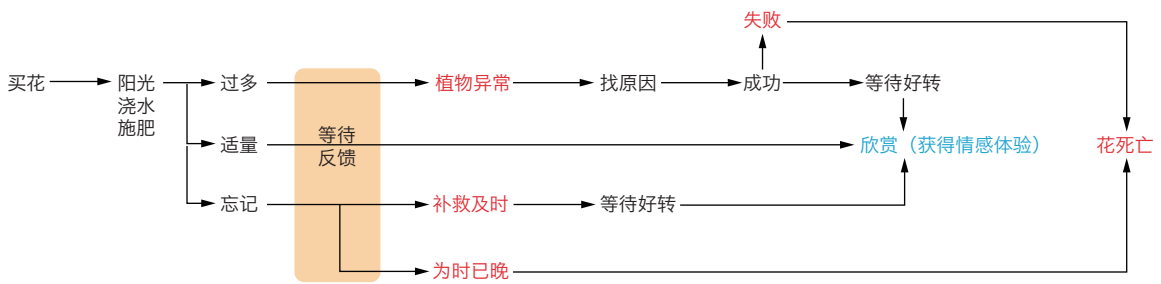


# CUSTOMER JOURNEY MAP

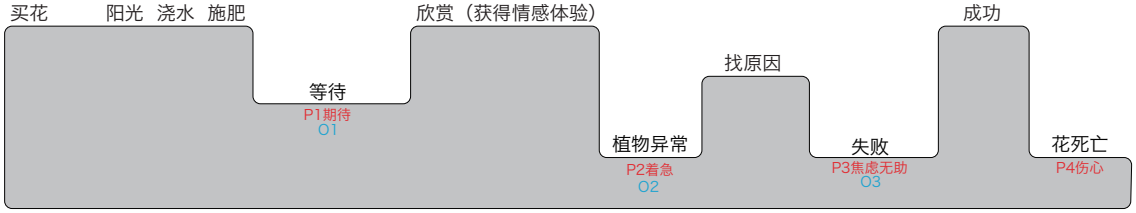
## 用户需求：

欣赏花　按时按量浇水　按时按量施肥　给花除虫　给花晒太阳　情感需求

## 行为：




## 情绪体验：



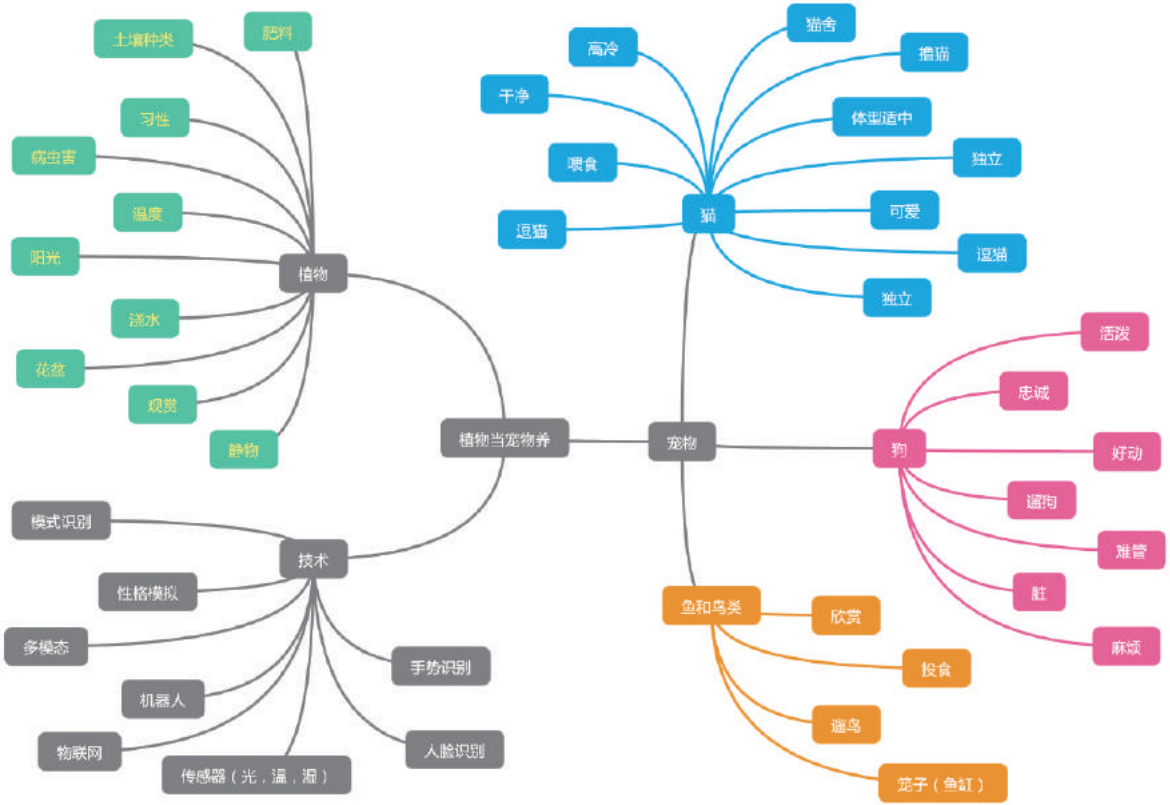

## 切入点：

P1植物反馈时间过长，易让人着急，且不知道照顾是否合适
P2过度照顾导致植物异常或疏于照顾导致植物异常
P3寻找解决办法时不知道如何筛选

## 机会：

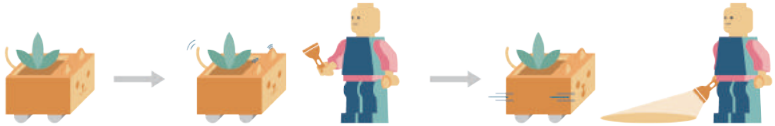
O1通过机器人辅助来放大植物的反馈
O2通过数据和机器人的反馈来矫正照顾度
O3针对植物的生理数据来制定植物异常的对策，或通过O2来避免异常

# MIND MAP


植物当宠物养
植物
土壤种类
肥料
习性
病虫害
温度
阳光
浇水
花盆
观赏
静物
技术
模式识别
性格模拟
多模态
机器人
物联网
传感器（光，温，湿）
手势识别
人脸识别
宠物
猫
猫舍
撸猫
体型适中
独立
可爱
逗猫
独立
逗猫
喂食
干净
高冷
狗
活泼
忠诚
好动
遛狗
难管
脏
麻烦
鱼和鸟类
欣赏
投食
遛鸟
笼子（鱼缸）

