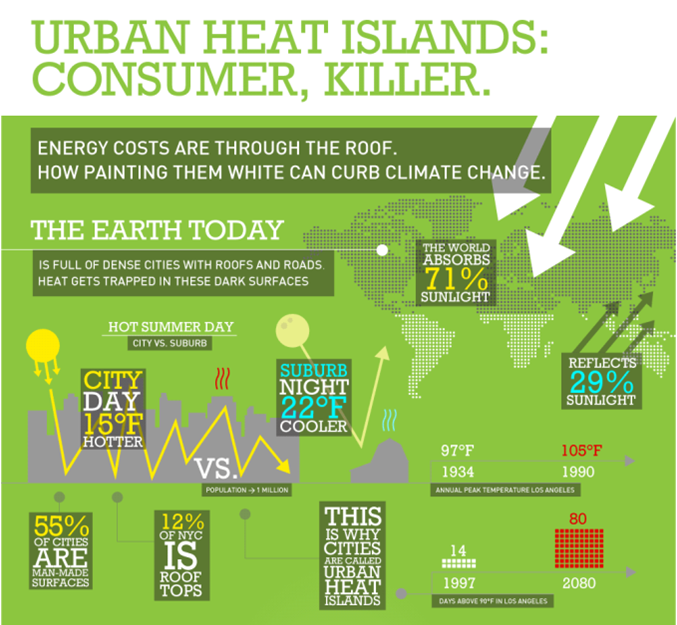
Starlogo TNG model for 2012 Supercomputer Challenge Kick-Off

Urban Heat Islands (UHI) are urban areas with significantly warmer temperatures than surrounding rural areas.

****

Energy consumption rises when the outside ambient temperature increases.

|  |  |
| --- | --- |
| **Urban Heat islands3.png** | **Urban Heat islands2.png** |

****

So, having a high percentage of pavement and black roofs contributes to global warming, makes the summer temperature hotter in cities both day and night, and causes an increased use of energy to cool buildings in cities. If that energy is from a fossil fuel source (coal, oil, gas), producing and using that energy may also independently contribute to global warming.

If we paint roofs white, they will reflect more sunlight and so result in lower summer temperatures in cities (day and night), lower energy use, and so decrease global warming. The Starlogo model we will build is intended to model the relationship between black roofs and energy absorption, and how that could be changed by painting roofs white.

**BUILDING THE MODEL:**

**1. Set-up:** Use Edit Breeds to choose a sphere, and create one sun, yellow, size 5, at an elevation of 30 in the center of the world. Use clear everyone.



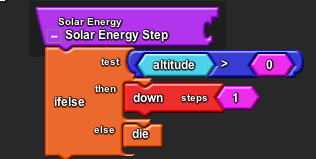
**SAVE PROJECT!**

**2) Procedures:**

**a) Create solar energy:** Next, create a new breed called solar energy (same shape), and then create a procedure for the sun to hatch the solar energy, make the agents small, and place them randomly in the sky.



**b) Make solar energy fall to earth**. Create a solar energy procedure to check altitude, and fall until the solar energy drops hit the ground. Be sure to make the solar energy die after it hits the ground, or you will quickly have too many agents to allow the program to run.

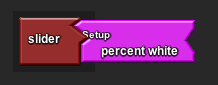


**3) Run:** In the setup and run drawer, get a run block, set a run time of 300 seconds, and call your procedures (in “My Blocks” get the procedure block from the drawer for each procedure created so far).

