# Table of Contents



Figure 1 – Sins of a Solar Empire, a popular scifi RT4x

Assessment 03

AI Project Report

Miles Whiticker

Gam302

[**Table of Contents** 0](#_Toc323227148)

[**AI** 2](#_Toc323227149)

[Overview 2](#_Toc323227150)

[Setting 2](#_Toc323227151)

[Agents 2](#_Toc323227152)

[Bayesian Networks and Fuzzy Logic 3](#_Toc323227153)

[Genetic Algorithms and Agent Spawning 3](#_Toc323227154)

[**Simulator** 4](#_Toc323227155)

[Ship Types 4](#_Toc323227156)

[Maintenance 4](#_Toc323227157)

[Spawning / despawning 4](#_Toc323227158)

[Trader characteristics 4](#_Toc323227159)

[Trader decisions 5](#_Toc323227160)

[Trader Companies 6](#_Toc323227161)

[Resource prices 6](#_Toc323227162)

[**Application** 6](#_Toc323227163)

[Technology 6](#_Toc323227164)

[Interface 7](#_Toc323227165)



Figure 2 - Aian starship

# AI

## Overview

This report outlines the system, mechanics and associated AI behaviour of the project, as well as relevant probability formulae used to determine actions by the simulation’s agents.

The project is an economic simulator set in a fictional, futuristic sci-fi environment that employs realistic virtual traders to interact with the world through trade. The aim of the project is to explore the emergent behaviour that arises from the various agent’s interactions with the world and each other.

The agents employ fuzzy logic to determine individual characteristics which are combined with Bayesian probability and environmental factors to make decisions. Genetic algorithms are employed on new generations of traders to slowly mutate traders over time to be more successful – successive generations are modelled after the richest currently active traders. [[1]](#footnote-1)

## Setting

The simulation is set on a galactic scale – there are multiple star systems with numerous planets each, which have different properties and resources, and some of which will be colonised. Each colonised planet produces certain resources, which are employed for various different purposes.

Colonies have a required upkeep of resources. If enough of the right resources are available, they are consumed and the colony grows a little bit. Larger colonies produce more materials, but also consume more materials. If enough materials are not provided, the colony can decay. Once population gets below a certain point, the colony despawns.[[2]](#footnote-2)

## Agents

Traders will move between systems, buying goods where they are cheap and selling them elsewhere for a profit. Traders can be in one of three ships, which vary in speed, range [maximum stored fuel] and carry capacity. All traders act entirely independently from each other.

Depending on trader characteristics, they may join together to form companies (pooling resources and ships) or start their own, buying additional ships to complement their current one.

If a trader runs out of money, they will go bankrupt and despawn to live out their life as a colony worker. Periodically, new traders will spawn at random shipyards with a corvette – their characteristics are copied from the wealthiest active trader with a random variation.

Colonies will dynamically change resource prices, in order to attract the resources that they need to survive and thrive.

Once colonies reach certain industrial conditions, they may also start producing colonists who are ferried to empty planets to start new colonies.

## Bayesian Networks and Fuzzy Logic

Bayesian Networks integrate Bayes Formula with a directed acyclic graph in order to determine the probable outcomes of a chained, interdependent series of events. Bayes Formula is employed here to determine outcome probability given multiple variable effectors, primarily involving one of the trader’s characteristics and some combination of one or more environmental variables. The basic formula is represented by:

Or, the probability of **∂** being true [given that **µ** is true] equals the probability of **µ** being true [given that **∂** is true] multiplied by the probability of **∂** being true, all divided by the probability of **µ** being true.

Where **∂** is a node with random probability and **µ** is [initially] assumed true / false data, which is substituted for its individual probability.

In the event that chained conditional nodes exist and must be taken into account, additional complexities can be introduced. For example, consider cases where **µ** has an additional conditional dependency **σ**, and where **∂** has an additional conditional dependency **σ**.

Case one: Pr() \* Pr() \* Pr()

Or, the probability of **∂** given **σ** [given **µ**] equals the probability of **µ** [given **∂**] multiplied by **∂** [given **σ**] multiplied by the probability of **σ**.

Case two:

Or, the probability of **∂** [where **µ** and **σ** are given] equals the probability of **∂** [given **µ**] multiplied by the probability of sigma [where **∂** and **µ** are given], all divided by the probability of **σ** [given **µ**].

Case two can be simplified further in order to present practical applications, but as both cases are not implemented in the simulator this is beyond the scope of the project.

In the simulation, the implementation of Bayes formula will likely not exceed the given prototype formula in complexity, and the various probabilities are given by the trader’s characteristics and environmental conditions.[[3]](#footnote-3)

## Genetic Algorithms and Agent Spawning

Genetic algorithms model the evolution of a set of data as it iterates over time, which the changes from one generation to the next being modelled off real-world biological evolution. Genetic algorithms follow four steps:

1. Initialisation, where the starting population is created.
2. Selection, where a certain part of the population is chosen to propagate the next generation.
3. Reproduction, where the previously selected population segment creates the next generation through genetic operators such as crossover and mutation.[[4]](#footnote-4)
4. Termination, repeat steps 2 and 3 until predefined ending conditions have been fulfilled.

When the simulation starts, a random number of traders with randomly determined characteristics[[5]](#footnote-5) are created at worlds with shipyards. At random intervals (between half a year and a year and a half), a new agent will spawn, with characteristics taken from the wealthiest active trader and then randomly modified with a variance of ±15%.

