

Interactive Growing Plant by Wang Jie

1. Concept

We sometimes use plants to make living room more lively.



But the growth of plants take a long time.



Actually, the decorative effect of common plants is poor.
We sometimes can't feel the growing process of plants.

2.Result

I have finished a devise to shorten the growth time.
Let life is full of the breath of life!



Introduction:

There are two forms of interactive plants.

There are some real plants and visual plants which are in the album.

You can interact with my installation in two methods.

a.Ambient light affects the plants.

b.Touch light affects the plants.

3.Hardware

a.Wii remote

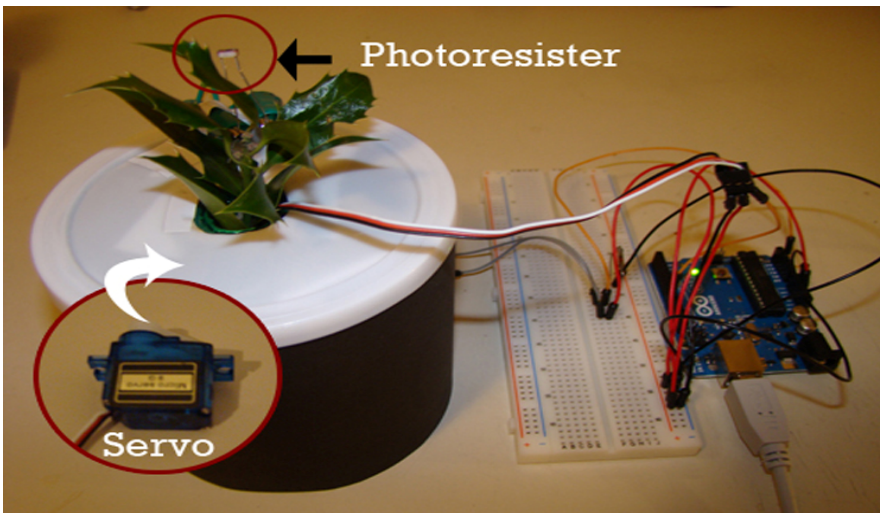
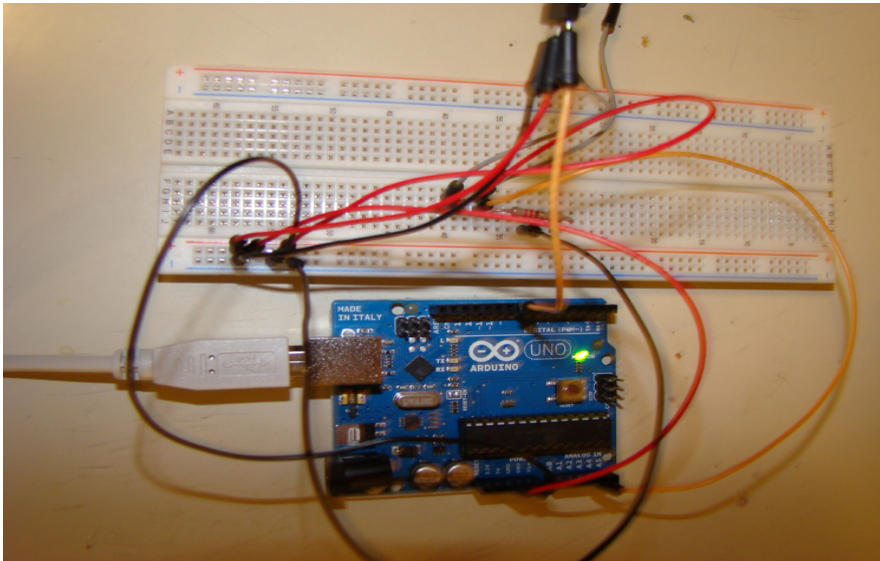
b.Servo

