Computational Biology

By Anish and Ishaan

Not to be confused with biological computation!

What is Computational Biology?

- The science of using **biological** data to develop algorithms and relations among various **biological** systems
- Has practical applications such as in DNA searching/matching
- A sub division of computer science

Project Introduction

Inspired by mathematician John Conway’s game of life, we thought if we could take the simulation to the next level - and simulate an entire ecosystem!

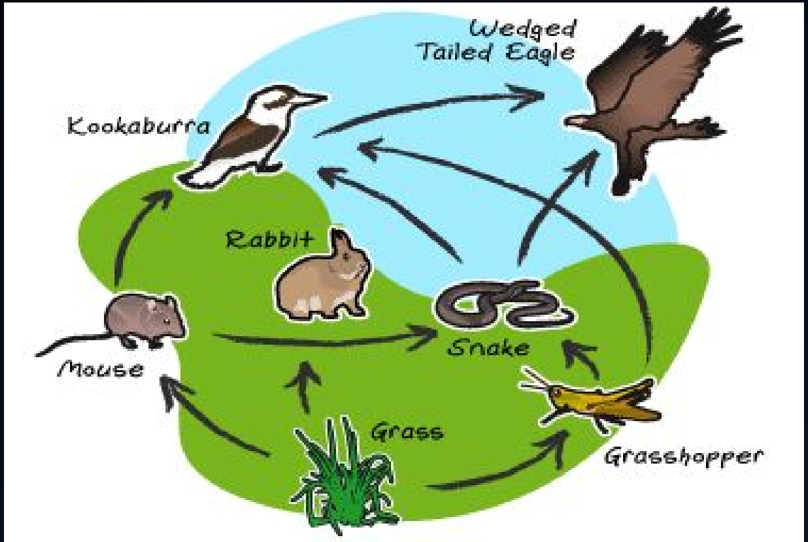
Some background: John Conway’s game of life is a cellular automaton in which based on certain parameters and rules, a cell will manipulate, stay alive, or die. The purpose of the simulation was to create extravagant patterns using mathematics. On the other hand, we are looking to accurately simulate real ecosystems!

Hypothesis:

We predict that one can efficiently and accurately model/simulate real ecosystems using computation and the manipulation of biotic/abiotic factors that affect the ecosystem.