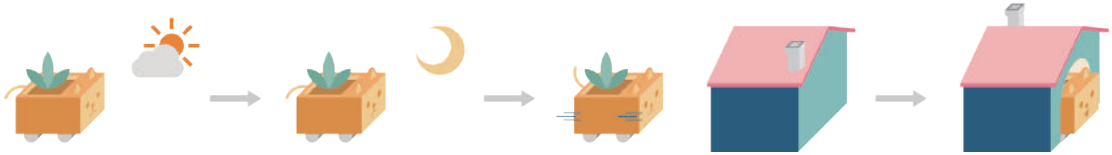
# FLOWCHART

缺水时　把盆叼到用户面前　把盆放下，用户给它倒水　用户把盆放到机器人嘴边　喝完自动把盆叼走



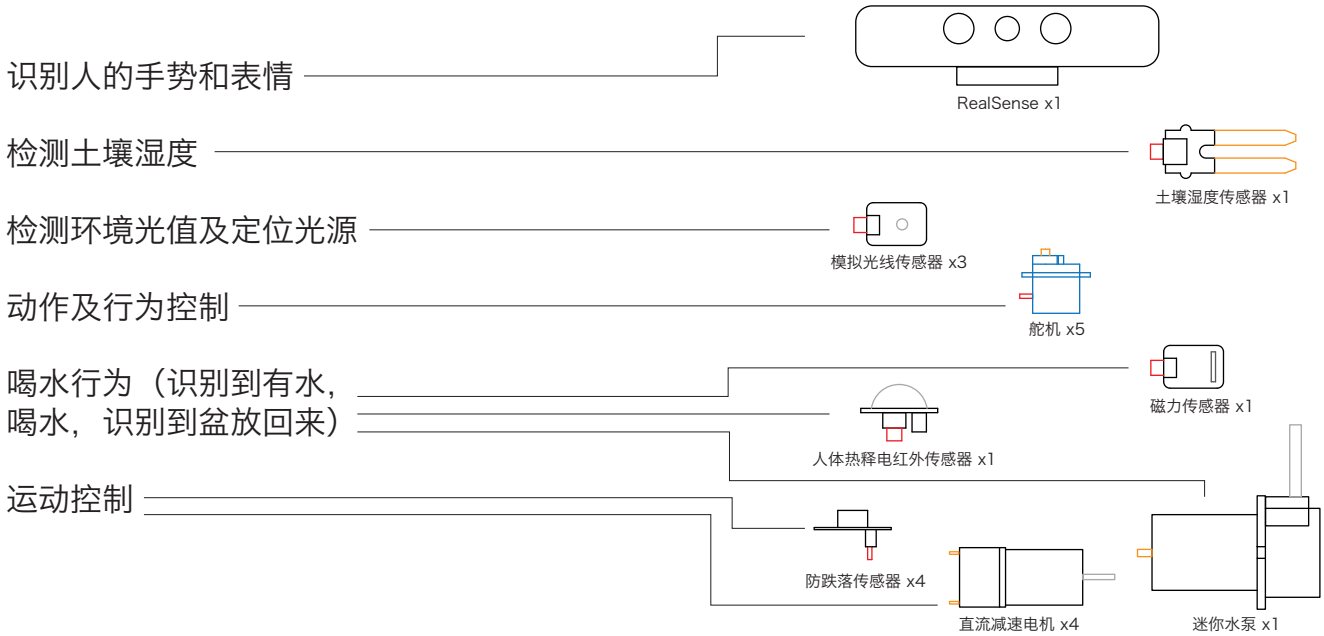
正常时　用户可以拿手电筒逗它　它会追着有光的方向跑去

缺光时　自动跑到阳光充足的地方　晒够或阳光太强时　自动跑到阴凉处

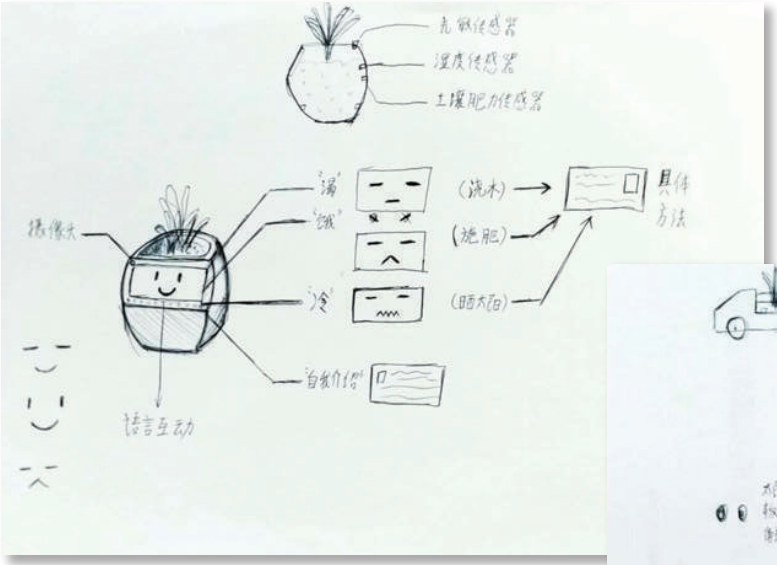


