Tyler Duryee
Kirk Hewitt
Jacob Westerback

Cats and Dogs Sandbox Game

Game Concept

Cats and Dogs takes place in a modern day house. Our concept started with cats chasing mice. You play as a cat, and the primary goal is to catch all of the mice. The mice will try and eat the cheese without getting caught. A rival cat found its way into the house and will try and catch the mice faster than you. Finally, there is a dog that will try to catch and eat both cats, so it's best to steer clear of it.

How to Play

Use the W-A-S-D keys to move the cat, and the mouse to look around. Catch the mice to increase your score and to prevent them from eating the cheese. If the mice end up eating all of the cheese, then the game will be over.

Game Roadmap

Path Following

Path following has been implemented for the dog. The dog follows a path between bed and food bowl, and will leave the path if a cat comes within his range. If he becomes hungry it will resume path following to go and get food. Likewise, if both cats go out of the dogs seeking range, it will do the same and go back to its bed. Path following was implemented using a doubly linked list.

Path Finding

Path finding has been implemented for the mice. They wander around the house until either they come within range of a piece of cheese or a cat gets too close to them. If they get too close to a wall when wandering, they will turn away from that wall and resume wandering in a more open space.

Decision Trees

Decision trees have been implemented for each of the three AI entities in the game. The enemy cat and mice have similar trees, and will firstly flee the entity trying to eat them, then seek the entity they are trying to eat, and finally wander if neither are happening. The dog performs path finding if there are no cats in range, and will seek and try to eat the cats if they are.

State Machine

The dog is the only entity that has a state machine implemented, with boolean values representing if it is hungry, angry, or sleepy. Depending on these values, the dog will perform various actions such as going to get food if it's hungry or chasing a cat if it's angry.

Behavior Tree

We did not directly implement this. Prolog code for a behavior tree will be submitted separately.

Genetic Algorithm

For the genetic algorithm we evolved the nodes along the path that the dog follows in an effort to find the optimal path. Each node is a Vector 3 point, the start and end points were not mutable as they were based on object location. Each generation had 10 members. Each "chromosome" had 2 "bits", an x and z; as we were using Vector3 points as our data. The y point stayed constant as there is no y-axis motion for the dog. The phenotype range we used for evolution was -10 to +10 units to each bit. Fitness was calculated based on how often the dog chased the cat and how many cats he caught.

When evolving each of them separately, It took a 20 rounds to develop decent behavior. In the end the "optimal values were somewhat close to what we had started with.

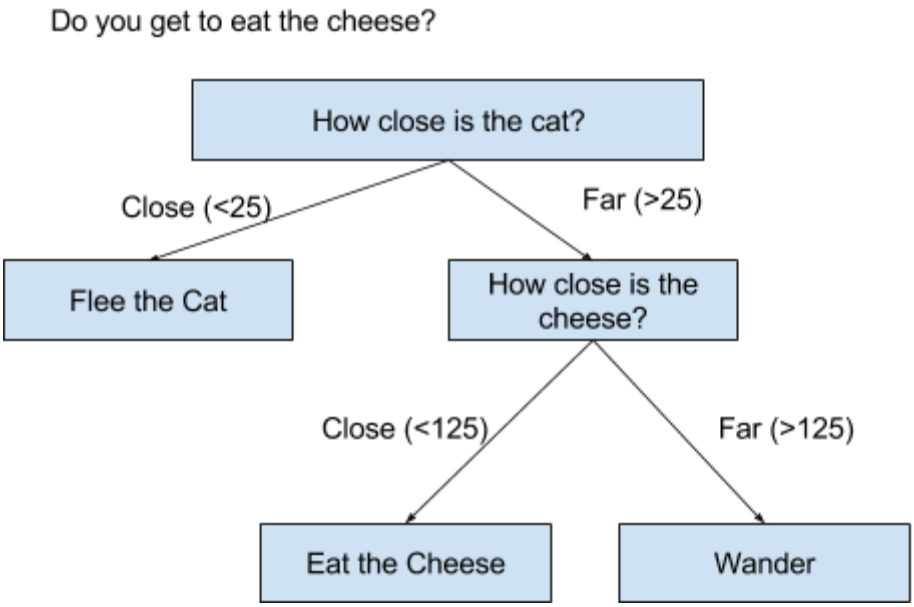
When evolving each of them separately, It took closer to 35 rounds to develop decent behavior. In the end the "optimal values were somewhat close to what we had started with.

This feature is not available in the game and was only used to develop the best path for the dog. Having the best path found, made the game better. There isn't really a way to make a workable playable integration of this algorithm.

Bayes Classifier

Bayes documentation has been provided in a separate file.

ID3-Tree



The tree makes sense, as you want the cat to be able to enter the fee radius with enough time for you to be able to run away effectively. This is very similarly to what we programed, and does capture the important decision points. If you are away from the cat, but not far enough that the cat will be able to get you while you are eating, then you should continue to flee.

A fair bit of generalization had to occur due to how the Bayes algorithm was run. The Bayes algorithm focused on finding the radii values and as such was not focused on making a decision about an action. As such the resulting ID3 tree is overgeneralized. None of the attributes were completely unnecessary but there were only two so if one were than it would not really be a tree anymore.

The Bayes algorithm was ultimately more effective at controlling the mice's behavior. But the results were able to be captured in the ID3 tree to demonstrate the resulting behavior.