Dynamic World: Practicum in Artificial Intelligence

Youtube Link

Constance Nozière (cmn72) and Leo Mehr (ljm262)

December 2015

Contents

1 Introduction

In this project, we have chosen to simulate the interactions between animals and their environment. We created a virtual world in which animals interact with each other and compete for resources. In order to survive animals must successfully eat, drink, sleep, and avoid predators. The world consists of a grid of cells. Each cell is defined by its terrain and resources. The amount of available resources changes dynamically over time as animals eat and the resources regenerate. We model the animals as rational agents such that we can effectively program their behaviors and draw parallels to areas in Artificial Intelligence.

Each animals is a random instance of its species, with various parameters such as hunger threshold, thirst threshold, energy threshold, and reproductive success. As animals reproduce (via a genetic algorithm) new animals with different combinations of the parameters are generated.

Our project culminates in a complex and fast-changing virtual world from which we draw observations that reflect qualities of our world formulation and phenomena in the real animal kingdom. We run a series of different experiments in the model world, and illustrate the power of Artificial Intelligence in improving the performance of the virtual animals.

2 Background and Theory

In this project, we apply the theory laid forth in Russell and Norvig to develop "successful" agents. In order to achieve this goal, we model the animals as rational, goal-based agents. Throughout the report, we use the words agent and animal interchangeably.

2.1 Properties of the Environment

We begin with a description of the agents' task environment.

Discrete

- The environment is divided into a grid of cells. Each cell has a terrain and a set of resources. Thus, the environment has a finite number of distinct states. An agent's location is characterized by cell the agent is currently in rather than taking on a range of continuous values. There is a finite set of actions that the agent can perform: move left, move right, move south, move north, eat, sleep, drink, and mate. There is also a finite number of percepts.

Partially Observable

- At the start of the simulation, the agent knows only of the cell it is currently in. As the agent moves around, it keeps track of all cells that it has visited. For each cell the agent encounters, it learns the terrain of the cell and the resources that were available at the time of its visit. However, the agent is unable to discern any surrounding cells. Moreover, the agent can perceive the animals in its current cell but cannot tell what the value of their characteristics (such as hunger and thirst level) are.

Sequential

- The decision an agent makes in a given time step can have consequences on future time steps.
- For example, if an animal with a low energy level perceives a predator approaching then the two most rational actions for the agent are i) rest ii) move away from the predator. If the animal chooses to rest then either the predator will catch up with the animal and eat it or the animal will be able to gain energy and use it to flee far from the predator. On the other hand, if the animal decides to move away from the predator without resting first then it might successfully escape or the predator might chase it until it runs out of energy and then eat it. Thus decisions made now can affect how things unravel and thus have an impact on future decisions.

Multi-agent

- There are many animals roaming the world and their amount changes with time. Agents can die from internal conditions (such as hunger) or other be killed by other agents. Agents can be born from two parent agents.
- A multi-agent environment is essential for competition between species. While each animal tries to maximize its performance measure, it may sometimes act at the cost of other animals.
- For instance, tigers can feed on both elephants and giraffes. Thus, a tiger trying to eat will minimize the performance measure of the animal it attacks (and kills). Moreover, all animals are ever-competing for basic resources such as water and fruit.

Stochastic

- There is some uncertainty in the world due to the random regeneration of resources and the birth of new animals.
- When animals mate, the algorithm that combines their parameters involves some randomization (reminiscent of a genetic algorithm).

Static

- The state of the world does not change within a single time step. Hence agents can decide on their actions without having to repeatedly examine their environment.

Known

- Each agent is familiar with the rules of the environment. It is aware of the actions it can perform and the effect those actions will have. An agent does not need to learn how the environment works in order to make good decisions. However, the agent does need to explore the environment so it can choose from a larger number of cells and thus make a better decision.

2.2 Properties of Agents

As previously alluded to, we model the animals as rational agents. "For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has."[1]. Such behaviour may come at the cost of the other animals.

