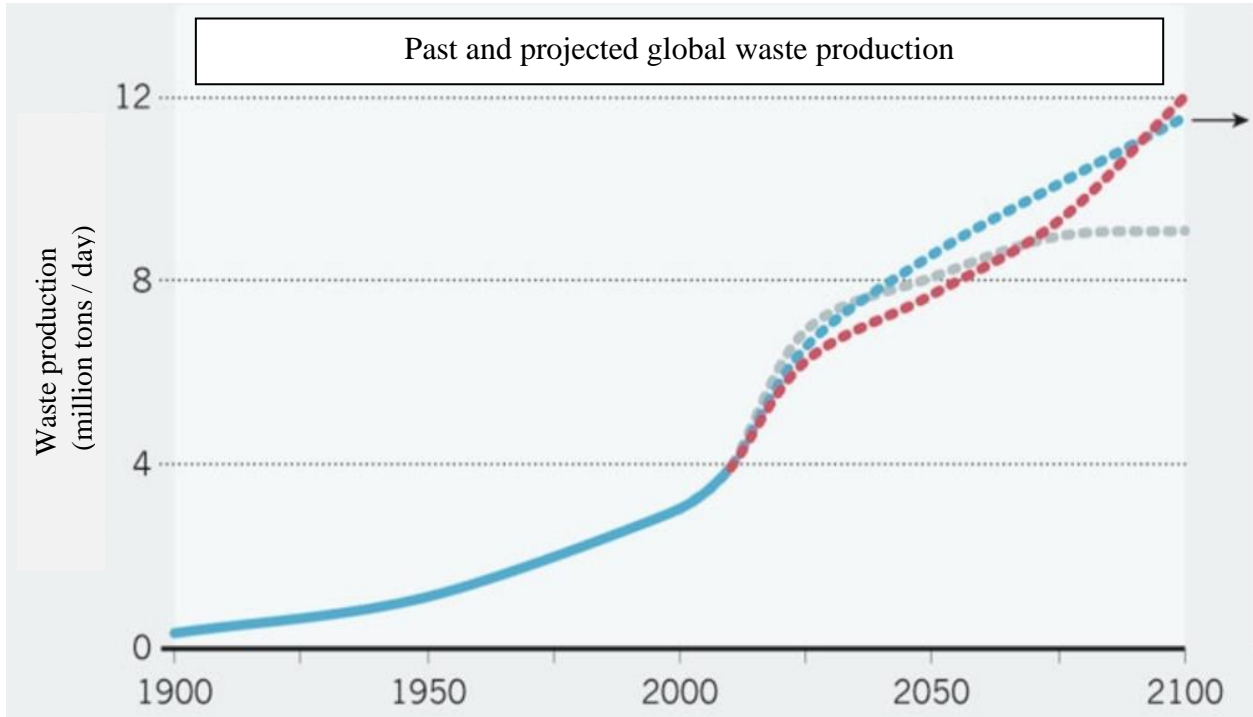
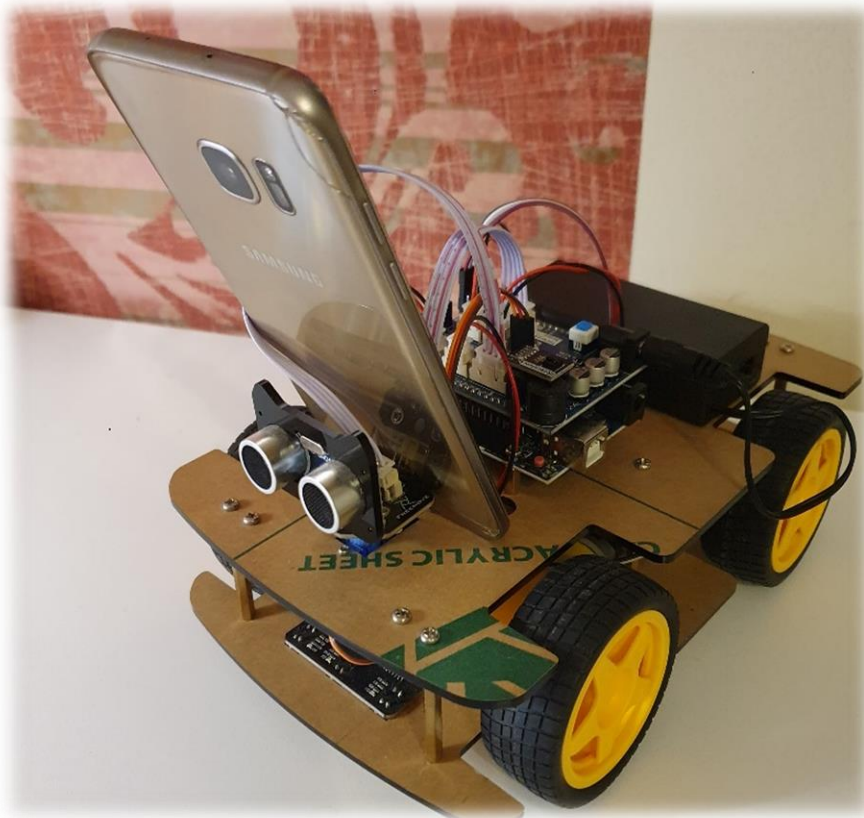


