**Can the Expected SARSA algorithm be used to create a dynamic and sustainable ecosystem for use in games?**

# Abstract

# Literature Review

## Dynamic AI in Games

## Reinforcement Learning Methods

## Policy Equations

## Predator Prey simulations

## Lotka-Volterra Equations

### Assumptions

# Tools and Methods

## Unreal Engine

### Blueprint

### C++ Classes

### Performance Profiling

### Artificial Intelligence tools

### Rama’s Victory Plugin

## Hardware Specifications

### Alienware 17R

### Harvey’s laptop

## Project Design

### Collision Detection

## Terrain

### Height Map

### Textures

## Agents

### Plants

The plants are modelled as a cube with a leaf texture applied to them. Plants do not inherit any AI behaviour and instead multiply periodically depending on external conditions.

#### Components

Plants are made up of a cube mesh object and a collision cube named “TouchCollision”. The collision component is used as a formality so that the prey detection functionality can be extended to the functionality of deer agents detecting plants.

### Variables

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** |
| State | Enumerator | The current state of the plant. This is based on overlap events. |
| Health | Integer | The current health value of the plant between 0 and 100. |
|  |  |  |
|  |  |  |

#### States

Plants have two states: MULTIPLY and BEING\_EATEN.

The current state of each plant has the potential to be modified when an overlap event is detected. If the

#### Functions

### Animals

Both Predator and prey inherit from the Animal\_AI class, as they share the same behaviour and functionality.

#### Components

Animals are made up of three components; the mesh component gives the animal shape and form as well as textures and material data, the touch collision sphere allows short distance detection of other object’s components, and the scent collision sphere allows long distance detection of other object’s components.

#### Variables

|  |  |  |
| --- | --- | --- |
| Variable Name | Data Type | Description |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

#### States

There are three sets of states for animal objects which describe their current position in relation to food, the current location in relation to predators, and the level of hunger.

Food States

Food states are based on the current collision status of the assigned type of prey objects with the touch and scent collision radii of the animal.

|  |  |  |  |
| --- | --- | --- | --- |
| Radius | Entering/Leaving | Agent type | State Change |
| Scent | Entering | Prey | NEAR\_FOOD |
| Scent | Leaving | Prey | Check if any other prey are colliding. If true, then NEAR\_FOOD. If false, then NO\_FOOD. |
| Touch | Entering | Prey | EATING\_FOOD |
| Touch | Leaving | Prey | Check if any other prey are colliding, If true, then EATING\_FOOD. If false, then NEAR\_FOOD. |

Predator states

Predator states are based on the current collision status of the assigned type of predator objectrs with the touch and scent collision redii of the animal agent.

|  |  |  |
| --- | --- | --- |
| Radius | Entering/Leaving | State Change |
| Scent | Entering | NEAR\_PRED |
| Scent | Leaving | Check if any other predators are colliding. If true, then NEAR\_PRED. If false, then NO\_PRED |
| Touch | Entering | BEING EATEN |
| Touch | Leaving | Check if any other predators are colliding. If true, then BEING\_EATEN. If false, then NEAR\_PRED. |

#### Animation

## States and Actions

## Rewards

## Boltzmann Softmax Equations

### Temperature Constant

The temperature constant, otherwise known as Tau, governs the weighting of the probabilities that each state may be selected. A tau value that tends to zero means that the action selection will be greedy (i.e. the action with the best Q value will be selected over the other possible actions). A tau value between 0 and 1 means that there will be a difference between the probabilities, based on their Q values and the value of tau. A tau value of 1 means that all action selections are equiprobable.

In this project, there will be two variations of tau; Greedy and mid-range (tau = 0.6).

## Expected SARSA

### Learning Rate

### Reward Discount

# Testing

The following tests provide insight on the performance and sustainability of the overall simulation. There are a number of control variables which will be used to initialise the simulation as it is executed. In order to receive a wide variation of results, each control variable has been iterated so that all combinations of these variables are used in each new test.

Before the core testing can proceed, a short pre-test phase has been implemented to perform basic checks on the functioning of the system. This will bring to surface any potential issues such as overlooked bugs so that the core test can be run smoothly.

The testing is divided into two categories: Performance testing and sustainability testing. This is due to a large data set being extracted from the system during the sustainability testing phase, which may affect the performance of the system. This would mean performance results that are not representative of the actual system speed that the system would run on.

## Control Variables

## Pre-Test Run

The pre-test run is designed to bring any potential issues to the surface before the core testing can begin. The simulation is run for test case 000 (see appendix …) for each of the three map sizes.

## Performance Testing

The performance testing will be run on two laptops capable of executing games at differing levels of performance.

## Sustainability Testing

This test will be run on the Alienware 17R laptop only, since the processing speed will ensure that the testing can run as smoothly as possible with a low risk of failing due to performance issues.

# Results and Analysis

## Performance Results

### Analysis

## Sustainability Results

### Analysis

# Conclusion

# Further Work

# Appendices

## Asset Licenses

## Class Diagram

## Enumerators

## Reward Data

## Blueprint Classes

## Text File Manager Class

## Test Data