Animals are competing for many of the same resources and thus one animal trying to decrease its hunger level by eating a fruit may in fact be depriving some other animal of that fruit. If there are not enough resources to go around then one animal tyring to maximize its performance measure by eating may cause the performance measures of the other animals to go down. Moreover, when a tiger decides to decrease its hunger level by eating either a giraffe or an elephant, it kills it prey and thus the eaten animal's performance measure goes to negative infinity.

The *performance measure* for a given agent is how many time steps the agent survives before being eaten or dying of thirst/hunger. That is, in surviving longer, we say the agent has performed *better*. However, when the agent dies, it is no longer an agent; it cannot make decisions or interact with the environment. Thus an agent cannot obtain direct feedback for its performance, yet it can approximate performance using certain heuristics on its state (some combination of hunger, energy, thirst, etc). The interesting part of the project is to create a parametrized heuristic for all agents such that agents with different parameters will have different performances.

The intelligent agent as introduced in Russell and Norvig has a simple algorithm to determine its actions (Algorithm 1). The agent is just a function that receives a percept and returns an action. The environment processes the action, updates its state, and then gives the agent a new percept. While the agent is determining its action, it retains some memory of its own state and the state of the environment.
We expand upon this Skeleton Agent by modeling animals as **Goal-Based Agents**. The goal-based agent uses *searching* and *planning* to determine the actions needed to achieve its goal.

When deciding on an action, an animal takes into account both goal information and knowledge gained from its percept sequence. At each time step, the animal determines a goal state and then performs an action that will help it reach that goal state. With each time step, the agent records information about the cell it is currently in, thus expanding its internal representation of the world. It then uses this updated representation to help choose an action. Algorithm 2 describes the function used by the animals to select an action:

Algorithm 1 Skeleton Agent, from Figure 2.4 in Russell and Norvig

function SKELETON-AGENT(*percept*) **returns** action
  static *memory* of state and environment

  memory ← UPDATE-MEMORY(memory, percept)
  action ← CHOOSE-ACTION(memory)
  memory ← UPDATE-MEMORY(memory, action)
  return *action*
end function

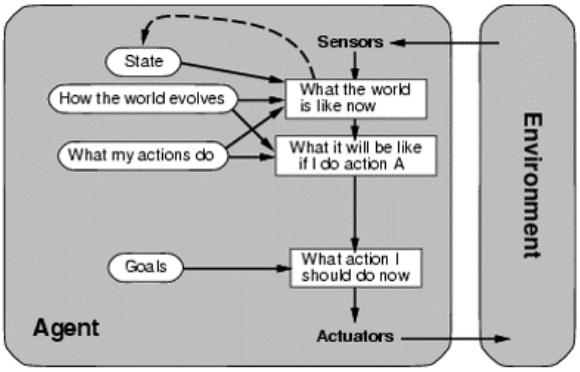


Figure 1: A model for a goal-based agent, from Figure 2.13 in Russell and Norvig

Thus, in order to create animals as goal-based agents, we need to specify the their *percepts*, *actions*, *goals*, and *memory*.

2.2.1 Percepts

At each time step, the agent perceives its current cell. The precepts from the cell are:

- Terrain - The agent learns the current cell's terrain (desert, plains, forest, or river)
- Resources - The agent learns how many of each type of resource (grass, fruit, leaves, water, and fish) the current cells contains.
- Animals - The agent learns how many of each type of animal (giraffe, elephant, and tiger) the current cell contains. However, the agent does not learn anything about the state of the other animals in the cell such as their hunger or energy level.

2.2.2 Actions

There are 8 different actions that an animal can perform: move north, move south, move east, move west, eat, drink, sleep, and mate.

Algorithm 2 Our implementation of Choose-Action

function CHOOSE-ACTION(memory) **returns** action **static** *parameters* the characteristics of the agent
  goal ← DETERMINE-GOAL(memory, parameters)
  action ← SEARCH-FOR-GOAL(goal, memory)
  return *action*
end function

Following from the possible actions, there are three ways an animal can die: thirst, hunger, or being eaten by a tiger (except of course for tigers!). We exclude exhaustion, but prevent an animal from being able to move if it runs out of energy.

