Beyond Sparkles | From Objects to Space: Light

Projektmodul Sommersemester 2013

Guanling Yuan

Chen Liang

Hsinyu Chuang

Professur

Interface Design: Interaction and

Experimental Interfaces (M)

Interaction and Motion Design (G)

Betreuung

Prof. Wolfgang Sattler

Prof. Jens Geelhaar

M. Sc. Kristian Gohlke

M. F.A. Kevin Lefeuvre

The first idea

04.2013

05.2013

The final plan

06 - 07.2013

Technical feasibility studies

Final technical solution

07 - 09.2013

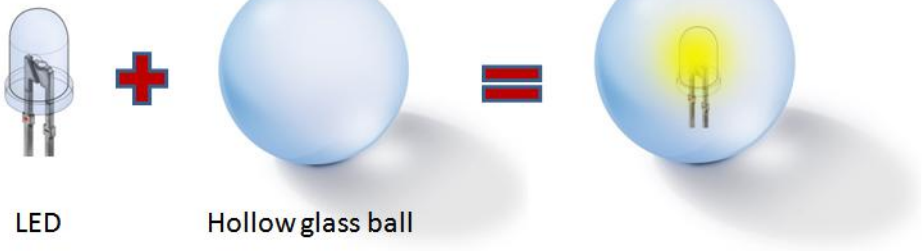
09 - 10.2013

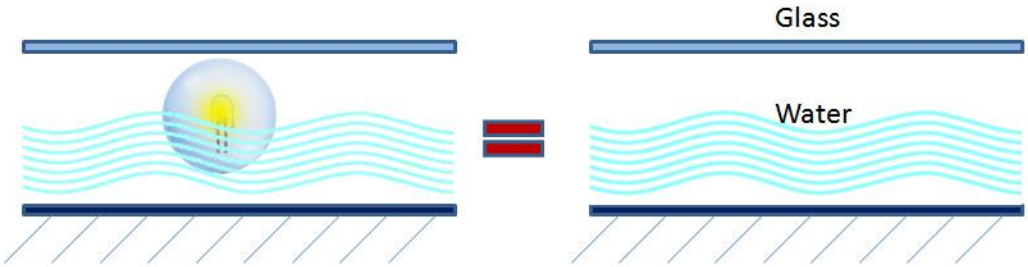
Initial prototype

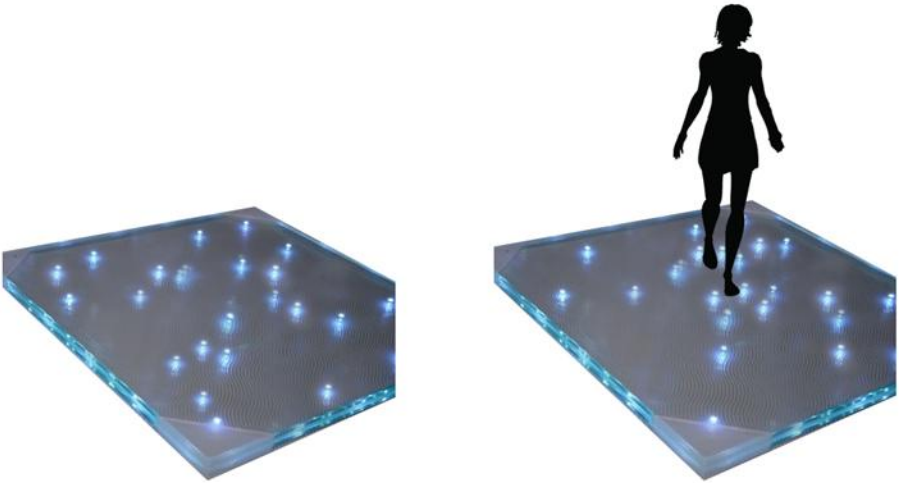
"Loi Krathong" - The name could be translated "Floating Crown" or "Floating Decoration", comes from the tradition of making buoyant decorations which are then floated on a river.

