



# CS 4701 Status Report

Kenta Takatsu | Pehuen Moure

## Summary

Our project is teaching an AI how to maneuver a bicycle to avoid an obstacle. We picked this agent because the bicycle requires multi-tasking: maneuvering and balancing. We designed this project with objectives that progressively become challenging. The first step we have taken is developing a genetic algorithm for navigation in a specific environment

## Completed Objectives

- Environment-aware agent
  - Objective: To navigate a given environment allowable using set of movements

## Agent

The bicycle agent contains following parameters:

- Position: cartesian coordinates in environment
- Direction: Normal vector from center of the bicyel
- Angle: The turning angle of the agent, where north, [0, -1], is 0, and counter clockwise rotation is positive.
- Chromosome: A sequence of "genes" where each individual gene is is a command for a specific fram in the simulation. The possible commands are left, right, or straight
- Fitness: the score which an agent has according to evaluation function

## Evaluation Functions

Evaluation function will encourage the gene selection of genetic algorithm (fitness). Our evaluation function is based on two factors: distance from target and number of steps necessary to reach the target. If the target is not reached then the distance is minimized, once an agent reaches the target it attempts to minimize the number of steps to reach the target.

## Algorithm

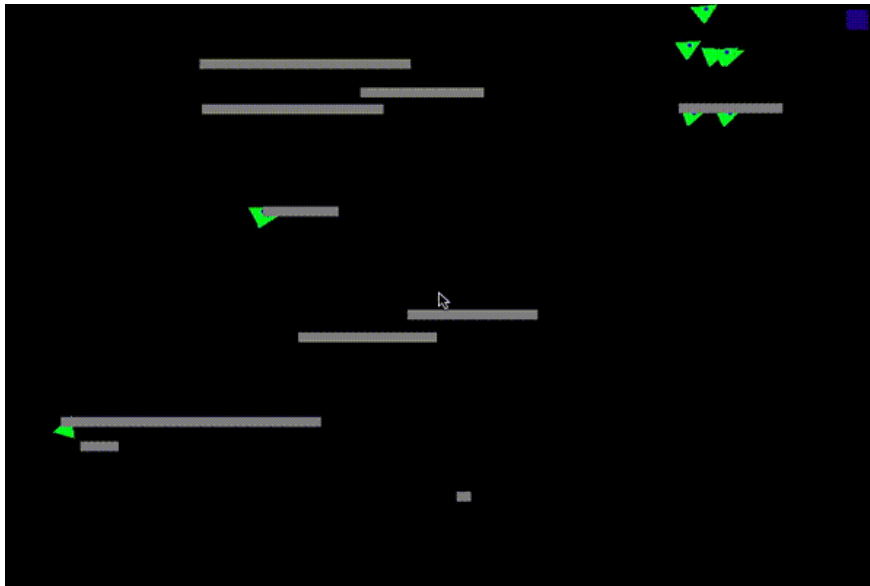
1. Initialize environment by setting the target destination and randomly distributing the walls

2. Initialize the agents with random chromosomes
3. Move set of live agents based on the next "gene" (command)
4. If the agents reached the target then calculate its fitness and return it.
5. Else if the agents died calculate its fitness and remove it from the live agents set.
6. Else if the set of live agents is not empty return to step 3
7. Else sort the agents based on their calculated fitness. Choose the top two agents and use their chromosomes as a basis for the next generations.

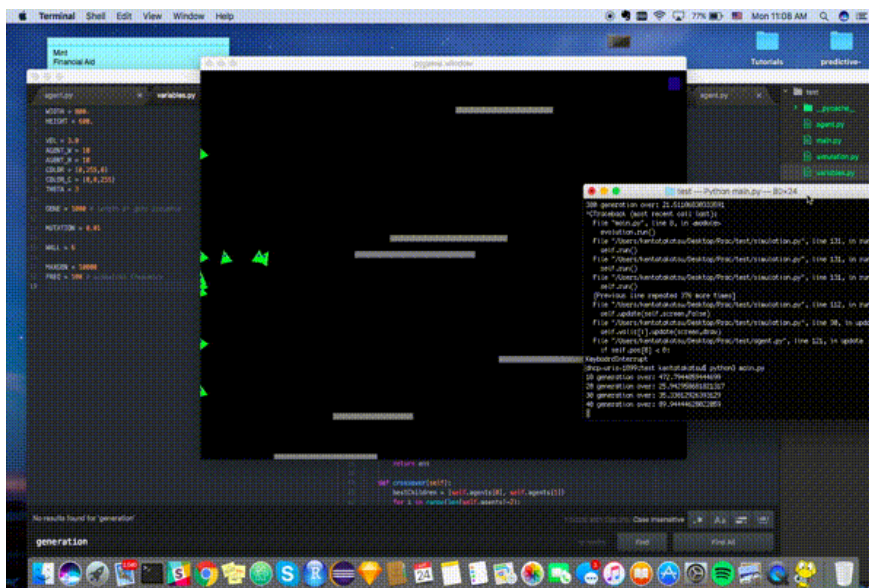
## Mutation

Once the two best agents are selected the next generations chromosomes are created as a 50% combination of both agents as well as some randomness.

## Simulation



We also developed a simailiton using moving obstacles:



## Upcoming Algorithms

- Feed-forward neural networks

- Convolutional neural networks
- Recurrent neural networks

Currently the bicycle is overfitting to the environment. The agent returned provides no assurances for any other environment, and would probably do very poorly. Our current algorithm also makes no assurances about conversion.

We plan on targeting this variance problem by using a recurrent neural network. The bicycle will get information about its surrounding in a form of the multi-dimensional vector space, feed that vector into neural network, and returns optimal operation.

state vector  $s$ :

- vector directions to the obstacles
- vector to the destination
- current bicycle state