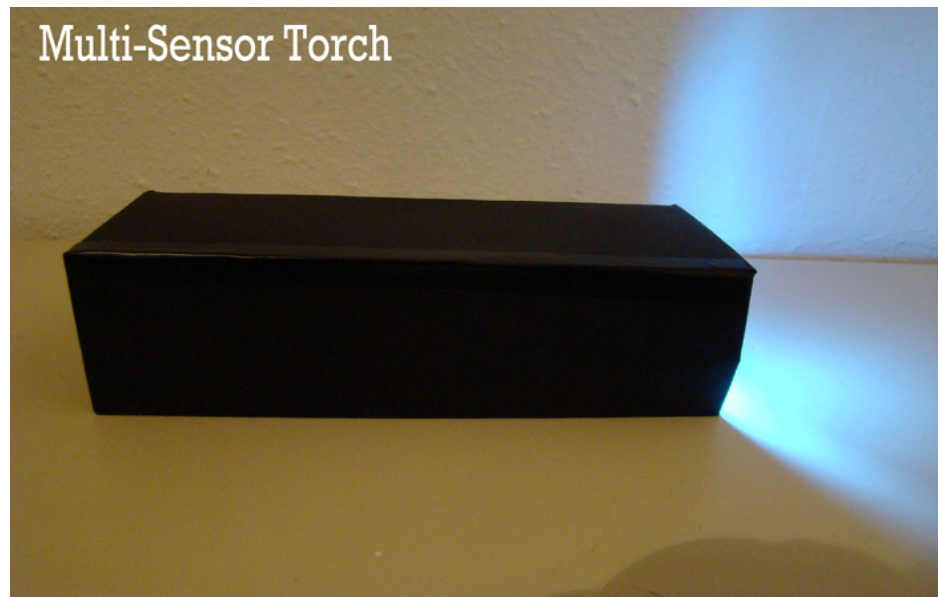
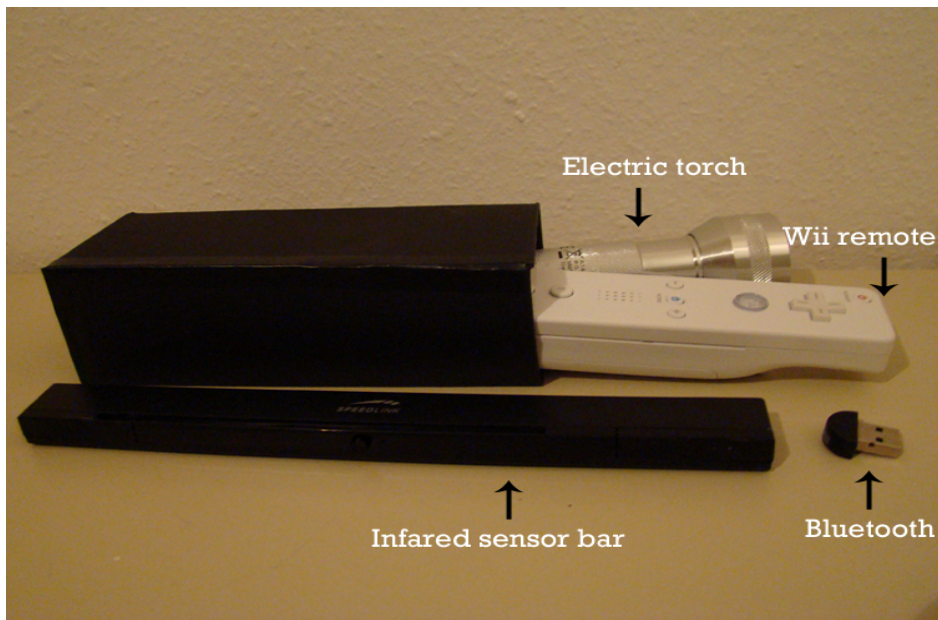
c.Photoresister

d.Arduino board

e.Torch

f.Infrared sensor bar





Technical Difficult Problems:

1. Multiple data communication between Arduino and Processing.

Arduino sends multiple data:

```
Serial.print("s"+String(sensorValue)+"-"+String(sensorValue2)+"-");
```

Processing receives and split the data:

```
int lf = 115;
```

```
myPort.bufferUntil(lf);
```

```
String[] val = split(inString,"-");
```

```
incomingVal =int(val[0]);
```

```
incomingVal2= int(val[1]);
```

```
void serialEvent(Serial p) {inString = (myPort.readString());}
```

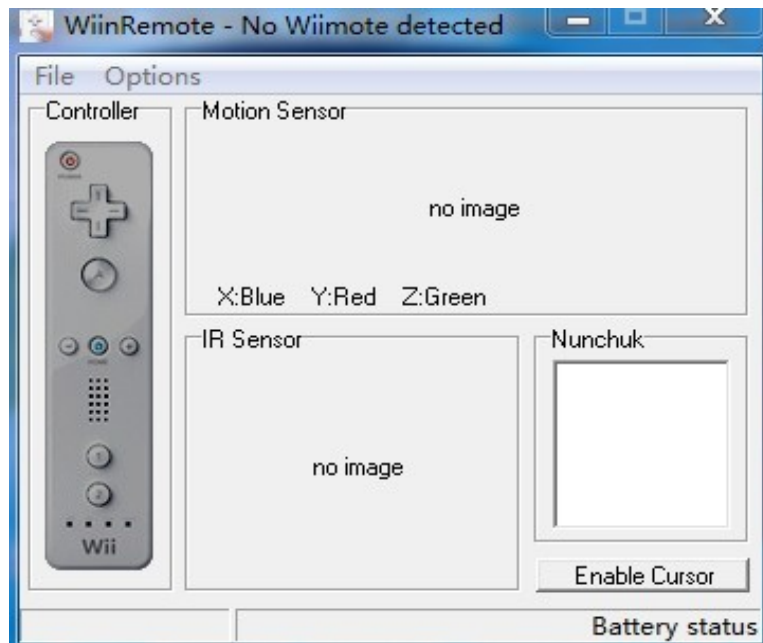
2. Communication between wii remote and infared sensor bar.

