instructables

Urban Farming Integrated With Beekeeping

by Sunny_surya

With the rising global population, the amount of food required to be produced every year is eye-opening. Farmers have been using technology to increase productivity via drones, various techniques, etc. We need not leave this to farmers, we as the next young generation can play our roles to contribute to help feed more people.

We need to close the gap between the agricultural industry and consumer's perception and come up with a successful model to deal with the global supply and demand. One of the effective ways is to integrate Urban farming with beekeeping. According to research, one-third of the food we eat wouldn't exist without honeybees. As pollinators, they help flowers produce fruits and seeds. Unsurprisingly they are an integral part of our food production ecosystem.

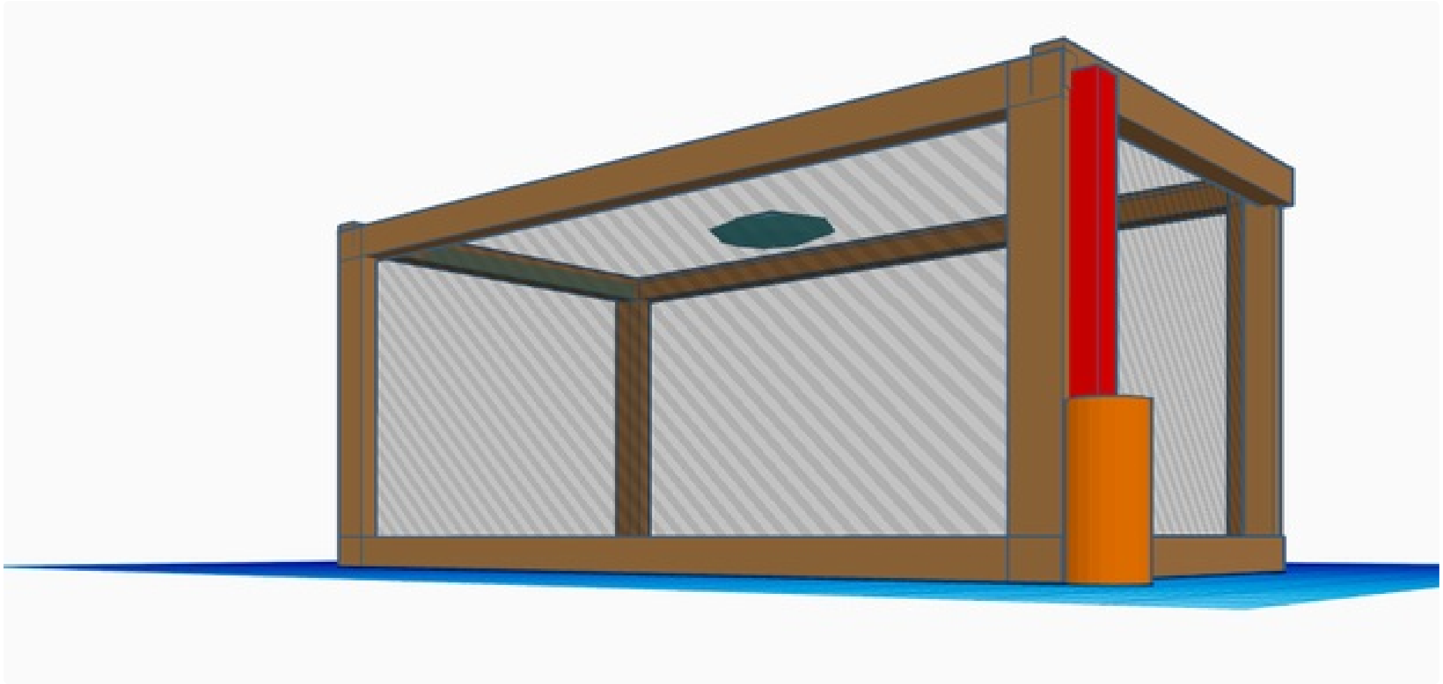
Our designed 3D structure's mission is to inspire individuals to think about the relationship between them and the food ecosystem and help them contribute towards the global food demand in a sustainable way. You can contribute even when you are living in populated cities or municipalities!

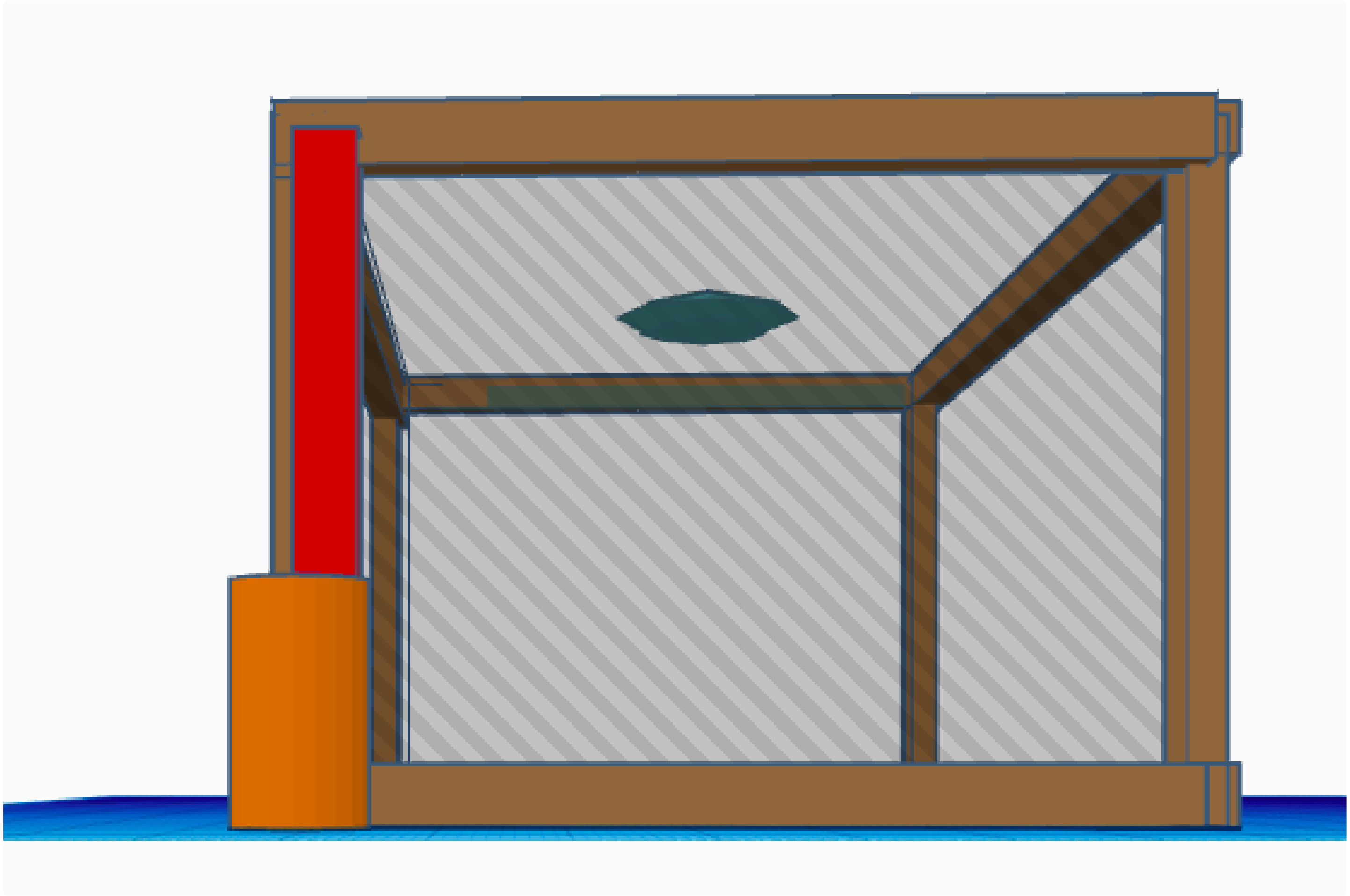
The main four components:

