Drake Seifert

CS475 Project 4

What your own-choice quantity was and how it fit into the simulation.

My choice was a UFO. The UFO checks if the graindeer population is above 5 (more graindeer means that they're easier to find), then has a 50% chance of abducting one. This potentially decreases the number of graindeer, and therefore indirectly affects the rate of grain growth.

A table showing values for temperature, precipitation, number of graindeer, height of the grain, and your own-choice quantity as a function of month number.

Note: I did not realize precipitation was measured in inches until after typing in all the numbers into Excel, hence the combination of units.

	Precipitation (in)	Temperature (°C)	Grain height (cm)	Num grainDeer	UFO (True = 1, False = 0)
Jan	7.28	-2.78	2.54	1	0
Feb	10	3.89	7.01	1	0
Mar	12.89	12.22	42.44	4	0
Apr	12.01	8.33	26.12	9	1
May	10.96	21.67	26.23	11	0
Jun	9.34	21.67	13.53	11	1
Jul	2.62	22.22	0	0	1
Aug	2.54	17.78	0	0	0
Sep	1.05	17.22	0.03	1	0
Oct	0.2	8.89	0.03	1	0
Nov	0.76	1.11	0.06	0	0
Dec	3.71	-6.11	6.79	2	0

Jan	5.92	-2.22	4.63	2	0
Feb	8.54	0.56	6.7	2	0
Mar	13.31	5.56	17.78	4	0
Apr	11.82	17.78	29.76	6	0
May	11.64	18.33	22.18	8	1
Jun	7.37	15	12.06	8	0
Jul	3.3	22.22	2.33	4	1
Aug	0.95	22.22	0	2	0
Sep	1.1	10.56	0	0	0
Oct	1.72	3.89	2.54	1	0
Nov	2.49	-1.11	4.09	2	0
Dec	2.99	-4.44	6.49	2	0
Jan	6.68	2.22	19	1	0
Feb	9.75	7.22	32.94	3	0
Mar	12.31	11.11	41.77	7	1
Apr	11.84	16.11	41.77	8	1
May	8.33	16.11	27.45	10	1
Jun	8.77	21.67	2.46	9	1
Jul	4.68	17.22	0	6	1
Aug	0.41	19.44	0	3	0
Sep	0.38	13.33	0	1	0
Oct	1.95	3.89	0.8	0	0
Nov	2.4	-1.67	0.8	1	0
Dec	4.51	-6.67	3.36	1	0
Jan	8.85	2.78	2.45	1	0

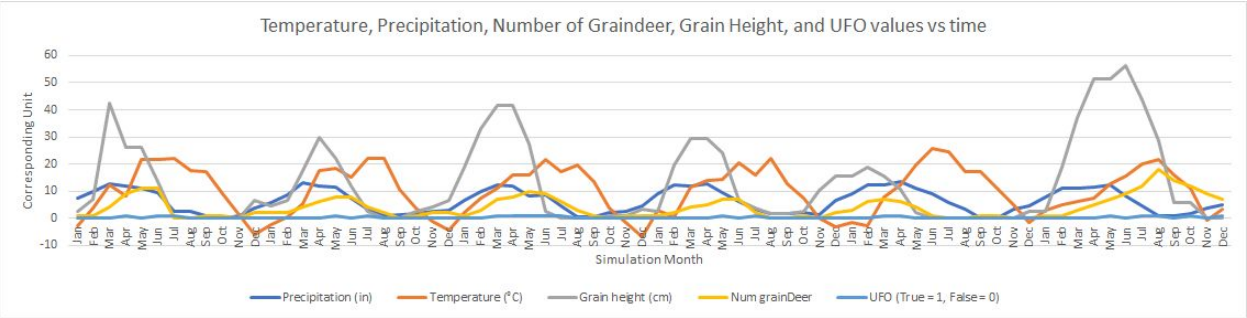
Feb	12.18	0.56	19.71	2	0
Mar	11.78	11.67	29.38	4	0
Apr	12.68	13.89	29.38	5	0
May	9.55	14.44	23.94	7	1
Jun	6.04	20.56	6.46	7	0
Jul	3.35	16.11	3.92	2	1
Aug	1.62	22.22	1.52	0	0
Sep	1.7	12.78	1.52	0	0
Oct	2.19	7.22	2.48	1	0
Nov	1.14	0	10.22	0	0
Dec	6.41	-3.33	15.75	2	0
Jan	8.89	-1.67	15.76	3	0
Feb	12.2	-2.78	18.86	6	0
Mar	12.18	7.78	15.48	7	1
Apr	13.5	12.22	10.82	6	1
May	11.09	19.44	2.22	4	0
Jun	9.11	25.56	0	1	0
Jul	5.79	24.44	0	0	0
Aug	3.37	17.22	0	0	0
Sep	0	17.22	0.05	1	0
Oct	0	11.11	0	1	0
Nov	3.37	5	0	0	0
Dec	4.51	-1.67	2.48	0	0
Jan	7.85	2.78	2.48	1	0
Feb	11.14	5	18.99	1	0

