

School of Computing

**Honours Report**

***Competing Cat and Mouse Game with AI Techniques***

**Sam Burns 2023**

This report is submitted as part of the requirements for the degree of

**BSc (Hons) in Computing: Application Software Development**

at Robert Gordon University, Aberdeen, Scotland

**I confirm that the work contained in this Honours project report has been composed solely by myself and has not been accepted in any previous application for a degree. All sources of information have been specifically acknowledged and all verbatim extracts are distinguished by quotation marks.**

Signed  Date 26/04/2023

Table of Contents

[Abstract 2](#_Toc133547109)

[Aim & Objectives 2](#_Toc133547110)

[Project Scope 2](#_Toc133547111)

[Literature Review 2](#_Toc133547112)

[Requirements Analysis 5](#_Toc133547113)

[Design and Implementation 7](#_Toc133547114)

[Testing & Results 14](#_Toc133547115)

[Test Series 1 – Base Values 14](#_Toc133547116)

[Test Series 2 – Cat Energy Reduced to 50 16](#_Toc133547117)

[Test Series 3 – Grid Size Increased to 40 x 40 19](#_Toc133547118)

[Test Series 4 – Cat Energy Reduced to 100, Grid Size increased to 40x40 21](#_Toc133547119)

[Test Series 5 – Cat Vision Reduced to 4.5 23](#_Toc133547120)

[Test Series 6 – No. Pickups increased to 9 25](#_Toc133547121)

[Test Series 8 – Capture Energy Set to 0 30](#_Toc133547122)

[Analysis 32](#_Toc133547123)

[Evaluation 33](#_Toc133547124)

[Conclusions & Future Work 34](#_Toc133547125)

[References 35](#_Toc133547126)

[Appendices 35](#_Toc133547127)

[Detailed Project Proposal 35](#_Toc133547128)

[Project title 35](#_Toc133547129)

[Background 36](#_Toc133547130)

[Aim & Objectives 37](#_Toc133547131)

[Tools & Technologies 37](#_Toc133547132)

[Project Plan 37](#_Toc133547133)

[Project Poster 38](#_Toc133547134)

[Project Log 38](#_Toc133547135)

[Ethics Form 40](#_Toc133547136)

[Source Code 44](#_Toc133547137)

# Abstract

The primary goal of this project was to develop a pair of AI agent that would compete against one another in a bid for resources. The eponymous Cat and Mouse – two teams with opposing goals. To survive, the Mice must collect food from the environment and attempt to outlast the Cat. The Cat, meanwhile, must hunt down the Mice quickly enough that it doesn’t starve to death.

Many examples of AI are purely collaborative, or work to oppose a human as often seen in video games. The interest behind this project is how these AI systems may interact with one another as they’re working towards their own goals.

# Aim & Objectives

To create a set of AI agents that compete against one another in a bid for resources

Objectives:

* Create a tile-based environment with barriers
* Define and implement behaviours for various types of agents
* Create AI algorithms to inform the agents’ decision-making processes

# Project Scope

## Literature Review

The topic of interest for this project is the interaction between different agents, and what behaviours develop as a result. To allow for emergent interactions to take place, I believe the best method is machine learning.

The topic of interest for this project is the interactions between competing agents, and what behaviours and tactics begin to develop as a result. To allow for these emergent interactions to take place, the best approach is likely to be machine learning over pre-programmed behaviours.

While some interesting behaviours can emerge from simpler systems consisting of pre-defined behaviours, it ends up being more consistent and limited than the ever-changing decisions of machine learning agents. An interesting example of a “zero player game” that follows a particular set of rules is John Conway’s Game of Life. It features a grid of cells that have one of two states - alive and dead. These cells follow a fairly simple set of rules, as follows:

* If a live cell has less than two live neighbours, it dies
* If a live cell has two or three live neighbours, it survives.
* If a dead cell has three live neighbours, it becomes live
* If a live cell has more than three live neighbours, it dies

With just these few rules, some very interesting behaviours and patterns emerge, such as those that can sustain themselves indefinitely, or those that travel across the grid. Some collections of cells may collide with one another and wipe each other out. The decision making in Game of Life is a lot less complex than that of which this project will be, however - the cells aren’t making decisions for themselves, but instead are simply subject to the rule system. This also means that their behaviours will never change - identical inputs will result in identical outputs.

More complicated decision trees may result in more interesting and varied behaviours - for example, AI in fighting games often simply reacts to the current situation by following a script. For example, the sf2platinum blog reveals that in Street Fighter II (1991), one strategy employed by the AI is specifically using the same move three times, and reacting to whether or not their opponent got hit by those three moves. This system operates using simple if/else statements, but still appears to be making complicated decisions from the outside thanks to the complexity of the game and variety of strategies.

There are three distinct approaches to machine learning algorithms - supervised learning, unsupervised learning, and reinforcement learning. Jo (2021) provides definitions for each of these categories - both supervised and unsupervised learning require a pre-defined set of data to train with. Supervised learning aims to “minimise the difference between the target output and computed output” (p. 11), and unsupervised attempts to do the same, albeit with an unlabelled set of data, so it relies on trial and error to determine the correct method of matching the desired output.

Both methodologies require input data, which they’ll attempt to match with their output. This project, however, is focused more on emergent behaviour - the interest is in what the results will end up being, and there therefore isn’t any expected output to attempt to match. As such, reinforcement learning will be the chosen approach for the agents’ learning models.

R. Sutton and A. Barto define reinforcement learning as “learning what to do - how to map situations to actions - so as to maximise a numerical reward signal” (p. 3), which matches well with the goal of the project. The intent here is to define a set of rules that the agents must abide by and adapt to, as well as modifying their own behaviour based on rewards and punishments, to allow for emergent behaviours to take place. Training for a specific behaviour would run in opposition to this goal.

A more complex example of emergent behaviour can be found in B. Baker et al.’s zero player game, where two teams of AI compete against one another in a game of hide and seek. In it, one team is tasked with avoiding the detection of the other team, and their means of doing so depends on the environment. Some environments allow the hiders to hide within an existing set of walls, blocking the entrances with objects they can manipulate. Others allow them to box themselves in using several different objects.

These agents acted on a reinforcement learning system, where they were punished or rewarded depending on their team’s success. Interestingly, it’s noted that there are “no direct incentives” (p. 6) for interacting with objects or exploring their environment, meaning the agents made the decision to start doing so entirely on their own, based entirely on the rewards they received for winning or losing matches.

B. Baker et al. detail that as more simulations were run, strategies and counter-strategies began to emerge. All agents would initially move randomly, before the seekers learned to chase the hiders. The hiders eventually learned to block off entrances to their shelter, to which the seekers would eventually climb over by using a ramp object. The hiders then learned to bring the ramp into their shelter with them before sealing off the entrances, thus preventing the seekers from making use of it, and so on.

One detail about the hide and seek game, however, is that both teams had similar sets of abilities. The hiders got a head start and could “lock” objects to prevent them from being grabbed, but aside from that they all shared the ability to drag objects and had similar movement abilities. This is one area where the cat and mouse project will differ, as the eponymous teams will have very different capabilities. Namely, that the cat agent will be able to hunt and kill mice agents, while all the mice agents can do is evade it.

The hide and seek game rewards all agents on a team the same way, and this project will follow suit. Each agent will have their own “hunger” metre that gradually ticks down, which can be refilled by gathering food (or in the cat’s case, hunting mice). The mouse agents will receive a negative reward when a mouse dies, whether by the cat’s hand or by starving.

The mice will act as a multi-agent system, while the cat will act as a single agent. However, if time permits, it may be possible to include the ability to add multiple cat agents to the environment. The mice agents will be incentivised to work together because of the collective rewards and punishment they’ll receive for occurrences such as a mouse being caught or food being gathered.

However, the cat and mouse game will also have different roles for the mouse agents - namely “gatherers”, who seek out food and collect it, and “distractors”, which will be suited for drawing the cat away from specific areas or its teammates. While in the hide and seek game, every team member is rewarded or punished equally, these two varieties of mouse will have different rewards for specific actions. For example, a gatherer will receive a punishment if it’s spotted by the cat, whereas a distractor won’t.