Move {North, East, South, West}

- The agent moves {up, right down, left} to the adjacent cell. If the agent attempts to move off the edge of the world, the agent simply remains in its current cell.

Eat

- The animal eats a resource or an animal from its current cell. What the animal can eat is determined by its species. If the animal eats a resource, the resource disappears from the cell (and might regenerate at some later time). If the animal (the predator) eats another animal (the prey), the prey dies and the predator's hunger level decreases. If the animal cannot eat anything in its current cell then the action has no effect.

Drink

- If there is water in the animal's current cell then the animal drinks and its thirst level goes down. The amount of water available in that cell also goes down (and might regenerate at some later time). If no water is present in the animals' cell then the action has no effect.

Sleep

- The animal sleeps for one time step meaning that it stays in its cell and its energy level increases.

2.2.3 Goals

As discussed at the start of this section, animals do not have a direct way of determining their performance. However, they can formulate goals that will improve their chances of survival. Thus, the primary goals are to drink, eat, or run away from a predator. Secondarily, the animal cannot move if it has no energy, so one other basic goal is to sleep.

Higher order goals are those which do not directly cause the animal to survive longer. Sleep is a simple, yet nonetheless, higher order goal. Other higher order goals are exploration, to expand the animal's representation of the world, and mating.

In summary, the possible goals are:

- Eat
- Drink
- Rest
- Explore
- Mate

2.2.4 Memory

Each animal maintains an internal representation of the world (based on its percept sequence), as well as information about its hunger level, thirst level, energy level, and the number of steps since it last ate, drank, and slept. The representation of the world consists of all cells the animal has visited as well as the state the cells were in at the time of the animal's visit. The animal then uses this stored information to decide on an action.

2.2.5 Intelligence

What makes our agents intelligent?
We define intelligence as the ability to act rationally in a variety of situations and process information so as to be able to act rationally when faced with new situations.

The animals are intelligent in that they attempt to survive as long as possible and take actions that help them achieve this goal. At every time step, an animal formulates a goal and then uses search/planning to find an action that brings it closer to its goal. Our animals use their internal representation of the world, knowledge of their current cell (acquired via percepts), and their current state to decide on a goal. As the animals explore and visit more cells, their internal representation of the world becomes increasingly rich. In turn, they are able to make better decisions as they have more cells to chose from when constructing a path to the desired goal.

3 Implementation and Performance

We implement the world as a grid of cells. Each cell can have one of four possible terrains: plains, forest, desert, or river. Each terrain has different resources which animals can consume. Plains have *grass*, forests have *leaves* and *fruit*, rivers have *water* and *fish*, and deserts have nothing.

The world is discretized spatially with cells, and temporally with integer time steps. In one time step, each cell calls the `act` function for each of its present animals. The cell processes the actions and changes the state as necessary.

As explained above, animals can either move around, consume resources in their cell, rest, or mate. Only eating or drinking actually change the state of the cell the animal resides in. In order to avoid complete resource exhaustion, we allow for resources to regenerate over time. Every time step, there is some small probability $p = 0.005$ that one of the types of resources in a cell will return. Further, each cell has a maximum amount for each resource it can contain. For each resource, we implement the maximum to be in the range 5-15.[1]

An animal can only move one cell at a time. When it moves or mates, its hunger and thirst increase, and its energy decreases. When it eats, its hunger decreases, but thirst increases (and analogously for drinking). Starting with no hungry/thirst, an animal will die from hunger/thirst after 50 time steps.

We have three animal species: Tigers, Elephants, and Giraffes. Tigers can eat Fish, Elephants, and Giraffes. Elephants and Giraffes can eat grass, fruit, and leaves.

3.1 The UI

We implement our project in Python, and use a 2D graphics library, TKinter, to create a simple visualization of our world (Figure 2).