First idea Water lights with their reflections, under the help of wind, floating and moving on the water.

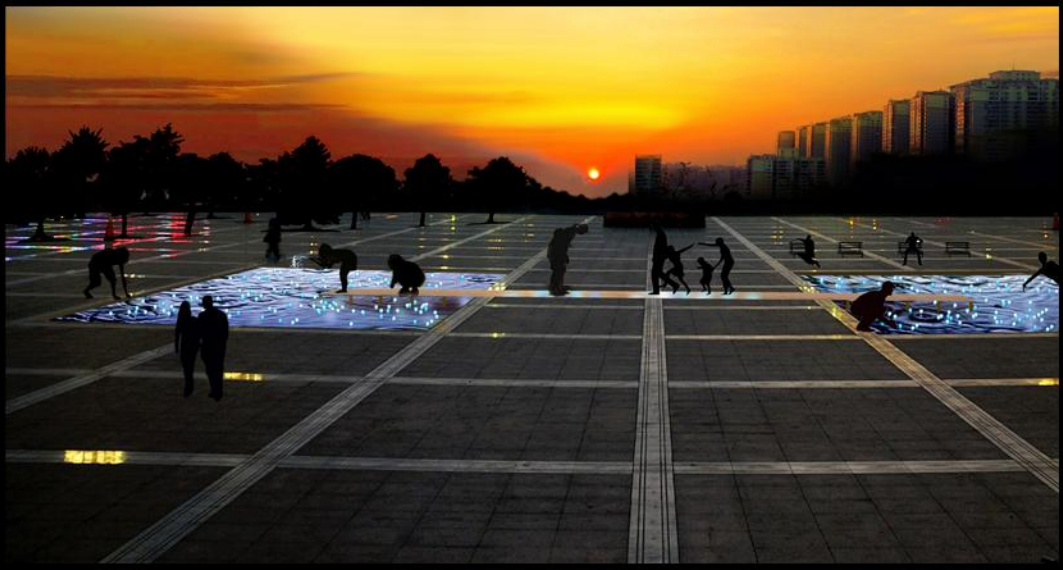
Initial prototype ideas







The small balls will follow with people



Final plan

Water Melody

„Water Melody “ is a public interactive installation, built in parks, plazas, shopping malls and other public places.

By sound, light and interactive design, „Water Melody “ creates a funny, tranquil and elegant atmosphere, to attract and encourage people to participate, and also to provide the opportunity for people to communicate with each other.

Entire interactive installation consists of three parts: pools, glass bridge and LED balls. Both length and width of the pool are 10m, and depth is 20cm. In order to make a better integration of the pool and the environment, surface of the water is flush with the ground.

And also there is no obstructions at the poolsides, to make it easier for people to get close. There isa glass „bridge" linking two pools. Glass bridge combined by a series of glass boxes, and each glass box has an independent electromagnet system. the electromagnet will be activated, when people set foot on the glass. The bridge deck is flush with the ground, and the waterway which under the glass connect two pools.

The protagonist of the whole installation are small plastic balls which with a diameter of 6cm. The small transparent balls can emit blue and white light. They wiil change colors when they collide with each other, and at the same time the balls sent out different chromatic tones.

Under ordinary circumstances, the balls floating freely in the pool water. But when someone moving on the glass bridge, the surrounding balls will be attracted by electromagnet, and follow the people move from one pool to another.