M. Wooldridge (2002) points out that different agents naturally have “spheres of influence” on their environment, which may coincide with one another. As such, it’s important for the mice to be able to communicate their decisions with one another to avoid situations like two mice attempting to collect the same piece of food. For simplicity’s sake, this would likely operate under the same vision system as is used for spotting the cat - if a mouse can see another mouse, it will know what its plan is. That way, they could decide to go after a different objective rather than wasting time pursuing the same one.

Wooldridge also mentions the relationships between agents, and how one agent may be considered the “boss” of another. Because victory is linked to the amount of food collected, the gatherer mice will likely be considered more important than the distractor mice, since they’re the only ones that can collect the food. If all gatherer mice were to be captured by the cat, then the distractors wouldn’t be able to refill their hunger metres and would simply have to attempt to outlast the cat, for example.

The hide and seek game also heavily features the agents affecting their environment to achieve their goals - pushing and locking objects as needed. In the cat and mouse game, the agents will have less influence on the environment itself - instead, focus will be placed on the agents. If two mouse agents are past a certain hunger threshold, they may reproduce to spawn a new mouse agent, which would give them a wider range to gather food from the area, but would also put more strain on the available resources, as well as providing more potential sources of food for the cat.

Agents in the hide and seek game are capable of examining their environment via a cone of vision in front of them - this cone gets blocked by walls or other objects. The cat and mouse agents will follow a similar principle, their vision being blocked by walls placed around the environment. Mice will also be able to “hear” the cat within a certain radius, allowing them to track its presence even if a wall is blocking their vision, provided they stay close enough to it.

As opposed to the 3D implementation of the hide and seek game, the cat and mouse game will be based on a 2D grid representing an overhead view of the environment. Agents will need to move throughout the grid, and as such will need a pathfinding algorithm. I. Zarembo and S. Kodors conclude that the Lifelong Planning A\* algorithm (LPA\*) is more efficient than the standard A\* algorithm when pathfinding on smaller grids - for example, LPA\* outpaced A\* on a 256x256 grid and smaller, but A\* was more efficient on 512x512 grids and above. While the difference is only a matter of milliseconds, these pathfinding calculations will be run often and by every agent present, so efficiency is important.

The larger a grid, the more unlikely the cat agent is to succeed in its task, so smaller grids are likely to be better for the simulation overall. As such, the LPA\* algorithm would be best for pathfinding purposes. Since agents will be occupying one space in the grid, a grid above 512x512 is certainly out of the question - it would take much too long for the cat to navigate, and the mice would have a much easier time avoiding it.

The actual decision making will be done in a similar method to the hide and seek game - agents will be placed in the environment and allowed to experiment, receiving rewards or punishments depending on their actions. They won’t have any particular programming encouraging them towards any specific behaviours, just a priority to desire rewards and avoid punishments.

Individual “rounds” are expected to take longer than those seen in the hide and seek game. The hide and seek rounds seem to last a matter of seconds, with the win condition being decided fairly quickly. The victory conditions in the cat and mouse game will be more complex, requiring the complete elimination of either party. These rounds will also naturally have a minimum duration thanks to the hunger system - even if the cat never catches a mouse, it will still last for however long its hunger metre takes to empty. The only fast method for a round to end would be for the cat to capture all the mice in quick succession.

In summary, this project will be focused on making use of machine learning algorithms to inform the decisions of two opposing AI teams. These AI teams will comprise of multi-agent systems which receive rewards and punishments as a team, though the team will be made up of agents with different roles that will receive different punishments for the same actions, to encourage particular strategies to form without explicitly programming them in.

## Requirements Analysis

Functional Requirements

Agents

* Must implement a set of interacting agents with reinforcement learning capabilities
* Must have two categories of agent - “mouse” agents and “cat” agents
* Mouse agents must operate as a multi-agent system, with multiple agents existing and collaborating
* The cat agent should act as a single-agent system
  + Could implement optional functionality for multiple cat agents to exist in one environment
* The cat and mouse agents must act in opposition to one another - the cat hunts the mice, and the mice attempt to evade capture
* Agents must be able to react to their environment and make decisions accordingly based on reinforcement learning
  + All agents should be able to see one another within a cone of vision
  + Mice agents should be able to hear the cat agent based on proximity, even without visibility
  + The cat agent should be able to move two spaces in one turn when it can see a mouse agent
* Agents must act on a turn-by-turn basis - once a decision is made, it’s executed and a new decision will be made based on the new circumstances
* All agents must operate on a “hunger” system - a value that goes down over time and can be replenished by gathering food
* Agents should be able to navigate between two points using the LPA\* pathfinding algorithm
* Mouse agents should be able to work together to achieve their goals
* Mouse agents should have two categories - one that focuses on completing their objective and one that focuses on distracting the cat agent
* Mouse gatherer agents must have the capability to collect and share food with other mice
* Agents should be able to determine one another’s goals when within line of sight, and evaluate which agent would be the most efficient if they both share the same goal (e.g. aiming to collect the same piece of food)

Environment

* Must implement an environment for the agents to populate
* Must be represented as a 2D overhead view
* Must use a tile-based grid to store environment details and agent positions
* Grid should be small - less than 512x512 - but large enough to give the agents room
* Must include “walls” that the agents can’t walk or see through
  + Should be randomly placed
  + Could optionally be manually placed by user instead
* Must include a system for randomly placing food for the mice to collect

Rules

* Must include victory states for the two types of agent - a cat victory occurs when all mice are eliminated (either by capture or depletion of hunger value), and a mouse victory occurs when the cat’s hunger value reaches zero
* Must include a rewards system for the agents to learn from - punishing actions like getting captured and rewarding actions like collecting food
* Rewards system should be distributed across the whole team when appropriate - e.g. a mouse being captured

Non-Functional Requirements

* A prototype with some agent functionality to act as a proof of concept should be completed before the Christmas holidays
* Should use Python, due to the range of libraries available for both machine learning and graphical display
* Could represent the grid in an appealing way - e.g. images to represent the cat and mice instead of coloured squares or letters

# Design and Implementation

During development, the design of several features ended up differing from initial plans, the most impactful one being the process behind the agents’ decision-making behaviours. The complexity of implementing reinforcement learning slowed development immensely, so my supervisor recommended pivoting away from that plan and focusing on an agenda-based system instead.

The agenda system still fulfils the project’s requirements – agents still make decisions based on their observations from their environment, they just do so based on a pre-programmed list of priorities, rather than on learned behaviour.

The first feature to be implemented was the 2D grid that acted as the environment. The grid is drawn using the pygame library – it draws a specified number of tiles on the screen, with margins between them so that each tile is distinct, even when adjacent to tiles of the same colour. Each tile’s colour is set according to the tile’s contents, as shown in the figure below.

|  |  |
| --- | --- |
| White | Empty |
| Black | Wall |
| Green | Food Pickup |
| Blue | Mouse |
| Red | Cat |

*Figure 1: A table listing what type of tile each colour represents.*

At this stage, tiles contents were simply set at random to test the colouration code.

A picture containing qr code

Description automatically generated

*Figure 2: The initial implementation of the grid, which colours tiles at random.*

The structure of the grid itself varied across development – initially, it was simply a 2D array of integers, each representing a single tile. This structure got altered to fit the needs of the program as time went on, culminating in the grid being a 2D array of objects, each storing the agent contained within the tile as well as whether the tile is a wall or not.

The next milestone was implementing an agent with basic movements. The grid was reworked to randomly place wall tiles throughout the environment, and a test agent was created. The agent was placed randomly somewhere on the grid, after first ensuring that it wasn’t being placed on top of a wall tile. Its behaviour in the first phase was simply to wander at random, checking its surrounding tiles for empty spaces and picking one to move to at random.

A picture containing text, crossword puzzle, bathroom, indoor

Description automatically generated

*Figure 3: The next implementation of the grid, which randomises wall placement and contains a randomly moving agent.*

The next iteration added pickups for the agent to collect, although it still moved completely randomly, so each instance of the agent collecting a pickup was down to chance. Whenever one is picked up, it’s placed in a random spot elsewhere on the grid.

Text, whiteboard

Description automatically generated

*Figure 4: The grid, containing a randomly moving agent and a pickup for it to collect. Each time it collects a pickup, its position is printed into the console.*

Agents were also given an energy value. Each turn, this value would decrease by one point, but picking up a food pickup would add to its energy. This was the introduction of the game’s endstate – if an agent runs out of energy, they’ll die and be removed from the environment. If all agents on a particular “team” (i.e. cats or mice) die, the game is ended.