Illustrations



(1. Diagram: Source: Daniel Hoornweg: Environment: Waste production must peak this century; The figure shows the change in the volume of three possible waste productions up to 2100. Gray is the case for a 7 billion more environmentally conscious population, blue is for 9.5 billion people and 80% for urbanization, and red is for the 13.5 billion people on Earth, 70% of whom live in extreme poverty in cities and moderate wealth.)



(2. Illustration: My photo: 4WD Car Kit and the Android based device)

```

65     toolbar.setSubtitle("Connecting to " + deviceName + "...");
66     progressBar.setVisibility(View.VISIBLE);
67     buttonConnect.setEnabled(false);
68     BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
69     createConnectThread = new CreateConnectThread(bluetoothAdapter, deviceAddress);
70     createConnectThread.start();
71 }
72 handler = new Handler(Looper.getMainLooper()) {
73     @Override
74     public void handleMessage(Message msg){
75         switch (msg.what){
76             case CONNECTING_STATUS:
77                 switch(msg.arg1){
78                     case 1:
79                     toolbar.setSubtitle("Connected to " + deviceName);
80                     progressBar.setVisibility(View.GONE);
81                     buttonConnect.setEnabled(true);
82                     buttonToggle.setEnabled(true);
83                     break;
84                     case -1:
85                     toolbar.setSubtitle("Device fails to connect");
86                     progressBar.setVisibility(View.GONE);
87                     buttonConnect.setEnabled(true);
88                     break;
89                 }
90             break;
91
92             case MESSAGE_READ:
93             String arduinoMsg = msg.obj.toString(); // Read message from Arduino
94             switch (arduinoMsg.toLowerCase()){
95                 case "led is turned on":
96                 imageView.setBackgroundColor(getResources().getColor(R.color.colorOn));
97                 textViewInfo.setText("Arduino Message : " + arduinoMsg);
98                 break;

```

(3. Illustration: This handler is responsible for reading and writing messages to the Arduino)

```

//motorok
#define irany_jobb 3
#define irany_bal 4
#define jobb_motor 5
#define bal_motor 6

//ultrahangos szenzor
const int trigPin = 7;
const int echoPin = 8;

long duration;
int distance;

void hatra() {
    digitalWrite(irany_bal, HIGH);
    analogWrite(bal_motor, 90);
    digitalWrite(irany_jobb, LOW);
    analogWrite(jobb_motor, 90);

    Serial.println("Hatra");
}

void elore() {
    digitalWrite(irany_bal, LOW);
    analogWrite(bal_motor, 90);
    digitalWrite(irany_jobb, HIGH);
    analogWrite(jobb_motor, 90);

    Serial.println("Elore");
}

```

```

void balra() {
    digitalWrite(irany_bal,LOW);
    analogWrite(bal_motor,90);
    digitalWrite(irany_jobb,LOW);
    analogWrite(jobb_motor,90);

    Serial.println("Balra");
}

void jobbra() {
    digitalWrite(irany_bal,HIGH);
    analogWrite(bal_motor,90);
    digitalWrite(irany_jobb,HIGH);
    analogWrite(jobb_motor,90);

    Serial.println("Jobbra");
}

void allj() {
    digitalWrite(irany_bal,HIGH);
    analogWrite(bal_motor,0);
    digitalWrite(irany_jobb,LOW);
    analogWrite(jobb_motor,0);

    Serial.println("Stop");
}