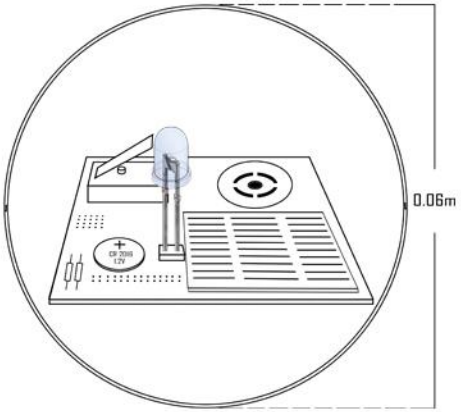
When night fell, the lights suspended in the water, gently sway with the water, exudes a kind of special charm. Light collide with each other slightly, change colors, make beautiful sound of the slight collision. People gathered at the the pool side, sitting or standing, Splash the water with hands or feet. Curious children picked up the ball from water and thrown back into the water.

A gust of wind blew, the pools are full of „Ding -Dong" sound. Occasionally someone walks on the glass bridge, light will flow from one pool to another.

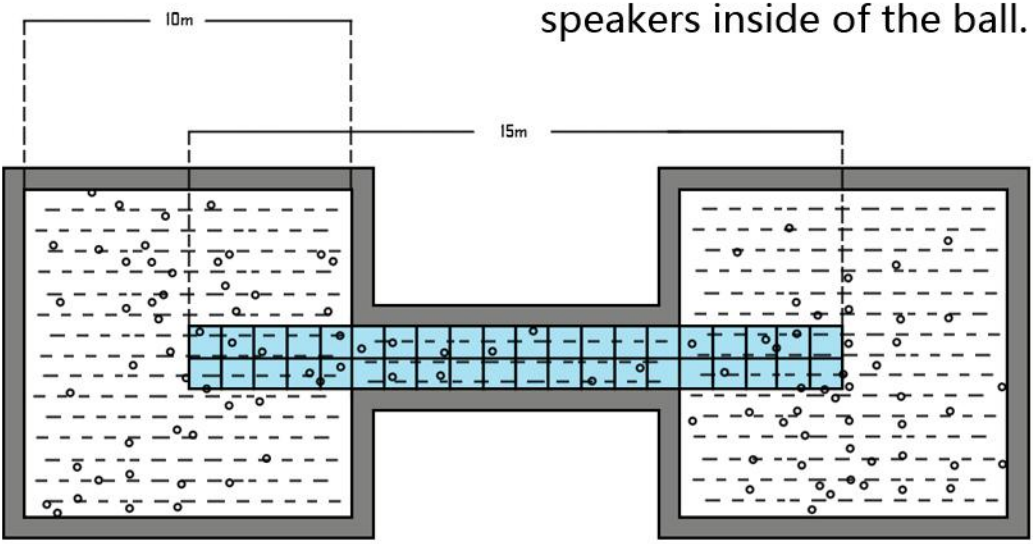
Technical feasibility studies

Aerial view of the overall structure: The pool is a square, both length and width are 10 meters. The glass bridge is 15 meters long, two ends of the bridge are in the center of the pools.

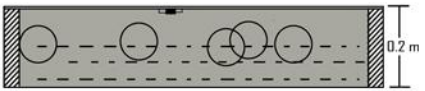
Internal structure of the LED ball: diameter is 6cm, arduino circuit board be fixed inside.

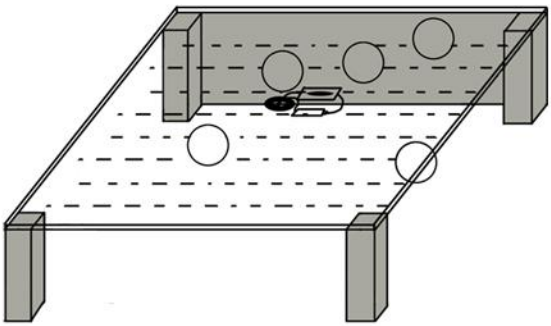


Solar power system includes rechargeable button battery, solar panels, charge controller. led light is the double colors light for blue and white. In addition, there are also collision sensors, electronic switches, iron, speakers inside of the ball.



