

Evolution of Predator Avoidance and Alarm Calls

Guanya Yang(guayang@ucla.edu)
ID: 204378262

Fan Li(fanli.cs.ucla@gmail.com)
ID: 304414453

Abstract—We implemented a predator-prey system in which prey animats escaped from predator animats while foraging for food in a 2D environment. The design of the animat brains include neural networks and reinforcement learning. Our results demonstrated prey learning in three stages: 1) learning how to avoid two types of predators 2) learning how to avoid a new type of predator 3) learning how to emit alarm calls while foraging for food in groups.

I. INTRODUCTION

In natural ecosystems, food webs are composed to multiple organisms on different trophic levels. Animals in the middle levels consume organisms in lower levels while escaping from predators in higher levels. We focus on prairie voles, a primary consumer whose main predators include hawks, foxes, and snakes. Prairie voles live in colonies, which we simulate in the situation of group foraging [1], [2].

A. Hypothesis

Given prey vole animats that must forage for nuts and avoid three types of predators to survive, they learn how to avoid the different types of predators while maintaining sufficient energy levels. After the voles adapted to the predators, we introduced the condition that the voles cannot sense prey while they are eating. Then they must also learn to respond to alarm calls from fellow voles in order to continue to evade predators. Over time, we hypothesize that the fitness of the prey animats would increase as they became skilled at evading predators.

B. Goals

- 1) Phase I: The prey vole learns how to avoid two types of predators, the hawk and the fox.
- 2) Phase II: A new snake predator is introduced, which the vole will have to learn how to avoid in addition to the hawk and the fox.
- 3) Phase III: Now, when the voles are eating nuts, they cannot sense predators at the same time. Thus, they have to find food in a group and any voles that are not eating learn to send warning signals to their fellow voles.

II. IMPLEMENTATION

A. Environment and Physics

We designed simple animats, specifically a pack of 30 prey animats and 3 predator animats, in a 2D world. Prey animats can sense both their own kind and the predators. All 3 predators can also sense the prey. We will not distinguish genders nor age among the animats.

The 2D world also includes various holes, bushes, trees, and nuts. To make it easier to observe interactions in our environment, we decided to make the total number of nuts present simultaneously to be 60; whenever the number of nuts drops below 60, we automatically replace it with another one placed in a random location on the screen.

B. Methodology

We concentrate of a lower level of animat complexity and a higher level of learning and adaptability. The specific abstractions used are:

- 1) Voles must eat nuts to survive, as measured by a hunger meter that decreases with time.
- 2) Voles die on contact from predators or when their hunger meter goes to zero. (For Phase II, to give the prey opportunities to learn how to escape from the snake, we set the probability of a snake eating the prey upon contact to be 50%)
- 3) Each type of animat will generate a specific signal. All other animats are able to sense the signal and infer the signal type, strength, and direction within a given radius.
- 4) The prey moves slower than the predator because the preys brain is more advanced than that of the predator.
All predators move at the same speed.
- 5) Animats do not mate or reproduce, but we artificially change the population size in our experiments.
- 6) The bodies of animats physically take up space but are allowed overlap with other each other, as well as with shelter.

III. ANIMAT DESIGN

A. Predator

There are three types of predators as follows:

- 1) Hawk: voles can evade the hawk by burrowing under a hole or hiding under a bush
- 2) Fox: voles can evade the fox by burrowing in a hole or climbing up a tree
- 3) Snake: voles can evade the snake by climbing up a tree or hiding under a bush, but theres a 50% chance of the snake catching the prey under a bush.

