PRO- SOLAR ENERGY UTILIZATION IN AUTOMATED GREENHOUSE

WhiteHat Jr
Live Online Coding for Kids

SOLAR ENERGY UTILIZATION IN AUTOMATED GREENHOUSE

What is our GOAL for this CLASS?

In this class, we learned the application of solar energy in agriculture in remote areas and how to make use of technology to automate the farming process via simulation.

What did we ACHIEVE in the class TODAY?

- Learned about climate issues faced in Ladakh.
- Understood the role of solar panels with the help of greenhouse Simulation.
- Used solar power to maintain the temperature in greenhouse

Which CONCEPTS/ CODING BLOCKS did we cover today?

- **overlap()**
- **image()**
- **Renewable Solar Energy.**

PRO- SOLAR ENERGY UTILIZATION IN AUTOMATED GREENHOUSE

How did we DO the activities?

Create a simulation to manage the temperature inside a greenhouse. During the daytime, the temperature needs to be maintained below 30 deg by powering exhaust fans with power generated from Solar energy.

1. Declare the variables for loading images, creating sprites and groups.
    - Create a variable for temperature as **temp** and set it to **10.**
    - Create two variables **panel1_volatage** & **panel2_voltage** assign them as **0**.
    - Create variable **power_gen = 0**; where we can store the total voltage generated.

```
//Global variables for images
var bg, sun, s_pan, fan_anim,fan_img,display, g_house_img;

//Global variables for Sprites
var g_house, pan1,pan2,fan,fan2;

//Creating a ray group
var rayGroup;

//Creating temprature and voltage variables
var temp = 10
var panel1_voltage =0;
var panel2_voltage = 0;
var power_gen = 0;
```

2. In function **preload()** load the images using **loadImage**.
    - For **fan_anim** create animation using multiple images and **loadAnimation.**

```
function preload()
{
  sunR = loadImage("sunrays.png");
  sunL = loadImage("sunrays1.png");
  bg = loadImage("bgimage.png")
  s_pan = loadImage("s_panel.png");
  fan_anim = loadAnimation("fan01.png","fan02.png","fan03.png","fan04.png","fan05.png");
  fan_img = loadImage("fan01.png");
  display = loadImage("disp.png");

  g_house_img = loadImage("greenhouse.png")
}
```

3. In function **setup()** declare the sprites for **g_house, pan1, pan2, fan1, fan2**.
   - Add images to each sprite.
   - Scale the images as required.
   - Create a group for **raysGroup.**

```
function setup()
{
  createCanvas(800, 500);

  g_house = createSprite(380,300,100,100);
  g_house.addImage(g_house_img);
  g_house.scale = 0.75;
  g_house.debug = true;
  g_house.setCollider("circle",-10,0,185)

  pan1 = createSprite(100,height-50,80,80);
  pan1.addImage(s_pan);
  pan1.scale = 0.75;

  pan2 = createSprite(width-150,height-50,80,80);
  pan2.addImage(s_pan);
  pan2.scale = 0.75;


  fan = createSprite(280,300,20,20);
  fan.addImage(fan_img);
  fan.scale = 0.3;
  fan.addAnimation('run',fan_anim);

  fan2 = createSprite(450,300,20,20);
  fan2.addImage(fan_img);
  fan2.scale = 0.3;
  fan2.addAnimation('run',fan_anim);
  textSize(15);

  raysGroup = createGroup()
```

4. Inside function **draw():**
    - Set the background image using the **image()** function.
    - Set the display image to display the temperature and power generated.
    - Use **text()** to display the values of temperature and power generated.
    - Use **push()** and **pop()** to retain styling given to text only for display text.
    - Call function **makeRay()** to generate random sun ray sprites.

```
function draw()
{
  background(220);

  image(bg,0,0,width,height);
  image(display,600,10,200,60)
  power_gen = panel1_voltage + panel2_voltage

  push();
  noStroke();
  fill(255,255,0)
  text("Voltage : ",620,37)
  text(power_gen,680,37)

  text("Temprature : ",620,56)
  text(temp,710,56);
  pop();

  makeRay();
  drawSprites();
}
```


To show animated sun rays, create two sprites and add a yellow line image to them, which will fall randomly over the entire simulation.