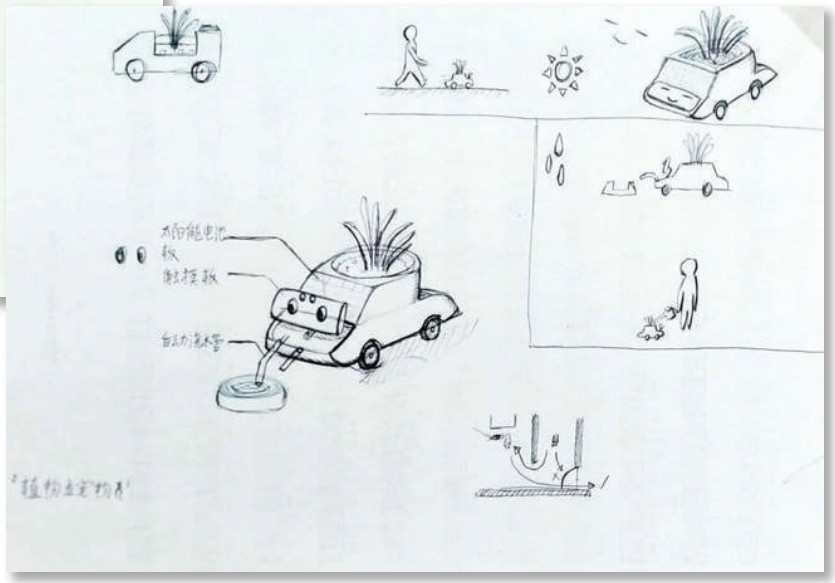
阴天时　到晚上累积光照也不足　跑回自己的小房子里　小房子里能补充光照并充电

# FLOWCHART


识别人的手势和表情
RealSense x1
检测土壤湿度
土壤湿度传感器 x1
检测环境光值及定位光源
模拟光线传感器 x3
动作及行为控制
舵机 x5
喝水行为（识别到有水，
喝水，识别到盆放回来）
磁力传感器 x1
人体热释电红外传感器 x1
运动控制
防跌落传感器 x4
直流减速电机 x4
迷你水泵 x1