*Figure 5: A simple end-of-game report in the console, showcasing how many turns the agent survived for and how many pickups it collected.*

Once basic movement was in, the next step was to allow the agent to observe its environment. The first step towards this was to define the agent’s maximum range of vision, which was adapted from A. Patel’s circle drawing algorithm for grids.

A picture containing calendar

Description automatically generated

*Figure 6: The agent’s defined maximum vision range, represented as a circle around the agent. For debug purposes, this information is printed into the console.*

Once the edges of its vision were defined, lines were drawn from the agent to each of these tiles, stopping wherever a wall was encountered. This allowed the agent to return the contents of the tiles surrounding it but prevented it from seeing anything past a wall.

Graphical user interface, text, application

Description automatically generated

*Figure 7: A debug representation of the agent’s vision. The X is the agent, the white squares are empty spaces, the black squares are the tiles in its vision range, and the W characters are wall tiles. Underneath, information on the single object within its vision is printed.*

The next step was to allow the agent to move towards the closest pickup that it could see – this was done simply by moving to whatever tile was closest to the pickup object in question.

Text, letter

Description automatically generated

*Figure 8: A list of objects the agent can see, along with the object that it determines is the closest.*

Finally, came the development of the agenda system. First, each agent was given a private version of the grid that represents their memory of the environment. As they explore and see the tiles, they’ll fill in the blanks on their memory grid.

With the agenda came the need for actions. One such action was to path to a specific spot, used to path towards any targets the agent has (food pickups for mice, mice for cats). The implementation’s structure was based on the one showcased in S. Lague’s video which covered the mechanics behind the algorithm. I opted to use the basic form of A\* initially, then convert it to the Lifelong Planning A\* Algorithm as outlined in the project scope. However, considering that it wasn’t guaranteed to have a noteworthy difference in performance compared to the standard A\* algorithm, I opted not to make this change in to focus on more necessary functionalities.

A picture containing table

Description automatically generated

*Figure 9: A representation of the agent’s memory of the environment. The question marks are tiles it hasn’t seen yet.*

Actions are set up as objects that the agents can call in order to receive instructions on what to do next. Tasks have two stages – the task is initialised, then the task follows instructions until its completion. The Move to Position task, for instance, instructs the agent to calculate a path to its destination when first called, then it follows that path until either the destination is reached or the task is abandoned.

Graphical user interface, text, application

Description automatically generated

*Figure 20: An excerpt of the code that controls the Move to Position action*

Each agent type has its own priority list to refer to when ordering its agenda.

Text

Description automatically generated

Text

Description automatically generated

*Figure 11: An excerpt of the code showcasing the lists that control the ordering of the agenda. The lists are ordered in importance from left to right.*

The dire priority list is used when the agent has low energy – since pickups restore energy, they prioritise gaining more if it’s low enough that acting otherwise would risk death anyway. This only applies to mice, however, as cats would have no reason to alter their behaviour in such an instance.

Each turn, an agent follows its current agenda, or reassesses its agenda if it’s been given reason to – for example, if they’ve completed all their tasks already (besides wandering at random) or if they’ve seen a new object in their environment. The actions available are listed in the figure below:

|  |  |
| --- | --- |
| MoveToPos | Used when they’ve spotted a pickup. They use the A\* algorithm to calculate a path towards the tile, then they follow it until their path is complete or something forces agenda reassessment. |
| Hide | Used when a threat is spotted. They’ll look for the nearest tile to them that isn’t within the threat’s line of sight and path to it. |
| Explore | Used when they aren’t aware of any threats or pickups. They’ll pick a random tile that isn’t within their memory and path towards it. |
| Wander | Used when all other options are exhausted. Picks a random adjacent tile and moves to it. |

*Figure 12: A table showcasing each type of action available for an agent to perform.*

Whenever the agenda is reassessed, the agent will examine its surroundings and memory, then pick its new agenda accordingly. It adds any possible actions to the agenda (e.g. it won’t add a “Move To” action if it doesn’t see anything), then these actions are ordered based on the priority lists.

Graphical user interface, text, timeline

Description automatically generated with medium confidence

*Figure 13: Console logs that showcase the agendas of two different Mouse agents.*

The final version also allows the user to pause the game by pressing P. When the game is paused, the tiles can be clicked to print any relevant information in the console. For standard tiles, this simply prints the coordinates and what the tile is.

Text

Description automatically generated

*Figure 14: Console logs showing the contents of the tiles that the user clicked.*

For tiles containing an agent, however, the console will print the information related to the agent, including its current location, what objects it can currently see, what objects it can remember, and its current agenda, as seen above. Clicking the same agent’s tile a second time will print more information, showing ASCII representations of the agent’s memory grid along with its current range of vision.

A picture containing table

Description automatically generated

*Figure 15: Console logs showcasing two ASCII representations of the grid. The left represents the agent’s memory of the environment, and the right represents the agent’s current vision.*

When the game is concluded (i.e. all the agents on one side of the conflict have been eliminated), the console will print out a list of stats related to each agent, including how many turns it lasted before death and what the cause of death was (either starving to death by failing to retrieve pickups/mice, or being caught by the cat)

Timeline

Description automatically generated

*Figure 16: Console logs showcasing the end-of-game reports, which tell the user each agent’s agenda, vision, memory, and other stats such as lifetime and killer.*

# Testing & Results

Testing was done by making changes to particular values and seeing what the result was. The possible variables and their default values are as shown in the following figure:

Text

Description automatically generated

*Figure 17: An excerpt from the code showing a list of properties and their default values.*

## Test Series 1 – Base Values

|  |  |
| --- | --- |
| Test No. | Result |
| Test 1 | Cat Victory  Cat   * Energy: 224 * Lifetime: 126 * Percentage of Grid Explored: 100%   Mouse 1   * Last Location: [1, 0] * Remaining Energy: 108 * Lifetime: 52 * Killer: Cat * Percentage of Grid Explored: 73%   Mouse 2   * Last Location: [4, 0] * Remaining Energy: 95 * Lifetime: 45 * Killer: Cat * Percentage of Grid Explored: 71%   Mouse 3   * Last Location: [1, 0] * Remaining Energy: 135 * Lifetime: 125 * Killer: Cat * Percentage of Grid Explored: 98% |
| Test 2 | Cat Victory  Cat   * Energy: 193 * Lifetime: 157 * Percentage of Grid Explored: 100%   Mouse 1   * Last Location: [2, 1] * Energy Remaining: 144 * Lifetime: 156 * Killer: Cat * Percentage of Grid Explored: 100%   Mouse 2   * Last Location: [1, 1] * Remaining Energy: 77 * Lifetime: 103 * Killer: Cat * Percentage of Grid Explored: 98%   Mouse 3   * Last Location: [0, 1] * Remaining Energy: 78 * Lifetime: 42 * Killer: Cat * Percentage of Grid Explored: 58% |
| Test 3 | Cat Victory  Cat   * Energy: 246 * Lifetime: 104 * Percentage of Grid Explored: 100%   Mouse 1   * Last Location: [0, 1] * Remaining Energy: 63 * Lifetime: 37 * Killer: Cat * Percentage of Grid Explored: 78 %   Mouse 2   * Last Location: [0, 0] * Remaining Energy: 197 * Lifetime: 103 * Killer: Cat * Percentage of Grid Explored: 100%   Mouse 3   * Last Location: [1, 0] * Remaining Energy: 85 * Lifetime: 35 * Killer: Cat * Percentage of Grid Explored: 44% |
| Test 4 | Cat Victory  Cat   * Energy: 195 * Lifetime: 155 * Percentage of Grid Explored: 100%   Mouse 1   * Last Location: [1, 0] * Remaining Energy: 72 * Lifetime: 68 * Killer: Cat * Percentage of Grid Explored: 67 %   Mouse 2   * Last Location: [0, 0] * Remaining Energy: 166 * Lifetime: 154 * Killer: Cat * Percentage of Grid Explored: 94%   Mouse 3   * Last Location: [0, 0] * Remaining Energy: 76 * Lifetime: 24 * Killer: Cat * Percentage of Grid Explored: 46% |
| Test 5 | Cat Victory  Cat   * Energy: 232 * Lifetime: 118 * Percentage of Grid Explored: 66%   Mouse 1   * Last Location: [0, 1] * Remaining Energy: 66 * Lifetime: 74 * Killer: Cat * Grid Explored: 83%   Mouse 2   * Last Location: [0, 0] * Remaining Energy: 63 * Lifetime: 117 * Killer: Cat * Percentage of Grid Explored: 100%   Mouse 3   * Last Location: [0, 0] * Remaining Energy: 116 * Lifetime: 44 * Killer: Cat * Percentage of Grid Explored: 67% |