The internal structure of the glass bridge: glass box is a square which both length and width are 1m. Infrared sensors and electromagnet are mounted under the glass. Height of the tunnel is 20cm, water and balls can pass through freely. When people set foot on glass, the signal which from infrared sensor activates electromagnet, balls will be attracted by a magnet. It looks like the light flow to follow the footsteps and track the people.





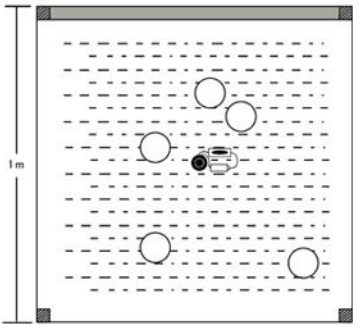
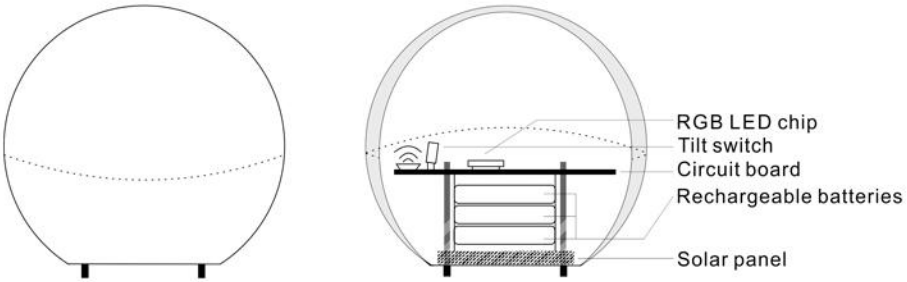
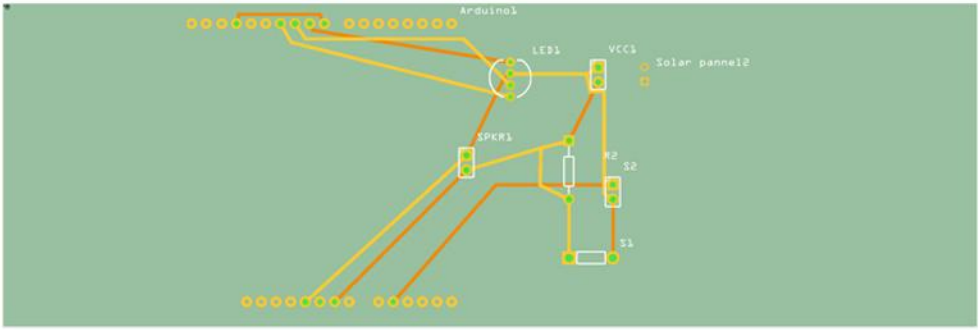
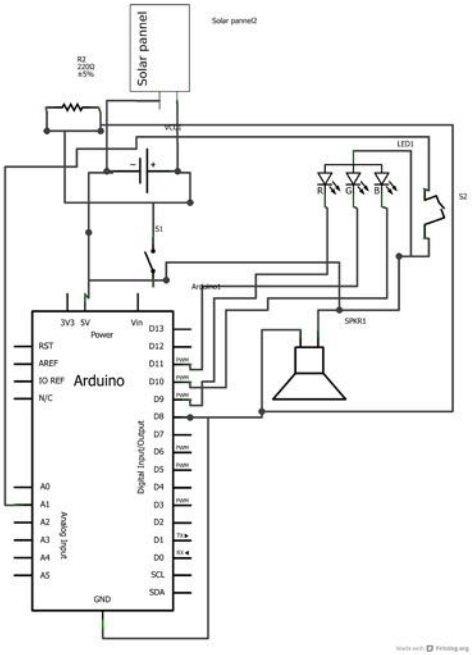


