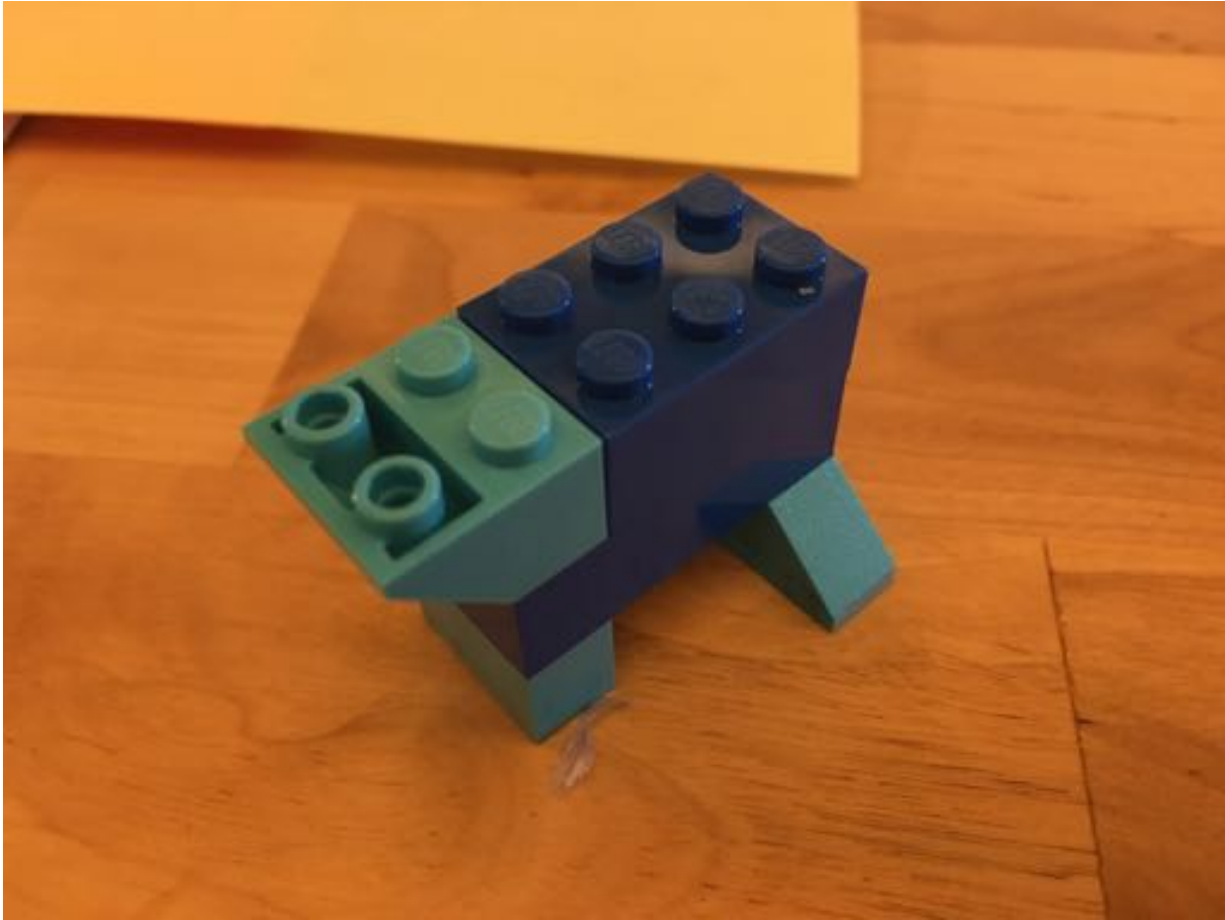
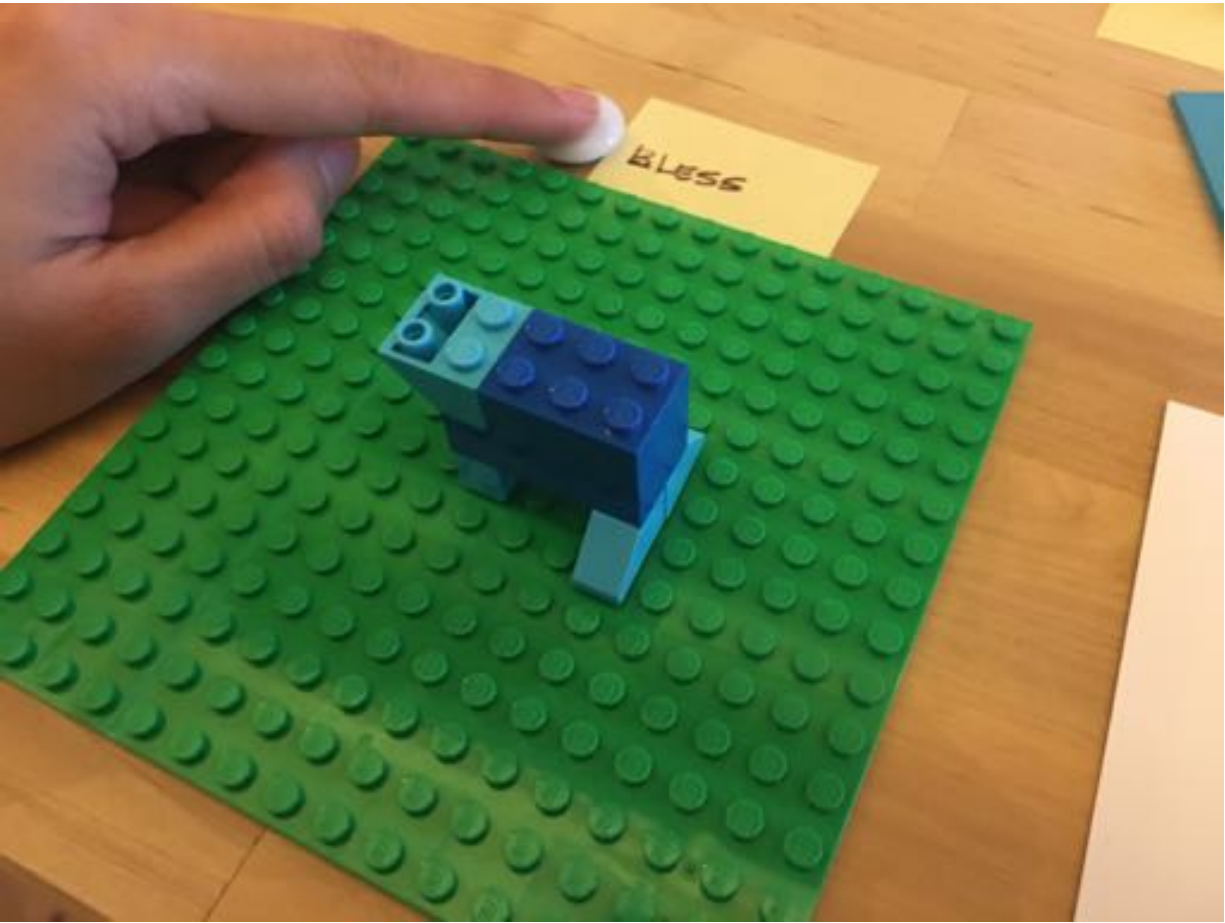