We implemented a Feedforward Neural Networks (FFN) as the controller. We chose an FFN as the brain controller for predator to make the predator smart enough to catch prey but not so smart as to wipe out the prey population. The FFN for predators is designed as follows:

- 1) Input Layer: Linear layer of size 6
- 2) Hidden Layer: Sigmoid layer of size 6

3) Output Layer: Linear layer of size 5

According to Hornik et. al., the sizes of the input layer and output layer should correspond to the size of the sensors and effectors of the brain [3]. Since there are 5 different actions to take (one for each direction of movement and one for predating), we set the output layers to 5. For the input layer, we could use as little as 3 inputs to represent all the parameters from the predator's sensors: two for the x- and y-coordinates of sensed prey and one for the hunger meter. Normalizing each parameter, they can cover all possible inputs. During our supervised learning process, we found that the FFN converges faster if we specified the inputs more concretely by increasing input layer of a size 6. Since the size of the hidden layer should be between the size of the input and output and a sigmoid function offers greater number of outputs, we implement the hidden layer as a sigmoid layer of size 6 to include as many hidden states as possible [4].

After the construction of the neural network, we designed the parameters. We generated a set of sample data to train the predator and a supervised learning process to set all the parameters in the network. With this FFN, once the predator is able to sense any prey animats in a given range and its own hunger level, it can decide on the shortest path to pursue the prey.

B. Prey

The brain is a Recurrent Neural Network (RNN) implemented with Q-learning, a type of RNN. We chose to use Q-learning to clearly implement reinforcement learning.

1) *State Representations:* The sensors for the prey are used to generate a representation of the state it is currently in. However, in addition to the perceptions of the prey, we include a variable that indicates whether or not the predator sensor was activated in the past 5 consecutive time steps. We represented our states as a 4-tuple (L, F, P, R) where L represents the location which indicates whether or not the prey is around the corner of the map with two different values, F represents the sense of food with two different values, P represents the exist of predators and their types with 8 different values for all 3 types of predators and R represents whether predators have recently appeared. Thus, we have an entire state space with $2 \times 2 \times 8 \times 2$ combinations of the percepts, which is 64 unique states in total for Phase I and Phase II. For Phase III, there is another sensor for the alarm calls and represented by two different values. However, the alarm sensor is active only if the predator sensor is disabled since voles that are eating cannot sense the predators and would need to avoid them by sensing alarm calls. Thus there are $2 \times 2 \times 2 \times 2$ which is 16 different states added.

2) *Actions:* There are five different effectors of the prey animat: *Up, Down, Left, Right, Stay, Eat* and *Call* additionally for Phase III. The prey should be capable of foraging for food or avoiding predators with only these five simple actions.

3) *Rewards:* The core of the entire Q-learning algorithm is based on the premise of a reward scheme [5]. In order for Q-learning to function correctly, it is imperative that the prey be able to determine the approximate value of an action immediately after performing the action. A simple reward scheme for our problem could be a positive reward after eating food or getting close to food; a positive reward when staying in or getting close to a shelter when chasing by a predator; a negative reward when getting close to a predator and so on. However, since the situation is quite complicated with finding food and escaping from at least two kinds of predators with different shelters, this simple solution will not work well so we introduce additional terms for reward. We define a value function for the prey as follows:

$$V = \alpha * E + \beta * S + \gamma * F \quad (1)$$

where α, β and γ are the weight coefficients of E , S , and F .

E is the estimate of possible space that the prey could travel to, with two special cases. The first case is when the prey is around the corner (Fig. 1). The prey should prefer not to go to the corner since it is neither good place to find food nor a safe place to escape from predators. This is important as the environment is bounded so the actions the preys take when it is in the middle of the environment between when it is around the corner can be quite different.

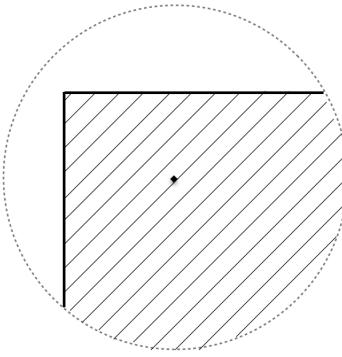


Fig. 1. The case when the prey is around the corner. The solid lines represent the boundaries of the environment and the circle represent the sense range of the prey. The shaded area is the estimation of possible space that the prey could go to around the corner.