5. Create a function **makeRay()** outside function **draw()**
    - Create 2 sprites **raysL** & **raysR** at random **x** positions, after every 60 **frameCount.**
    - Add images to both the sprites.
    - Add both sprites to the same group, **raysGroup().**
    - Set velocity x to random from **(-1,1)**.
    - Assign velocity x and y to all the sprites in the group.
    - Assign **lifetime** to each sprite in the group.
    - Use the **overlap()** function to check collision between **raysGroup** and greenhouse and two panels.

```
function makeRay()
{

  if (frameCount % 60 === 0)
   {
    var x = Math.round(random(10,350));
    rayL = createSprite(x,50,10,10);
    var xr = Math.round(random(350,750));
    rayR = createSprite(xr,50,10,10);
    rayL.addImage(sunL);
    rayR.addImage(sunR);
    rayL.scale = 0.08;
    rayR.scale = 0.08;
    vx = random(-1,1);
    raysGroup.add(rayL);
    raysGroup.add(rayR);
    raysGroup.setVelocityYEach(2)
    raysGroup.setVelocityXEach(vx)
    raysGroup.setLifetimeEach(134)
   }
  raysGroup.overlap(pan1,charge1);
  raysGroup.overlap(pan2,charge2);
  raysGroup.overlap(g_house,temp_rise)
}
```

Once the collision is detected between **raysGroup** and the panels or greenhouse, we want to remove that particular sprite from the simulation. The **overlap()** function allows us to do that.

raysGroup.overlap(pan1,charge1); This statement checks if the **raysGroup** is overlapping / touching **pan1**, then moves to the **charge1()** function.

6. Create functions **charge1(), charge2()** and **temp_rise()** after the **makeRay()** function, to take the necessary action based on the overlap between sprites.
    - Create global variables for **absorbed1** and **absorbed2**, which will be used to increase **power_gen**

```
var absorbed1= 0;
var absorbed2= 0;
```

- If the overlap is detected between **raysGroup()** and **pan1** or **pan2** Increment the value of **absorbed** by one and remove the particular sprite from the group using the **.remove()** function.

- If the overlap is detected between **raysGroup()** and **g_house** Increment the value of temp by one and remove the particular sprite from the group using the **.remove()** function.

- **sprA.remove()** and **sprb.remove()** will remove the sun rays from the group after the collision.

```
function charge1(sprA)
{
    sprA.remove()
    absorbed1+=1;
}
function charge2(sprA)
{
    sprA.remove()
    absorbed2+=1;
}
function temp_rise(sprb)
{
  sprb.remove();
  temp+=1;
}
```

- Inside function **draw()** multiply it by **0.15** and assign it to **panel1_voltage** & **panel2_voltage** voltage to slow down the increase in voltage.

```
panel1_voltage = round(absorbed1* 0.15);

panel2_voltage = round(absorbed2* 0.15);
```

7. Add the conditions to start the fans inside function **draw()**:
    - Start one fan if we have **power_gen >= 4** and temp **>= 30**.
    - Start both the fans if the **power_gen is >= 8** and **temp >= 30.**
    - When the conditions are met,
        i. Change animation of the fan to moving.
        ii. Reduce **temp** by **1**.

iii. Reduce panel voltage by **1**.

```
panel1_voltage = round(absorbed1* 0.15);

panel2_voltage = round(absorbed2* 0.15);


if(power_gen>=8 && temp>=30)
{
  fan.changeAnimation('run');
  temp-=1;
  panel2_voltage-=1
}

if(power_gen>=4 && temp>=30)
{
  fan2.changeAnimation('run');
  temp-= 0.5;
  panel1_voltage-=1
}
```

OUTPUT: *Link*.

PRO- SOLAR ENERGY UTILIZATION IN AUTOMATED GREENHOUSE

WhiteHat Jr
Live Online Coding for Kids

What's NEXT?

In the next class, we will continue to work on the Physics Engine and build the game Pirate Invasion.

Expand Your Knowledge:

To know more about the use of Solar Energy at the School campus in Ladakh : https://secmol.org/about/eco-friendly-living/renewable-energy.