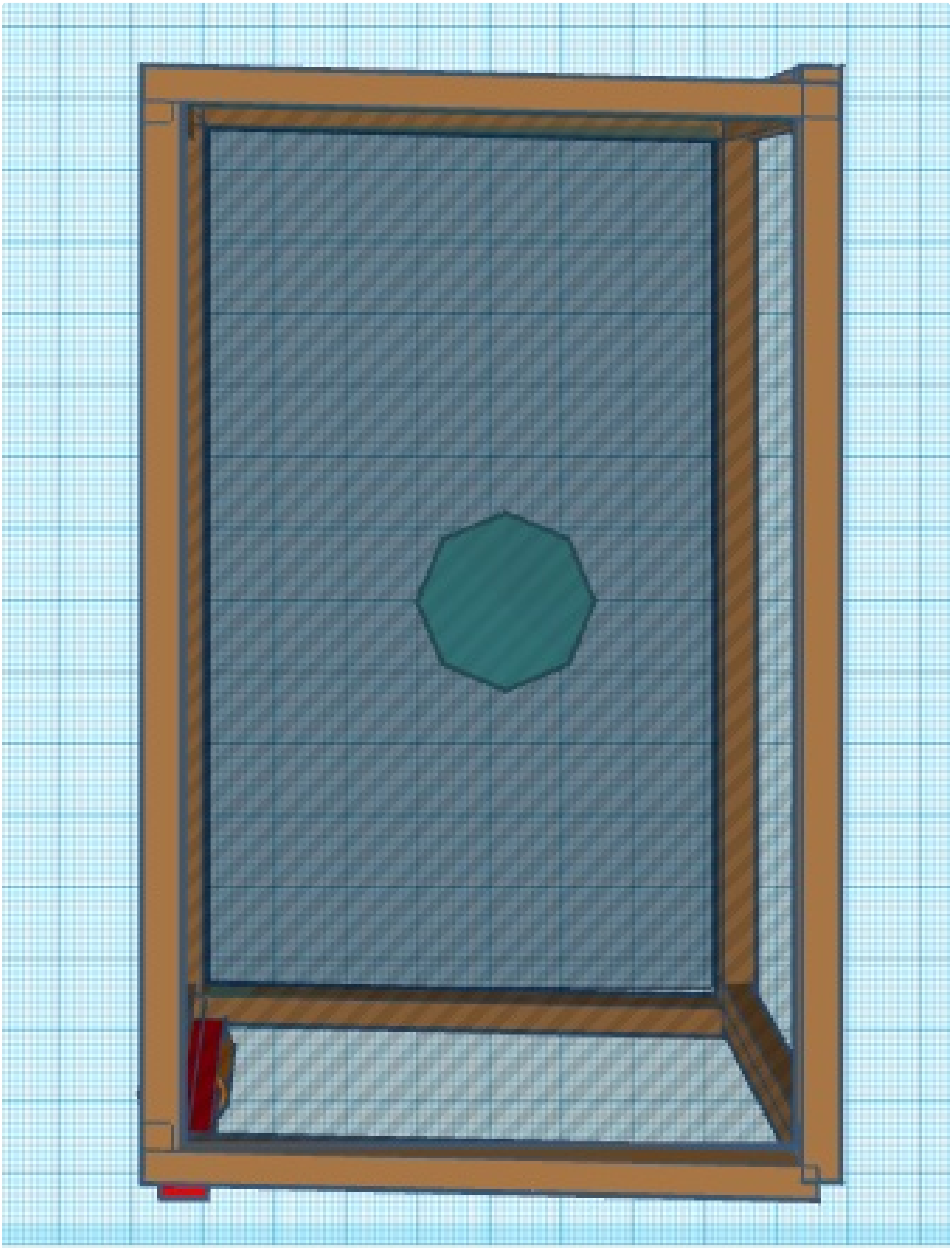
Step 1: 1) Flower Tank/ Terrarium:

Flower terrariums are usually plastic or glass containers that are filled with plants. They are compatible when it comes to storing small indoor plants without creating a mess with minimum effort and care. They are either open or closed, open ones are good for humanity and temperature control.