## Test Series 2 – Cat Energy Reduced to 50

|  |  |
| --- | --- |
| Test No. | Result |
| Test 1 | Mouse Victory  Cat   * Last Location: [0, 0] * Remaining Energy: 0 * Lifetime: 150 * Killer: Starvation * Percentage of Grid Explored: 84%   Mouse 1   * Last Location: [0, 0] * Remaining Energy: 54 * Lifetime: 46 * Killer: Cat * Percentage of Grid Explored: 53%   Mouse 2   * Last Location: [0, 1] * Remaining Energy: 85 * Lifetime: 35 * Killer: Cat * Percentage of Grid Explored: 74%   Mouse 3   * Energy: 10 * Lifetime: 150 * Percentage of Grid Explored: 84%   Note: Cat and Mouse 3 both got stuck until the Cat ran out of energy |
| Test 2 | Cat Victory  Cat   * Energy: 59 * Lifetime: 141 * Percentage of Grid Explored: 98%   Mouse 1   * Last Location: [1, 0] * Remaining Energy: 85 * Lifetime: 15 * Killer: Cat * Percentage of Grid Explored: 20%   Mouse 2   * Last Location: [0, 1] * Remaining Energy: 60 * Lifetime: 140 * Killer: Cat * Percentage of Grid Explored: 100%   Mouse 3   * Last Location: [1, 0] * Remaining Energy: 63 * Lifetime: 77 * Killer: Cat * Percentage of Grid Explored: 94% |
| Test 3 | Mouse Victory  Cat   * Last Location: [10, 0] * Remaining Energy: 0 * Lifetime: 50 * Killer: Starvation * Percentage of Grid Explored: 95%   Mouse 1   * Energy: 50 * Lifetime: 50 * Percentage of Grid Explored: 68%   Mouse 2   * Energy: 50 * Lifetime: 50 * Percentage of Grid Explored: 65%   Mouse 3   * Energy: 70 * Lifetime: 50 * Percentage of Grid Explored: 74% |
| Test 4 | Cat Victory  Cat   * Energy: 88 * Lifetime: 112 * Percentage of Grid Explored: 83%   Mouse 1   * Last Location: [1, 0] * Remaining Energy: 82 * Lifetime: 38 * Killer: Cat * Percentage of Grid Explored: 46%   Mouse 2   * Last Location: [5, 0] * Remaining Energy: 149 * Lifetime: 111 * Killer: Cat * Percentage of Grid Explored: 100%   Mouse 3   * Last Location: [15, 1] * Remaining Energy: 113 * Lifetime: 67 * Killer: Cat * Percentage of Grid Explored: 88% |
| Test 5 | Cat Victory  Cat   * Energy: 146 * Lifetime: 54 * Percentage of Grid Explored: 96%   Mouse 1   * Last Location: [15, 1] * Remaining Energy: 107 * Lifetime: 53 * Killer: Cat * Percentage of Grid Explored: 87%   Mouse 2   * Last Location: [3, 17] * Remaining Energy: 97 * Lifetime: 3 * Killer: Cat * Percentage of Grid Explored: 15%   Mouse 3   * Last Location: [4, 2] * Remaining Energy: 79 * Lifetime: 21 * Killer: Cat * Percentage of Grid Explored: 55% |

## Test Series 3 – Grid Size Increased to 40 x 40

|  |  |
| --- | --- |
| Test 1 | Cat Victory  Cat   * Energy: 110 * Lifetime: 140 * Percentage of Grid Explored: 51%   Mouse 1   * Last Location: [0, 1] * Remaining Energy: 33 * Lifetime: 67 * Killer: Cat * Percentage of Grid Explored: 22%   Mouse 2   * Last Location: [33, 13] * Remaining Energy: 0 * Lifetime: 140 * Killer: Starvation * Percentage of Grid Explored: 52%   Mouse 3   * Last Location: [25, 7] * Remaining Energy: 0 * Lifetime: 140 * Killer: Starvation * Percentage of Grid Explored: 55% |
| Test 2 | Cat Victory  Cat   * Energy: 110 * Lifetime: 140 * Percentage of Grid Explored: 43%   Mouse 1   * Last Location: [3, 0] * Remaining Energy: 0 * Lifetime: 140 * Killer: Starvation * Percentage of Grid Explored: 61%   Mouse 2   * Last Location: [21, 11] * Remaining Energy: 0 * Lifetime: 120 * Killer: Starvation * Percentage of Grid Explored: 35%   Mouse 3   * Last Location: [14, 0] * Remaining Energy: 29 * Lifetime: 71 * Killer: Cat * Percentage of Grid Explored: 40%   Note: Mouse 3 did still corner itself, just against a wall and not the left boundary of the grid |
| Test 3 | Cat Victory  Cat   * Energy: 90 * Lifetime: 160 * Percentage of Grid Explored: 79%   Mouse 1   * Last Location: [27, 15] * Remaining Energy: 0 * Lifetime: 100 * Killer: Starvation * Percentage of Grid Explored: 27%   Mouse 2   * Last Location: [23, 11] * Remaining Energy: 0 * Lifetime: 160 * Killer: Starvation * Percentage of Grid Explored: 78%   Mouse 3   * Last Location: [1, 0] * Remaining Energy: 39 * Lifetime: 81 * Killer: Cat * Percentage of Grid Explored: 47% |
| Test 4 | Cat Victory  Cat   * Energy: 120 * Lifetime: 180 * Percentage of Grid Explored: 70%   Mouse 1   * Last Location: [33, 21] * Remaining Energy: 0 * Lifetime: 120 * Killer: Cat * Percentage of Grid Explored: 38%   Mouse 2   * Last Location: [5, 0] * Remaining Energy: 65 * Lifetime: 55 * Killer: Cat * Percentage of Grid Explored: 29%   Mouse 3   * Last Location: [23, 24] * Remaining Energy: 0 * Lifetime: 180 * Killer: Starvation * Percentage of Grid Explored: 56% |
| Test 5 | Cat Victory  Cat   * Energy: 130 * Lifetime: 120 * Percentage of Grid Explored: 51%   Mouse 1   * Last Location: [8, 3] * Remaining Energy: 0 * Lifetime: 120 * Killer: Starvation * Percentage of Grid Explored: 54%   Mouse 2   * Last Location: [27, 20] * Remaining Energy: 0 * Lifetime: 100 * Killer: Starvation * Percentage of Grid Explored: 36%   Mouse 3   * Last Location: [0, 1] * Remaining Energy: 41 * Lifetime: 59 * Killer: Cat * Percentage of Grid Explored: 34% |