I mocked up a way to program a cat-litterbox simulation in Ometropolis.



I made a cat.



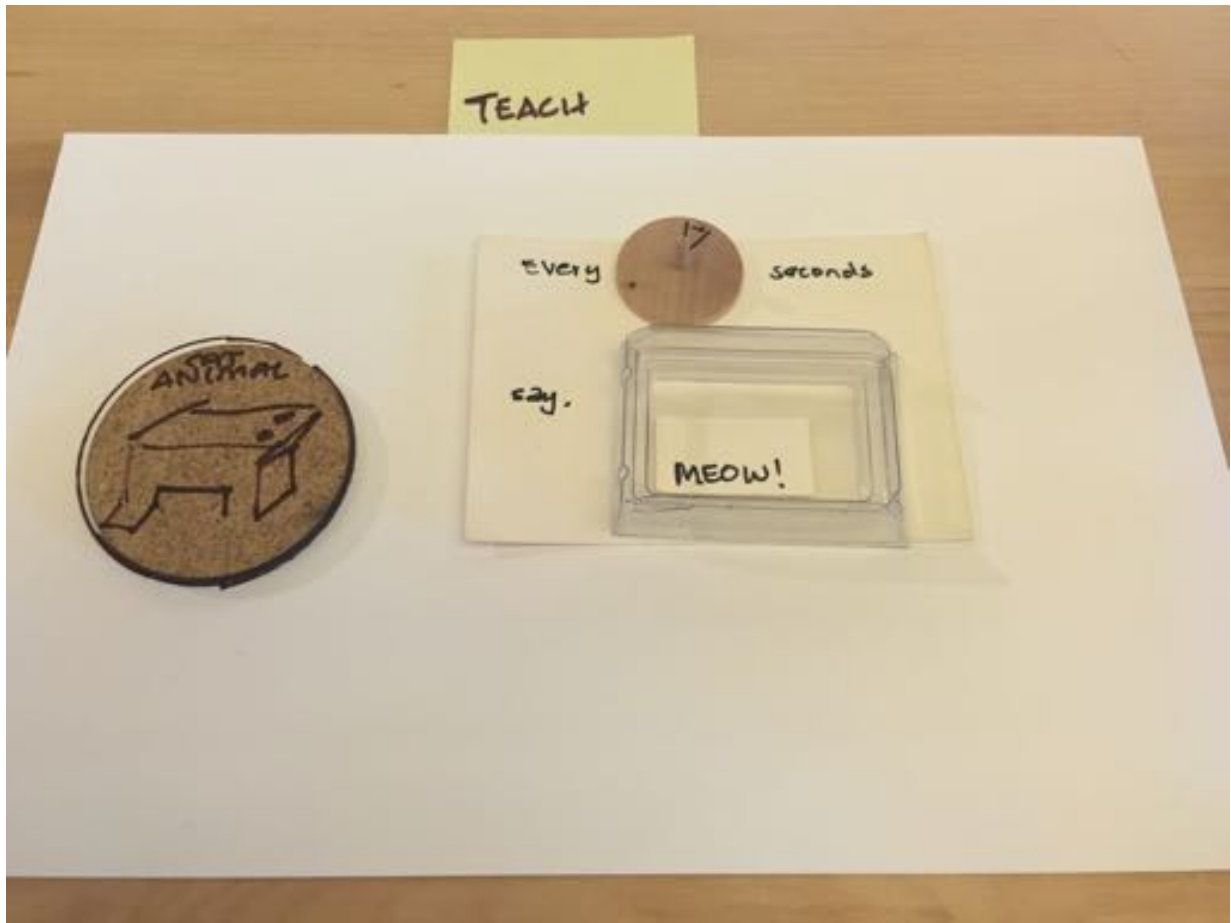
I put it on the "blessing" mat and press the button. This makes it a living agent.



“blessing” the object makes the system spit out a token/icon, which you can name by writing on it. I named it “cat animal.”



I came up with two scenarios for “programming” the agent. One is the “teaching” placemat, where you teach the agent how to do a thing.



Program a rule on a card. The card has a container for “sound files.” So I can record a sound, it spews out as a sound file, and I can label it “Meow!”



Adding more sound files to the container.

Another scenario is that the agents have “drives” or “goals.” So you don’t teach them what their drives or goals are, you just specify what they are.



This cat wants to meow very often.