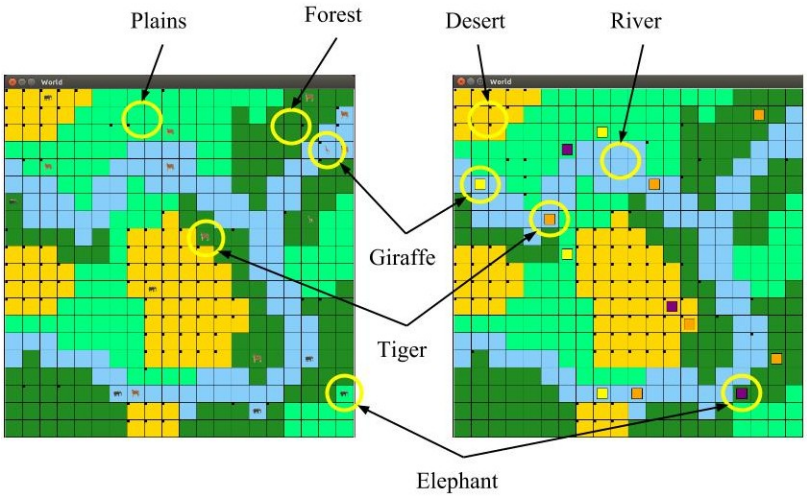
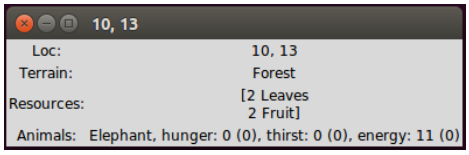


Figure 2: Our standard 20x20 world

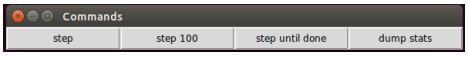
[1]Here and throughout, many of the chosen values are arbitrary. There are infinite possibilities for the implementation of the environment and animal behaviors. In our project, we often defined parameters that seemed reasonable, and adjusted them to fit our desired vision of the project.

The visualization features a grid of cells. The colour of a cell represents the cell's terrain. By clicking on the cell the user is able to obtain information such as the cell's coordinates, terrain, as well as lists of the resources and animals present in the cell. The cell listing provides the details of the status of each animal that exists on the cell: the hunger (and time of least meal), thirst (and time of last drink), and energy (and time of last sleep). See Figure 3a. If a cell's resources are exhausted, a small black square is drawn in its top left corner.

Our simulator also features an auxiliary command bar (Figure 3b). [Step] runs one step, [step 100] runs 100 steps, [step until done] runs the simulator until all of the animals in the world are dead, and [dump stats] writes statistics about the animals in the simulation to a local file.



(a) Information about cell at row 10, column 13

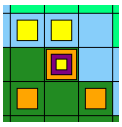


(b) Command bar widget

Figure 3: Additional widgets for observing and running the world

There are two options for visualizing animals: pictures or colored boxes. In both cases, the animal visualizations are overlaid on the cells, and adjusted to display multiple animals. The pictures provide a more appropriate, accurate representation, yet the boxes provide faster performance during simulation. See Figure 4.

(a) An elephant and tiger share a cell



(b) An elephant, tiger, and giraffe all in one cell

Figure 4: Animal overlapping in both rendering style

3.2 The Code

Link to Github.
Below we give a high-level overview of each of the files/components in our project.

`main.py`

- Provides the UI via TKinter widgets and key bindings. Provides a `--no_images` flag to run the colored boxes animal visualization and a `--no_display` flag to speed up computation when running experiments. This file only depends on `world_control`.

`world_control.py`

- Maintains an instance of a `world` object, handles stepping and drawing the world on a canvas.

`world.py`

- Stores a 2D array of cells and a set of all animals in the world. Also provides some utility functions for writing statistics about the world to file.

`cell.py`

- Implements the operation of a single cell in the environment. It stores local animals and handles their actions, provides a draw function used by `world_control`, and includes various utility functions used to create cells, get distances between cells, get neighbours, etc.

`terrain.py`

- Implements the different Terrains as objects. Each terrain has a current and maximum amount of each resource. Also provides functionality for regenerating resources.

`animal.py`

- The most complex of the files, provides the implementation of our animals as rational agents. There is a complex parent class `Animal`, which classes `Elephant`, `Tiger`, and `Giraffe` all inherit from.