```

```

void ultrahangos(){
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    // Calculating the distance
    distance= duration*0.034/2;
    // Prints the distance on the Serial Monitor
    Serial.print("Distance: ");
    Serial.println(distance);

    const long interval = 2000;
    unsigned long kezdet = 0;

    if(distance<=10){
        kezdet=millis();
        while(millis()-kezdet<=interval){
            hatra();
        }
        while(millis()-kezdet<=interval){
            hatra();
        }
    }
}

```

(4. Illustration: The Arduino program, specifically the parts responsible for the robot's movement; drives or reverses the robot if the ultrasonic sensor detects less than a certain distance between the robot and the obstacle or wall in front of it.)

```

app.post('/createNewCoordinate', (req,res,next) => {
  var coord = req.body;
  console.log(req.body);
  //var query = 'Insert into trash_coordinates(' + coordinates.x_coordinate + ',' + coordinates.y_coordinate + ')'
  var query = 'Insert into trash_coordinates(x_coordinate, y_coordinate) VALUES('+coord.x_coordinate+', '+coord.y_coordinate+')'
  console.log(query);
  pool.query(query)
    .then(result => {
      console.log('Insert was succesfull')
      res.send(result.rows);
    })
})

```

(5. Illustration: post request; Android-Server)

Tables (1)

- trash_coordinates
 - Columns (5)
 - id
 - x_coordinate
 - y_coordinate
 - is_cleaned_up
 - garbage_type

	Data Output	Explain	Messages	Notifications
	id [PK] bigint		x_coordinate double precision	y_coordinate real
1		18	46.89039	23.49233
2		19	46.7712101	23.623636
3		20	46.7712101	23.623636
4		21	46.78127	23.612696
5		22	46.7783656	23.612032
6		23	46.77906	23.607569
7		24	46.7772369	23.599295
8		25	46.7903842	23.610954

(6. Illustration: The PostgreSQL database structure)

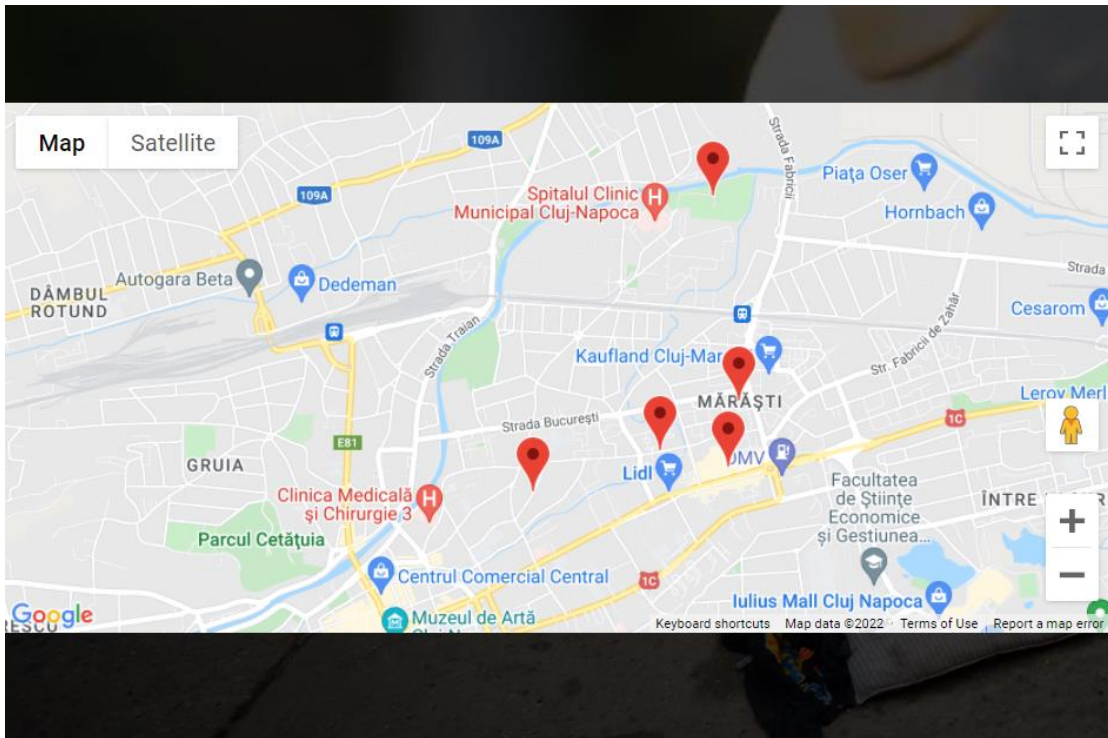
Home About **Trashpoint** Garbage collection Log in