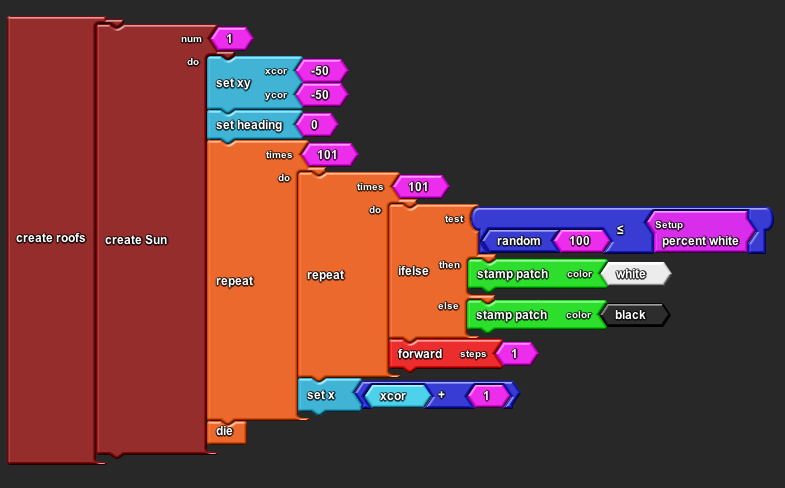
**TIME TO SAVE NEXT VERSION!**

**4. Heat variables:** Get a shared number variable from the variables drawer, and name it white energy, and get a second one and name it black energy. Add a set block for each variable to the set-up procedure.

**5. Slider for white roofs:** Create a slider (get a slider button from the set-up and run drawer, attach a shared number black from the variables drawer), and name it percent white.

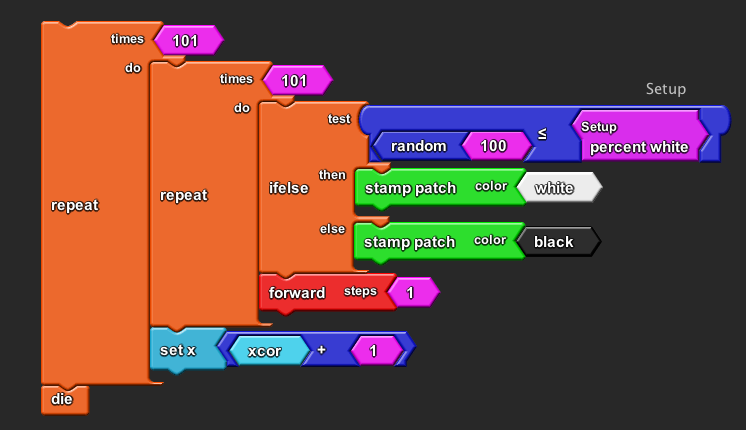


**6. Terrain builder**: Here and in class at the kickoff, we created a separate set up block for the terrain builder. Once you are ready to run experiments with your model, you might combine the two into a single setup block, so you are sure that all setup procedures are executed every time you restart the model. (Delete the second set-up “create roofs” block and move the “create sun” and everything inside it right under the first “create sun” block in setup).



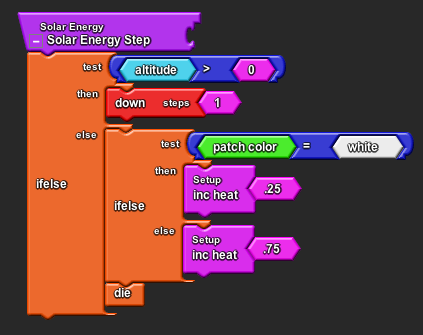
**TIME TO SAVE NEXT VERSION!**

In case that is too hard to read, here are just the repeat blocks:



Be sure to use stamp patch instead of stamp --if you don’t see stamp patch, you are using Starlogo 1.2 instead of Starlogo 1.5. It should still work, but you should download the newer version anyway.

**7. Create heat:** In the solar energy step procedure, add a conditional block to make the agents check the patch color, add a small fraction of heat, and die:



**8. Graphs and monitors**: Next, get a bar graph and two monitors from the setup and run drawer, and put them into the runtime column. Get the black and white energy blocks from My Blocks, runtime drawer and put them into the graph and monitors. Don’t forget to label the monitors, the graph itself, and the two variables to measure the black and white energy. If you want the graph and monitor to clear on set-up, add a “reset clock button” from the setup and run drawer to the end of the set-up procedure.