`action.py`

- A simple file that provides the abstract representation actions: Eat, Sleep, Drink, Mate, and Move.

`resources.py`

- A very simple file that abstracts the resources: Grass, Fruit, Leaves, Water, and Fish.

`generate_world.py`

- This is a utility file that we initially used to generate random instances of our world, and save them to file. In our experiments, we use a single initial configuration consistently.

`world_config/`

- This directory has text files that represent different world configurations. When the simulation starts up, it reads the cell terrains from `world_config/world.txt` and reads the animal placement from `world_config/animals.txt`.

`run_experiment.sh`

- A bash script that runs the simulator without a display, and saves the output of animal statistics into a file. One experiment runs 10 simulations in order to get more data for the plots.

`run_world.sh`

- Simple bash script that runs the main file with just boxes.

`data/`

- Stores the data and plots from experiments.

`constance.py` (Constants)

- Stores two variables: pixels per cell, and the dimension of the world (in cells). This metadata would have lived in `world_control`, but would then have caused a circular dependency.

4 Experiments

We ran a series of experiments in the model world, to see how the animals behave and interact when they were programmed differently. There are three degrees of intelligence, determined by the function that an animal uses to determine its action: random_determine_action, naive_determine_action, and determine_action_by_goal.

4.1 Initial Experiment

Figure 5 shows the initial setup for our experiments. There are 20 animals of each species (60 animals total), placed sparsely throughout the world.

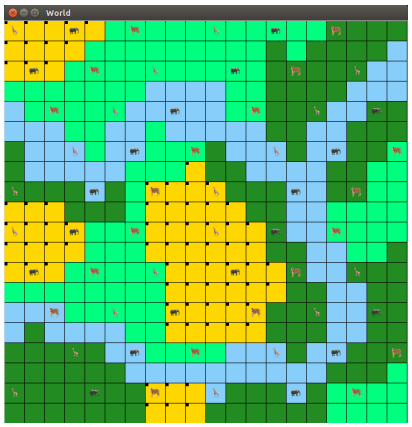


Figure 5: Caption

First, we program all of the animals to behave randomly (Figure 6). We run the simulation of the world 10 times until all animals are dead, and aggregate the results.

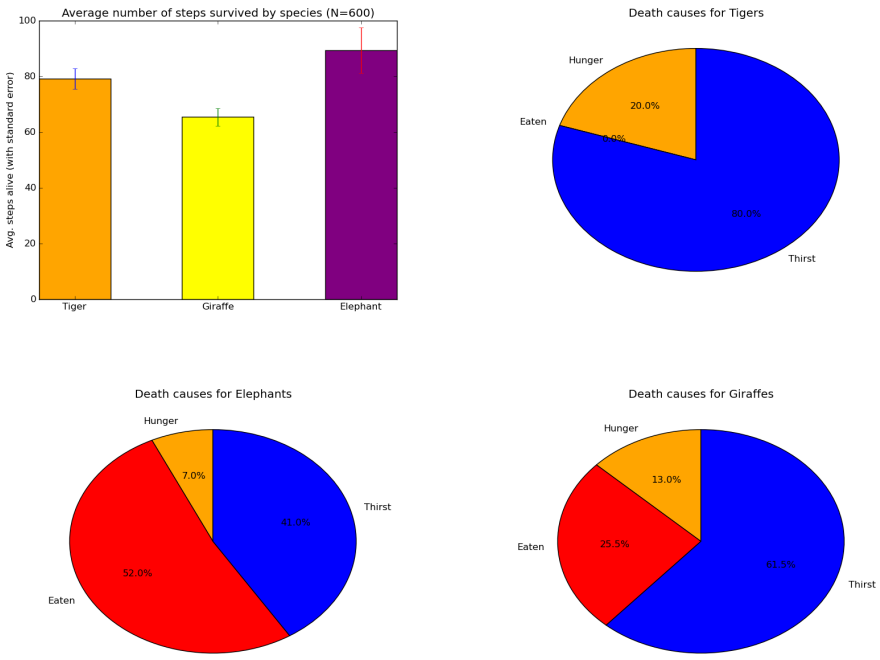


Figure 6: Random actions

If tigers had the same diet as elephants and giraffes, all of the plots would be roughly symmetric. However, tigers can eat elephants and giraffes, which explains why the average number of steps survived by elephants and giraffes is significantly lower than that of tigers. Further, as expected, the giraffe and elephant death causes/survival are almost completely symmetric.