The estimation of E also indicates the distance between the prey and the predators. As shown in Fig. 2, the possible space is divided by the median line between the prey and the predator in this case. The estimation space to travel decreases as the distance from predators decreases. We use the estimation space because we found out that in order to escape while several predators exist in different positions, the best strategy is to move to the places where the prey can take the greatest number of possible actions. In general, using this estimation instead of the distance between the prey and the predator allows the prey to calculate more precisely the reward for situations when it

is around the corner of the environments and when predators appear in different positions.

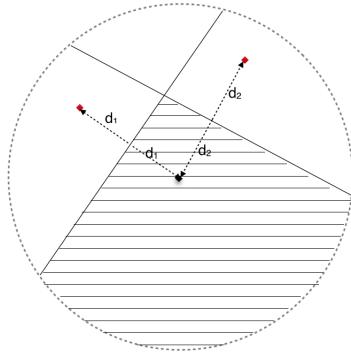


Fig. 2. The case when the predators exist. Predators are represented by red points and the solid lines is the median line between the predators and prey. The shaded area is the estimation of possible space that the prey could go to around the corner.

S indicates whether the prey should escape to shelter to avoid any predators sensed and is defined as:

$$S = \sum_{s \in S} p_s^i * |p^i| \quad (2)$$

where $|p^i|$ is the amount of predator with type- i sensed and the value p_s^i represent whether s is a correct shelter for predator p^i . In Phase I, the prey is engineered with knowing the correct shelter for hawks and foxes and in Phase II the prey does not know the coefficient $p_s^3, \forall s$. We introduced a reinforcement process to change the coefficient as long as it is attacked by the snake.

The food function F in Eqn. 3 describes the distance between the prey and the food, which is defined as:

$$F = \max_{f \in S} \frac{1}{\text{Dis}(\text{prey}, f)} \quad (3)$$

We only estimate the food whose coordinates is within the range of the possible space that the prey can travel to that is also away from any sensed predators.

In summary, our reward is a combination of the varying value function representing the combination of the preys distance from predators, shelter, and food. With the appropriate weights for each part of the equation, the prey is able to consider and decide on the appropriate action to take given a combination of factors from the its surroundings.

4) *Evolution*: For Phase III, we originally intended for the prey to somehow learn to forage for food in groups if they are unable sense predators while eating. However, under our given design of the states and actions of the prey animat, it is not possible to let them learn how to forage in groups since there is no such behavior to reward them for foraging in groups in the first place. We needed to engineer the group foraging

behavior and then let the voles adapt to prefer the group foraging behavior to the original solitary foraging behavior. To demonstrate such an adaptation, we modeled an evolutionary process instead of relying solely on reinforcement learning. During the evolution process, we generated a mutation of our prey at the beginning of the experiment with the recessive genotype aa ; the initial population contained the homozygous dominant genotypes AA . We chose the mutation to be recessive because we did not want the voles to adapt to group foraging too quickly, making the problem too easy. The prey with the mutated recessive genotype would have knowledge of forging food in a group of two and is stimulated by an additional positive reward. Upon reaching food, we randomly one of the voles to eat and the other to warn its companion if appears within its sense range.

IV. PROBLEMS ENCOUNTERED

In the beginning, we contemplated whether to implement a simple design of actions as moving in either four directions, staying in one place, or eating, or to implement a more abstract design of actions representing an overall goal, such as foraging for food or escaping from a predator. Although the animat learned more quickly when we implemented the abstract design, we realized that this contradicted our hypothesis because this was merely a form of supervised learning. In order to stay true to our plan of reinforcement learning, we ended up choosing the actions to be simple, letting the animats learn how to pursue complicated goals by taking a set of simple actions.

Designing the predators to find food was relatively straightforward, but adding the additional goal of escaping from a generic type of predator was much more difficult. For example, the prey do not have memory of where it traveled or what it encountered before, so it would try to move towards the predator once the predator was out of its sense range. We solved this problem by adding internal states to specify what the prey has sensed. We also tried a lot effort in coming up with a good estimation of the reward function to make the prey more capable in learning different kinds of skills.