a. Bluetooth

Ensure that your Bluetooth connectivity is enabled. The Wiimote uses Bluetooth in order to send and receive signals to and from your computer. Calibrating the Wiimote to your PC should be relatively easy. It is done by pressing the sync button in the battery housing on the Wiimote while your Bluetooth device is enabled.

b. Wiimote application

Download and open the application, click the button of "enable cursor".



c. Infrared sensor bar

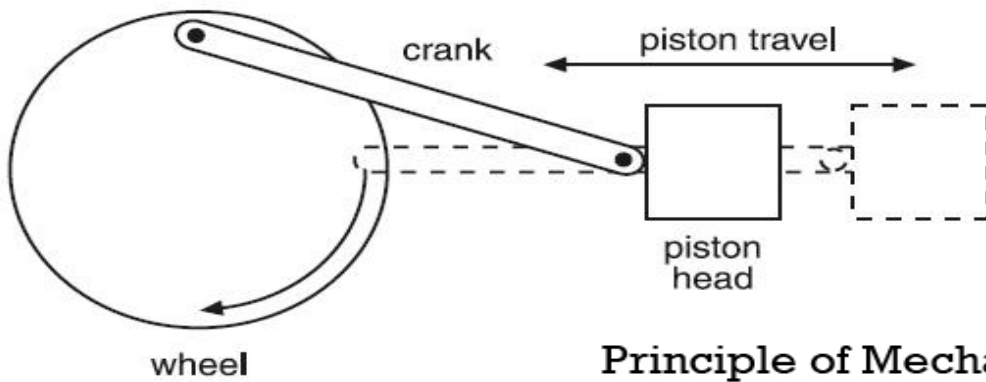
open the infrared sensor bar

d. Wiimote

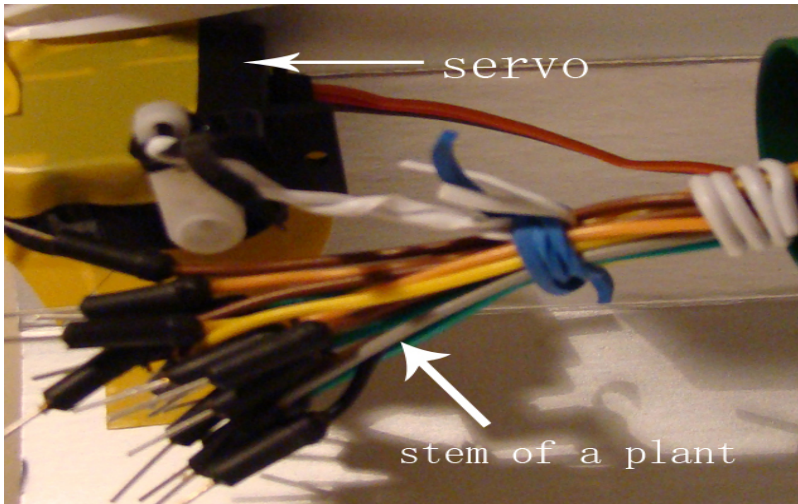
Configure the Wiimote buttons to work comfortably. You can use the "Z" trigger or the "1" button for left-clicks while enabling the "2" button for the right-click function. This is the default control layout for both applications.

3.principle of Mechanics

the wing of servo control the plants growing follow this principle.



Principle of Mechanics



The fine steel wire is tied with the servo and the stems of plant

4.Translate multiple centre point

The two digital plants have two transformation point.

If the light affect the left plant, the transformation centre point in the bottom of the left plant.

If the light affect the right plant, the transformation centre point in the bottom of the right plant.

This is the code:

```
translate(width*3/5,555); // tranlate the centre to the left plant
```

```
scale(a);
```

```
frameF = (frameF+1) % numFrames; // Use % to cycle through frames
```

```
frameRate(value);
```

```
j=(j+k) % numFrames;
```

```
h=(h+k) % numFrames;
```

```
image(images[j],-100,-200);
```

```
scale(1/a);
```

```
translate(-width*3/5,-555); // tranlate the centre to right place
```

```
translate(width/4,555); // tranlate the centre to the right plant
```

```
scale(z);
```

```
image(images[h],-100,-200);
```

```
scale(1/z);
```

```
translate(-width/4,-555); // tranlate the centre to right place
```