Note that tigers die primarily from thirst because water is more scarce a resource. While water only appears in river cells, tiger food appears in both rivers (as fish) and anywhere on the map where there are giraffes/elephants.

4.2 Intelligent Actions

We now give elephants superior intelligence by having them use the `naive_determine_action` function. This function uses the animal's current state to determine which action would maximize its chances of survival, but without any consideration of the environment. If the animal is hungry it tries to eat, if it is thirsty it drinks, etc. However, if it cannot perform the action because the cell does not have the correct resources, it simply sleeps.

Thus, we hypothesized that elephants will outlive their competitors, and die less frequently from natural causes. Figure 7 shows the results.

As we expect, performance of the elephants improved over that of the other species (even tigers). The cause of death for elephants was less frequently natural because whenever possible, they acted in such a manner as to avoid being hungry and thirsty.

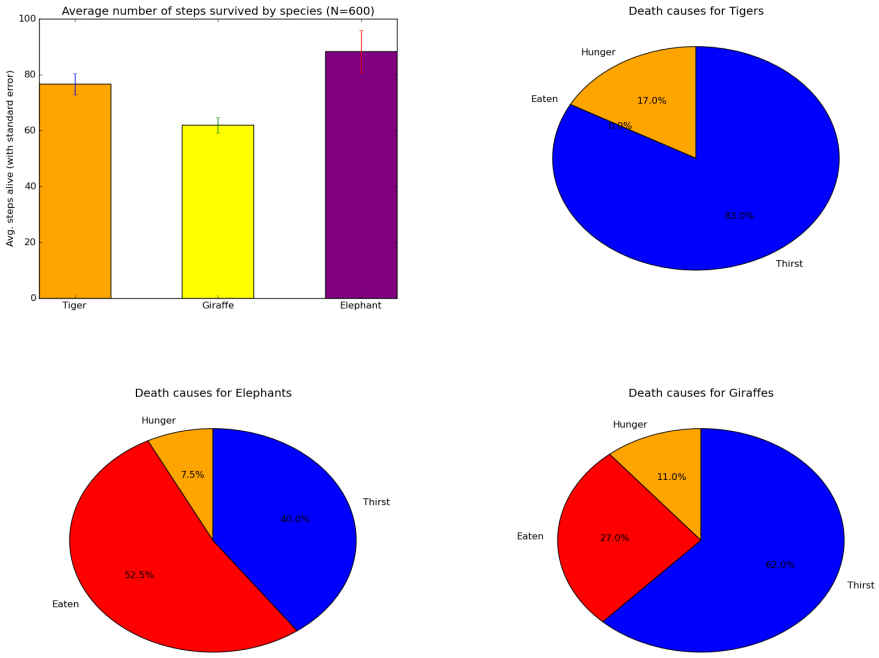


Figure 7: Elephants are naive agents, and other species are random

We next give tigers and giraffes naive action capability, and then further improve the performance of elephants, by making their actions goal-based. As outlined in the theory section, the goal-based agent first picks a goal based on its survival needs (similar to the naive determine action). However, it also uses the knowledge of its environment to search for the best move to achieve its goal. This means that when an elephant is hungry, it will search through its internal representation of the world and find the nearest cell that has food. Furthermore, the goal-based agent, if it is neither hungry, thirsty, nor tired, will explore to enrich its knowledge of the world.

We expected this to be a significant improvement over the naive action agents, because with the knowledge of where food and water is, there is ample time for an agent to make its way and find it. See Figure 8 for results.

We indeed found this change to be a massive improvement in the elephants' performance. The results are similar to the previous experiment, yet much more exaggerated. Elephants die less frequently from natural causes, and significantly outperform their "less intelligent" peers.