DRIVES  
AMBITIONS



i have

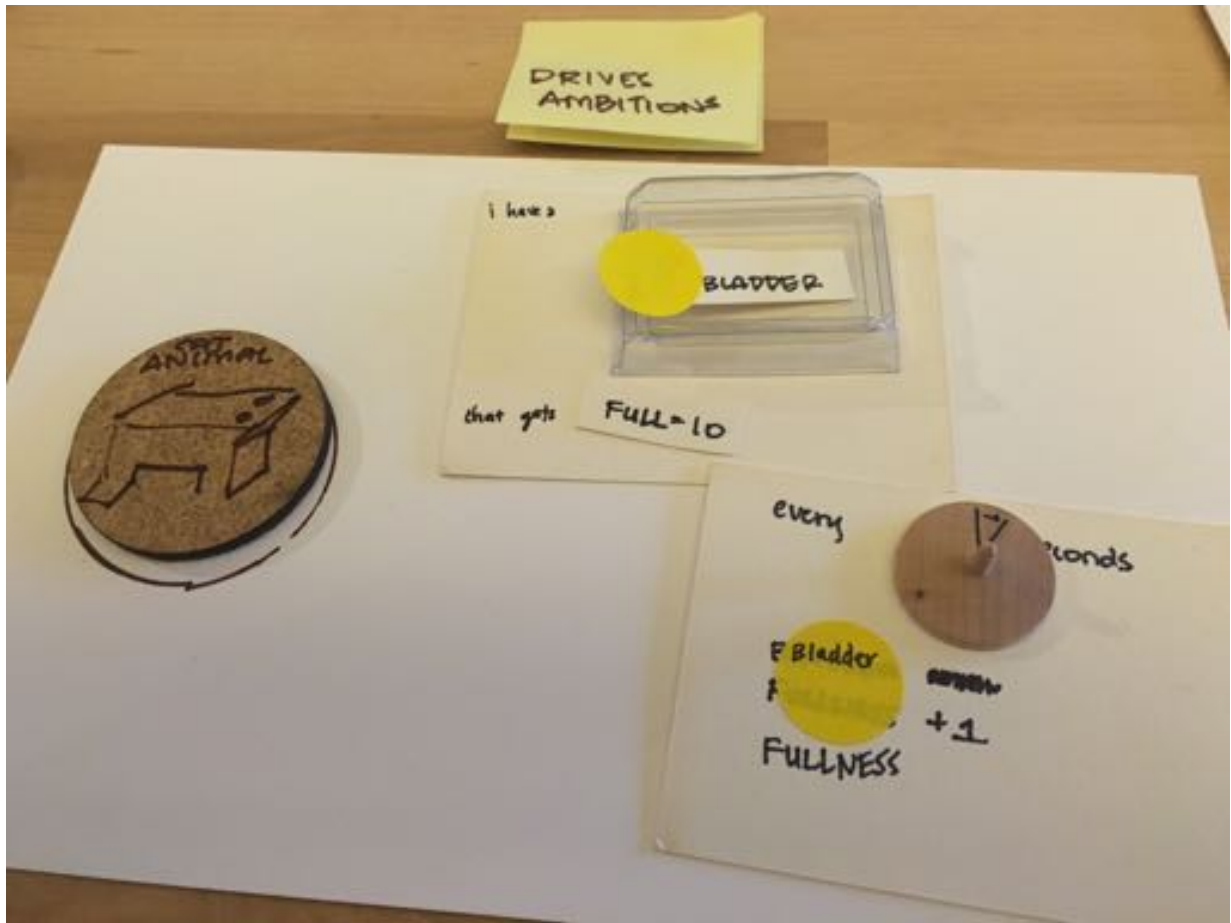


that gets

FULL = 10

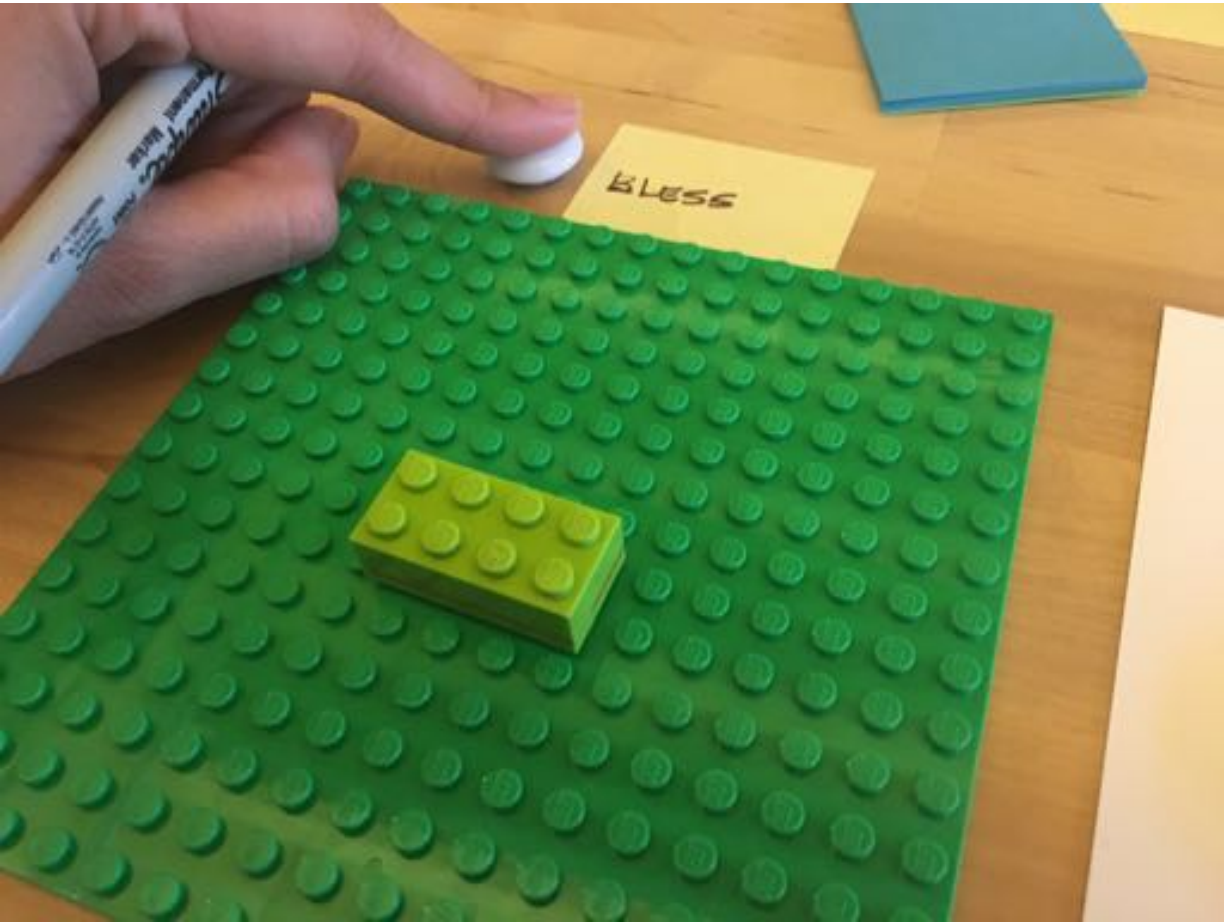
Now we show that the cat has a bladder that gets full at "10." Don't know what the units are here.