Ecosystem Simulation

Ecosystem Simulated



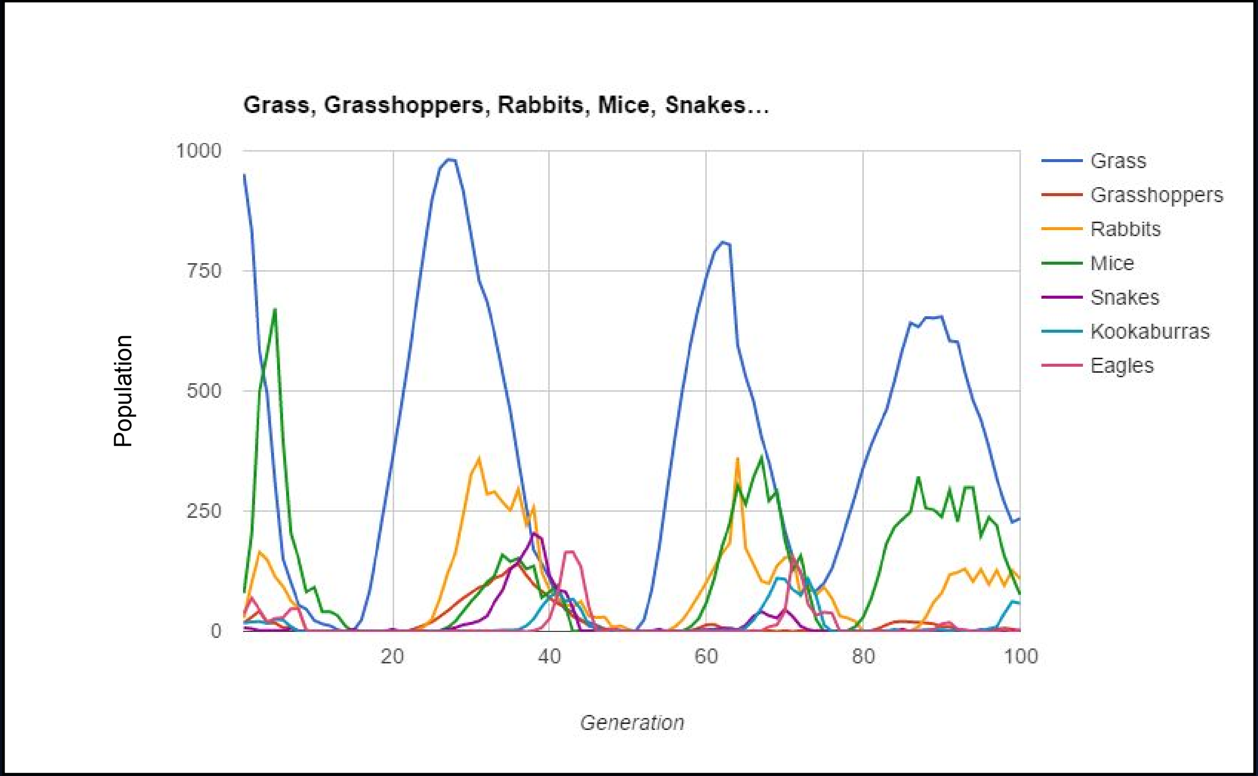
Information

In our ecosystem, we set parameters of radius, food, and mate for each species, based on researching how these animals survive in real life.

Example Simulation Data (100 Generations)

Generation	Grass	Grasshoppers	Rabbits	Mice	Snakes	Kookaburras	Eagles
1	953	18	28	79	8	18	37
2	834	30	103	204	6	20	69
3	585	43	165	497	2	21	45
4	494	18	149	579	2	17	20
5	309	18	114	672	2	25	27
6	151	8	94	398	2	24	29
7	102	10	63	204	6	10	47
8	55	0	48	151	2	0	47
9	46	2	0	82	0	0	3
10	23	0	2	92	1	0	0
11	16	2	0	41	0	0	0
12	12	0	2	41	1	0	0
13	4	2	0	33	0	0	0
14	0	0	2	11	1	0	0
15	2	0	0	0	5	0	0
16	24	2	0	0	0	0	0
17	82	0	2	0	1	0	0
18	170	2	0	0	0	0	0
19	266	0	2	0	1	0	0
20	362	2	0	0	5	0	0
21	461	0	2	0	0	0	0
22	566	2	0	0	1	0	0
23	679	7	2	0	0	0	0
24	792	13	10	2	1	0	0
25	898	20	26	0	0	2	0
26	964	31	68	2	1	0	0
27	982	43	122	9	5	2	0
28	980	56	162	21	8	0	2
29	917	69	242	42	14	2	0
30	824	80	327	62	17	0	2
31	730	92	359	80	22	2	0
32	687	97	286	103	32	0	2
33	620	111	291	116	61	2	0
34	539	117	270	159	85	3	2
35	458	132	252	145	129	3	0
36	357	140	296	152	144	4	2
37	260	117	222	130	171	12	0
38	169	98	258	136	204	26	2
39	141	84	124	70	193	51	8
40	111	69	87	82	116	71	27
41	68	58	69	96	87	82	76
42	45	50	56	51	82	62	164
43	42	31	51	0	49	67	165
44	22	25	62	2	1	46	135
45	19	12	32	0	2	13	52
46	13	8	29	2	0	5	0
47	5	5	30	0	2	1	1
48	6	3	9	2	0	3	5
49	3	1	12	0	2	0	0
50	0	3	5	2	0	0	0
51	2	0	0	0	1	0	0
52	24	2	0	0	0	0	0
53	82	0	2	0	1	0	0
54	176	2	0	0	5	0	0
55	294	0	2	0	0	0	0
56	406	2	10	0	1	0	0
57	510	0	26	2	0	0	0
58	599	2	48	10	1	0	0
59	675	7	74	26	3	2	0
60	738	14	103	60	4	3	2
61	790	15	133	111	5	3	0
62	810	7	163	179	7	3	2
63	805	4	184	225	7	3	0
64	595	3	362	303	4	3	2
65	531	0	173	265	13	8	0
66	479	2	139	321	32	24	0
67	404	0	105	361	41	49	2
68	353	2	100	272	33	74	10
69	283	0	136	292	28	111	14
70	210	2	153	191	46	109	41
71	154	0	156	128	31	87	153
72	121	2	75	157	11	75	123
73	86	0	98	87	4	109	56
74	85	2	75	25	1	73	34
75	101	2	92	0	1	14	40
76	131	0	68	2	0	0	38
77	180	2	32	0	1	0	0
78	230	0	28	2	0	0	0
79	284	2	21	10	1	0	0
80	342	0	0	30	0	2	0
81	388	2	0	68	1	0	0
82	427	7	2	121	0	0	0
83	462	14	0	184	2	1	0
84	522	20	2	217	1	5	0
85	587	21	0	233	5	0	2
86	642	20	2	248	0	0	0
87	634	19	10	322	2	1	0
88	653	18	30	257	2	4	1
89	652	16	61	253	4	3	4
90	655	10	79	238	2	4	16
91	605	10	118	294	2	1	19
92	603	5	123	228	2	4	5
93	539	0	130	299	0	0	4
94	481	2	104	299	1	0	0
95	441	0	129	199	4	2	0
96	384	2	98	237	0	6	0
97	318	5	127	220	2	10	1
98	269	7	96	157	0	37	5
99	227	5	127	116	2	62	0
100	235	1	110	76	3	58	2