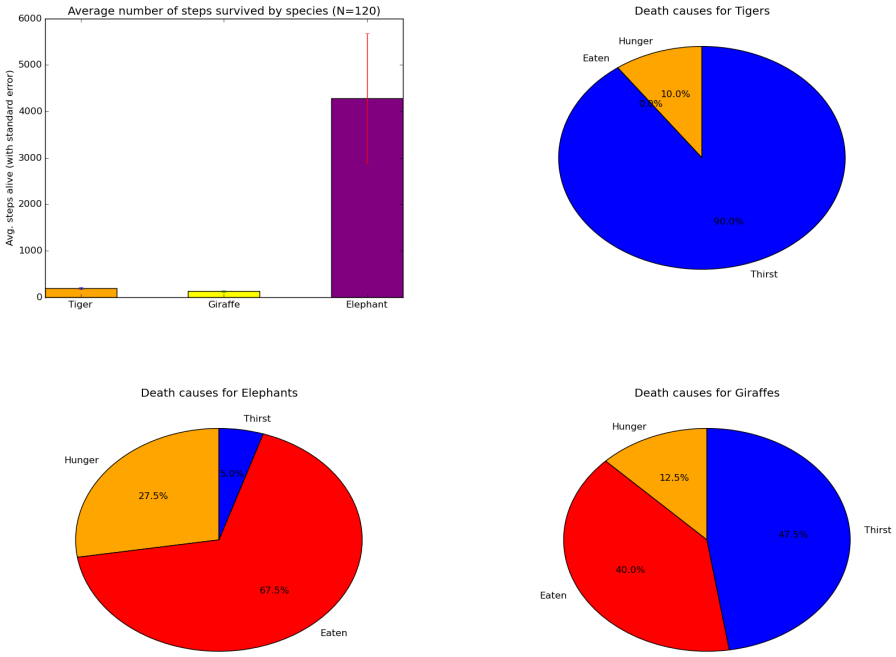


Figure 8: Elephants are goal-based, and other species are naive

The last experiment we ran is to use goal-based decision making for all the animal species. Now, all animals are adept at finding food and water for themselves. However, this makes tigers a big threat to the other species. As we see in the results, Figure 9, tigers outlive the other species, but still die out relatively fast (compared to goal-based elephants from the previous experiment, who had an average of around 4,500 steps survived). The tigers die primarily from hunger, because their food resources become significantly more scarce once the other animals are dead.