Diagram of the light ball





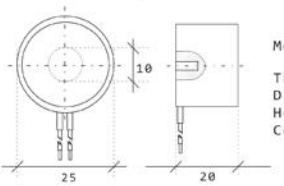
Made with Fritzing.org



Demo_ pcb of the light ball & Circuit diagram of the light ball

Detail of electromagnet in glass panel &
Detail of RGB LED Chip in the light ball

12V DC 11 LB / 5kg Electric Lifting Magnet Electromagnet Solenoid Lift Holding

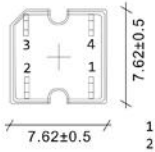


Model: ZYE1-P25/20

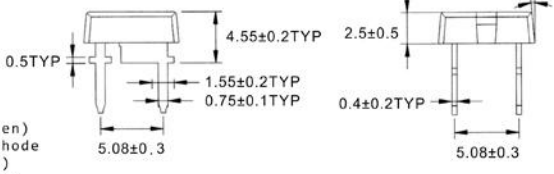
Thread Size: M4
Diameter: Approx. 25 mm
Height: Approx. 20 mm
Center Diameter: Approx. 10 mm

Input Voltage: 12V DC
Current: 0.33A
Power Consumption: 4W
Holding Force: 50N / 11lbs
Thread Size: M34
Material: Metal
Color: Silvery
Weight:50g

Super-Flux RGB LED (3-Chip Technic)



1 Anode (Green)
2 Common Cathode
3 Anode (Red)
4 Anode (Blue)



Diameter: 7,6x7,6mm

Viewing Angle: 130

Height: 4.55mm
Width: 7.62mm
Depth: 7.62mm

Emitting Color: red / green / blue
Housing Color: clear
Nanometer: R=626 / G=532 / B=475

mA test.: 20 mA
mA typ.: 30 mA
V typ.: R=2,0 / G=4,0 / B=4,0V V

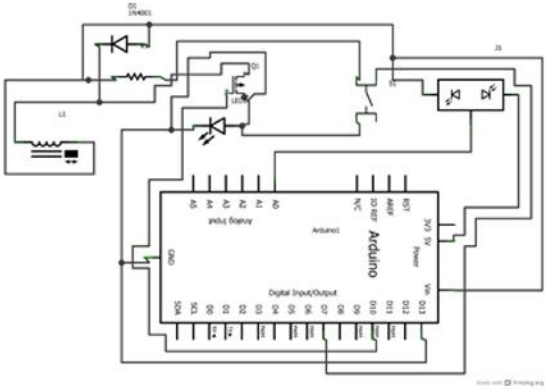
RGB full color superflux LED with common cathode.
This LED works with many RGB application.

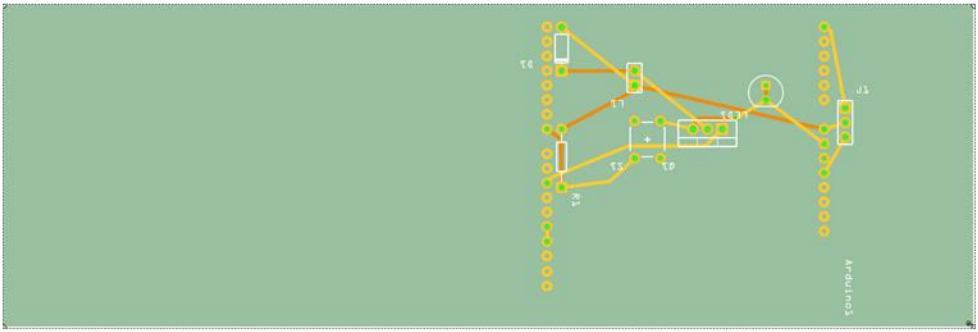
- MCD: R=900 / G=2100 / B=700mcd
- nm: R=626 / G=532 / B=475nm
- V typ.: R=2,0 / G=4,0 / B=4,0V