Population Chart For All Species



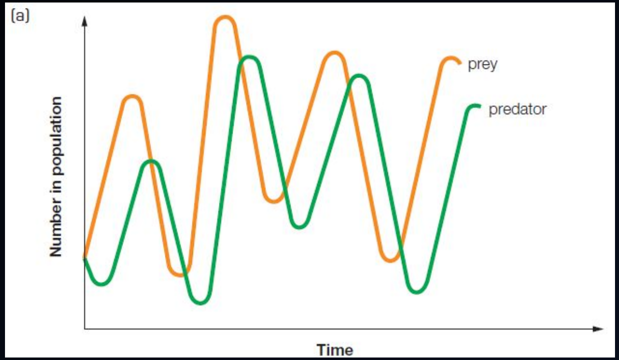
Data Analysis

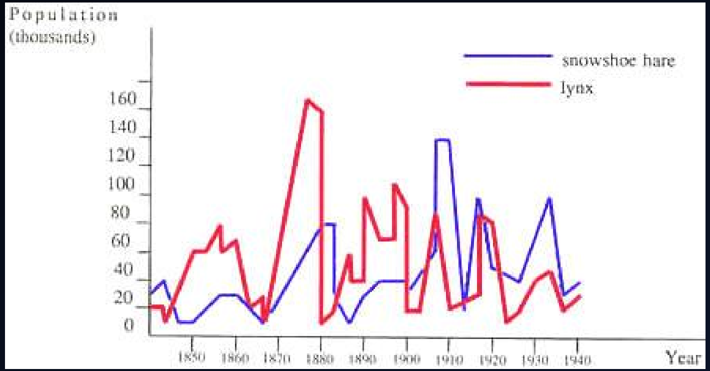
- Lots of cycling in data
- Small population so every species is significantly influential
- Great fluctuations in populations
- Predator/Prey Relationships
- Since grass is the single autotroph in the ecosystem, all other species depend on its population
- Each species started with a different initial population
- The greater the population is, the higher chance and faster it will fall
- The prey population was greater than the predator population, the majority of the time
- We can visually see statistical features of ecosystems, such as the peak -------------------->

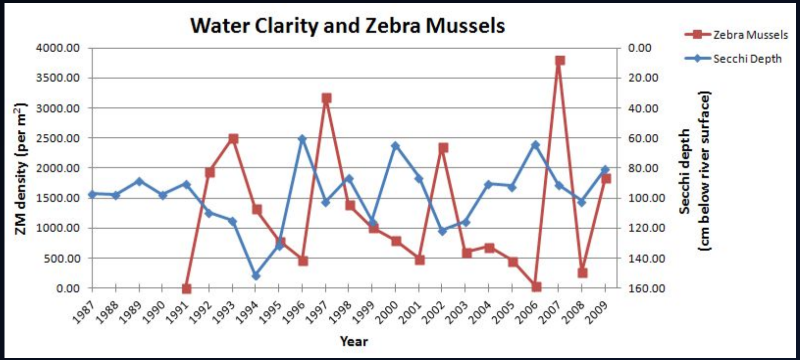
Animal	Average Population	Plateau at Peak (approx)	Carrying Capacity (Max)
Grass	375	750	980
Grasshoppers	19	80	640
Rabbits	84	175	375
Mice	118	260	574
Snakes	17	30	204
Kookaburras	15	50	109
Eagles	15	15	165