Every x seconds, the cat gets an increase in bladder fullness.

Now the cat needs a litterbox to empty its bladder when it is full.



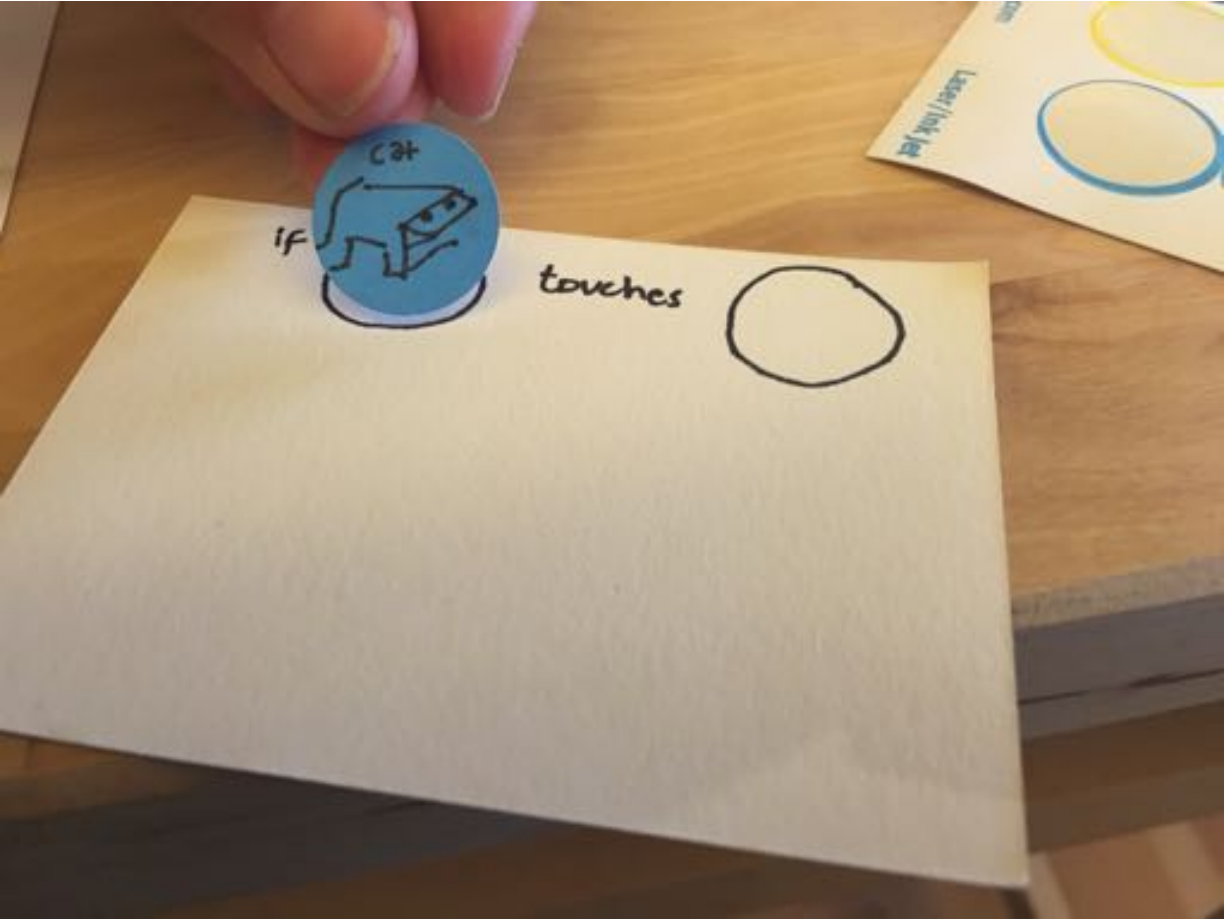
We bless the green object and name it litterbox.