Mar	11.18	6.11	37.36	3	0
Apr	11.3	7.22	51.39	5	0
May	12.2	12.78	51.39	7	1
Jun	8.04	15.56	56.28	9	0
Jul	4.36	20	43.85	12	1
Aug	0.99	21.67	28.62	18	1
Sep	1	16.11	5.76	14	0
Oct	1.75	11.11	5.76	12	1
Nov	3.68	-0.56	0	9	0
Dec	4.96	3.33	0	7	1

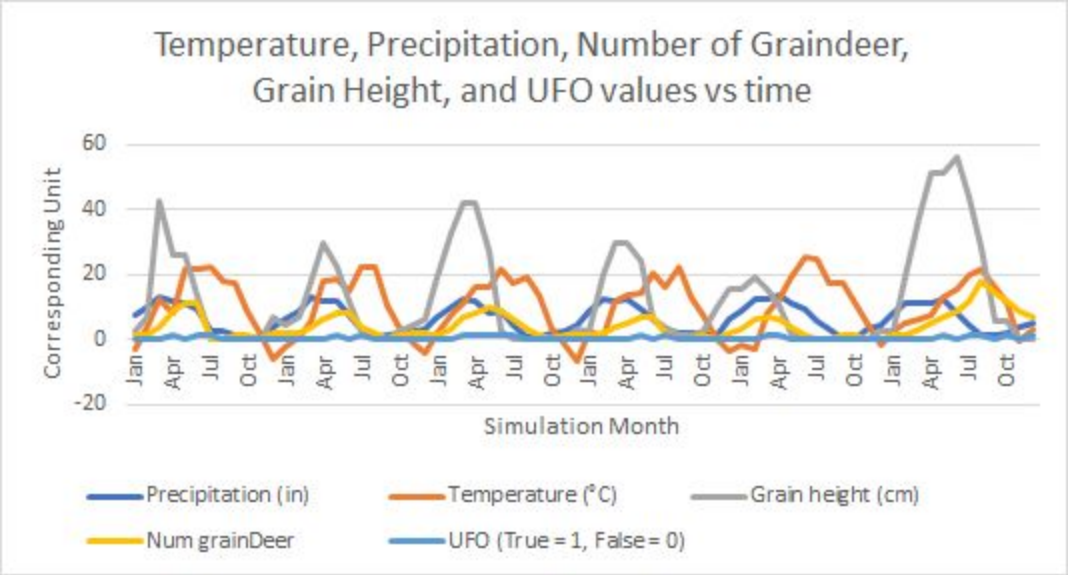
A graph showing temperature, precipitation, number of graindeer, height of the grain, and your own-choice quantity as a function of month number.

Note: Excel wouldn't let me see all values unless I made the chart obnoxiously wide. Below the 6 year chart is a 5 year and 4 month chart which contains a bit less data but is much further zoomed in.

6 years:



5 years, 4 months:



A commentary about the patterns in the graph and why they turned out that way.

It seems that grain height peaked when the temperature was around 16 degrees Celsius and the precipitation was around 10 inches. These conditions make sense because the temperature is optimal for grain growth and the grain is receiving plenty of water. The number of graindeer appears to increase as the grain height increases. This makes sense because the graindeer then have plenty of food. The UFO didn't have a large effect on the number of graindeer, but is still a factor at play.

This general pattern repeated itself every year. In general this makes sense; As the temperature rises and precipitation increases, the grain will grow more rapidly. This in turn will spawn more graindeer. When the graindeer reach a certain population size, the temperature gets hot enough, and the precipitation levels begin to decrease, the grain amount begins to suffer, and with it the graindeer population. Then, the precipitation and temperature begin to increase and the cycle begins once again.

Bonus section: Looping outside or inside of threads

Upon designing this program I was faced with where to put the while loop that checks the year to exit the program. The two options were to place the while loop within each individual thread or outside of the "#pragma omp parallel sections". I wasn't sure which would be quicker; having to respawn threads may be costly, but would it be more costly than for each individual thread to be running its own while loop, thus performing the same operation NUMTHREAD times? To come to a conclusion I exercised what this class has taught me and ran some benchmark tests.

The following had foregone 1000 individual tests each, used the same seed, and only used the Watcher, Grain, and GrainDeer threads. The test can be found in **proj4TestLoop.cpp**.

While loop outside of “#pragma omp parallel sections”:
Quickest time: 0.004673 seconds
Average time: 0.030158 seconds

While loop within each thread:
Quickest time: 0.001996
Average time: 0.031573

This was a bit disappointing as there was no clear conclusion on the winner, however in hindsight it seems to make sense. There is overhead to be had in both cases, but in general the processes are so quick that the costs and benefits of both methods seem to cancel each other out.