Our designed flower terrarium is a rectangle-shaped structure which is open to provide air and humidity to plants as well as bees. It has two additional components: Led sensor light and water pump. Led sensor light's functionality is to keep the bees warm during low temperatures and water pump pumps fresh cold water to keep the bees cold during extremely hot temperatures as well as water the plants.



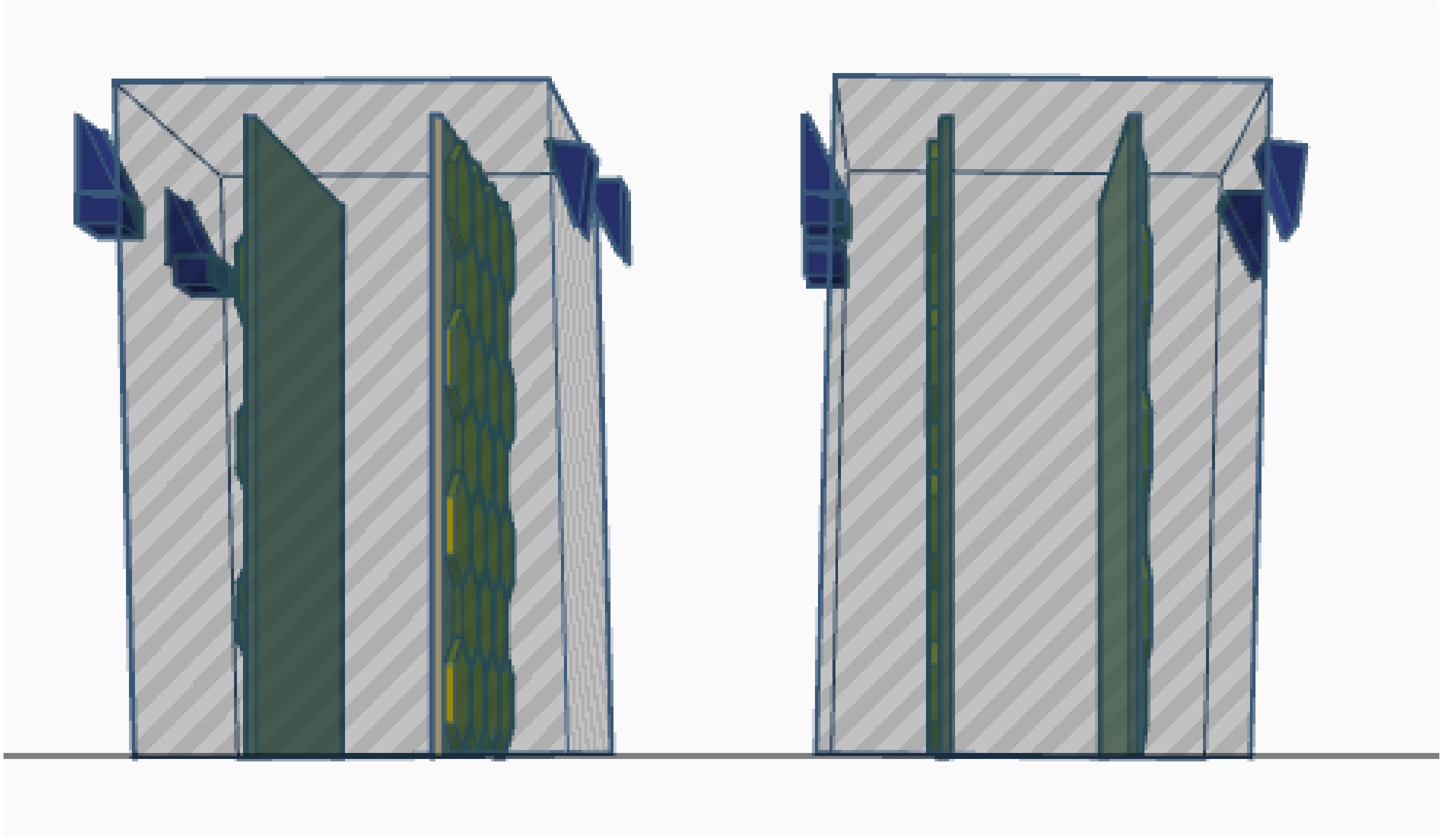