Somewhere the system's also spewed out some sticker icons for us to use while programming.

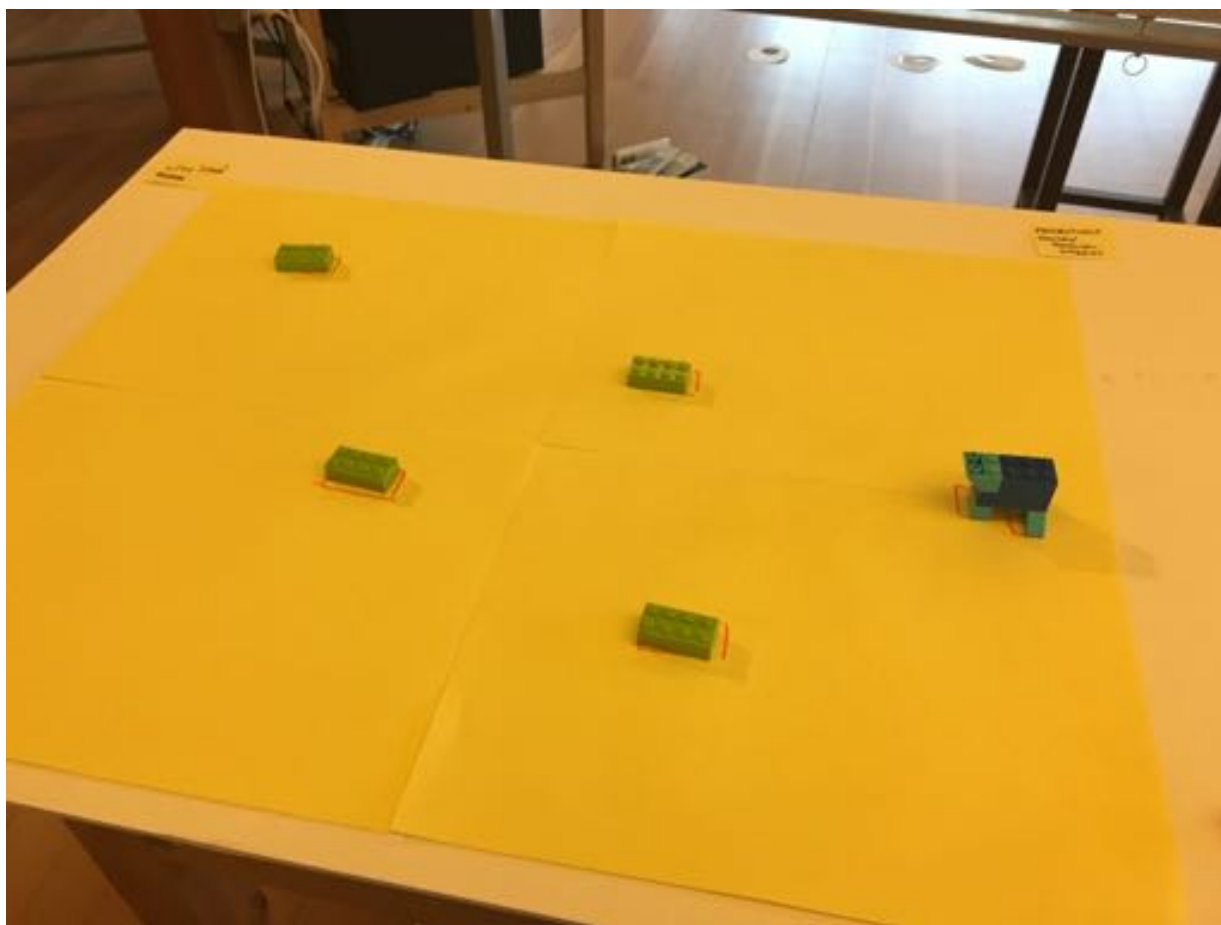


So now we can program interactions between agents.