**TIME TO SAVE NEXT VERSION!**

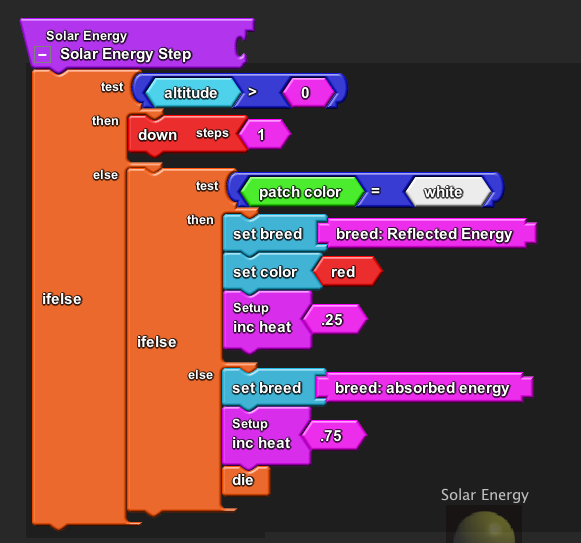
You should have agents and graphs cleared on setup, black and white terrain based on your slider setting when you hit create roofs, and on run, you should see solar energy being hatched and falling to the ground, and black and white heat being added to your graphs and monitors.

Some possible modifications of this model, to fit the real world:

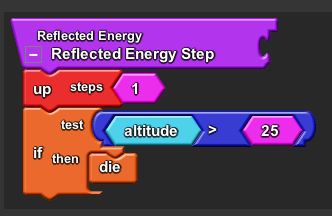
**1. Modifications to make the model more visual:**

There are two easy ways to make this model look flashier, and they might also help with some of the substantive modifications discussed later. First, instead of simply having the solar energy add heat then die when it hits the ground, you can have it change into a different breed and act accordingly. First, create two new breeds using edit breeds: reflected energy and absorbed energy.

**1. Show reflected energy:** In the solar energy step, have the solar energy set its breed to reflected energy when it hits a white patch, and set the color to red. If the roof is not white (in the else), have the solar energy set its breed to absorbed energy, and move the die block to apply to this breed only.



Then, make a procedure for the reflected energy to go up from the ground before it dies, and call this procedure in your run block.



This shows the sunlight bouncing up off the white roofs.

**2. Show heat in terrain**. Another way to make the program more visual is to start with gray and white roofs, and have the gray roofs turn black as the solar energy hits them. First, change the “create roofs” to create gray instead of black roofs. Then, add a block to the solar energy step procedure so the absorbed energy changes the color (add a stamp patch color patch color patch color -2) in the else procedure so that the gray roofs will look blacker each time one is hit by solar energy. Again, this is just a visual change to show what’s happening in the model, it doesn’t change the data collected. You have to use gray roofs because black is as dark a color as there is in Starlogo, it can’t get any darker. **Suggestions for some substantive modifications:**

1. **Heat agent:** Instead of heat variables you could use the absorbed energy as a heat agent, which moves along the surface for a while (you would probably want to hide this agent). If two heat agents collide, they could hatch another heat agent. You would need to have some limit to these heat agents, such as an age procedure after which they die, to make the model work. You would also need to change your system of data collection, so you are counting heat agents, rather than a heat variables.
2. **Albedo slider**: Vary the heat absorbed by black and white roofs (and gray, if you made that change above) to a percentage that is based on research. Use a slider to control those percentages to reflect this research and make experiments easy. The model is visually showing this in an unrealistic way (turning gray to black), but is not capturing information about how heat accumulates during the day.
3. **Cooling variable**: The model only shows heat increasing over a 5 minute period – in the real world, the urban heat islands heat up more than the surrounding rural areas during the day, and cool off less at night. Find a way to show this reality, by adding an age variable to the heat energy so it decreases slowly over time, or by adding a cooling variable which affects it.
4. **Show the cost of white roofs:** Research how much it costs to cool the urban buildings by traditional means (the use of traditionally produced electricity to power air conditioners to cool the temperature inside the buildings), and compare that to the cost of painting the roofs white. This might give some interesting feedback information, about the overall “cost” of black and white roofs, financially or in terms of climate change costs.
5. **Cooling roofs**: Alter this model to have black (absorbing) roofs, white (reflecting) roofs, and some type of green roofs -- either roofs that collect solar voltaic energy (to power air conditioning cooling) or literally green roofs, which grow plants that cool the atmosphere and absorb carbon dioxide. This requires research into how such roofs work in the real world, but would not require many programming changes to the model (you would need to create another slider for the percentage of cooling roofs, and change the terrain builder procedure to include it, as well as changing the graphs, or you could use it as a cooling variable (see #3)).