Another issue was that the animat would move back and forth in one spot and seem to forget how to find food. Eventually, we discovered the cause was due to how we set up the possible actions for each state to take. For example, we define each state to randomly choose an action with probability p_a . Each state will converge to an action if there is one action with a large enough $p_{a'}$ to dominate all other possible actions. However, we had set several states such that there was a great chance of choose between two actions (i.e. moving up or left if it sensed a predator in the upper left quadrant), making it impossible for each state to converge to one action. Since a state can lead to any other state in the Q-table, the non-convergence of one state will also lead to the non-convergence of the other states, which prevents learning from finishing. We solved this issue by a deeper examination of all the states

that the prey may encounter and the transformation probability between completely different states and handle them with carefully designed reward function.

V. EXPERIMENTAL RESULTS

A. Visualized Simulator

We implemented a visualized simulator for a better understanding and testing whether our animats are acting in the way we desire. Fig. 3 shows a simulator with the size 800×600 . The simulator is used for testing whether our algorithm is working or not, however, it is not a good way to show the results when dozens of animats added. Thus, we carried out other experiments for different phases.



Fig. 3. A visualization simulator for our environment: animals in red circles are predators, while animals in blue circles are prey; circles represent their sense ranges. Nuts and three types of shelters are randomly distributed throughout the environment.

B. Phase I

The experiment parameters are set as follows:

Parameter	Food	Predator	Shelter
Value	40	3 (Hawk and Fox)	15 (Each)

We determined the carrying capacity of the environment with the above given parameters by initially creating a population of 300 voles. The result is shown in Fig. 4. Over time, as the number of nuts is held constant, the voles compete for food while their hunger levels decrease. Since all voles start with the same initial hunger level, a majority of them die from hunger at the same time of around 475 ticks. Immediately afterwards, the population stabilizes to around 30 voles. Thus, we decided to set our initial population for the three phases to 30 voles. Any further changes in the population can then be assumed to be solely due to predation.

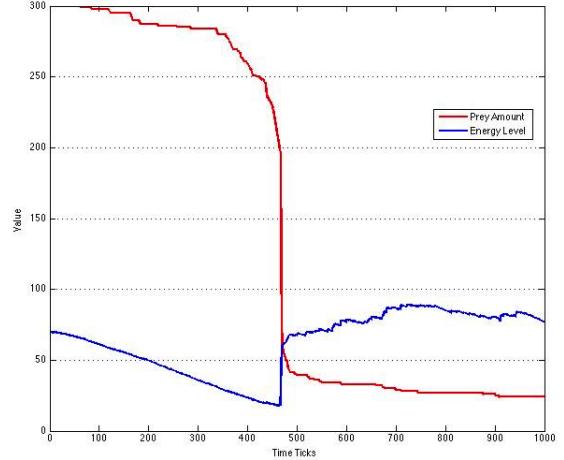


Fig. 4. The carrying capacity of the environment for 40 nuts is around 30 voles. All voles were initially placed in the environment with the same hunger level. At around 475 ticks, there is a steep decline in the population as many voles die from hunger, right before the population stabilizes.

Without adding any new prey to the environment, the population steadily decreases over time due to predation (Fig. 5). However, the population never reaches 0 even after 1000 ticks, indicating that the voles were able to successfully learn how to evade the first two predators, the fox and the hawk.

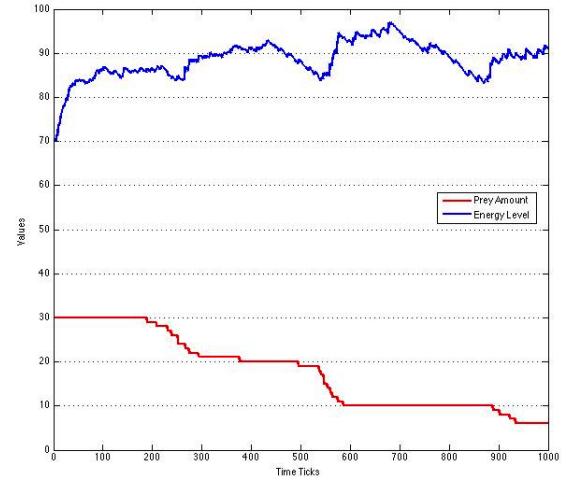


Fig. 5. A population of 30 voles without any artificial reproduction in a 2-predatory environment results in a steady decline in population over time.