Now I made a cat that has a bladder, which gets full every x seconds, and is emptied when it touches a litterbox.



You can place the cat on the playing field with the litterboxes. I didn't get very far in thinking about running the simulation, but we talked about it afterwards (which I'll get to later).

Here are some other fanciful features:

Agents can get behaviors added to them - like special abilities or personalities. This social cat hat makes the cat meow when it touches another cat.

SPECIAL ABILITIES / personal abilities



SOCIAL CAT HAT

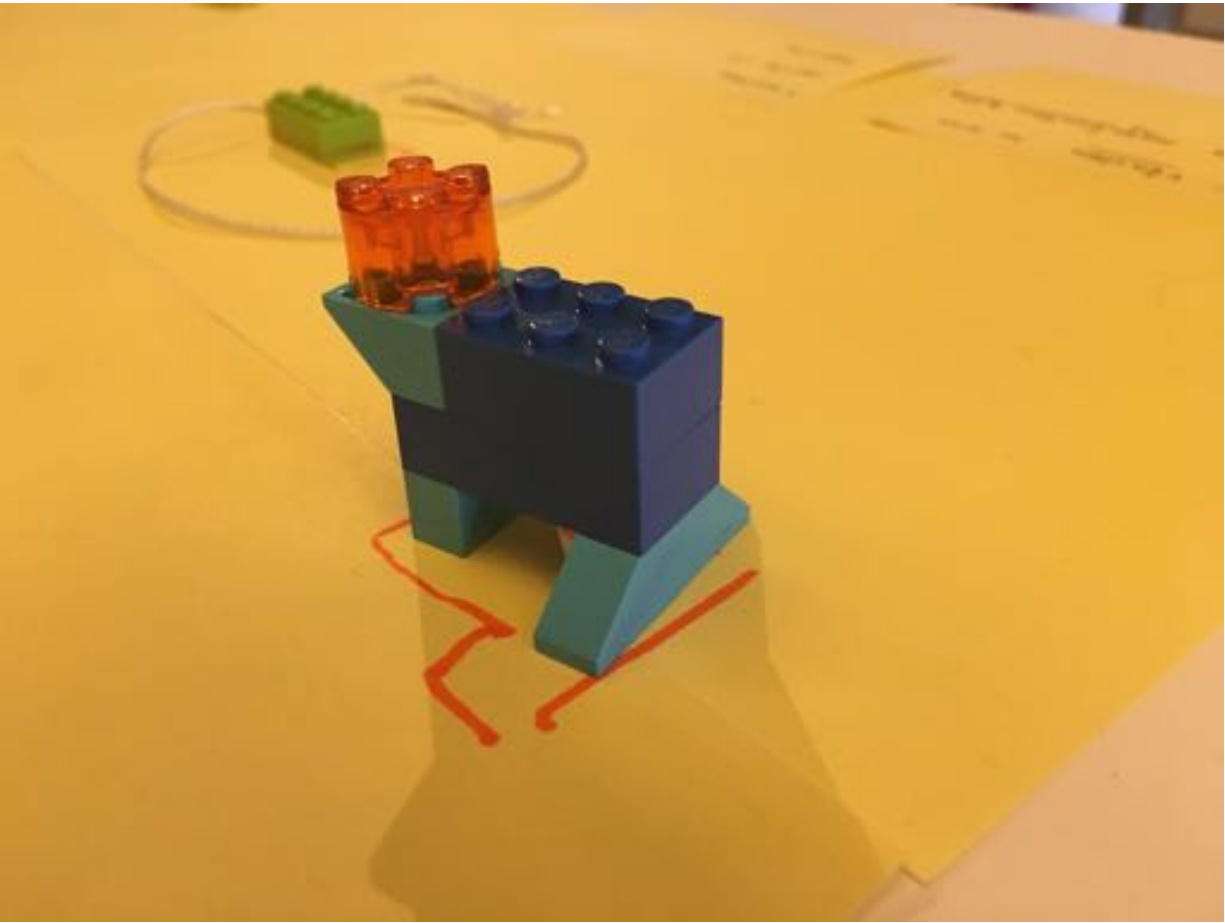
say "meow" when

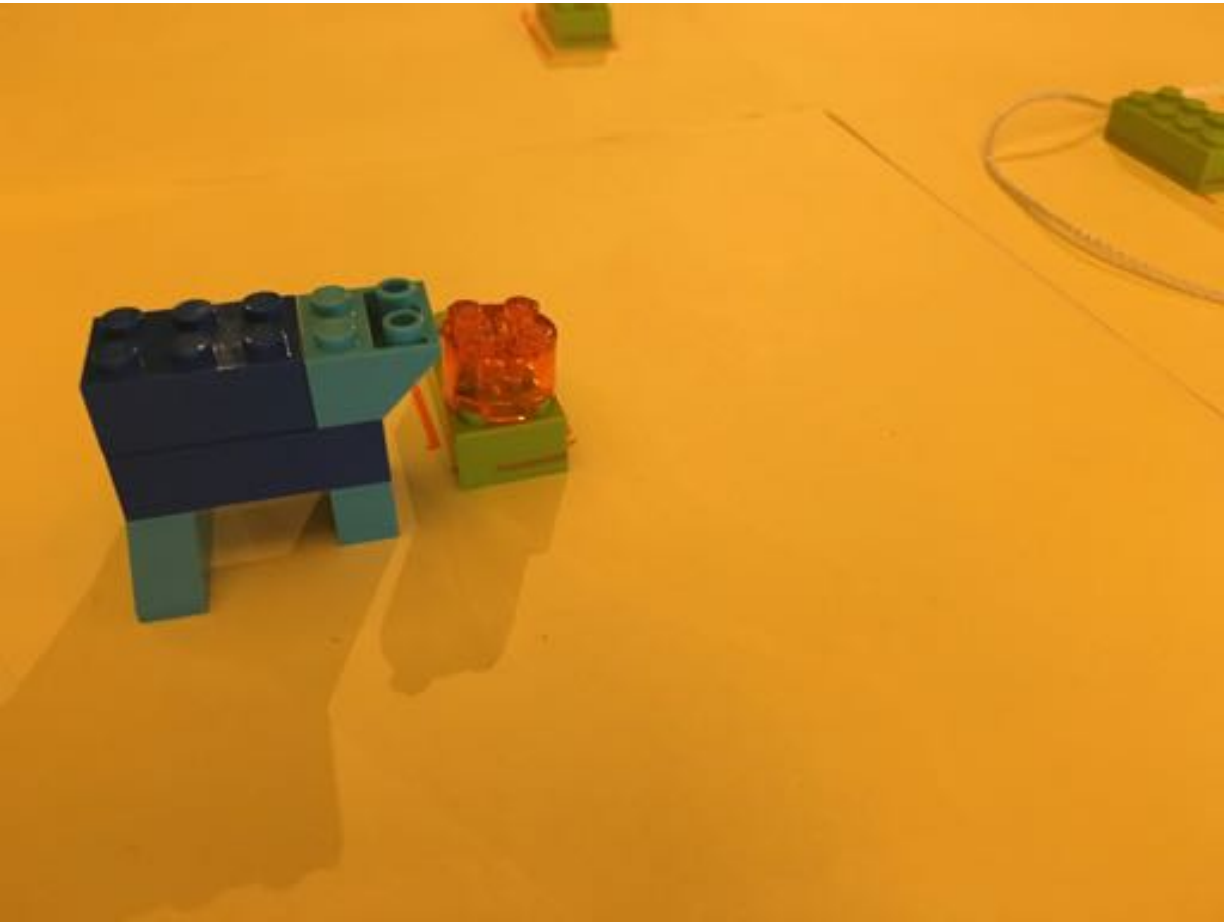


touches









Does the litterbox say meow now?



Then finally all the rules can be placed into a project book which I can use to “save” if other people need to use the space, etc.

For the simulation, we talked about how the playing field can be “captured” and the cat can be placed at its starting point for the simulation. Then the cat’s movement in the simulation would be projected on to the playing field with the litterboxes being physical objects.

We also talked about being able to run through the simulation first by hand, like papert’s turtle. Much like a board game, the cat’s movement rules would be written out, and a child would have to first play the simulation by hand. Then the computer could be used as a tool to speed up or slow down the simulation once the child understands what is running the cat’s behavior. It’s a way of “reverse scaffolding,” providing more powerful computational tools after the child can understand what they do.

We talked about how the system could track objects - one way to do it would be by color, which as bret said is the most visible state of the object and very etoys-like. So you could make your litterbox with a green lego or with green construction paper, and they’d both be litterboxes in the simulation system.