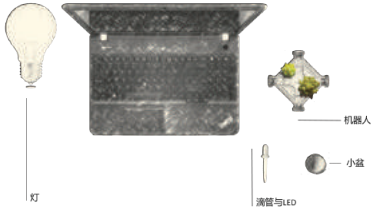
# SKETCH


光敏传感器
湿度传感器
土壤肥力传感器
'渴'
'饿'
'冷'
(浇水)
(施肥)
(晒太阳)
具体方法
语言互动


太阳能电池板
激控装置
自动浇水管

# PAPER PROTOTYPE

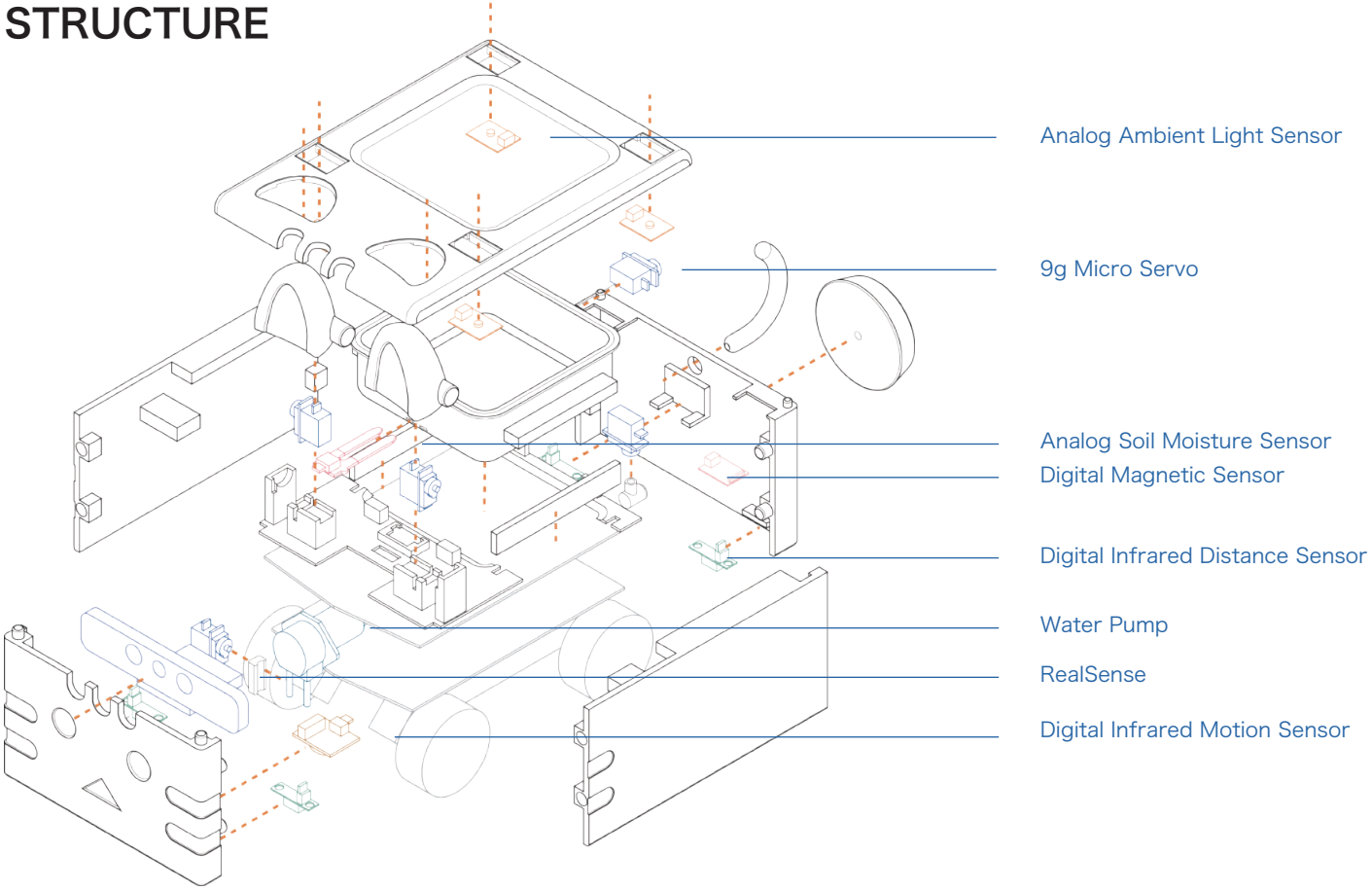
在有了初步方案后，我们制作了一些纸原型并邀请了三个用户来体验，根据他们的反馈我们总结出了以下几点，并进行了一些修正。