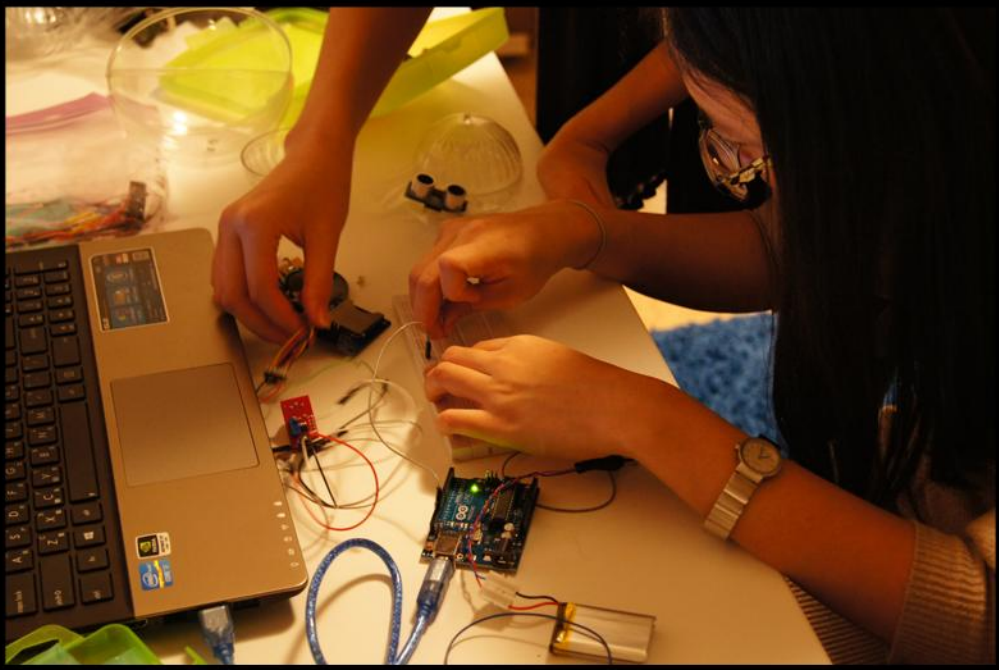
Main advantages are

- small housing
- wide viewing angle
- long lifetime
- indiv. useability

Circuit diagram of the glass panel & Demo_ pcb of the glass panel with infrared switch and electromagnet