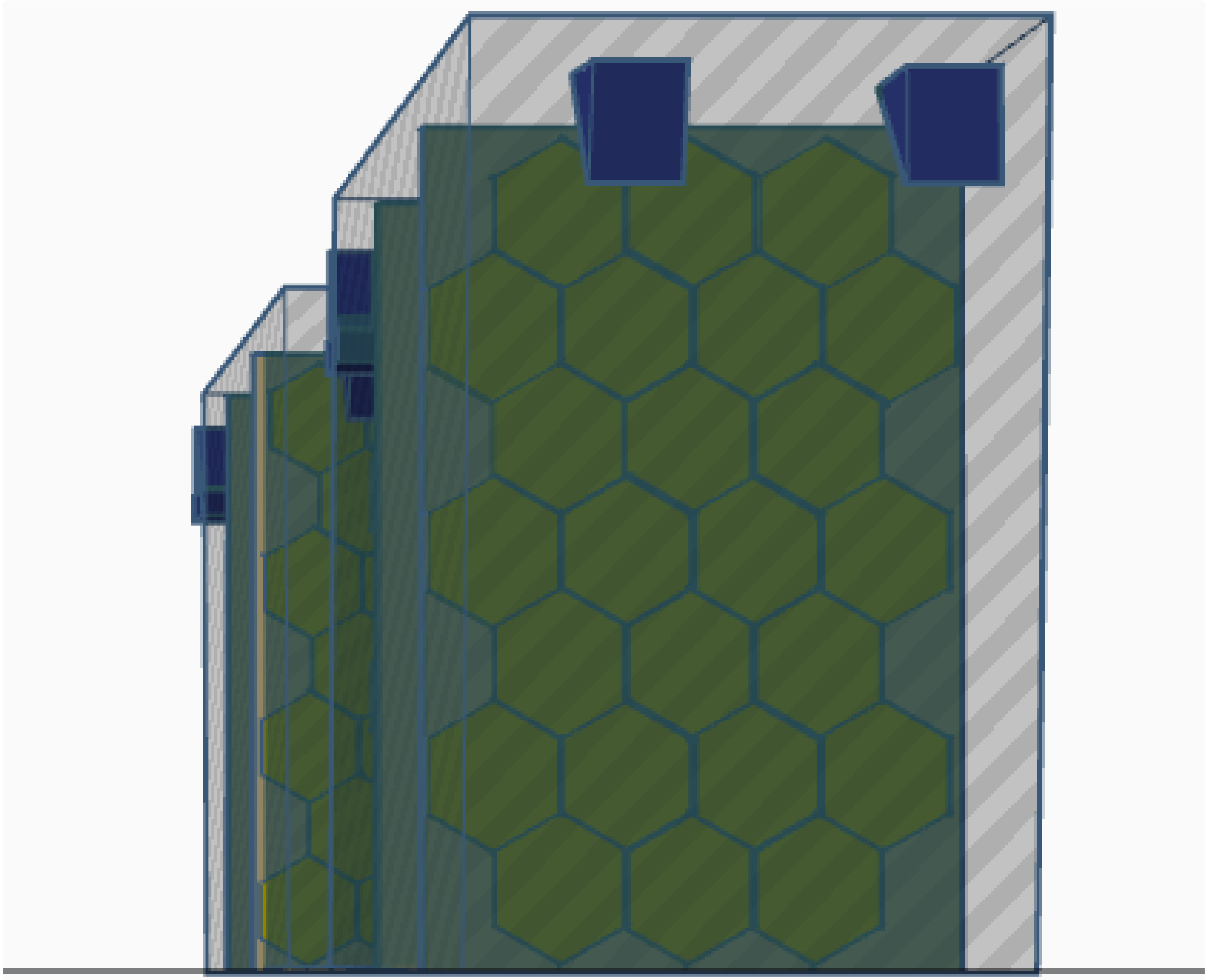


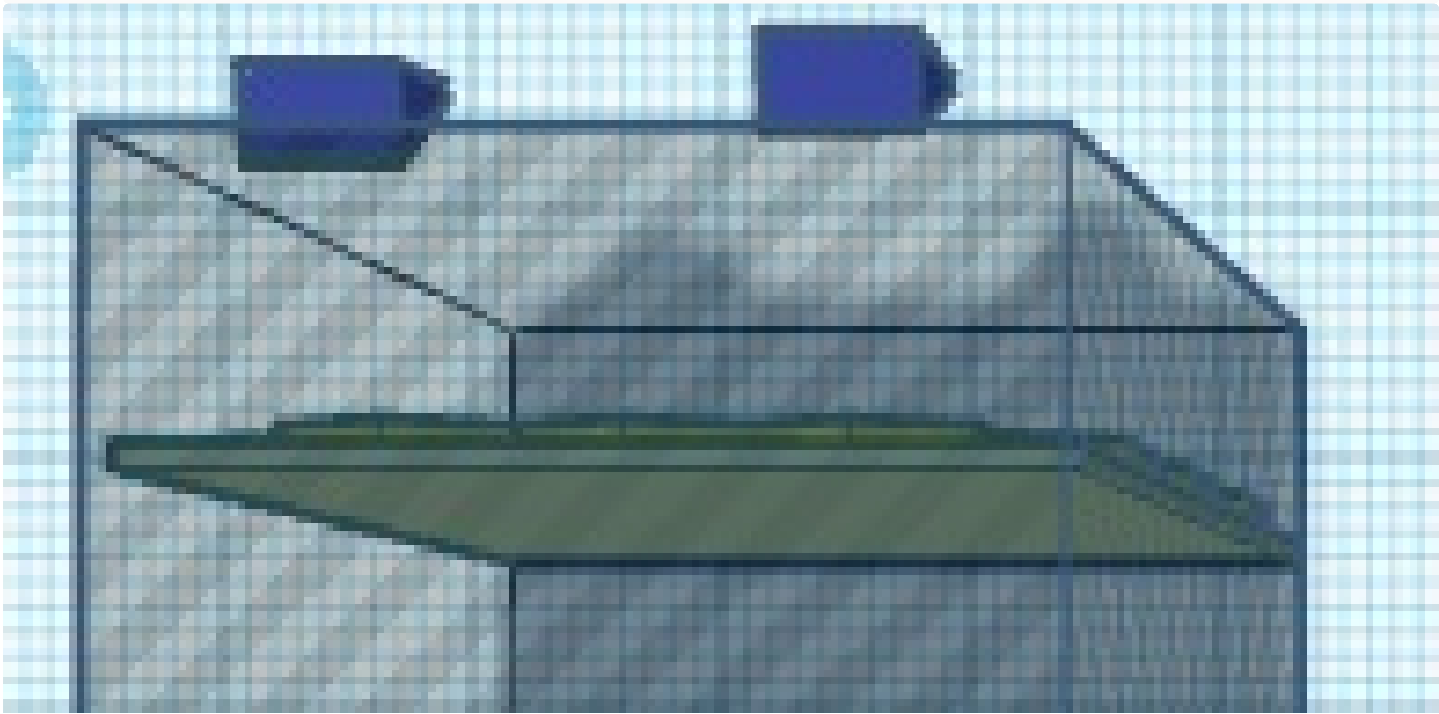
Step 2: 2) the Bee-hive

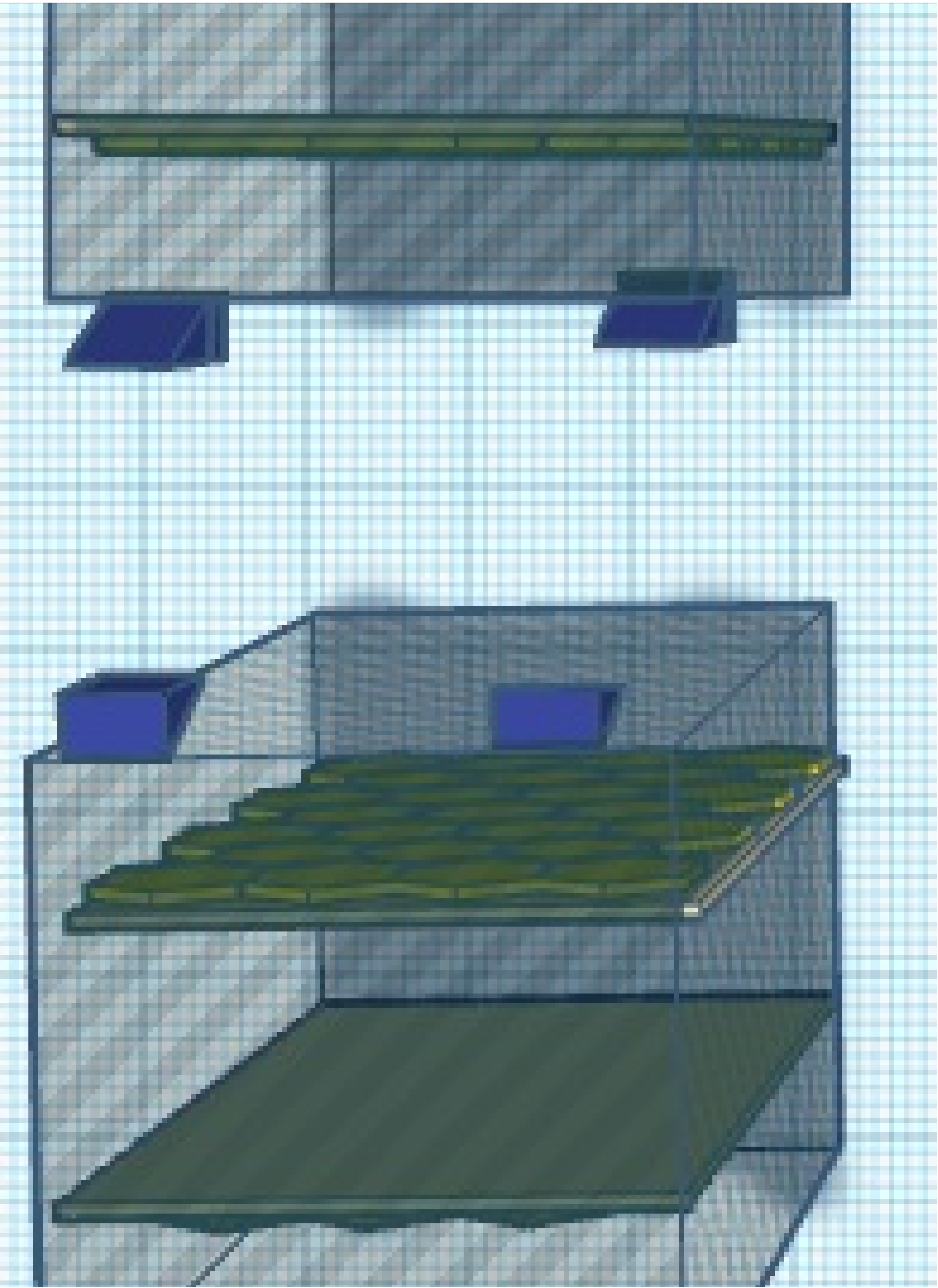
Beehives are colonies for the bees where they stay and store honey. There are various types of man-made beehives. Our bee-hive is a vertical structure with aerated openings to provide air to the bees.

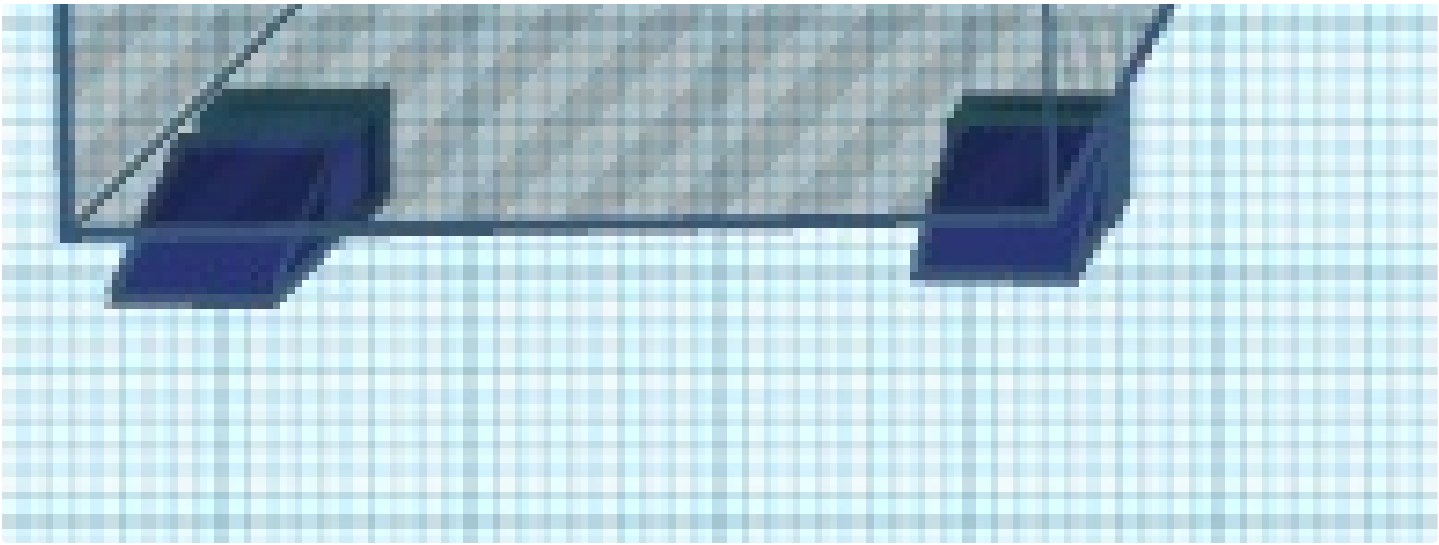
The vertical bee-hives are the most commonly used ones when it comes to bee-keeping. They can be easily designed and are readily available. They are easy to maintain with interchangeable parts. Its ideal for honey production and the honey can be easily collected without disturbing the bee colony. We added an additional feature where the outer structures of our bee-hives have clips where they can be attached together with the beehives to lock with each other and maintain stability.