4. Software

a. Processing Code:

```
import processing.serial.*;
import processing.opengl.*;
import fullscreen.*;

FullScreen fs;
PImage img;
Serial myPort;
int incomingVal ;
int incomingVal2;
int numFrames = 50; // The number of frames in the animation
int frameF = 0;
PImage[] images = new PImage[numFrames];
int value = 20;
int k = 1;
int u = 0;
int j = 0;
PImage b;
PImage c;
PImage d;
float a = 1;
float z = 1;
int i = 0;
int h = 0;
String inString; // Input string from serial port:
int lf = 115;

void setup()
{ fs = new FullScreen(this); // enter fullscreen mode
  fs.enter();
  size(1250, 740,P3D);
  frameRate(value);
  b = loadImage("bg.jpg");
  c = loadImage("light.png");
  d = loadImage("line.png");
  noStroke();
  myPort = new Serial(this,"COM8",9600);
  myPort.bufferUntil(lf);
  for(int i=0; i<numFrames;i++ ) {
    String imageName = nf(i, 4) + ".png";
    images[i] = loadImage(imageName); }
}

void draw()
{noCursor();
  background(255);
  image(b, 0, 0);
```

```

translate(width*3/5,555);
scale(a);
frameF = (frameF+1) % numFrames; // Use % to cycle through frames
frameRate(value);
j=(j+k) % numFrames;
h=(h+k) % numFrames;
image(images[j],-100,-200);
scale(1/a);
translate(-width*3/5,-555);
translate(width/4,555);
scale(z);
image(images[h],-100,-200);
scale(1/z);
translate(-width/4,-555);
if(mouseX>50&mouseX<1150&mouseY> 50&mouseY<650){
image(c, mouseX-350,mouseY-350); }
image(d,150,500);
String[] val = split(inString,"");
incomingVal =int(val[0]);
incomingVal2= int(val[1]);
println("val1: "+str(incomingVal));
println("val2: "+str(incomingVal2));
if(incomingVal>400)
    {if(a<3){
        a=a+0.005;}
    }
if(incomingVal<400)
    {if(a>1){
        a=a-0.005;}
    }
if(incomingVal2>40)
    {if(z<3){
        z=z+0.005;}
    }
if(incomingVal2<40)
    {if(z>1){
        z=z-0.005; }
    }
}
void mouseMoved()
{if(abs(mouseX-720)/2-80<j&abs(mouseY-500)/2-80<j)
    {if(a<3)
    { a=a+0.005; }
    }
}

```

```
if(abs(mouseX-720)/2-80>=j&abs(mouseY-500)/2-80>=j)
{if(a>1){
  a=a-0.005; }
}
if(abs(mouseX-100)/2-80<j&abs(mouseY-500)/2-80<j)
{if(z<3)
{ z=z+0.005; }
}
if(abs(mouseX-100)/2-80>=j&abs(mouseY-500)/2-80>=j)
{if(z>1)
{ z=z-0.005; }
}
}
}
void serialEvent(Serial p) {
  inString = (myPort.readString());}
```


b.Arduino Code:

```
unsigned long time;
#include <Servo.h>
int servoPin=8;
int analogPin=3;
int val;
Servo myServo;
float light=0;
int ledPin=13;
int sensorPin=0;
int sensorValue;
int servoPin2=7;
int analogPin2=4;
int val2;
Servo myServo2;
float light2=0;
int sensorPin2=1;
int sensorValue2;
int threshold1;
int threshold2;
int tolerance=5;

void setup(){
  myServo.attach(servoPin);
  myServo.write(0);
  myServo2.attach(servoPin2);
  myServo2.write(0);
  pinMode(ledPin,OUTPUT);
  Serial.begin(9600);
  threshold1=analogRead(analogPin);
  threshold2=analogRead(analogPin2);
}

void loop(){
  val=analogRead(analogPin);
  val2=analogRead(analogPin2);
  if(val>(threshold1+tolerance))
  {light=light+0.1;
  if(light>179){
    light = 180; }
  }
  if(val<=(threshold1+tolerance))
  {light=light-0.1;
  if(light<1){
    light = 0; }}
}
```

```
if(val2>(threshold2+tolerance))
{light2=light2+0.1;
if(light2>179){
light2 = 180; }
}
if(val2<=(threshold2+tolerance))
{light2=light2-0.1;
if(light2<1){
light2 = 0; }
}

myServo.write(light);
myServo2.write(light2);
sensorValue= analogRead(sensorPin);
if(sensorValue>512){
digitalWrite(ledPin,HIGH);}
else{
digitalWrite(ledPin, LOW);}
int scaledValue=map( sensorValue, 0,1023, 0, 255);

sensorValue2= analogRead(sensorPin2);
int scaledValue2=map( sensorValue2, 0,1023, 0, 255);
time= millis();
Serial.print("s"+String(sensorValue)+","+String(sensorValue2)+",");
delay(5);}
```