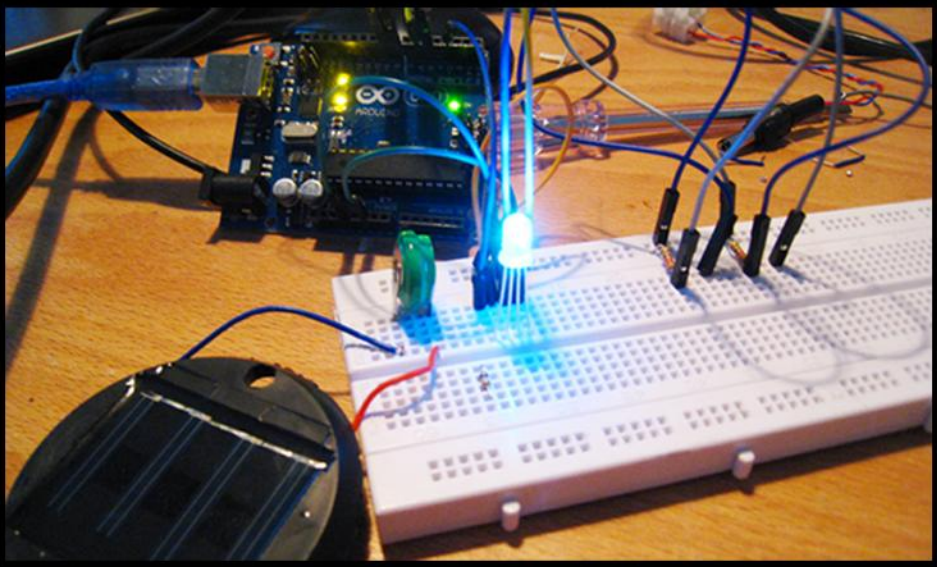




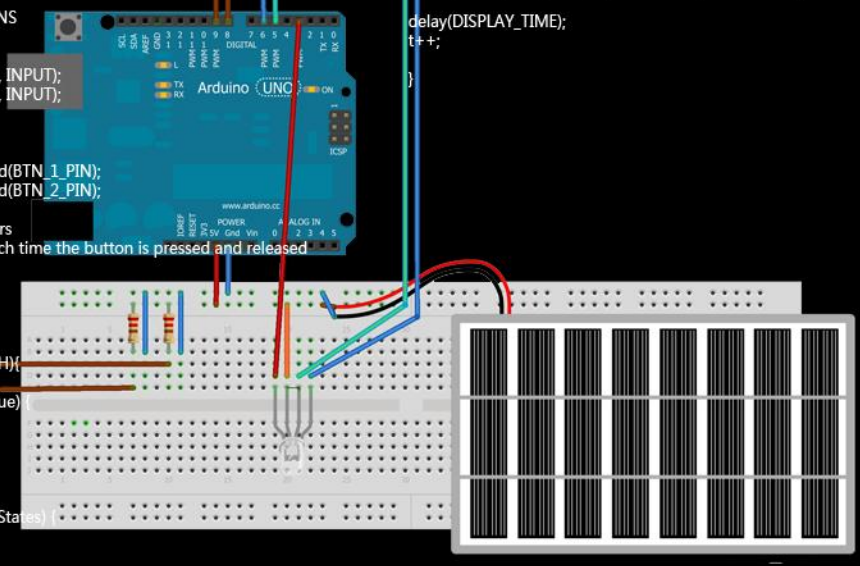


Final technical solutions

Arduino circuit experiment

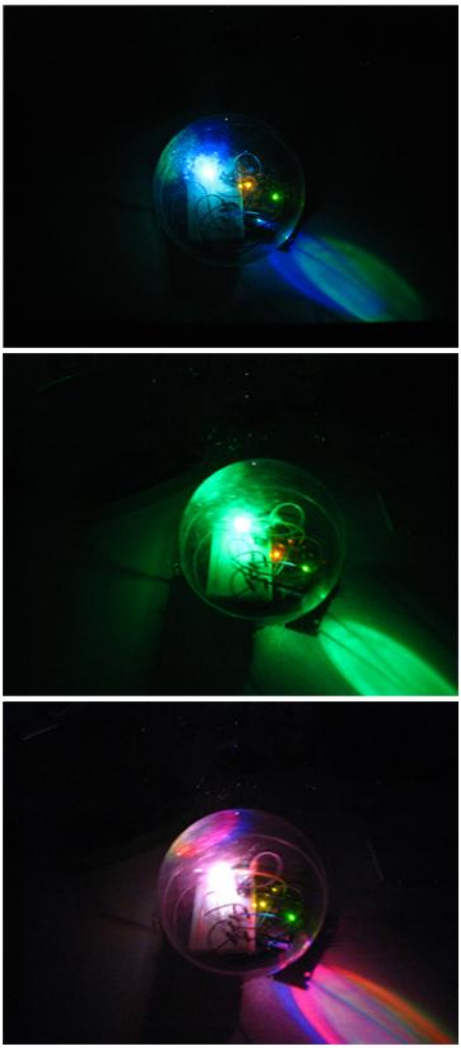


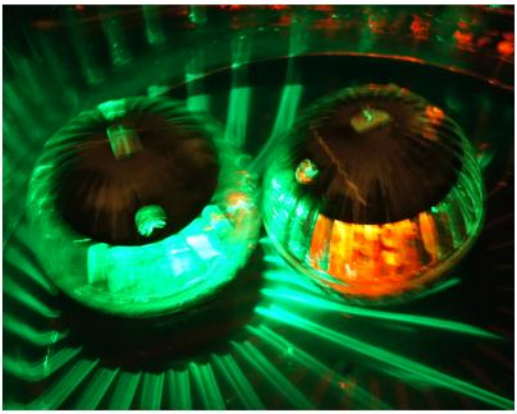
```
const boolean VERBOSE = true;
// LED leads connected to PWM pins
const int RED_LED_PIN = 3;
const int GREEN_LED_PIN = 5;
const int BLUE_LED_PIN = 6;

/* Button pins */
const int BTN_1_PIN = 8;
const int BTN_2_PIN = 9;

/* Thresholds for intensity of LED */

const int MAX_INTENSITY = 255; //Full brightness
const int MIN_INTENSITY = 5; //Minimum Brightness

// Used to store the current intensity level of the individual LEDs
int redIntensity = 0;
int greenIntensity = 0;
int blueIntensity = 0;
// All of those levels are controlled by this one. See the loop code.
int intensity = 255;
//Channel on/off state - 0 = off, 1 = on
int redState = 1;
int greenState = 1;
int blueState = 1;

//The bounced the master intensity between MIN_ and MAX_INTENSITY
int dir = -1;

//This variables represents time - +1 each tick. Used for Trig.
int t = 0;

// Length of time we spend showing each color
const int DISPLAY_TIME = 50; // In milliseconds

/* States to toggle thru RGB on/off */
int states[6][3] = {{0,0,1},
{0,1,0},
{0,1,1},
{1,0,0},
{1,0,1},
{1,1,1}};
int currState = 5; //Set to the one above, all colors on.
int numStates =6;

//Used to cause the button to make an action once each time it is pressed.
boolean btn1_DOWN = false;

//SETUP THE BUTTONS
void setup() {
Serial.begin(9600);
pinMode(BTN_1_PIN, INPUT);
pinMode(BTN_2_PIN, INPUT);
}