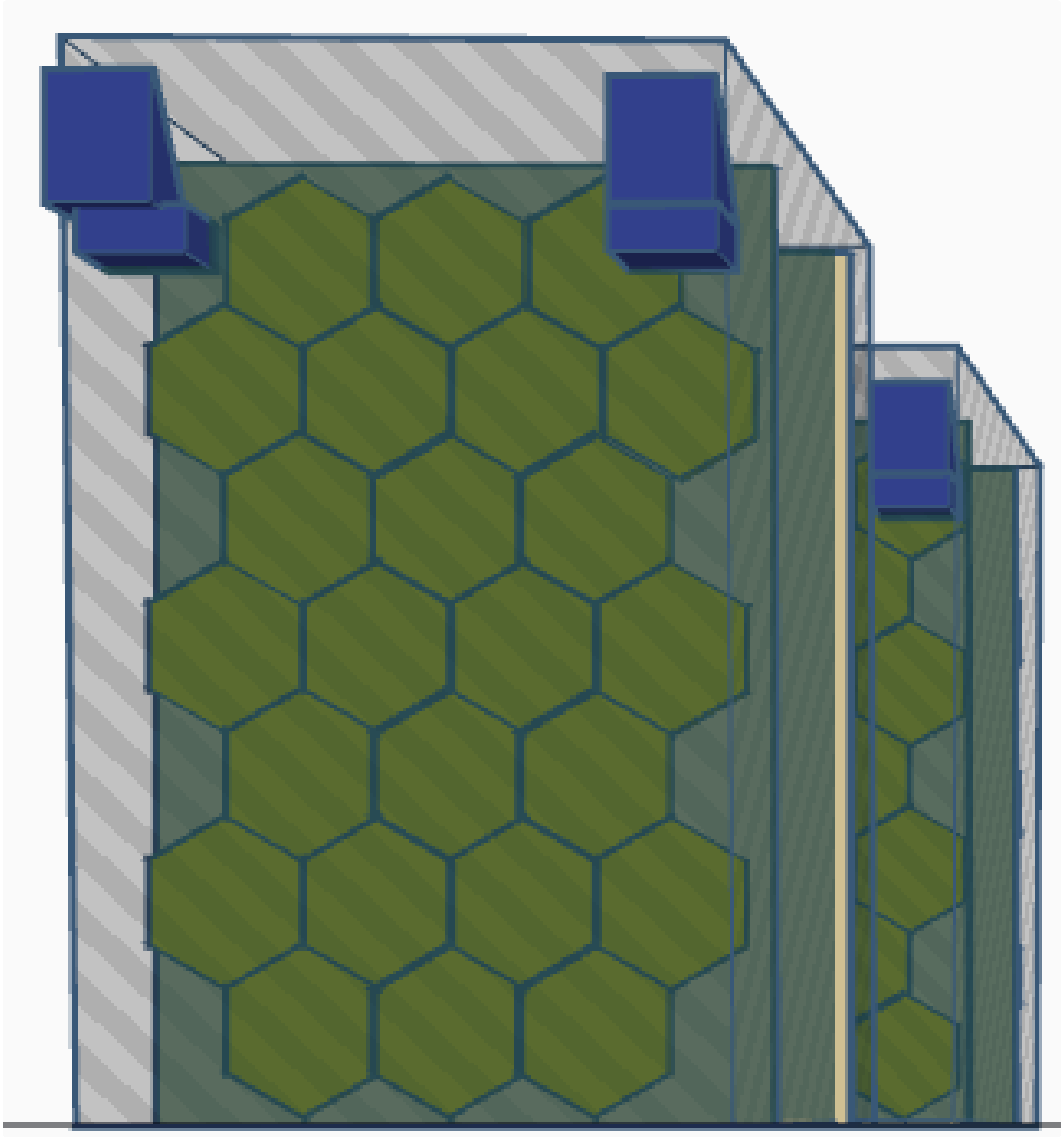












Step 3: 3) Water+fertilizer Pump With a Sensor

A relay controlled automated water pump and a temperature control.

Here is an example of an automated water pump and a heater unit controlled by a moisture and temperature sensor. Let's talk about the water side of things first.

The hardware required is the wiring, a 9V battery, a moisture sensor, a dc water pump (DC motor used in the example), a spdt relay module, an Arduino uno and a potentiometer (optional). Here's the how to build:

Connect the Arduino ground (GND) pin to the lower common pin (terminal 12 in TinkerCad spdt relay).

Connect the Arduino pin No. 13 to the upper coil pin (terminal 5).

Connect the Arduino 5V pin to the lower coil pin (terminal 8).

Connect the positive pin from the 9V battery to the upper common pin (terminal 1).

Connect potentiometer VCC to the NO pin (terminal 6).

Connect the potentiometer out to the DC pump positive (wiper in TinkerCad).

Connect the DC pump to the 9V negative pin.

Connect the moisture sensor ground pin to Arduino GND.

Connect the moisture sensor power pin to the Arduino 3.3v pin .

Connect the moisture sensor out pin to Arduino pin 8.

Then we will talk about fire. Or heat in this instance. Here's what you will need: the wiring, a heating unit (lightbulb used in the example), a spdt relay module and a temperature sensor. Here's the build:

connect the relay as explained earlier except the upper coil pin goes to Arduino pin 12.

Connect the heating unit to the 9V negative pin.

Connect the temperature sensor ground to Arduino GND.

Connect the temperature sensor power pin to the Arduino 3.3v pin.

Connect the temperature sensor out pin to the Arduino pin analog 0 (A0).