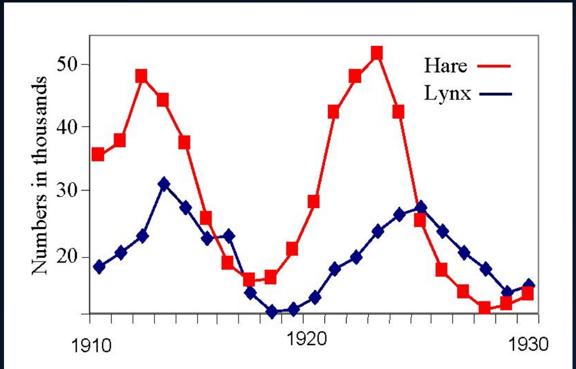
Some other statistical findings

Example of Real Population Graphs (To Compare)







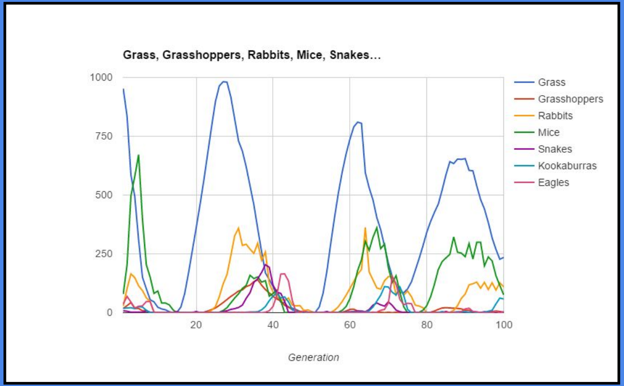


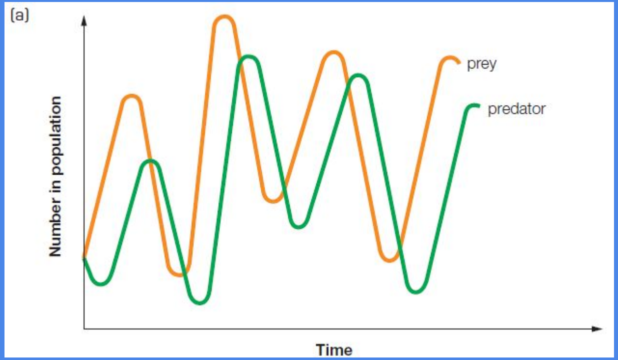
Results

Direct Comparison of Our Graph

Was the experiment successful?

Yes! First of all, the we can see the direct relation between trophic level and population (the higher up on the trophic level, the lower the population). Second, we know that the smaller an ecosystem is, the more species rely on each other, which we can see in the data. Lastly, the raw numbers are quite reasonable as well (as we predicted they would be).





About The Program

Program Features

- Automatic Data Generation
- Scalable Parameters
- Multiple Assessment Factors
- Pseudo Randomness
- Regeneration
- Natural Disasters

Automatic Data Generation

Thanks to the power of computation and data structures, we can efficiently calculate and store data values which we can access any time!

```
Enter number of generations to skip (1..n) or 0 to end:  -1
Would you like to view max (M), averages (A), or generations (G)?:
 M
 max grass: 1434
 max grasshoppers: 229
 max rabbits: 303
 max mice: 566
 max snakes: 143
 max kookaburras: 159
 max eagles:180
If you would like to continue inquiring enter any integer other than 0. Else enter 0:
```

```
Would you like to view max (M), averages (A), or generations (G)?:
 A
 avg grass: 616
 avg grasshoppers: 39
 avg rabbits: 75
 avg mice: 156
 avg snakes: 22
 avg kookaburras: 18
 avg eagles:16
If you would like to continue inquiring enter any integer other than 0. Else enter 0:
```

Scalable Parameters

In order to make the program much more user friendly, the parameters for each animal’s variables can easily be changed. This is because objects and general variables were wisely used. So, in the event that one notices an irregularity in the simulation, they can spend more time optimizing rather than having to deal with changing large portions of code.