void loop() {
int btn1 = digitalRead(BTN_1_PIN);
int btn2 = digitalRead(BTN_2_PIN);

/*Change active colors
Only do this once each time the button is pressed and released
*/
if(btn1 == LOW) {

btn1_DOWN = true;

} else if(btn1 == HIGH){

if(btn1_DOWN == true) {

btn1_DOWN = false;

currState++;

if (currState == numStates) {
currState = 0;
}

redState = states[currState][0];
greenState = states[currState][1];
blueState = states[currState][2];

f (VERBOSE) {
Serial.print("::: redState ");
Serial.print(redState);
Serial.print(" greenState ");
Serial.print(greenState);
Serial.print(" blueState ");
Serial.print(blueState);
Serial.println();
}
}
}

/*CHANGE INTENSITY
Changed continuously while button is pressed.
*/
if(btn2 == LOW) {
if (dir == -1) {
intensity-=5;
if(intensity <= MIN_INTENSITY) {
intensity = MIN_INTENSITY;
dir = 1;
}
} else {
intensity+=5;
if(intensity >= MAX_INTENSITY) {
intensity = MAX_INTENSITY;
dir = -1;
}
}
if (VERBOSE) {
Serial.print(" INTENSITY::");
Serial.print(intensity);
Serial.println();
}
}
//Use trig functions to make the channels pulse at different rates.
redIntensity = abs(int(intensity * sin(t*0.01)));
greenIntensity = abs(int(intensity * cos(t*0.02)));
blueIntensity = abs(int(intensity * sin(t*0.03)));

analogWrite(RED_LED_PIN, redIntensity * redState);
analogWrite(GREEN_LED_PIN, greenIntensity * greenState);
analogWrite(BLUE_LED_PIN, blueIntensity * blueState);

delay(DISPLAY_TIME);
t++;

}
```



Initial prototype

Under the action of collision sensor, the ball will transform different colors.





To be continued ...