To determine the parameters for artificial reproduction of voles in an environment with predators, we tested different rates of reproduction at each time tick (Fig. 6). We define the rate of reproduction as the chance of a vole mating with another vole and producing one offspring at each time tick, imitating sexual reproduction whereby two voles mate to produce one surviving offspring. Although real voles tend to have much litters of 5 to 10 each year, we limited the litter to only one offspring since there isn't enough predation in our simulation to control such a high reproduction rate [1]. Eventually, we found that a mating rate of 0.005 imitates

natural prey population fluctuations and sustains our vole population in the presence of the initial two predators.

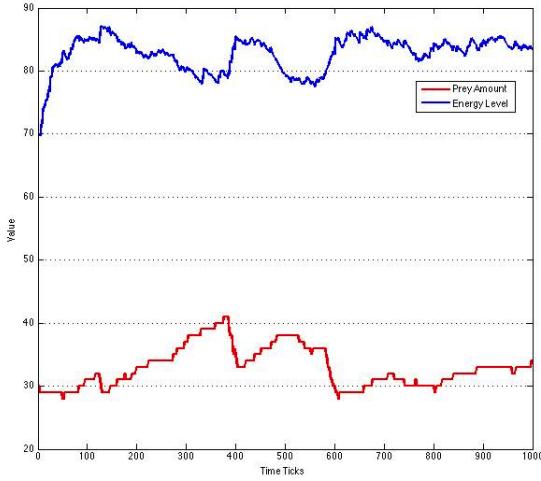


Fig. 6. After introducing artificial reproduction at a rate of 0.005 in a 2-predatory environment, the vole population stays relatively constant around 30. (The environment capacity)

C. Phase II

We now introduce a new predator, the snake, into the environment and the experimental results are shown in Fig. 7.

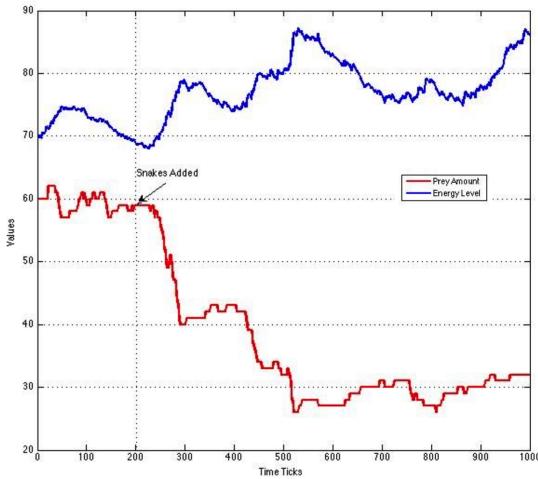


Fig. 7. After introducing the new snake predator at 200 ticks, the vole population initially steadily decreased, eventually reaching a stable size of around 30 at 500 ticks.

Although the voles can sense that the snake is a dangerous predator, they do not know which type of shelter to choose to escape to. By trial-and-error and reinforcement learning, the voles eventually learn to choose the bush and the hole, with a preference for the hole. For the experiment, we set the Food Amount to be 60 and increase the initial number of preys to 60 because many of them will die before they learn to avoid the snake. The slope of population decrease is large initially as the voles have not yet learned which shelter to escape to. As the voles learn successfully over time, the

slope becomes less steep. There is even a slight increase in population after 800 ticks, which shows how the voles are beginning to flourish just a couple hundred of ticks after the appearance of a new predator.

Next, we examined the relative positions of the prey animats compared to the locations of the shelters dispersed in the environment. An assortment of the three types of shelters were randomly distributed throughout the environment (Fig. 8).