Here's the code. "//" means that the words after these don't affect the code and are there just so the code can be explained.:

```
int Relay = 13; //relay module connected to arduino pin 13
int sensor = 8; //soil sensor connected to arduino pin 8
int sensor2 = A0; //temp sensor connected to arduino pin a0
int relay2 = 12; //second relay connected to arduino pin 12
int sensorInput; //The variable we will use to store the sensor input
double temp; //The variable we will use to store temperature in degrees.
int val; //this variable stores the value received from the soil moisture sensor
```

Then we set the pins to do the desired job by setting up the void (scary, huh).

```
void setup() {
Serial.begin(9600); //for monitoring the temperature values
pinMode(12,OUTPUT); //this sets the Arduino pin 12 to send data outwards, to a relay in this case.
pinMode(A0,INPUT); //this sets the Arduino pin Analog 0 (A0) to receive data.
pinMode(13,OUTPUT); //Set pin 13 to send data outwards, to a relay in this case.
pinMode(8,INPUT); //Set pin 8 as input pin, to receive data from Soil moisture sensor.
}
```

Now we'll code the system to constantly loop a code:

```
void loop() {
```

After that we'll make the Arduino read the data that the moisture sensor is delivering (pin 8). If the value is "low" the Arduino will send a signal to the relay which will let power through to the DC water pump.

```
val = digitalRead(8);
if(val == LOW)
{
digitalWrite(13,HIGH); //if soil moisture sensor provides LOW value send LOW value to relay
}
else
{
digitalWrite(13,LOW); //if soil moisture sensor provides HIGH value send HIGH value to relay
}
```

Then we will include the temperature part. This one requires some math as well. The data from the temperature sensor is not readable as is so we have to make the Arduino run some calculations to make it convert the data to degrees Celsius.

```
}
sensor2 = analogRead(A0);
temp = (double)sensor2 / 1024; //find percentage of input reading
temp = temp * 5; //multiply by 5V to get voltage
temp = temp - 0.5; //Subtract the offset
temp = temp * 100; //Convert to degrees
```

Then we will set the Arduino to read the analog pin 0. The Arduino has been set to run the incoming data through the calculation prior to reading it. If the read value is under the set value (18c in this case) the Arduino will send a signal to the relay and it will let power through to the heating unit.

```
val = analogRead(A0);
if(temp < 18)
{
digitalWrite(12,LOW); //if temp is below 18 send LOW value to relay
if(temp == 18) {
digitalWrite(12,HIGH);
}
```

If the data is not under the set value (18c) the Arduino will close the relay so the heating unit will not receive any power. After that we have a small delay so the code doesn't clog itself up.

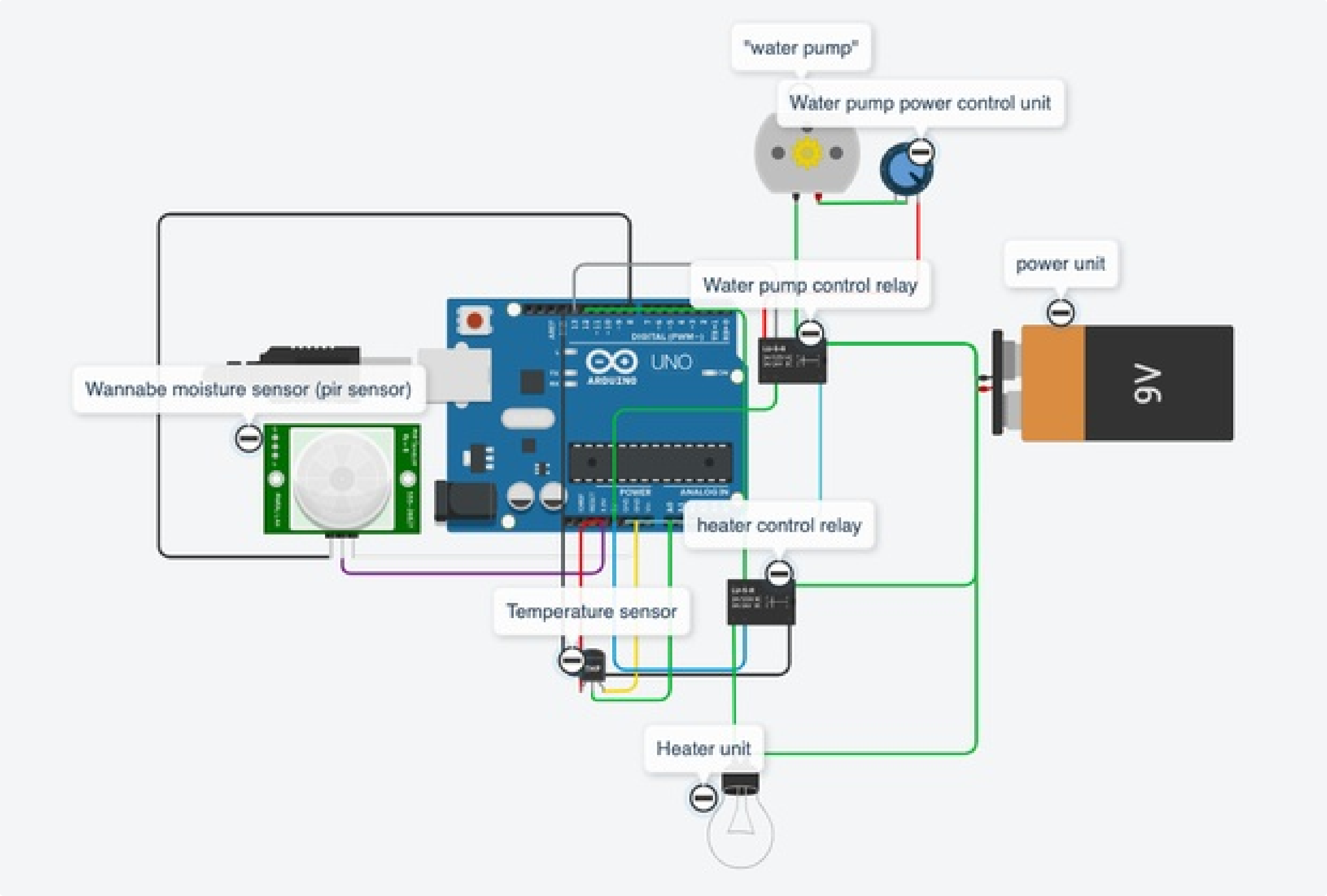
```
} else {
digitalWrite(12,HIGH); //if temp is over 18 send HIGH value to relay
}
delay(400);
```

This line feeds the data of the temperature sensor to the serial monitor so you can check everything is ok.

```
Serial.print("Current Temperature: ");
Serial.println(temp);
}
```