## Test Series 4 – Cat Energy Reduced to 100, Grid Size increased to 40x40

|  |  |
| --- | --- |
| Test 1 | Mouse Victory  Cat   * Last Location: [0, 1] * Remaining Energy: 0 * Lifetime: 100 * Killer: Starvation * Percentage of Grid Explored: 59%   Mouse 1   * Energy: 20 * Lifetime: 100 * Percentage of Grid Explored: 52%   Mouse 2   * Energy: 60 * Lifetime: 100 * Percentage of Grid Explored: 39%   Mouse 3   * Energy: 40 * Lifetime: 100 * Percentage of Grid Explored: 47% |
| Test 2 | Mouse Victory  Cat   * Last Location: [18, 25] * Remaining Energy: 0 * Lifetime: 200 * Killer: Starvation * Percentage of Grid Explored: 58%   Mouse 1   * Energy: 20 * Lifetime: 200 * Percentage of Grid Explored: 77%   Mouse 2   * Last Location: [24, 6] * Remaining Energy: 45 * Lifetime: 75 * Killer: Cat * Percentage of Grid Explored: 53%   Mouse 3   * Last Location: [28, 0] * Remaining Energy: 0 * Lifetime: 140 * Killer: Cat * Percentage of Grid Explored: 55% |
| Test 3 | Mouse Victory  Cat   * Last Location: [0, 7] * Remaining Energy: 0 * Lifetime: 150 * Killer: Starvation * Percentage of Grid Explored: 54%   Mouse 1   * Energy: 10 * Lifetime: 150 * Percentage of Grid Explored: 55%   Mouse 2   * Last Location: [9, 0] * Remaining Energy: 43 * Lifetime: 57 * Killer: Cat * Percentage of Grid Explored: 37%   Mouse 3   * Last Location: [25, 30] * Remaining Energy: 0 * Lifetime: 120 * Killer: Starvation * Percentage of Grid Explored: 34% |
| Test 4 | Cat Victory  Cat   * Energy: 30 * Lifetime: 120 * Percentage of Grid Explored: 70%   Mouse 1   * Last Location: [28, 20] * Remaining Energy: 0 * Lifetime: 120 * Killer: Starvation * Percentage of Grid Explored: 41%   Mouse 2   * Last Location: [1, 1] * Remaining Energy: 34 * Lifetime: 86 * Killer: Cat * Percentage of Grid Explored: 44%   Mouse 3   * Last Location: [25, 26] * Remaining Energy: 0 * Lifetime: 100 * Killer: Starvation * Percentage of Grid Explored: 32% |
| Test 5 | Mouse Victory  Cat   * Last Location: [1, 1] * Remaining Energy: 0 * Lifetime: 100 * Killer: Starvation * Percentage of Grid Explored: 41%   Mouse 1   * Energy: 20 * Lifetime: 100 * Percentage of Grid Explored: 30%   Mouse 2   * Energy: 40 * Lifetime: 100 * Percentage of Grid Explored: 48%   Mouse 3   * Energy: 40 * Lifetime: 100 * Percentage of Grid Explored: 38% |

## Test Series 5 – Cat Vision Reduced to 4.5

|  |  |
| --- | --- |
| Test 1 | Cat Victory  Cat   * Energy: 243 * Lifetime: 107 * Percentage of Grid Explored: 70%   Mouse 1   * Last Location: [1, 0] * Remaining Energy: 104 * Lifetime: 36 * Killer: Cat * Percentage of Grid Explored: 67%   Mouse 2   * Last Location: [11, 2] * Remaining Energy: 123 * Lifetime: 77 * Killer: Cat * Percentage of Grid Explored: 93%   Mouse 3   * Last Location: [0, 0] * Remaining Energy: 94 * Lifetime: 106 * Killer: Cat * Percentage of Grid Explored: 86% |
| Test 2 | Cat Victory  Cat   * Energy: 186 * Lifetime: 164 * Percentage of Grid Explored: 74%   Mouse 1   * Last Location: [1, 2] * Remaining Energy: 17 * Lifetime: 103 * Killer: Cat * Percentage of Grid Explored: 56%   Mouse 2   * Last Location: [8, 10] * Remaining Energy: 42 * Lifetime: 118 * Killer: Cat * Percentage of Grid Explored: 90%   Mouse 3   * Last Location: [1, 2] * Remaining Energy: 17 * Lifetime: 163 * Killer: Cat * Percentage of Grid Explored: 84%   Note: Game came to a halt for a while as Cat and Mouse 1 got stuck moving back and forth beside each other. Eventually broke out of it when Mouse 1 fell below 20 energy and began to prioritise food instead of hiding. |
| Test 3 | Cat Victory  Cat   * Current Location: [19, 2] * Energy: 40 * Lifetime: 260 * Percentage of Grid Explored: 99%   Mouse 1   * Last Location: [0, 0] * Remaining Energy: 113 * Lifetime: 67 * Killer: Cat * Percentage of Grid Explored: 77%   Mouse 2   * Last Location: [1, 7] * Remaining Energy: 0 * Lifetime: 260 * Killer: Starvation * Percentage of Grid Explored: 99%   Mouse 3   * Last Location: [1, 0] * Remaining Energy: 55 * Lifetime: 105 * Killer: Cat * Percentage of Grid Explored: 89%   Note: Cat and Mouse 2 both got trapped looping failed A\* pathing attempts until M2 starved to death |
| Test 4 | Cat Victory  Cat   * Energy: 147 * Lifetime: 203 * Percentage of Grid Explored: 94%   Mouse 1   * Last Location: [0, 0] * Remaining Energy: 68 * Lifetime: 112 * Killer: Cat * Percentage of Grid Explored: 76%   Mouse 2   * Last Location: [0, 1] * Remaining Energy: 46 * Lifetime: 94 * Killer: Cat * Percentage of Grid Explored: 74%   Mouse 3   * Last Location: [3, 0] * Remaining Energy: 118 * Lifetime: 202 * Killer: Cat * Percentage of Grid Explored: 100% |
| Test 5 | Cat Victory  Cat   * Energy: 186 * Lifetime: 164 * Percentage of Grid Explored: 91%   Mouse 1   * Last Location: [19, 0] * Remaining Energy: 121 * Lifetime: 119 * Killer: Cat * Percentage of Grid Explored: 80%   Mouse 2   * Last Location: [7, 1] * Remaining Energy: 120 * Lifetime: 20 * Killer: Cat * Percentage of Grid Explored: 67%   Mouse 3   * Last Location: [13, 0] * Remaining Energy: 57 * Lifetime: 163 * Killer: Cat * Percentage of Grid Explored: 100% |

## Test Series 6 – No. Pickups increased to 9

|  |  |
| --- | --- |
| Test 1 | Cat Victory  Cat   * Energy: 274 * Lifetime: 76 * Percentage of Grid Explored: 82%   Mouse 1   * Last Location: [1, 0] * Remaining Energy: 69 * Lifetime: 31 * Killer: Cat * Percentage of Grid Explored: 52%   Mouse 2   * Last Location: [1, 0] * Remaining Energy: 73 * Lifetime: 47 * Killer: Cat * Percentage of Grid Explored: 86%   Mouse 3   * Last Location: [19, 1] * Remaining Energy: 305 * Lifetime: 75 * Killer: Cat * Percentage of Grid Explored: 85% |
| Test 2 | Cat Victory  Cat   * Energy: 283 * Lifetime: 67 * Percentage of Grid Explored: 72%   Mouse 1   * Last Location: [0, 1] * Remaining Energy: 102 * Lifetime: 18 * Killer: Cat * Percentage of Grid Explored: 37%   Mouse 2   * Last Location: [0, 0] * Remaining Energy: 194 * Lifetime: 66 * Killer: Cat * Percentage of Grid Explored: 82%   Mouse 3   * Last Location: [12, 0] * Remaining Energy: 89 * Lifetime: 31 * Killer: Cat * Percentage of Grid Explored: 37% |
| Test 3 | Cat Victory  Cat   * Energy: 222 * Lifetime: 128 * Percentage of Grid Explored: 99%   Mouse 1   * Last Location: [0, 0] * Remaining Energy: 124 * Lifetime: 56 * Killer: Cat * Percentage of Grid Explored: 81%   Mouse 2   * Last Location: [0, 0] * Remaining Energy: 75 * Lifetime: 85 * Killer: Cat * Percentage of Grid Explored: 58%   Mouse 3   * Last Location: [16, 0] * Remaining Energy: 353 * Lifetime: 127 * Killer: Cat * Percentage of Grid Explored: 99% |
| Test 4 | Cat Victory  Cat   * Energy: 253 * Lifetime: 97 * Percentage of Grid Explored: 90%   Mouse 1   * Last Location: [1, 1] * Remaining Energy: 118 * Lifetime: 62 * Killer: Cat * Percentage of Grid Explored: 73%   Mouse 2   * Last Location: [3, 2] * Remaining Energy: 99 * Lifetime: 21 * Killer: Cat * Percentage of Grid Explored: 60%   Mouse 3   * Last Location: [2, 5] * Remaining Energy: 284 * Lifetime: 96 * Killer: Cat * Percentage of Grid Explored: 86% |
| Test 5 | Cat Victory  Cat   * Energy: 293 * Lifetime: 57 * Percentage of Grid Explored: 85%   Mouse 1   * Last Location: [1, 2] * Remaining Energy: 144 * Lifetime: 56 * Killer: Cat * Percentage of Grid Explored: 81%   Mouse 2   * Last Location: [10, 3] * Remaining Energy: 120 * Lifetime: 40 * Killer: Cat * Percentage of Grid Explored: 61%   Mouse 3   * Last Location: [0, 10] * Remaining Energy: 97 * Lifetime: 23 * Killer: Cat * Percentage of Grid Explored: 32% |