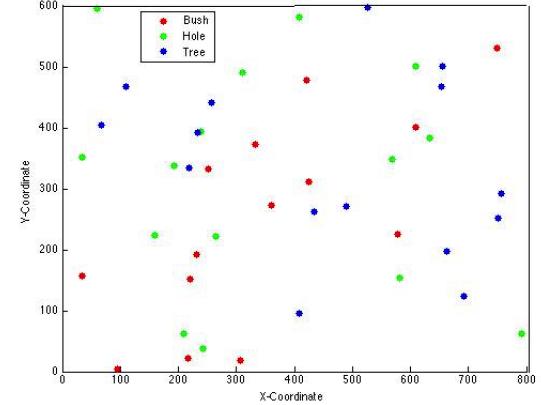


Fig. 8. The map of shelters in the Phase II experiment.

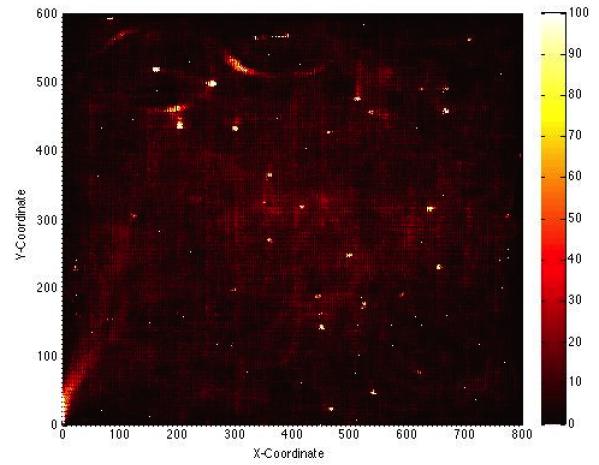


Fig. 9. The heat map of locations the prey animats visited. Yellow indicates locations visited most frequently while black indicates locations visited least frequently.

As expected, the locations the animats visited (Fig. 9) roughly corresponded to the locations of the shelters as they tended to move to those areas to escape from predators. Other red areas most likely correspond to nut positions. One aspect of the result we weren't expecting was the high frequency that the animats were visiting the lower left corner. Even though we attempted to prevent the voles from moving towards corners in our experimental setup, our approach was not foolproof. It is possible that a vole may have moved to a corner to eat a nut, but also begins to sense a predator, which may be able to corner the prey and capture it.

D. Phase III

We initially created a population of voles that composed of 0.9 expressing the native phenotype and 0.1 expressing the recessive gene for group foraging. Over time, as is shown in Fig. 10, the voles that did not exhibit the group foraging gene died off and failed to reproduce, while those that were recessive for the gene flourished, eventually making up almost 100 percent of the final population at 5000 ticks. The voles that participated in group foraging exhibited greater fitness by adapting to the handicap of being unable to sense predators while eating.

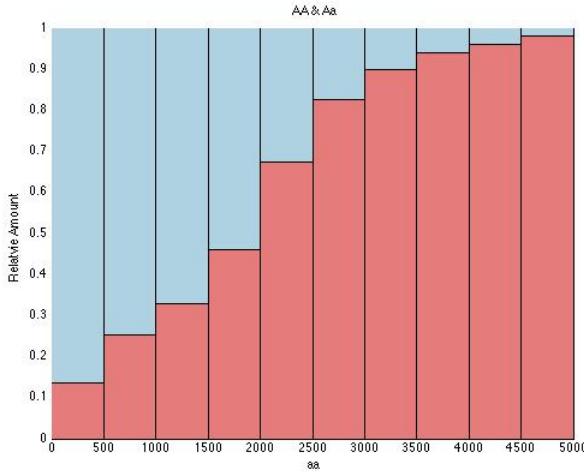


Fig. 10. The relative population of voles with the dominant (Red) and recessive (Blue) phenotypes for the group foraging gene. The amount is the average taken over every 500 ticks

Fig. 11 shows the energy level and prey amount after 4000 ticks in experiment for Phase III.