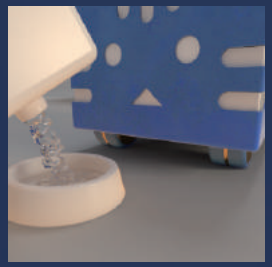
情景一：植物缺水模拟
用滴管给它滴水时，它会自动跟着滴管的前端，这样更像宠物
用小盆子倒担心倒在桌子上
希望能够加上一些无意识的动作，这样更加有生命感
如果每次发出指令得到的回馈都一样，感觉会比较单调

情景二：植物缺光模拟
晚上开一宿灯给植物补充阳光感觉影响休息
灯光对人眼会不会有伤害，感觉会晃眼
形态不要是花盆和花的关系，可以更像一个整体
可以像激光笔逗猫一样去逗它

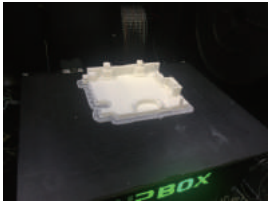
# STRUCTURE


Analog Ambient Light Sensor
9g Micro Servo
Analog Soil Moisture Sensor
Digital Magnetic Sensor
Digital Infrared Distance Sensor
Water Pump
RealSense
Digital Infrared Motion Sensor

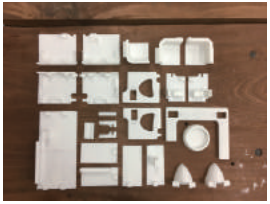
# VISUAL DESIGN



# DEMO



3D打印

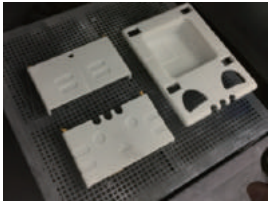


全部零件

胶水粘合

腻子打磨

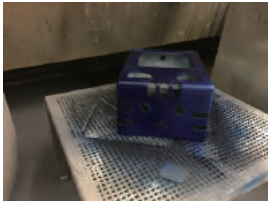
初步组装



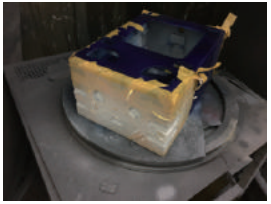
喷水补土

喷白底漆

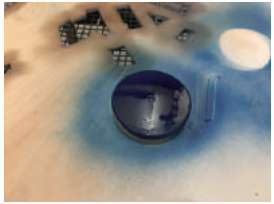
贴遮盖带



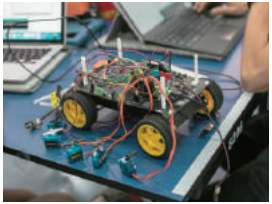
喷蓝色漆



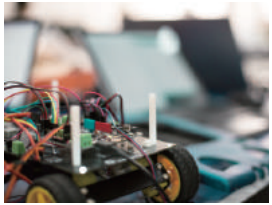
喷白色漆



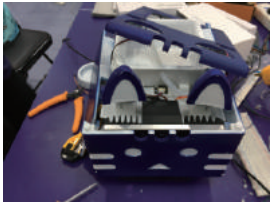
喷保护漆



连传感器



修改调试



结构组装

组装完成

# CODE α

由于系统复杂，而且需要大量调试，我们大量用变量替代数值，并放在开头，在程序中用少数变量和算法来替代大量数值，以此来降低调试难度。为了尽可能地降低一天中不同时刻对初始光值的影响，在不被挑逗的状态下一定时间后会重新测量环境光值并初始化，为了提高测量精度采用了测多次并取平均值的做法。触发挑逗程序后，通过光敏传感器阵列之间的光值增量差来控制小车的运动状态。

```
//舵机参数
int earlPosRange=20;        //左耳朵舵机的运动范围角度
int earlPosSet=150;         //左耳朵舵机的初始角度
int ear2PosRange=20;        //右耳朵舵机的运动范围角度
int ear2PosSet=20;          //右耳朵舵机的初始角度
int tailPosRange=90;        //尾巴舵机的初始角度
int tailPosSet=60;          //尾巴舵机的运动范围角度
int mouthPosSet=160;        //嘴舵机的初始角度
int mouthPosRange=90;       //嘴舵机的运动范围角度
int magnetPosSet=0;         //磁铁舵机的初始角度
int magnetPosRange=90;      //磁铁舵机的运动范围角度

int magnetPos=0;                        //磁铁舵机的角度变量
int earlPos=earlPosSet-earlPosRange;    //左耳朵舵机的角度变量
int ear2Pos=ear2PosSet+ear2PosRange;    //右耳朵舵机的角度变量
int tailPos=tailPosSet-tailPosRange;    //尾巴舵机的角度变量
int mouthPos=0;                         //嘴舵机的角度变量

int earSlow=30;        //耳朵角度书写时间间隔（慢）
int earNormal=20;      //耳朵角度书写时间间隔（正常）
int earFast=10;        //耳朵角度书写时间间隔（快）