Multiple Assessment Factors

Assessment factors in order from top-down:

- The eagles radius; the range in can hunt and mate in
- Prey the eagle can eat, in this case 6 represents kookaburras and 5 represents snakes
- Booleans for food and mate, to check if offspring will be born for the given eagle
- Check if mate and food are true, if so push to next phase
- Otherwise kill animal

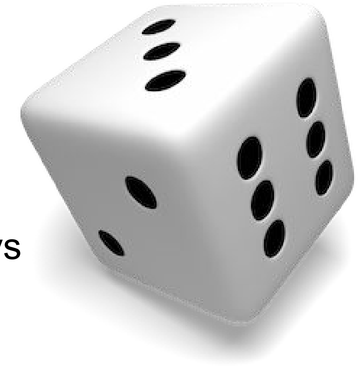
```
} else if (grid[y][x] == 7) {
    eagles++;
    //eagles are capable of eating more than 1 animal per gen, so we will set it to eat a max of 2. But, they still
        must eat at least 1 to stay alive
    int eagleDigested = 0;
    for (int i = y - eagle.radius; i < y + eagle.radius && i < width; i++) {
        for (int z = x - eagle.radius; z < x + eagle.radius && z < width; z++) {
            if (i < 0) {
                i = 0;
            }
            if (z < 0) {
                z = 0;
            }

            if ( (grid[i][z] == 6 || grid[i][z] == 5 /*|| grid[i][z] == 3*/) && eagleDigested < 1) {
                grid[i][z] = 0;
                eagleDigested++;
                if (eagleDigested >= 1) {
                    food = true;
                }

            } else if (grid[i][z] == 7) {
                mate = true;
            }
            if (mate && food) {
                eagleLocations.push_back(temp);
                i = 100000;
                z = 100000;
            } else if ((i == y + eagle.radius - 1 && z == x + eagle.radius - 1) || (i == width - 1 || z == width - 1
                )) {
                grid[y][x] = 0;
            }
        }
    }
    //eagleLocations.push_back(temp);
}
```

Actual Code Taken From The Program!

Pseudo Randomness

The program is capable of generating random initial values so that it displays a different ecosystem each time (although it is possible to change it to set values, if need be), but at the same time the randomness is controlled. With controlled randomness we can align statistics to what we find in nature, based on ranges and domains. When combined we get pseudo-randomness, which gives us the best representation from both factors.

A die is pseudo random. Although the outcome is random, we know it's always between 1-6 (inclusive)

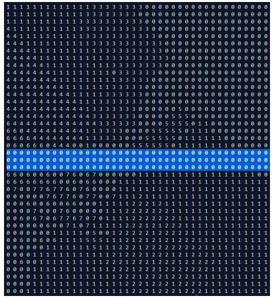


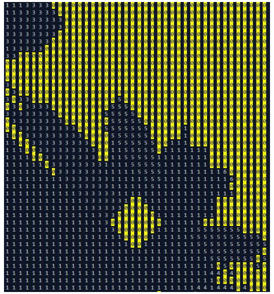
Regeneration

Because species are highly dependent on each other in small ecosystems, with one species going extinct it led to chain reaction for other species to die out as well. In order to counteract this phenomenon, we implemented regeneration based on the rule that if a species is extinct, we can import 2 animals from that species in a random location in the ecosystem. This idea may not be too far off from a real ecosystem as when a species is close to extinction there is a chance for more/less competition, which leads to immigration or emigration of species. Overall, with the implementation of this regeneration function, we can observe ecosystems over significantly larger spans of time.

Natural Disasters

Another environmental factor that affects an ecosystem and its population are natural disasters. Although they are rare, they do happen. This is why we programmed our simulation to endure a natural disaster at least once every 100 generations. The different types of natural disasters present at the moment are fissures, and lightning strikes.





It's Expandable!

