```
#include <Adafruit_NeoPixel.h>


#define trigPin 12
#define echoPin 11
#define PIN 6
#define NUM_LEDS 64


Adafruit_NeoPixel strip1 = Adafruit_NeoPixel(NUM_LEDS, PIN, NEO_GRB + NEO_KHZ800);
Adafruit_NeoPixel strip2 = Adafruit_NeoPixel(NUM_LEDS, PIN, NEO_GRB + NEO_KHZ800);


int eyes_x = 3;            // x-coördinaat van de ogen
int eyes_y = 3;            // y-coördinaat van de ogen
int eyes_radius = 2;     // Grootte van de ogen
int delay_time = 100;    // vertraging tussen bewegingen in milliseconden


void setup() {
  strip1.begin();
  strip2.begin();


Serial.begin (9600);
pinMode(echoPin, INPUT);
pinMode(trigPin, OUTPUT);


}
```

```
void loop() {


  long duration, distance;

  digitalWrite(trigPin, LOW);

  delayMicroseconds(10);

  digitalWrite(trigPin, HIGH);


  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);

  distance = (duration/2) / 29.1;



if (distance < 50) { //ogen worden rood

  // Beweeg de ogen in een willekeurige richting

    int x_direction = random(-1, 2);     // -1 is links, 0 is stilstand, 1 is rechts

    int y_direction = random(-1, 2);     // -1 is omhoog, 0 is stilstand, 1 is omlaag

    eyes_x += x_direction;

    eyes_y += y_direction;


    // Controleer of de ogen de randen van de matrix raakt en keer het om als dat het geval is

    if (eyes_x < eyes_radius) {

        eyes_x = eyes_radius;

    }

    if (eyes_x > NUM_LEDS / 8 - eyes_radius - 1) {
```

```
    eyes_x = NUM_LEDS / 8 - eyes_radius - 1;
}
if (eyes_y < eyes_radius) {
    eyes_y = eyes_radius;
}
if (eyes_y > NUM_LEDS / 8 - eyes_radius - 1) {
    eyes_y = NUM_LEDS / 8 - eyes_radius - 1;
}

// Wis de matrix en teken de ogen opnieuw
strip1.clear();
strip2.clear();
for (int y = eyes_y - eyes_radius; y <= eyes_y + eyes_radius; y++) {
    for (int x = eyes_x - eyes_radius; x <= eyes_x + eyes_radius; x++) {
        if ((x - eyes_x) * (x - eyes_x) + (y - eyes_y) * (y - eyes_y) <= eyes_radius * eyes_radius) {
            int red = random(256);
            int green = random(0);
            int blue = random(0);
            strip1.setPixelColor(y * 8 + x, red, green, blue);
            strip2.setPixelColor(y * 8 + x, red, green, blue);
        }
    }
}

strip1.show();
```

```
    strip2.show();
    delay(delay_time);
}
  else { //ogen worden groen
// Beweeg de ogen in een willekeurige richting
    int x_direction = random(-1, 2);    // -1 is links, 0 is stilstand, 1 is rechts
    int y_direction = random(-1, 2);    // -1 is omhoog, 0 is stilstand, 1 is omlaag
    eyes_x += x_direction;
    eyes_y += y_direction;

    // Controleer of de ogen de randen van de matrix raakt en keer het om als dat het geval is
    if (eyes_x < eyes_radius) {
        eyes_x = eyes_radius;
    }
    if (eyes_x > NUM_LEDS / 8 - eyes_radius - 1) {
        eyes_x = NUM_LEDS / 8 - eyes_radius - 1;
    }
    if (eyes_y < eyes_radius) {
        eyes_y = eyes_radius;
    }
    if (eyes_y > NUM_LEDS / 8 - eyes_radius - 1) {
        eyes_y = NUM_LEDS / 8 - eyes_radius - 1;
    }

    // Wis de matrix en teken de ogen opnieuw
  strip1.clear();
  strip2.clear();
  for (int y = eyes_y - eyes_radius; y <= eyes_y + eyes_radius; y++) {
      for (int x = eyes_x - eyes_radius; x <= eyes_x + eyes_radius; x++) {
          if ((x - eyes_x) * (x - eyes_x) + (y - eyes_y) * (y - eyes_y) <= eyes_radius * eyes_radius) {
              int red = random(0);
              int green = random(256);
              int blue = random(0);
              strip1.setPixelColor(y * 8 + x, red, green, blue);
              strip2.setPixelColor(y * 8 + x, red, green, blue);
          }
      }
  }

  strip1.show();
  strip2.show();
  delay(delay_time);
}

if (distance >= 200 || distance <= 0){
Serial.println("Out of range");
}
else {
Serial.print(distance);
Serial.println(" cm");
```

```
  }



  delay(500);

}
```