Test Series 7 – Set wall percentage to 30

|  |  |
| --- | --- |
| Test 1 | Cat Victory  Cat   * Energy: 200 * Lifetime: 100 * Percentage of Grid Explored: 94%   Mouse 1   * Last Location: [18, 12] * Remaining Energy: 0 * Lifetime: 100 * Killer: Starvation * Percentage of Grid Explored: 5%   Mouse 2   * Last Location: [2, 0] * Remaining Energy: 83 * Lifetime: 17 * Killer: Cat * Percentage of Grid Explored: 41%   Mouse 3   * Last Location: [4, 16] * Remaining Energy: 67 * Lifetime: 53 * Killer: Cat * Percentage of Grid Explored: 83%   Note: Mouse 1 spawned surrounded by walls and could not move |
| Test 2 | Cat Victory  Cat   * Energy: 283 * Lifetime: 67 * Percentage of Grid Explored: 69%   Mouse 1   * Last Location: [19, 1] * Remaining Energy: 77 * Lifetime: 43 * Killer: Cat * Percentage of Grid Explored: 62%   Mouse 2   * Last Location: [7, 4] * Remaining Energy: 94 * Lifetime: 66 * Killer: Cat * Percentage of Grid Explored: 64%   Mouse 3   * Last Location: [9, 7] * Remaining Energy: 114 * Lifetime: 26 * Killer: Cat * Percentage of Grid Explored: 33% |
| Test 3 | Cat Victory  Cat   * Energy: 294 * Lifetime: 56 * Percentage of Grid Explored: 84%   Mouse 1   * Last Location: [13, 2] * Remaining Energy: 78 * Lifetime: 22 * Killer: Cat * Percentage of Grid Explored: 43%   Mouse 2   * Last Location: [2, 6] * Remaining Energy: 45 * Lifetime: 55 * Killer: Cat * Percentage of Grid Explored: 43%   Mouse 3   * Last Location: [13, 17] * Remaining Energy: 99 * Lifetime: 1 * Killer: Cat * Percentage of Grid Explored: 12% |
| Test 4 | Cat Victory  Cat   * Energy: 246 * Lifetime: 104 * Percentage of Grid Explored: 87%   Mouse 1   * Last Location: [17, 0] * Remaining Energy: 57 * Lifetime: 103 * Killer: Cat * Percentage of Grid Explored: 74%   Mouse 2   * Last Location: [1, 1] * Remaining Energy: 69 * Lifetime: 31 * Killer: Cat * Percentage of Grid Explored: 39%   Mouse 3   * Last Location: [17, 6] * Remaining Energy: 94 * Lifetime: 6 * Killer: Cat * Percentage of Grid Explored: 23% |
| Test 5 | Cat Victory  Cat   * Energy: 267 * Lifetime: 83 * Percentage of Grid Explored: 66%   Mouse 1   * Last Location: [1, 2] * Remaining Energy: 50 * Lifetime: 70 * Killer: Cat * Percentage of Grid Explored: 73%   Mouse 2   * Last Location: [1, 0] * Remaining Energy: 88 * Lifetime: 52 * Killer: Cat * Percentage of Grid Explored: 60%   Mouse 3   * Last Location: [3, 12] * Remaining Energy: 18 * Lifetime: 82 * Killer: Cat * Percentage of Grid Explored: 57% |

## Test Series 8 – Capture Energy Set to 0

|  |  |
| --- | --- |
| Test 1 | Cat Victory  Cat   * Energy: 77 * Lifetime: 123 * Percentage of Grid Explored: 99%   Mouse 1   * Last Location: [0, 1] * Remaining Energy: 60 * Lifetime: 40 * Killer: Cat * Percentage of Grid Explored: 84%   Mouse 2   * Last Location: [0, 1] * Remaining Energy: 18 * Lifetime: 122 * Killer: Cat * Percentage of Grid Explored: 91%   Mouse 3   * Last Location: [0, 1] * Remaining Energy: 45 * Lifetime: 75 * Killer: Cat * Percentage of Grid Explored: 88% |
| Test 2 | Cat Victory  Cat   * Energy: 59 * Lifetime: 141 * Percentage of Grid Explored: 99%   Mouse 1   * Last Location: [1, 12] * Remaining Energy: 140 * Lifetime: 140 * Killer: Cat * Percentage of Grid Explored: 100%   Mouse 2   * Last Location: [1, 0] * Remaining Energy: 96 * Lifetime: 64 * Killer: Cat * Percentage of Grid Explored: 74%   Mouse 3   * Last Location: [0, 1] * Remaining Energy: 71 * Lifetime: 29 * Killer: Cat * Percentage of Grid Explored: 43% |
| Test 3 | Cat Victory  Cat   * Energy: 50 * Lifetime: 150 * Percentage of Grid Explored: 100%   Mouse 1   * Last Location: [0, 1] * Remaining Energy: 69 * Lifetime: 71 * Killer: Cat * Percentage of Grid Explored: 89%   Mouse 2   * Last Location: [0, 1] * Remaining Energy: 131 * Lifetime: 149 * Killer: Cat * Percentage of Grid Explored: 100%   Mouse 3   * Last Location: [8, 2] * Remaining Energy: 93 * Lifetime: 7 * Killer: Cat * Percentage of Grid Explored: 41% |
| Test 4 | Cat Victory  Cat   * Energy: 98 * Lifetime: 102 * Percentage of Grid Explored: 93%   Mouse 1   * Last Location: [7, 0] * Remaining Energy: 167 * Lifetime: 93 * Killer: Cat * Percentage of Grid Explored: 91%   Mouse 2   * Last Location: [1, 0] * Remaining Energy: 59 * Lifetime: 101 * Killer: Cat * Percentage of Grid Explored: 100%   Mouse 3   * Last Location: [1, 0] * Remaining Energy: 53 * Lifetime: 47 * Killer: Cat * Percentage of Grid Explored: 62% |
| Test 5 | Cat Victory  Cat   * Energy: 63 * Lifetime: 137 * Percentage of Grid Explored: 95%   Mouse 1   * Last Location: [0, 0] * Remaining Energy: 22 * Lifetime: 98 * Killer: Cat * Percentage of Grid Explored: 95%   Mouse 2   * Last Location: [0, 0] * Remaining Energy: 144 * Lifetime: 136 * Killer: Cat * Percentage of Grid Explored: 96%   Mouse 3   * Last Location: [0, 0] * Remaining Energy: 70 * Lifetime: 30 * Killer: Cat * Percentage of Grid Explored: 60% |

## Analysis

Out of forty total games, the mice won six games overall. Two wins took place in Test Series 2, where the Cat’s base energy was set to fifty, and four took place in Test Series 4, where the Cat’s energy was set to a hundred (the same as the mice) and the grid size was set to 40x40.

Mice captures overwhelmingly took place in the upper few rows of the grid. Across ninety-four captures, forty-five took place on Y = 0, twenty-five took place on Y = 1, and nine took place on Y = 2. The rest took place on various Y levels from 3 to 21.

Test Series 1 generally resulted in quick captures for the Cat – the Mice often had short lifetimes, despite having high energy levels. The Cat generally had an overabundance of energy, at times having over three times what a Mouse may have.

Test Series 2 resulted in two wins for the Mice – once because of faulty pathfinding resulting in both parties being unable to move and allowing Mouse 3 to outlast the Cat, and once because the Cat simply never caught up to any of the Mice.

Test Series 3 resulted in many more starvation deaths than usual – nine starvations to six captures. This means that often the Cat simply outlasted the Mice due to its higher starting energy reserves – nearly half of the Mice never managed to pick up any food before either starving or being captured.

Test Series 4 is the only test series where the Mice won the majority of the games. In two games, the Cat didn’t catch a single Mouse, and it only managed to win one game out of the five.

Test Series 5 didn’t seem to have a major impact on the results, with the statistics overall resembling Series 1 rather closely.

Test Series 6 allowed the Mice to store more energy overall, but their lifetimes often still ended up being similar to those from Series 1.