int tailSlow=30;       //尾巴角度书写时间间隔（慢）
int tailNormal=20;     //尾巴角度书写时间间隔（正常）
int tailFast=10;       //尾巴角度书写时间间隔（快）
int timeInterval=20;   //磁铁和嘴书写时间间隔

int count=20;    //角度顺次计数
int i=0;         //计数器
```

舵机组运动控制调试变量

```
void resetLight(){//每隔一段时间重置环境光值
  if(state==1){
    time0= millis();
    state=0;
  }
  time1=millis()-time0;
//  Serial.print("time0=");
//  Serial.print(time0);
//  Serial.print("time1=");
//  Serial.print(time1);
//  Serial.println();
  if(time1>period){
      for(int i=0;i<10;i++){
      lgS10=analogRead(lgSP1)+lgS10;
      lgS20=analogRead(lgSP2)+lgS20;
      lgS30=analogRead(lgSP3)+lgS30;
      delay(10);
      }
      lgS10=lgS10/10;
      lgS20=lgS20/10;
      lgS30=lgS30/10;
      state=1;
      print0();
      Serial.println(" success");
    }
}
```

环境光定时初始化函数

```
//**********************主程序**********************

void setup() {//设置pinMode，环境光初始化，舵机初始化等

void loop() {//根据土壤湿度的变化来进行各个动作的判断

//**********************函数**********************

//————光感引部分————
void resetLight()[//每隔一段时间重置环境光值
void print0()[//打印重置环境光值
void printLR()[//打印左右环境光值差
void printFB()[//打印前后环境光值差
void swerve()[//转向判断函数
void walk()[//前进判断函数

//————小车前进函数————
void carStop()[                    //  Motor Stop
void carAdvance(int leftSpeed, int rightSpeed)[          //Move FORWARD
void carBack(int leftSpeed, int rightSpeed)[        //Move BACKward
void carTurnLeft(int leftSpeed, int rightSpeed)[        //Turn Left
void carTurnRight(int leftSpeed, int rightSpeed)[        //Turn Right

//————动作流程————//
//放音动作
void putpot()[//耳朵+尾巴+磁铁舵机
//喝水动作
void drink()[//放管子，开水泵，持续，收管子

//————神态控制————//
void shaking(int earTimeInterval, int tailTimeInterval)[//基本摇动函数，后面的状态调用函数调参数
void hungery()[//饥饿状态摇动参数较慢
void happy()[//开心状态摇动参数正常
void glut()[//吃撑状态摇动参数急促
void waitingDrink()[//等待喝水摇动参数停顿
void waitingPot()[//等待放盆摇动参数停顿
void change()[//状态切换摇动参数恢复

//————realsense————
int distance()  //realsense测距函数
int kinaValue()  //realsense杂音测量函数
```