And so we close off the code. Simple as that. There's a picture of the circuitry at the end and here's the code as a whole:

```
int Relay = 13; //relay module connected to arduino pin 13
int sensor = 8; //soil sensor connected to arduino pin 8
int sensor2 = A0; //temp sensor connected to arduino pin a0
int relay2 = 12; //second relay connected to arduino pin 12
int sensorInput; //The variable we will use to store the sensor input
double temp; //The variable we will use to store temperature in degrees.
int val;
void setup() {
Serial.begin(9600);
pinMode(12,OUTPUT);
pinMode(A0,INPUT);
pinMode(13,OUTPUT); //Set pin 13 as OUTPUT pin, to send signal to relay
pinMode(8,INPUT); //Set pin 8 as input pin, to receive data from Soil moisture sensor.
}
void loop() {
val = digitalRead(8);
if(val == LOW)
{
digitalWrite(13,HIGH); //if soil moisture sensor provides LOW value send LOW value to relay
}
else
{
digitalWrite(13,LOW); //if soil moisture sensor provides HIGH value send HIGH value to relay
}
sensor2 = analogRead(A0);
temp = (double)sensor2 / 1024; //find percentage of input reading
temp = temp * 5; //multiply by 5V to get voltage
temp = temp - 0.5; //Subtract the offset
temp = temp * 100; //Convert to degrees
val = analogRead(A0);
if(temp < 18)
{
digitalWrite(12,LOW); //if temp is below 18 send LOW value to relay
if(temp == 18) {
digitalWrite(12,HIGH);
}
} else {
digitalWrite(12,HIGH); //if temp is over 18 send HIGH value to relay
}
delay(400);
Serial.print("Current Temperature: ");
Serial.println(temp);
}
```

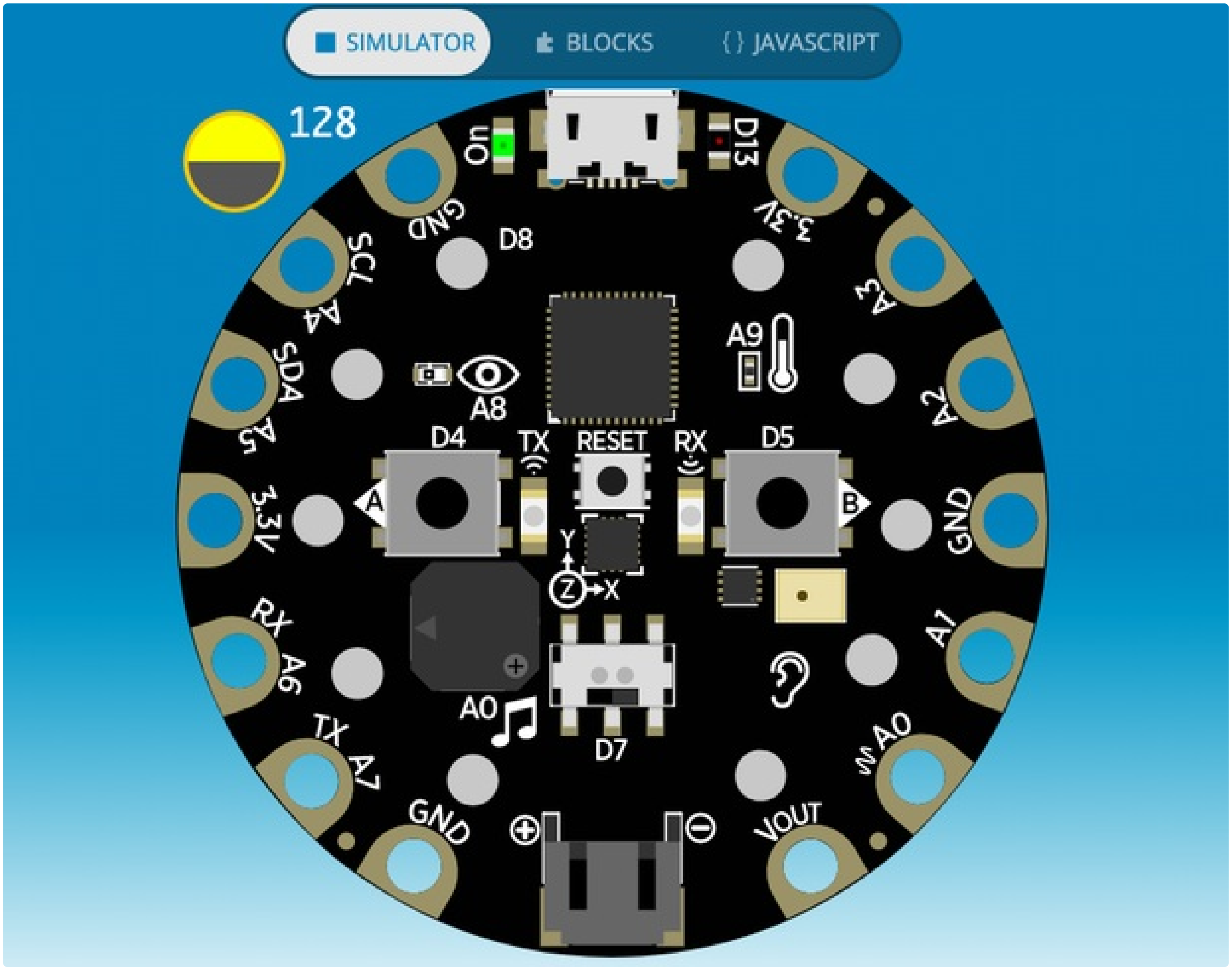


Step 4: 4) LED Light With Sensor

One solution to providing proper lighting for plants could be an LED- strip fitted with an ambient light sensor. We came up with a simple way of doing it with an Adafruit circuit:

Here a light sensor will constantly be looping a line of code. It's checking the light levels and when the light level goes below 40 units, it will tell the LEDs to light up.

This device just uses very simple and basic codes, so you can craft it in no time. Here is the link for the Makecode Adafruit codes: https://makecode.com/_DgkeTuaDdTue



Thanks for sharing!