Animal AI Models in Simulated Environments

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

4850 - 03

Spring 2023

Professor Perry

February 24th 2023

INDY-1 Team

Name	Role	Email
Selene Espinoza	Owner, Project Manager, Misc Coder.	sebae97@gmail.com
JaDante Hendrick	Developer	jdpackers544@gmail.com
Jake Dunkley	Developer	jakeddunkley@gmail.com
Anna Alquisiras	Documentation, Researcher	annaalquisiras@gmail.com

Table of Contents

1.0 Introduction

This project will create an agent-based simulation of animal group movements in response to environmental factors. Populations are modeled by multiple data points, each of which models a unit of population (i.e. a herd or pack). Each species' population representatives are controlled by one neural network, which has been trained on historical ecological data. The simulation takes place on a 2D grid, with each square representing a small area of land of a preset scale to the animal model. Users are allowed to insert "human interference" or other environmental changes into the simulated area to see how the local fauna will respond. Simulated environments and animal models can be saved and loaded independently, and data from the entire simulation can be saved for later study.

With animal populations changing due to alterations to ecology and climate from human interaction, it is more important than ever to understand how animals react to specific environmental variables. Using an AI “cell” that analyzes its environment, packs of animals may be tracked more efficiently in theoretical environments. This software seeks to test that possibility, and hopefully create a foundation from which humans can further learn of their influences on animals and coexist with animals.

The purpose of this document is to represent the design and system requirements in a more readable way so that clients can understand the direction of the project in detail, and the developers can follow and implement from this.

1.2 Project Goals

The three main goals of our project include:

1. Developing a proof of concept
    a. Testing the theory that animals may be able to be tracked in the future using AI and ML
    b. Greater understanding for nature preservation organizations, scientists, or other groups that may wish to observe or change the environment
2. Provide demo models to help clients understand the functions of the AI
3. Provide an environment to showcase how the program works and a foundation for future development

1.3 Definitions and Acronyms

Learning phase - Where an animal model learns from real world data during long period of time

Reacting phase - Finished model interacts with a simulated environment to understand possible reactions

Animal model - The AI data set based off real-world ecological data that reacts to a simulated environment

AI FNN – Artificial Intelligence feed forward neural network

Animal cell - Instantiated object of the animal model in an environment; Al animal cells of a given species share the same model, and use environmental variables and distance from other cells of the same species as inputs to process

Simulated Environment - Digital recreation of terrain and other environmental factors to create inputs for the animal model

1.4 Assumptions

It is assumed that the clients during the Learning Phase will have a large enough pool of data to let an AI model learn.

It is assumed that the clients during the Reacting phase will have created or otherwise obtained premade AI models.

It is assumed that the clients during the Reacting phase will have a computer to operate the software on with the proper technical specifications needed.

2.0 Design Constraints

2.1 Environment

We are using ImGUI, GLFW and OpenGL.

Our data will come from previous research done and list them as we further develop our sources.

2.2 User Characteristics

Understanding the environment and the environmental properties of the simulated region.

People with access to greater research or pre-made animal AI models.

People with understanding of the importance of animal presences in ecologies and the effects of human habitation and interference.

2.3 Further System Developments

It may be possible to add more models and simulation tools to further track or edit the simulation (Reacting Phase).

In the future, merging Learning Phase into the software to streamline AI modeling.

A further understanding and implementation of timelines and reactions to various seasons and their effect on animal behavior.

More animal models and environmental layers being created for the sake of further research/experimentation.

3.0 Requirements

1. Learning Phase

1.1. Animal model (Model A)

1.1.1. Contains multiple inputs for the model (A_I)

1.1.1.1. Environmental layers, four inputs of each to match cardinal directions (A_{En}, A_{Ew}, A_{Ee}, A_{Es} for each layer A_E)