舵机运动控制算法

```
while(i<count){
    earlPos=earlPos+earlPosRange/count;
    earlSv.write(earlPos);
    delay(earTimeInterval);
```

光敏控制算法

```
  while(lgS1>threshold||lgS2>threshold||lgS3>threshold){
    delay(100);

    //
    lgS1=analogRead(lgSP1)-lgS10;   //左光值增量
    lgS2=analogRead(lgSP2)-lgS20;   //右光值增量
```

主程序及所用函数

# CODE *β*

## 参数设置

```
#include <ArduinoJson.h>
#include <Servo.h>
#include <Metro.h>

String json = "";

//多线程设置
Metro sweepServo = Metro(20);
Metro eargearMetro = Metro(20);
Metro eargearMetro01 = Metro(20);
Metro eargearMetro02 = Metro(20);
Metro tailgearMetro = Metro(20);

int pos01 = 30;
int pos02 = 30;
int pos = 30;

//控制小车
int speedPin_M1 = 5;     //M1 Speed Control
int speedPin_M2 = 6;     //M2 Speed Control
int directionPin_M1 = 4;     //M1 Direction Control
int directionPin_M2 = 7;     //M1 Direction Control
unsigned long actualDistance = 10;

Servo myservo;  // create servo object to control a servo

//数字防跌落传感器
//设置数字防跌落传感器的pin脚，需要看机子调整
int sen03 = 2; //车子右上角，第一象限
int sen01 = 3; //车子左上角，第二象限

//尾巴防跌落
```

## 整体线程

```
void setup() {                                    // Serial initialization

  //script02  找到人的代码
void findPeople() {

  //script04  放送
void putput() {

  //script05  喝水
void drink() {

int distance() {

void loop() {

void carStop() {                  //  Motor Stop

void carAdvance(int leftSpeed, int rightSpeed) {         //Move FORkward

void carBack(int leftSpeed, int rightSpeed) {       //Move BACKrward

void carTurnLeft(int leftSpeed, int rightSpeed) {     //Turn Left

void carTurnRight(int leftSpeed, int rightSpeed) {     //Turn Right

  //车转身180度
void carTurnBack() {
```

## 主函数

```
void loop() {

  unsigned long timer = millis();

  if (measureDistance.check() == 1) {

  //script-1
  //土壤湿度传感器和耳朵尾巴的互动

  if (analogRead(moisture) < 300) {

    Serial.println(actualDistance);
    //耳朵角度快速运动

    if (eargearMetro01.check() == 1) {
  } else {

  //小车四个角的防跌落

  //车子右上角，digitalRead(8)表示pin脚读取的是二进制的值，1为检测到10cm内的物
  if (digitalRead(sen03) == 0) {

  //车子左上角，digitalRead(9)表示pin脚读取的是二进制的值，1为检测到10cm内的物
  if (digitalRead(sen01) == 0) {

  if (sweepServo.check() == 1) {

  Serial.print("all progress:");
  Serial.println(millis() - timer);
}
```

## 细节展示

```
Servo mouthServo;
//mouth
int mouthspeed = 10;
void mouthgear(int pos) {

int state = 1;
Servo tailServo;//tail
int tailspeed = 11;
//int taildirection=20;
void tailgear(int maxpos03, int perpos03) {
  if (state) {
    if (pos >= 0 && pos <= maxpos03) {
      pos = pos + perpos03;                              // in steps o
      tailServo.write(pos);                            // tell servo to g
    }
    if (pos > (maxpos03 - 1))  state = false;                    // assi
  }
  else {
    if (pos >= 0 && pos <= maxpos03) {
      pos = pos - perpos03;
      tailServo.write(pos);
    }
    if (pos < 1)  state = true;
  }
}
Servo magnetServo;//磁铁的舵机

int magnetspeed = 12;
void magnetgear(int pos) {
    magnetServo.write(pos);
}
```

# FINAL VISUAL