Costal Ecosystem Simulation.

By Mark Emmanuel k21009628 and Vaibhavkumar patel k21076223.

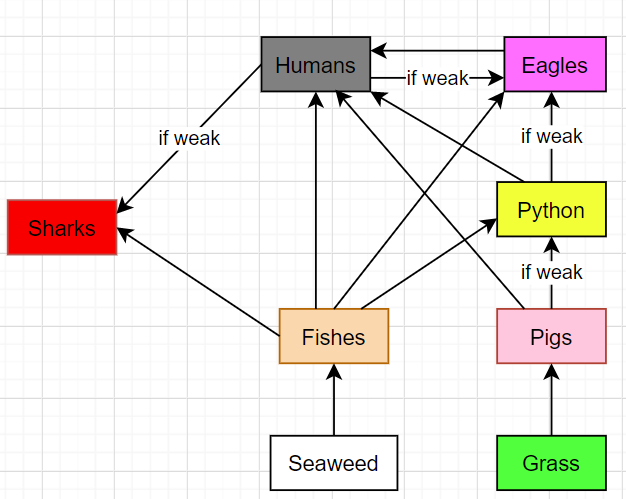
Contents

[Simulation Summary 2](#_Toc97022853)

# Simulation Summary

The Simulation is based around a costal ecosystem where we have two animals in the ocean and four animals on the land. The animals that are simulated are fishes, pigs which are the prey in the simulation. The other four which are predators are sharks, humans, pythons, and eagles.

# Food web

Below is the food web diagram of the interaction made between the species. There also special cases where some animals can eat other animals if they are weak. Weak is a special attribute that is related to how old an animal is and if they have disease or not.

The colours indicate the colours of the species in the simulation for clarity.

# Breeding.

All the animals only propagate if they find a partner who is of opposite gender. This is done by checking if the animal adjacent to this animal is not of the same gender when giving birth.

# Time – Day and night.

The day and night affect both animals and plants. The animals on land all sleep during night-time and are only able to move during daytime. The sea creatures can move at night-time as well. The plants such as grass and seaweed only grow during daytime. The current length of day and night is equal and is of 5 steps.

Challenge tasks

Island generation

The simulation is about a costal ecosystem, so we wanted to implement islands as well. To do so we have used Perlin noise algorithm which outputs us a 2d array of floats. We then use this array which is random enough but still to a human eye looks natural and coherent that it is suitable to use for island generation. This algorithm takes in few parameters’ octaves, persistence, and the seed number. The octaves and persistence are set manually to create the effect of island that we were trying to aim for. The seed number is selected randomly out of 100 every time we are using this Perlin noise object. So currently the algorithm will output 100 unique different maps for the Perlin noise.

Then there is a map class which uses the noise map created by the Perlin noise and maps the value them to an integer these integers are used as an ID for our land classes which are ground(0), shallow water(1) and water(2). The Noise Map through two for loops which then check if the float value is 0.5 then in the map 2d array 1 is inputted. If the value is between 0.5 and 0.66 then 2 is inputted and lastly 0.66 to 1 float value will be mapped to 2. By doing so we ensure that ground and water will be separated by shallow water most of the times like real life. This is because the likelihood of two location next to each other have similar values to one and another.

Land classes – Plants, Weather