Test Series 7 often resulted in very short games overall.

Test Series 8 didn’t have much impact compared to Series 1, besides the Cat usually being left with lower energy reserves of around 50-100.

# Evaluation

It’s clear that the Mice’s best chance of success is when the Cat has lower starting energy overall. Because the Cat moves at the same speed and with the same capabilities as the Mice, it means that once a Mouse is spotted, it has a difficult time escaping from the Cat – the best chance of survival is never meeting it in the first place.

Many Mouse deaths are a result of them getting cornered by the Cat – almost exclusively in the top of the grid at Y = 0. For some reason, the code for escaping from a threat seems to be weighted towards moving up and left, which quickly results in a capture. However, due to both agent types having the same movement abilities, cornering a Mouse is often the *only* way the Cat can manage to capture them.

It's clear that the logic for dealing with a threat needs work. Currently, the agent attempts to find the nearest space that the threat can’t see and moves to it, but this just results in the threat moving to where it last remembers seeing the agent, which typically results in getting spotted again. Agents also have no follow-up after breaking line of sight – they’ll move onto the next task in their agenda after doing so, essentially forgetting the threat’s presence and moving back out of cover again.

A better solution would be to attempt to leave the threat’s overall range of vision entirely, but this would be difficult to do when the threat’s moving at the same pace as the agent. This is where communication between Mice could come into play – one Mouse could get close enough to the Cat to divert its attention away from the threatened Mouse, allowing it to get away safely.

This would require the ability to assess threat severity – if a Mouse has high energy, it has more time to run away from the Cat compared to one with low energy that risks dying of starvation as it attempts to evade the threat. It would also need to compare how far the Cat is from the other Mouse and determine if it can get close enough to distract its attention without immediately being captured.

I feel the Mice could also do with an ability to escape from the Cat in exchange for sacrificing some energy – say, for example, it can sacrifice 30% of its base energy level in order to stun the Cat for a few turns when it’s about to be captured. This would give the Mice greater ability to flee from the Cat, as well as giving them a benefit to stockpiling a high amount of energy. It would also increase the Mice’s ability to distract the Cat – allowing itself to be captured with high energy so it can then stun it and let it and its fellow agents flee.

Perhaps the Cat could also opt to expend energy at a greater rate in order to move faster, or to perform a “pounce” where it moves very quickly but only in a straight line.

# Conclusions & Future Work

Currently, the game is very heavily weighted in the favour of the Cat, and I believe that’s largely down to a lack of cooperation between the Mice. The Cat can just chase down one Mouse at a time, cornering them and easily killing them, and the other Mice will simply attempt to avoid the Cat’s vision and pick up food for themselves, ignoring their fellow Mouse’s plight.

Implementing the “abilities” mentioned previously would certainly help to improve the agents’ individual capabilities, but it would require the ability to perform teamwork to really shine. As it stands, there’s no teamwork at all between the Mouse agents – the first feature I’d implement would be to add a little communication between them.

When Mice are in vision of one another, they should be able to relay to one another the positions of items they know about, such as food pickups or where they last saw the Cat. This might require the implementation of a timer so they can determine who has the most up-to-date information. They should also be able to coordinate – the Mouse with the least energy should go for the closest pickup, while the Mouse with more energy should prioritise others. Being able to reference how old information is would also give them a better option once the grid is fully explored – rather than wandering at random, they could opt to revisit areas they haven’t checked in a while to look for changes.

I’d also like to alter the A\* pathing code to instead be LPA\*, as planned initially. Grids are generally on the smaller side for this program, primarily because extremely large grids (over 50x50) are too large to fit on my monitor, although this could be resolved by dynamically shrinking each individual tile’s size in pixels once a threshold is passed.

The movement code for both agent types could also be made more efficient. Currently, the Move to Position action always calculates a path using A\*, but that isn’t necessary if the agent can currently see the target in question – if it’s visible, that means no obstacles are present between it and the agent, meaning there’s no need for complex pathfinding. As such, the move action should first check if the target tile is in the agent’s list of currently visible tiles and, if so, it should move directly there instead of calculating a path. Adding direct movements would also make the program less reliant on A\*, which would reduce the number of times the pathfinding freezes up.

I’d also like to give the user the ability to place tiles for themselves, rather than relying on walls to be randomly placed. This could reuse the pause screen’s functionality of detecting which tile the user has clicked on. The program could prompt the user to click to place walls, then pickups, then Mice, then Cats.

Naturally, this would also require some actions that respond to a path being unreachable. The Mouse in Test Series 7 that started surrounded by walls on all sides had no way of reacting to its situation. This should be remedied, even if it’s only something like giving it a “give up” action for when literally no moves are possible.

# References

Conway, J., 1970. *Game of Life.*

CAPCOM CO., LTD, 1991. *Street Fighter II*. Osaka : Capcom Co., LTD

sf2platinum, 2017. *The AI Engine* [online]. Available from: <https://sf2platinum.wordpress.com/2017/01/20/the-ai-engine/> [Accessed 7th November 2022]

Jo, T., 2021. *Machine learning foundations : supervised, unsupervised, and advanced learning*. Cham: Springer

Sutton, R.; Barto, A., 1998. *Reinforcement Learning: An Introduction*. 1st ed. Cambridge: MIT Press

Baker, B., et al., 2019. *Emergent Tool Use From Multi-Agent Autocurricula*

Wooldridge, M., 2002. *An Introduction to Multiagent Systems*, 1st ed. New York : John Wiley and Sons, Inc.

Zarembo, Imants and Kodors, Sergejs, 2013. *Pathfinding Algorithm Efficiency Analysis in*

*2D Grid*

Patel, A., 2019. *Circle fill on a grid*. [online]. Available From: <https://www.redblobgames.com/grids/circle-drawing/> [Accessed 5th March 2023]

Lague, S., 2014. *A\* Pathfinding (E01: algorithm explanation).* [online video]. 14 December. Available from: <https://www.youtube.com/watch?v=-L-WgKMFuhE> [Accessed 7th April 2023]

# Appendices

## Detailed Project Proposal

|  |  |
| --- | --- |
| First Name: | Samuel |
| Last Name: | Burns |
| Student Number: | 1706269 |
| Supervisor: | Kit-Ying Hui |

### Project title

|  |
| --- |
| **Competing Cat and Mouse Game With AI Techniques** |

### Background

|  |
| --- |
| The concept behind this project is to pit two artificial intelligences against one another in a "cat and mouse" game, to observe how they adapt to one another's actions. These two entities - the cat and mouse - have opposing goals. Namely, the cat aims to catch the mouse, and the mouse aims to avoid the cat for as long as possible.  This will be represented by a 2D grid with walls spread throughout. These walls will block the line of sight of both entities. The mouse will be able to learn the cat's position within a certain radius by hearing it, even without line of sight. The cat won't be able to hear the mouse's movements, but when the mouse is within its line of sight it will move twice as quickly as the mouse can. When out of sight, it will return to its regular speed. When line of sight is broken, both entities will be able to remember their last seen position, and their AI would be able to inform their decisions based on that.  They will operate in "turns", where both entities move one tile. When the mouse is in line of sight, the cat will be able to move two tiles at once.  I'm interested in seeing how these entities will end up interacting, and what strategies will be developed. For instance, the mouse may stay near the cat to be constantly aware of its position and avoid it that way, it may work to avoid letting itself get cornered, or it may stick close to walls to break line of sight as soon as it can so the cat won't get its speed advantage. The cat should then be able to learn to counter these strategies, which will cause the mouse to change its behaviour, and so on.  I intend to keep a record of how long the mouse can last, as well as taking note of the general strategy of each entity, if one can be observed. During development, the attributes of each entity may need to be reevaluated if one has a clear advantage over another - e.g. the mouse being able to hear the cat, or the cat's bolstered speed when chasing. |

### Aim & Objectives

|  |
| --- |
| Aim: To create a system where two artificial intelligences competitively adapt to one another  Objective 1: Create tile-based environment with barriers  Objective 2: Define and implement behaviours for each entity  Objective 3: Create AI algorithms that inform the entities’ decision-making processes  Objective 4: Observe the entities as they adapt, recording statistics and making note of any emergent strategies |

### Tools & Technologies

|  |
| --- |
| I expect Python to be the primary tool for this project - it has a wide array of libraries that should be helpful for visualising the cat and mouse game, as well as for the AI side of the project. |

### Project Plan

|  |
| --- |
| October: Refinement of concept, evaluate project scope, research Python libraries  November: Create design for program - interface, etc.  December: Work towards creation of proof of concept  January: Present proof of concept, continue development based on feedback  February: Finalise development, run tests and observe behaviour  March: Begin writing report  April: Prepare demo and poster  May: Complete and submit final report |

## Project Poster

A picture containing text

Description automatically generated

## Project Log

Tile Test – 13/01/23

Creates a grid with randomly coloured tiles

Tile Test 2 – 16/01/23

Above, divided into functions

Agent Test – 18/01/23

Creates an agent that navigates randomly placed blocks, moving at random

Agent Test 2 – 30/01/23

Creates an agent that navigates randomly placed blocks, searching for items to pick up

Max Vision – 24/02/23

Agent vision range finder

LOS – 06/03/23

Line of Sight initial implementation

Pause – 20/03/23

Implementation of pause feature

Removal of above class

Ability to move towards targets, rather than simply moving at random

Implementation of subclasses with base agent.