Lastly, there is no "end of the road" for the program. We can always add extra features and species later (say for simulating an environment in Russia, where we might add climate factors). Some features we plan to implement in the future are the following (but not limited to):

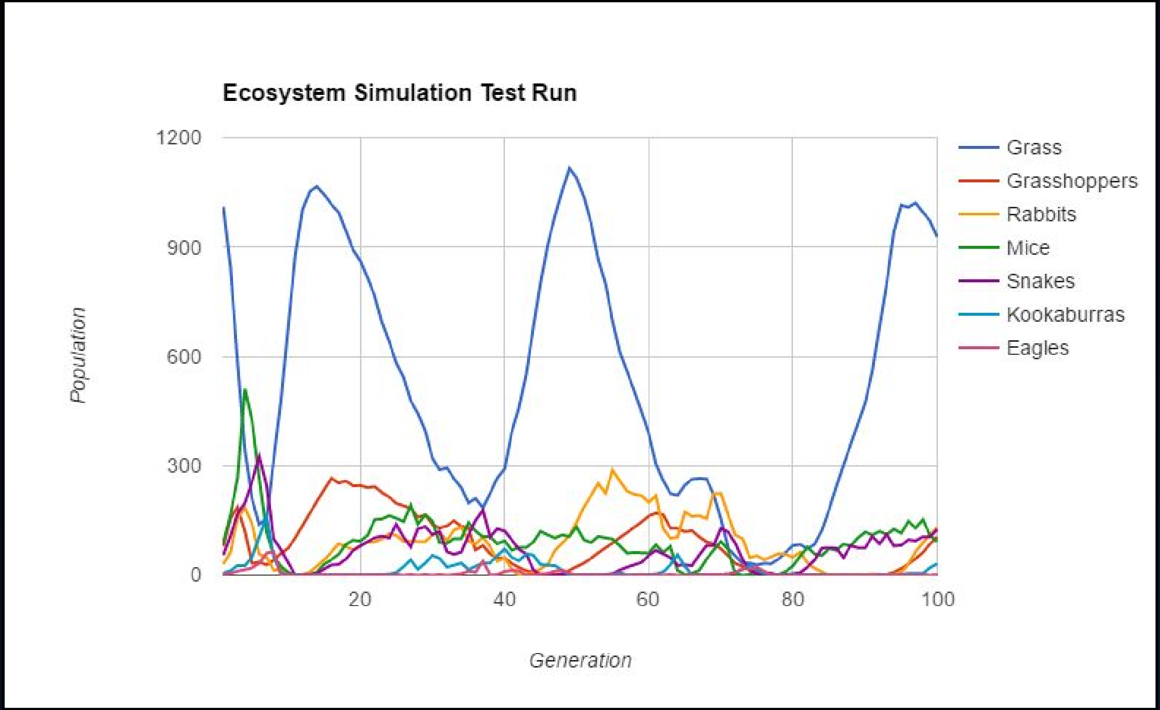
- Climate factors
- Human factors
- Pollution/global warming factors
- Disease factors
- Lifespan factors

About The Process

For this experiment, we went through 4 **main** phases, as outlined below:

1. **Research**: Understanding the relationships, characteristics, and behaviours of different ecosystems
2. **Programming**: Developing the software itself. Probably the longest part
3. **Debugging**: Experimenting with software, and locating possible errors, bugs, or bias in the parameters/”rules”
4. **Data Analysis**: Actually taking samples of generated data, graphing them, and learning from the results

Example Data From Debugging Phase



Challenges Faced

Computational:

- Grid traversal
- Easily accessible I/O
- Debugging
- Calculations / Planning
- Data Structures

Biological:

- Configuring parameters
- Researching and understanding how ecosystems work
- Understanding different biological factors
- Overall, getting as much **accuracy** as possible

And Finally...

How Will Our Creation Help The World?

"We strongly believe this kind of technology can play a significant role in predicting and simulating ecosystems, in order to observe, research, learning, teaching and/or preventing disasters from occurring (ex. Extinction, or how to deal with invasive species)"

- *Anish & Ishaan, Difference Makers*

Benefits to Society

- can predict what will happen to other species, the severity, patterns or trends after the introduction of the species
- Can help against epidemics like the Asian Longhorn beetle
  - Able to create trend of the different species

- Would help in scenarios that we could not predict

Importance and Impacts

- Ecosystems are dynamically changing
  - For better and worse due to many reasons
- It could help with data for assessing ecosystems
- Would be able to do more before than after

Impacts

- Data is helpful as all ecosystems change
- Gives an accurate perception of an ecosystem

Conclusion

- Results were inline with predictions
- Have developed a program that is useful in real life scenarios
- The experiment had positive impact
- Overcame some of the challenges that were given