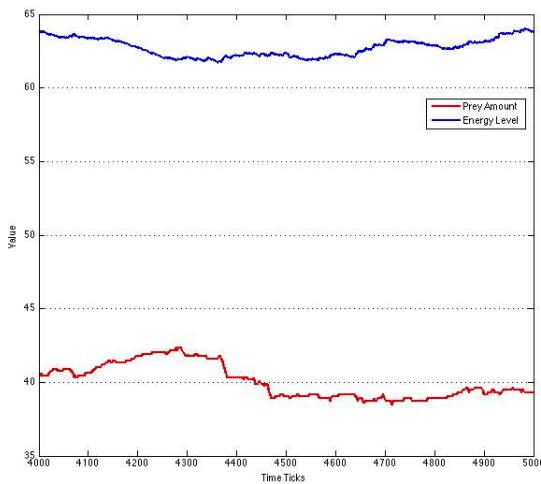


Fig. 11. The energy level and prey population stays relatively constant throughout the whole time period of Phase III.

Compared to an environment with only two predators in Fig. 6, environment with three predators results in lower

average energy but higher survival rate. The lower average energy is likely due to the increased complexity of foraging for food in groups. Conversely, the higher survival rate suggests that the prey have learned to better escape from their predators by working together. By evoking an alarm call to warn their neighbors, the voles effectively increase each others sense range by sharing the information within all of their sense ranges.

VI. CURRENT STATUS AND FUTURE DIRECTIONS

Currently our Q-learning prey animats are able to successfully escape from three types of FFN predator animats by escaping to appropriate types of shelter, while foraging for nuts to survive. Through reinforcement learning, the voles utilize the information they sense from the environment to decide on the best action to take at each time tick; over time, we can regard the summation of set of simple individual actions as broad, abstract goals.

If we had more time, we would have liked to explore Phase III more, using strictly reinforcement learning to let the prey animats adapt to group foraging, without relying on evolution. This would entail adding more states and actions to the design of the prey animat to support group foraging behaviors.

It would also be interesting to further investigate how the prey animats would adapt to smarter predator animats that are also implemented with Q-learning, to group hunting with more than two prey animats at a time, or to other more complex environmental situations. These experiments would more closely mimic predator-prey interactions in nature. We hope that our results contribute to a greater understanding of animal group hunting and to the field of animat research.

VII. LANGUAGE/TOOLS/PACKAGES:

- 1) Language: *Python*
- 2) FFN network using *PyBrain*
www.pybrain.org
- 3) Visualized simulation based on *Pygame*
www.pygame.org

VIII. CONTRIBUTIONS

Both authors contributed equally to the project, from the coding and implementation to the presentation and written report.

REFERENCES

- [1] Solomon, Nancy G., and Joseph J. Jacquot. "Characteristics of resident and wandering prairie voles, *Microtus ochrogaster*." *Canadian Journal of Zoology*, 80.5 (2002): 951-955.
- [2] Lowell L. Getz, Betty McGuire, Theresa Pizzuto, Joyce E. Hofmann, Barbara Frase. Social Organization of the Prairie Vole (*Microtus ochrogaster*). *Journal of Mammalogy*, Vol. 74, No. 1, 1993, pp. 44-58.
- [3] Kurt Hornik, Maxwell Stinchcombe, Halbert White. Multilayer feedforward networks are universal approximators, *Neural Networks*, Vol. 2, Issue 5, 1989, pp. 359-366, ISSN 0893-6080, [http://dx.doi.org/10.1016/0893-6080\(89\)90020-8](http://dx.doi.org/10.1016/0893-6080(89)90020-8).
- [4] Anil K. Jain, Jianchang Mao, K.M. Mohiuddin. Artificial Neural Networks: A Tutorial, *Computer*, Vol. 29, no. 3, pp. 31-44, March, 1996
- [5] Dini, Steve, and Mark Serrano. "Combining Q-Learning with Artificial Neural Networks in an Adaptive Light Seeking Robot." (2012).