Trashpoint

In the interest of Rusty's work to have an even more significant impact, those who follow Rusty's instructions may communicate with Rusty on this website, moreover, for their work in the public interest, they can collect digital points called Trashpoint. Later, these points could be used for charitable purposes or various rewards. In summary, a user can get Trashpoint in two ways:

1. If you pick up the trash indicated by Rusty, then throw it in a selective bin in the right place, and finally scan the QR code leaked on Rusty's back.
2. The second way to collect Trashcoins is to look for one of the trashes shown on the map on the right and then submit a picture of it while you still have the trash on the ground and another picture after you pick it up (and of course you throw it in the right trash).

Log in to collect and track your Trashcoins!



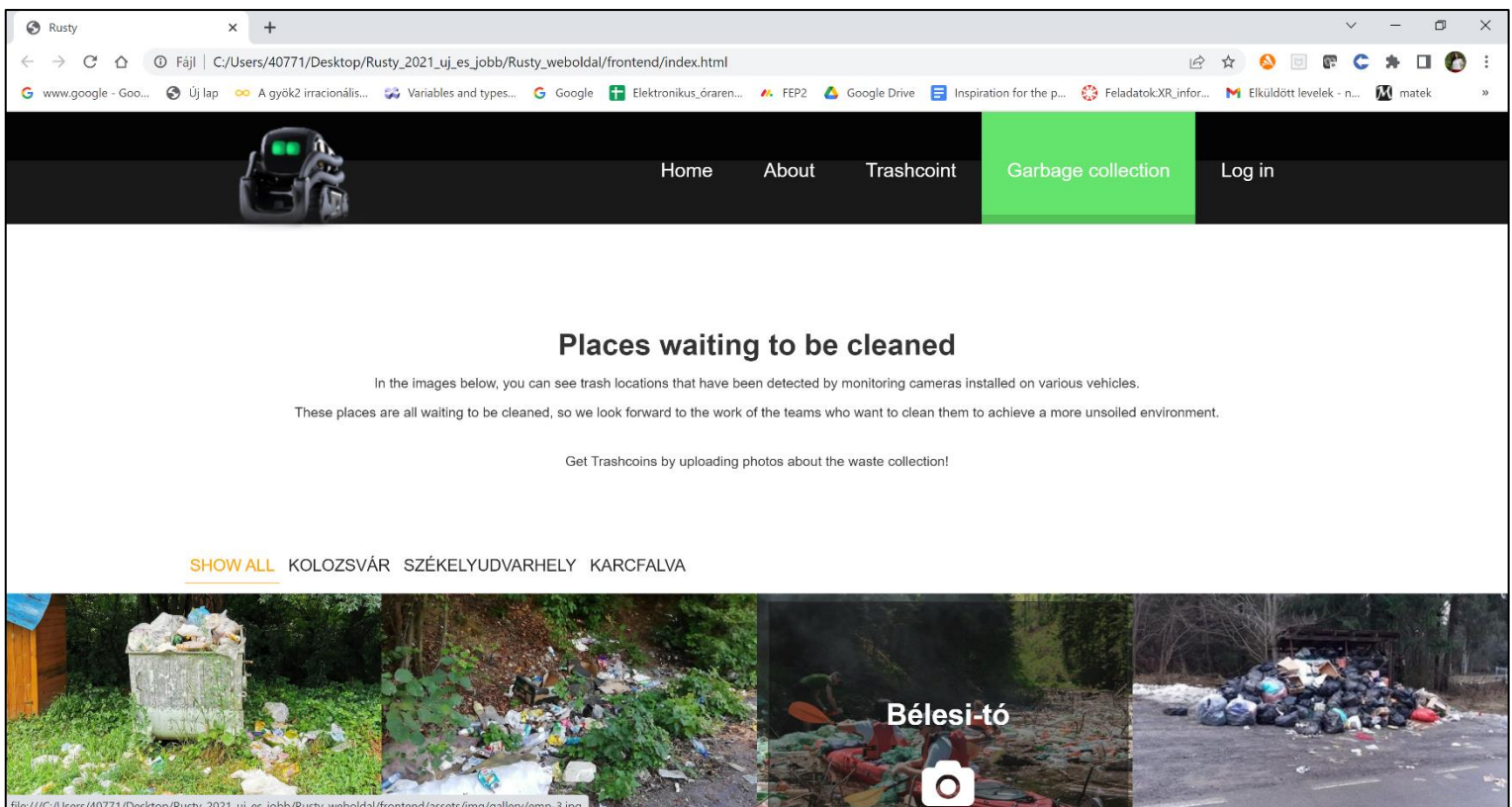
(7. Illustration: Garbage location map)

```

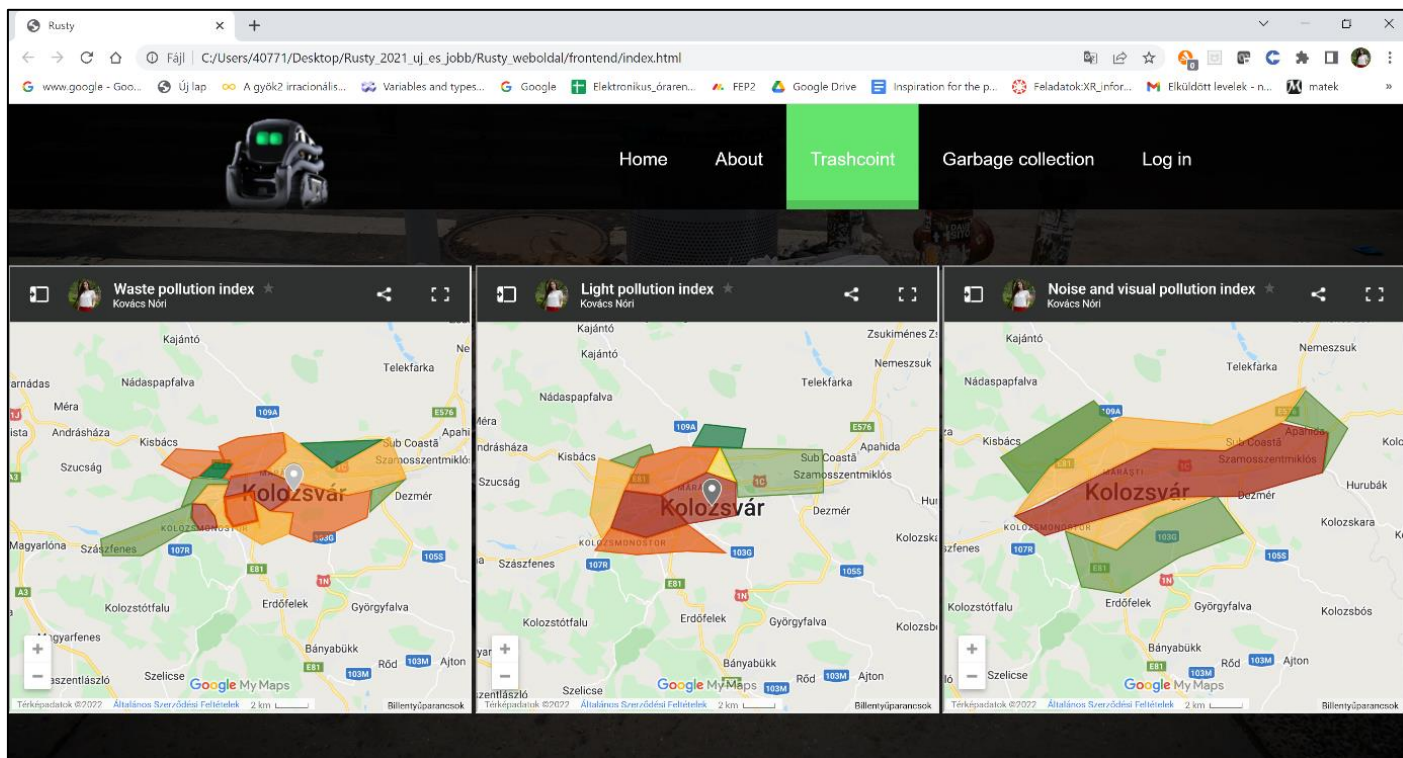
app.get('/getAllCoordinates', (req, res, next) => {
  pool.query('Select * from trash_coordinates')
    .then(testData => {
      console.log(testData);
      res.send(testData.rows);
    })
})

```

(8. Illustration: Retrieve the coordinates from the server with the get request)



(9. Illustration: My website: Problem reporting section)



(10. Illustration: Pollution level by region)