

& What Have You Done

A Meta-Mess and Process Summary

by Geoff Merson

Findings:

Balancing simulations is hard

The system tends to favor extremes

Agents will swarm in such a way that negativity is multiplied to a nearly unstoppable degree. The same has worked for positivity.

'Grinch'-like agents are interesting in particular, as they can affect many other agents before becoming insignificant, which then creates a positive feedback loop

For Another Iteration:

Agents would be vector based particles

The mental attributes of an agent would be its own class instead of their current state as a part of the agent class

First responder agents who would go out to 'rescue' exhausted agents and give some good will

Or a buddy system, wherein two particular agents may recover faster when together, or may become 'buddies' from recovering together

Conditions can worsen or improve without being directly on or off.

Changes in energy drain depending on where the agent is going

Agents being able to have more control and change their destination (already used with square agents, but for the purpose of creating more pathways between hives and places to save stranded agents.

Challenges. What was hard:

Real issues aren't numbers

How do you even show them on screen?




Catching outlying modifiers

Creating a synthesis of the entire simulation as an easy to read “result”



The numbers, what do they mean?

Differences in my attitude and how that affected the project

	 master	 origin/master	 origin/HEAD	final push pre-class	47a299b	Homebase <geoff...	Today, 11:28 AM
Positive Real life events				HomeStretch push 3	c2063c7	Homebase <geoff...	Today, 10:09 AM
				font change	4d2fab	Homebase <geoff...	Today, 10:05 AM
				Home stretch push 2	9feb64f	Homebase <geoff...	Today, 1:05 AM
				Home stretch push 1	1a93006	Geoffrey Merson...	Yesterday, 3:10 PM
				premove	1bcde5c	Homebase <geoff...	Dec 2, 2017, 12:0...
Lots of stress & Confusion				Eo(work)D push	8e37909	Geoffrey Merson...	Nov 29, 2017, 4:5...
				ppush?	289e7b3	Homebase <geoff...	Nov 29, 2017, 12:...
				fix conflict	089da99	Homebase <geoff...	Nov 29, 2017, 12:...
				prevent super energy loss, adjust other values	b8a6a51	Geoffrey Merson...	Nov 21, 2017, 2:2...
				agent mood modifications	4866182	Homebase <geoff...	Nov 28, 2017, 7:2...
Anger				doot doot for presentation	fdd2f26	Homebase <geoff...	Nov 20, 2017, 1:0...
				weekend update	42837fd	Homebase <geoff...	Nov 20, 2017, 9:0...
				Push it you idiot	9a28014	Geoffrey Merson...	Nov 18, 2017, 6:0...
				Update values	cc10917	Geoffrey Merson...	Nov 15, 2017, 2:1...
				Agents go Places now! also places exist. Very very WIP	277f19f	Homebase <geoff...	Nov 12, 2017, 8:5...
Keeping Busy & Reorganizing				Updates and start on work class	a8fa190	Geoffrey Merson...	Nov 10, 2017, 5:0...
				Add Hive, allow them to spawn agents, reconfigure agents	176b20d	Geoffrey Merson...	Nov 9, 2017, 9:37...
				Starting on hive_class	455bf87	Geoffrey Merson...	Nov 7, 2017, 3:33...
				biiliig reshuffling	22705f6	Homebase <geoff...	Nov 6, 2017, 1:09...

Needing & Wanting

The vast majority of my time working on the simulation was spent trying to make the agents within feel absolutely terrible. I was tweaking numbers to increase rates of exhaustion, likelihood of stress or depression occurring in a given agent.

Towards the end of the project's build, I found myself trying to get the fewest upset or exhausted agents per run. Once I realized what I was doing, I put a balance back, but made sure to make recovery possible for affected agents

For what it might be worth there were many changes between the red and the green

```
for (agent o : globalAgents) {  
-   if (o.ID != this.ID && dist(o.x, o.y, this.x, this.y) < (aSize/4)*2) {  
+   if (o.ID != this.ID && dist(o.x, o.y, this.x, this.y) < aSize*1.25) {  
       if (o.agentHome == this.agentHome) {  
           if (o.aStateVal > this.aStateVal) {  
-               adjustState(1);  
+               adjustState(.46);  
               //println(ID + " has interacted with " + o.ID + " negatively");  
           } else {  
-               adjustState(-.9);  
+               adjustState(-.7);  
               if (aEnergy < 100) {  
-                   aEnergy += baseBurn*6;  
+                   aEnergy += baseBurn*4;  
               }  
               //println(ID + " has interacted with " + o.ID + " positively");  
           }  
       } else {  
           if (o.aStateVal > this.aStateVal) {  
-               adjustState(1);  
+               adjustState(.41);  
               //println(ID + " has interacted with " + o.ID + " negatively");  
           } else {  
-               adjustState(.8);  
+               adjustState(-.55);  
               if (aEnergy < 100) {  
-                   aEnergy += baseBurn*5;  
+                   aEnergy += baseBurn*2;  
               }  
           }  
       }  
   }  
}
```

Prediction & Accuracy

My original timeline served as a good starting point for the project, but I can admit it was more of a starting checklist than a solid timeline.

As a result of the vague nature of the timeline, it was not strictly adhered to. There were weeks where a task would not be entirely completed, this task was then worked on the next week until deemed acceptable, at which point the next task would be started.

In terms of learning, I've realized that it may be better to go in either direction more, whether that means to have very loose timelines or have very detailed and strict timelines.

Comparing the timeline to the git history shows Inconsistency.