1.1.1.2. Distance from another cell (A_T)

1.1.2. Contains multiple outputs for the model (A_O)

1.1.2.1. Two outputs for vector movement (A_{Vx}, A_{Vy})

1.1.2.2. Four outputs for action to take next iteration ($A_m, A_s, A_d, A_n \in A_A$)

1.1.3. Has weights in fully connected hidden layers (Nodes in A_N)

1.1.3.1. Randomized weights at first

1.1.4. Loss function uses known observance values of specific packs/groups of animals and their location (Loss of A_L)

1.1.5. For each seasonal iteration, the AI is adjusted to learn to be more accurate with its predictions

1.1.5.1. Summer & Winter seasons running concurrently but separately, and operating off the same model

1.1.6. Range variable to tell cell how far to "see" (A_R)

1.1.6.1. Range is scaled based off of biological data of average travel distance per day and the scale of the map

1.1.7. Saved as an independent XML file

1.1.7.1. Contains the name and length of the name

1.1.7.2. Contains the “range” variable”

1.1.7.3. Contains the scale of maps

1.1.7.4. Contains the input names

1.1.7.5. Contains the node weights of the hidden layers

1.2. Animal cell (Cell object C)

1.2.1. Contains AI model (C_A)

1.2.1.1. AI is shared with all other cells of the species ($C^1{}_A, C^2{}_A, \ldots C^n{}_A = A$)

1.2.2. Observes the surrounding environment for model inputs

1.2.2.1. Cardinal direction inputs are created that scale with direction using range's calculations (C_{En}, C_{Ee}, C_{Es}, C_{Ew} becomes A_{En}, A_{Ee}, A_{Es}, A_{Ew} respectively for each layer of C_E)
1.2.2.2. Range from next closest cell (C_T becomes A_T)
1.2.3. Chooses actions based on its surroundings (C_0)
1.2.3.1. Vector-based movement, moves in a direction if it exceeds a threshold, horizontal or vertical, positive = East, North (C_{0x} and C_{0y} takes from A_{Vx}, A_{Vy})
1.2.3.2. Merge, Split, Die, and doing nothing (C_{0a} takes highest value from A_A)
1.2.4. Checks for proximity to expected movement and adds each of these distances to make a final loss value (C_L becomes A_L)
1.3. Learning environment (Environments E)
1.3.1. Two dimensional grid (Grid E_{xy})
1.3.2. Contains layers of environmental values and the observance values of packs ($E^1, E^2, \ldots \in E$)
1.3.3. Follows commands from cells
1.3.3.1. Merge deletes adjacent cell and itself and creates a new cell, Split creates a new cell, Die also deletes a cell (C_{0a})
1.3.3.2. Has a move function that a cell can call to move itself to a new location if possible (Uses C_{0x} and C_{0y} if not traveling beyond the edge of the map)
1.3.4. A timeline that has the environment change slightly so as to understand how cells will react and compare to real observance values and pack actions (E_T)
1.3.4.1. 95% of maps in the timeline will be for testing
1.3.4.2. 5% will be for evaluation and understanding model accuracy

2. Reacting Phase

2.1. Animal models (A^1, A^2, etc.)
2.1.1. Loadable from XML
2.1.2. Can automatically scale weights if some input data is not given in order to attempt to reproduce results with fewer inputs
2.1.3. Does not change or "learn" in this phase

2.1.4. Interacts with the simulation map using cells and model much like in learning phase, but without the loss function and change to weights

2.2. Simulated environment (Map M)

2.2.1. Map is loadable from file

2.2.2. Contains layers and input names (Set of M_E)

2.3. Software presentation (Program S)

2.3.1. Can display a simulated map from file (Map M)

2.3.1.1. Can load environmental layers from map and allow the user to "activate" and "deactivate" them (Boolean M_A array for whether each layer is in use)

2.3.1.2. Layer displays can be turned on and off to be seen on the environment simulation or not (Boolean M_V array for whether each layer is visible)

2.3.1.3. Layers are visible through monochrome color gradient on map (M_V)

2.3.1.4. Mouse over squares for info on each visible layer grid space

2.3.1.5. Layers can be connected manually if needed (M_{E1}, M_{E2}, … can connect to A_E)

2.3.2. Can load animal models from file (Model A^1, A^2, etc.)

2.3.2.1. Displays backpropagation ratios of model AI in subwindow

2.3.2.1.1. Selectable inputs and color gradients that represent effects on output (A_O)

2.3.2.2. Fully displays all nodes (A_I, A_N, A_O)

2.3.2.2.1. Hides hidden nodes by default (A_N)

2.3.2.3. Allows placement of Animal cells onto map or randomly scatters a set amount

2.3.3. Can create animal cells on the simulation grid

2.3.3.1. Placeable cells by clicking

2.3.3.2. Placeable "scatter" cells of an input amount randomly around the grid

2.3.4. Operates from a timeline (S_T)

2.3.4.1. Animal models react over time from starting positions of their cells and can be placed specifically in different positions

2.3.4.2. Map does not change (Phase 2 addition of timeline change)

2.3.4.3. Timeline is split between Summer and Winter (S_{Ts} and S_{Tw} respectively)

3. Version Control

3.1. GitHub Desktop

3.1.1. Local file changes reflected in environmental-simulation.github.io repository

3.2. Documents

3.2.1. Google Docs

APPENDICES

Github:

https://github.com/environmental-simulation/simulation-engine

ImGUI:

https://github.com/ocornut/imgui

OpenGL:

https://www.glprogramming.com/red/