Agenda – 01/04/23

Implementation of agenda and decision making code

Implementation of memory grids

Ability to move away from threats

Introduction of basic cat agent for testing (e.g. mouse move away from threat)

Agenda Gridrework – 04/04/23

Options implemented - move to position, explore, wander, evade

Implementation of A\* algorithm

Cat fully implemented

Death handling added

Grid reworked to better handle agents

Gridrework2 – 11/04/23

Grid reworked again to use objects to represent individual tiles instead of arrays

Task progress - started & completed trackers

"Evade" removed and replaced with "Hide" action - agent seeks nearest tile outside of threat's vision and moves there

Proper end-of-game report that considers every agent

Final – 18/04/23

Removal of several unused pieces of code

Better system for tile occupation

Tile click system for paused games

Properties list of easier editing of variables

## Text Description automatically generatedEthics Form

**STUDENT PROJECT ETHICAL REVIEW (SPER) FORM**

**The aim of the University’s *Research Ethics Policy* is to establish and promote good ethical practice in the conduct of academic research. The questionnaire is intended to enable researchers to undertake an initial self-assessment of ethical issues in their research. Ethical conduct is not primarily a matter of following fixed rules; it depends on researchers developing a considered, flexible and thoughtful practice.**

**The questionnaire aims to engage researchers discursively with the ethical dimensions of their work and potential ethical issues, and the main focus of any subsequent review is not to ‘approve’ or ‘disapprove’ of a project but to make sure that this process has taken place.**

The *Research Ethics Policy* is available at  [www.intranet.rgu.ac.uk/credo/staff/page.cfm?pge=706](http://www.intranet.rgu.ac.uk/credo/staff/page.cfm?pge=7060)0

|  |  |
| --- | --- |
| **Student Name** | Samuel Burns |
| **Supervisor** | Kit-Ying Hui |
| **Project Title** | Cat and Mouse Neural Network Learning |
| **Course of Study** | Application Software Development |
| **School/Department** | Computing |

|  |  |  |  |
| --- | --- | --- | --- |
| **Part 1 : Descriptive Questions** | | | |
| 1 | Does the research involve, or does information in the research relate to: | Yes | No |
|  | (a) individual human subjects |  | ✔ |
| (b) groups (e.g. families, communities, crowds) |  | ✔ |
| (c) organisations |  | ✔ |
| (d) animals? |  | ✔ |
| Please provide further details: | | |
|  | The project won’t be dealing with information relevant to people or things outside its own functions |  |  |
| 2 | Will the research deal with information which is private or confidential? | Yes | No |
|  |  | ✔ |
| Please provide further details: | | |
| All data processed by the project will be self-produced, meaning there’s no risk related to potentially sensitive information |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Part 2: The Impact of the Research | | | |
| 3 | In the process of doing the research, is there any potential for harm to be done to, or costs to be imposed on | Yes | No |
|  | (a) research participants? |  | ✔ |
| (b) research subjects? |  | ✔ |
| (c) you, as the researcher? |  | ✔ |
| (d) third parties? |  | ✔ |
| Please state what you believe are the implications of the research: | | |
| Because the project is focused on its own behaviour, it won’t have any negative impact on any of these groups. | | |
| 4 | When the research is complete, could negative consequences follow: | Yes | No |
|  | (a) for research subjects |  | ✔ |
| (b) or elsewhere? |  | ✔ |
| Please state what you believe are the consequences of the research: | | |
| This project is self-contained - it’s focused on the results of two AIs interacting and learning from one another, and all interactions will be entirely simulated | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Part 3: Ethical Procedures | | | |
| 5 | Does the research require informed consent or approval from: | Yes | No |
|  | (a) research participants? |  | ✔ |
| (b) research subjects |  | ✔ |
| (c) external bodies |  | ✔ |
| If you answered yes to any of the above, please explain your answer: | | |
| 6 | Are there reasons why research subjects may need safeguards or protection? | Yes | No |
|  |  | ✔ |
| If you answered yes to the above, please state the reasons and indicate the measures to be |  |  |
| 7 | Has PVG membership status been considered? |  |  |
|  | (a) PVG membership is not required. | ✔ |  |
| (b) PVG membership is required for working with children. |  | ✔ |
| (c) PVG membership is required for working with protected adults. |  | ✔ |
| (d) PVG membership is required for working with both children and protected |  | ✔ |
| If you answered yes to (b), (c) or (d) above, please give details: | | |
| 8 | Are specified procedures or safeguards required for recording, management, or storage of data? | Yes | No |
|  |  | ✔ |
| If you answered yes to the above, please outline the likely undertakings: | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Part 4: The Research Relationship | | | |
| 9 | Does the research require you to give or make undertakings to research participants or subjects about the use of data? | Yes | No |
|  |  | ✔ |
| If you answered yes to the above, please outline the likely undertakings: | | |
| 10 | Is the research likely to be affected by the relationship with a sponsor, funder or employer? | Yes | No |
|  |  | ✔ |
| If you answered yes to the above, please identify how the research may be affected: | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Part 5: Other Issues | | | |
| 11 | Are there any other ethical issues not covered by this form which you believe you should raise? | Yes | No |
|  |  | ✔ |
|  |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Statement by Student  I believe that the information I have given in this form is correct, and that I have addressed the ethical issues as fully as possible at this stage. | | | |
| Signature |  | Date | 28/09/2022 |

**If any ethical issues arise during the course of the research, students should complete a further Student Project Ethical Review (SPER) form.**

The *Research Ethics Policy* is available at  [www.intranet.rgu.ac.uk/credo/staff/page.cfm?pge=706](http://www.intranet.rgu.ac.uk/credo/staff/page.cfm?pge=7060)0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Part 6: To be completed by the supervisor | | | | |
| 12 | Does the research have potentially negative implications for the University? | | Yes | No |
|  |  | X |
| If you answered yes to the above, please explain your answer: | | | |
|  | | | |
| 13 | Are any potential conflicts of interest likely to arise in the course of the research? | | Yes | No  X |
|  | If you answered yes to the above, please identify the potential conflicts: | | | |
|  | | | |
| 14 | Are you satisfied that the student has engaged adequately with the ethical implications of the work? [In signifying agreement, supervisors are accepting part of the ethical responsibility for the project] | | Yes  X | No |
|  | If you answered no to the above, please identify the potential issues: | | | |
|  | | | |
| 15 | **Appraisal:** Please select one of the following | | | |
|  | The research project should proceed in its present form – no further action is required | | X | |
| The research project requires ethical approval by the School Ethics Review Panel | |  | |
| The research project needs to be returned to the student for modification prior to further action | |  | |
| The research project requires ethical review by an external body. If this applies please give details | |  | |
| Title of External Body providing ethical review |  | | |
| Address of External Body |  | | |
| Anticipated date when External Body may consider project |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Affirmation by Supervisor | | | |
| **I have read the student’s responses and have discussed ethical issues arising with the student. I can confirm that, to the best of my understanding, the information presented by the student is correct and appropriate to allow an informed judgement on whether further ethical approval is required.** | | | |
| **Signature** | K. Hui | **Date** | 29/09/2022 |

## Source Code

Available to view at <https://github.com/sjburns99/honoursproject>