Week	Date	Task
1	06/11/2017	Interaction Map + Base Agent class
2	13/11/2017	Hive (agent emitter) + Work Classes
3	20/11/2017	Mingling interactions
4	27/11/2017	Personalities and Illnesses
5	04/12/2017	Documentation and final tweaks

master	origin/master	origin/HEAD	final push pre-class
HomeStretch push 3	47a299b	Homebase <geoff...	Today, 11:28 AM
font change	c2063c7	Homebase <geoff...	Today, 10:09 AM
Home stretch push 2	4d2fab	Homebase <geoff...	Today, 10:05 AM
Home stretch push 1	9feb64f	Homebase <geoff...	Today, 1:05 AM
remove	1a93006	Geoffrey Merson...	Yesterday, 3:10 PM
Eo(work)D push	1b0de5c	Homebase <geoff...	Dec 2, 2017, 12:0...
ppush?	8e37909	Geoffrey Merson...	Nov 29, 2017, 4:5...
fix conflict	289e7b3	Homebase <geoff...	Nov 29, 2017, 12:...
prevent super energy loss, adjust other values	089da99	Homebase <geoff...	Nov 29, 2017, 12:...
agent mood modifications	b8a6a51	Geoffrey Merson...	Nov 21, 2017, 2:2...
doot doot for presentation	4866182	Homebase <geoff...	Nov 28, 2017, 7:2...
weekend update	added	Homebase <geoff...	Nov 20, 2017, 1:0...
Push it you idiot	42837fd	Homebase <geoff...	Nov 20, 2017, 9:0...
Update values	9a28014	Geoffrey Merson...	Nov 18, 2017, 6:0...
Agents go Places now! also places exist. Very very WIP	cc10917	Geoffrey Merson...	Nov 15, 2017, 2:1...
Updates and start on work class	277f19f	Homebase <geoff...	Nov 12, 2017, 8:5...
Add Hive, allow them to spawn agents, reconfigure agents	a8fa190	Geoffrey Merson...	Nov 10, 2017, 5:0...
Starting on hive_class	176b20d	Geoffrey Merson...	Nov 9, 2017, 9:37...
billiig reshuffling	455bf87	Geoffrey Merson...	Nov 7, 2017, 3:33...
	22705f6	Homebase <geoff...	Nov 6, 2017, 1:09...

Week	Date	Task
1	06/11/2017	Interaction Map + Base Agent class
2	13/11/2017	Hive (agent emitter) + Work Classes
3	20/11/2017	Mingling interactions
4	27/11/2017	Personalities and Illnesses
5	04/12/2017	Documentation and final tweaks

Just within the first 2 weeks, I'm already diverting from the timeline's intent and letting myself get swayed by whatever happens in real life.

A great example of how skewed the timeline became is the mingling interactions. Which ended up being started a week later and ended up being worked on until the date of submission.

How it works

As agents pass other agents, they influence the mood of one another. The value changes depending on if these two agents were from the same hive. This was done to try to reflect the difference between having a positive interaction with a stranger compared to someone you are familiar with.

Once a given agent reaches either end of the mood scale, which ranges from positive at -10 to negative at 10, it checks on mental state conditions. While this seems counter-intuitive, it was originally used as a way to measure susceptibility to a condition. Conditions are given when they reach a value of 10, and they can be healed if an agent -- after receiving a condition -- reaches the minimum value of -10.

When an agent is given a condition, one is chosen at random out of the three potential conditions, it is then applied to the agent. Agents can acquire multiple conditions, and conditions can be worsened, though this is still a rough idea. If a condition has a color associated to it, and another condition is applied, the most recent condition's color will appear on the agent as a result.

Balance

The simulation's system is delicately balanced. Like ridiculously so.

For example, two agents from the same hive can raise each other's moods by .46 or lower it by .7 .

While raising their moods by .5 instead of .46, the system was almost constantly in a negative mood. It was lowered by hundredths until some sort of balance could be had without reaching a state of constant positivity.

Exhaustion

An agent's energy is one of the most visible elements in the simulation. It allows for agents in need to be highlighted, as well as supplying a level of control over how an agent's life will be overall.

Agents receive very small amounts of energy from interacting with other agents, and as such, allows an agent to become 'rested' during a rest period or potentially rescue an exhausted agent

If an agent becomes exhausted, its mood will gradually worsen until it contracts a condition. At which point it will either stay there, in a stranded state until it is rescued by either a moving hive or another agent passing by.

So in the end

The simulation did find it's own life, and it is now able to attain very different moods all in the same run. Albeit, more runs favour happier times now, which I'm willing to accept as a message for this iteration.

It took a lot of process and realization to separate my bias from the simulation and as a creator I am far from removed from it. I've reached a sort of plateau in my emotional turmoil, so I no longer am driven to point the simulation in specific directions. Though my interpretations of conditions are still a driving force in the simulation, and the fact that those are the conditions present in the simulation at all are markings I've left on the project.

The Autobiography Through The Masses

After any mood based tweak, I found an obvious link between how I was feeling and the overall mood of the simulation. The seemingly less impactful changes though were where I felt a bigger autobiographical influence. In the first few steps of the simulation, exhausted workers were fading away to nothing, effectively being forgotten. At the time trying to get through the issues I was having meant to push distress away and pretend it wasn't there. The current iteration has the inverse, with emotionally stable agents having a softer transparency applied to them to signify that they are carrying on, you can pay attention to them should you wish, but the main point of interest is the distressed agents.

As stated before, this autobiography was not intended at first, in fact the realization came towards the very end when observing the project as a process.

To an extent, I could say that the simulation is a biographical experience instead of an autobiography. I gave the system the information and it gave me a story that reflected my life as it was on that given day.