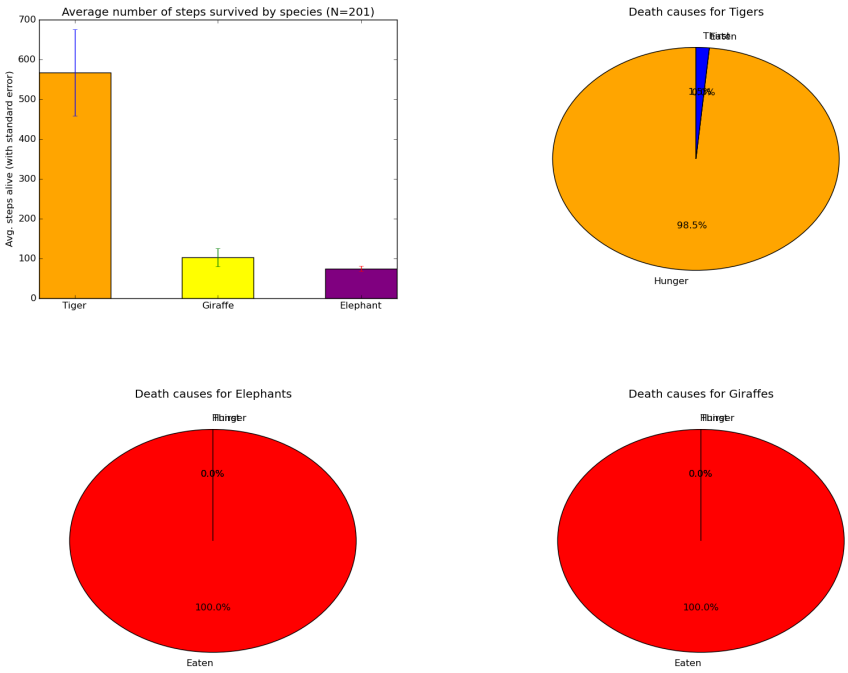


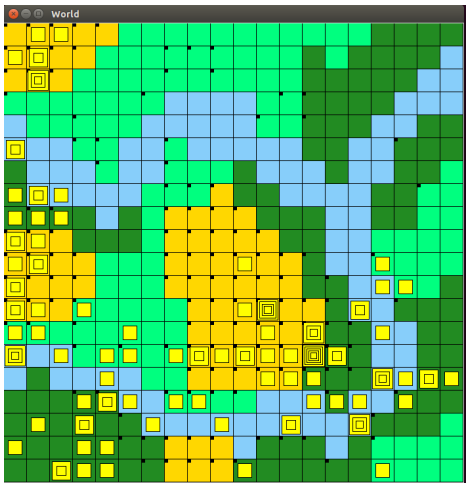
Figure 9: Animals from all species are goal-based agents

4.3 Mating

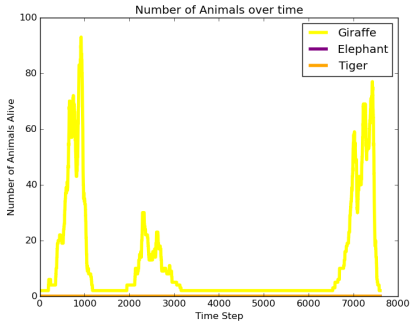
We isolated the previous experiment from mating, so as to avoid the world running forever, and more complex behaviour that mating introduces.

We ran a simple experiment to see the effect of having a large mating parameter when an animal decides which goal to pursue. When the mating parameter is too large, the world population blooms in waves, then cuts down when resources run out.

A mating parameter of 50% for a goal-based agent means that after the agent has satisfied its need for food, it will try to mate 50% of the time, and explore the rest of the time. In a world starting with just two giraffes, the results of a 50% mating parameter are astonishing. The giraffe population experiences exponential growth, followed by exponential decay, as waves of animals are born, and then die from resource exhaustion. See Figure 10.



(a) A bloom in the population



(b) A time series of the boom/bust cycle

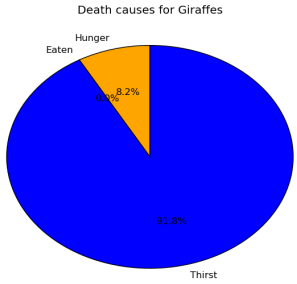


Figure 10: Animals from all species are goal-based agents

5 Onward

There are many possible additions to the project to make it more realistic and complex. One example, is to implement more animal characteristic randomization.

There could also be a learning aspect to this project where animals receive feedback on their actions and use this feedback to make better decisions in the future.

It would be interesting to explore and run more tests on the genetic algorithm. One could tinker with the parameters and see how the world responds. When implementing the algorithm, we hoped that it would generate very successful animals and weed out those that were not as "fit".

6 Conclusions

In this project we simulated the interactions between animals and their environment. We drew on search strategies, knowledge representation, and rule-based reasoning.

The aim of this project was not to gain insight into animal decision making but rather use artificial intelligence and object oriented programming to model an ecosystem.

We successfully modeled a world in which animals compete for resources and there are predator-prey interactions. Our world mimics the circle of life with animals dying from hunger, thirst, etc. and reproducing to create offspring. We were able to program animals such that they survived much longer than the control group that took random actions.

This project suggests that artificial intelligence can be a very powerful tool in modeling natural systems and can help researchers make strides in the field behavioral ecology. [2]

References

[1] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.

[2] H Saarenmaa, ND Stone, LJ Folse, JM Packard, WE Grant, ME Makela, and RN Coulson. An artificial intelligence modelling approach to simulating animal/habitat interactions. *Ecological Modelling*, 44(1):125–141, 1988.