# Simulator

## Ship Types

* Corvette [fast speed, low cargo capacity, short fuel range, light maintenance]
* Frigate [medium speed, average cargo capacity, medium fuel range, moderate maintenance]
* Bulk liner [slow speed, high cargo capacity, long fuel range, high maintenance]

## Maintenance

* Maintenance decays over time.
* A lower maintenance level decreases speed and cargo capacity and increases fuel usage (both by a percentage modifier).
* Maintenance consumes food, plates and components at shipyards depending on the amount of maintenance performed [trader pays, shipyard provides resources].

## Spawning / despawning

* If the average wealth of traders is rising, new traders will spawn. Traders can run out of money and go bankrupt, which makes the market easier for all other traders.
* New traders spawn with a small amount of money and a corvette at a random shipyard with enough resources.
* If a trader runs too low on money, they will downsize (sell cargo prematurely / trade in ship for a cheaper one / sell a ship).
* Once a trader runs out of money and can’t downsize any more, they will go bankrupt [despawn – their last ship will disappear].

## Trader characteristics

These are ranked on a scale of 0 < x < 1 and determines a trader’s probable choice towards certain binary choices.

* Restlessness ; a decrease means more likely to wait after a trade.
* Decisiveness; a decrease means more likely to change jobs halfway through.
* Caution; an increase means trader will repair more frequently.
* Pride; an increase means a higher profit threshold is required for jobs.
* Individuality; an increase means less likely to join a conglomerate / start a company.
* Ambitious; an increase means will be more likely to start companies.
* Risk taker; an increase means more distant jobs will be considered.

## Trader decisions

When a trader has no currently active goal, they must first decide to do something according to Pr(restlessness), then they will go through the following three in order of priority.

First, [if the trader has a less than full tank] they will decide whether and where to purchase fuel.

1. Get a list of all places selling fuel within range, and sort them according to price, ascending.
2. Starting with the cheapest place, loop through checking each destination until there are none left or the probability check returns true.

* Pr(need for fuel | acceptable distance)
  + Or, how much fuel is needed as compared to how far the trader is willing to go to buy it.
  + Need for fuel = Max( 1 – (amount of fuel / fuel max) + 0.1, 1).
  + Acceptable distance = risk taker + ( (distance to current – distance to furthest) / (distance to closest – distance to furthest)) \* (1 – risk taker).

1. Travel to destination and purchase fuel.

Second, [if the trader’s ship is damaged] the trader must decide whether to repair their ship at a nearby shipyard.

* Pr(need for repairs | caution)
  + Or, how much repairs are needed given that the trader is cautious enough to want repairs.
  + Need for repairs = Max ( 1 – (ship ‘health’ / ship max ‘health’), 1 )

Thirdly, the trader must decide whether and where the next trade destination will be.

1. Get list of all possible jobs in range and sort them by profit. If jobs are not profitable, exclude them.
2. Starting with the highest profit, determine whether to choose the current destination according to:
   1. Pr(acceptable profit | acceptable distance)
      1. Or, the probability of an acceptably high profit, given that the distance is within the acceptable radius.
      2. The acceptable distance is determined thusly: ‘risk taker’ stat gives the probability of choosing the furthest potential job, which scales up to 100% at the nearest potential job.
      3. The acceptable profit is determined thusly: anything equal to or greater than the agent’s average trade profit is 100%, which scales down to (1 – pride) when the trade breaks even.
   2. Loop through potential jobs (starting with the most profitable). If the check for the current potential job fails, then it is tried again with the next highest profit margin and so on.

Once a trader has arrived at a destination to purchase goods, there is a chance they will change their mind about the current job.

1. Get a list of all jobs in range with a higher profit than the current one, and sort them by profit descending.
2. Starting with the highest profit job, loop through all and if a check returns true then have it replace the current one.

* Pr(acceptable profit | acceptable distance, 1 - decisiveness)
  + Or, the probability of an acceptably high profit, given that the distance is within the acceptable radius and the trader is indecisive.

## Trader Companies

Companies are groups of traders pooling finances. There are no changes to decision making, but with pooled finances traders are able to upgrade faster and are less likely to go bankrupt.

Periodically (about once a year) a trader will attempt to either form a company [if they have enough money], or join an existing one:

1. Check once to see if we want to start a company, according to:

* Pr(ambition | financial wellbeing)
  + Or, the general financial performance compared to other traders, if the trader is sociable enough to work with others.
  + Financial wellbeing = Max( Avg(average trader profit / global trader profit, average trader wealth / global trader wealth), 1 ).

1. If we haven’t started a company and our profit is above the global average, loop through all companies in order of Avg(relative global wealth, relative global profit) [descending] checking to join each with:
2. Pr(financial wellbeing of target company | 1 - individuality)
   * Or, how well off the company is, given that the trader doesn’t mind working with others.
   * Financial wellbeing of target company is relative to other companies; the wealthiest company is at 100%, the least wealthy company is at 50%, everything scales between them according to how many companies there are. Wealth is determined by Avg(relative global wealth, relative global profit).

## Resource prices

These dynamically alter according to stored resource quantity and demand at the colony. More developed colonies can produce more raw materials, and turn more raw materials into finished goods per update. [[6]](#footnote-6)

# Application

## Technology

The simulator will be written in C++ using Haaf’s Game Engine to display all graphical and textual elements, and employing HGE’s inbuilt GUI system to function as the player interface.

## Interface

The screen is split into four sections – top bar, side bar, bottom bar and main view. The top bar contains options to enter the main menu, view the date, and change the rate of passage of time. The side bar contains list data, including all traders, settled planets and companies. The bottom bar contains a detailed view of individual traders, planets and companies.[[7]](#footnote-7)



Figure 3 - GUI mockup

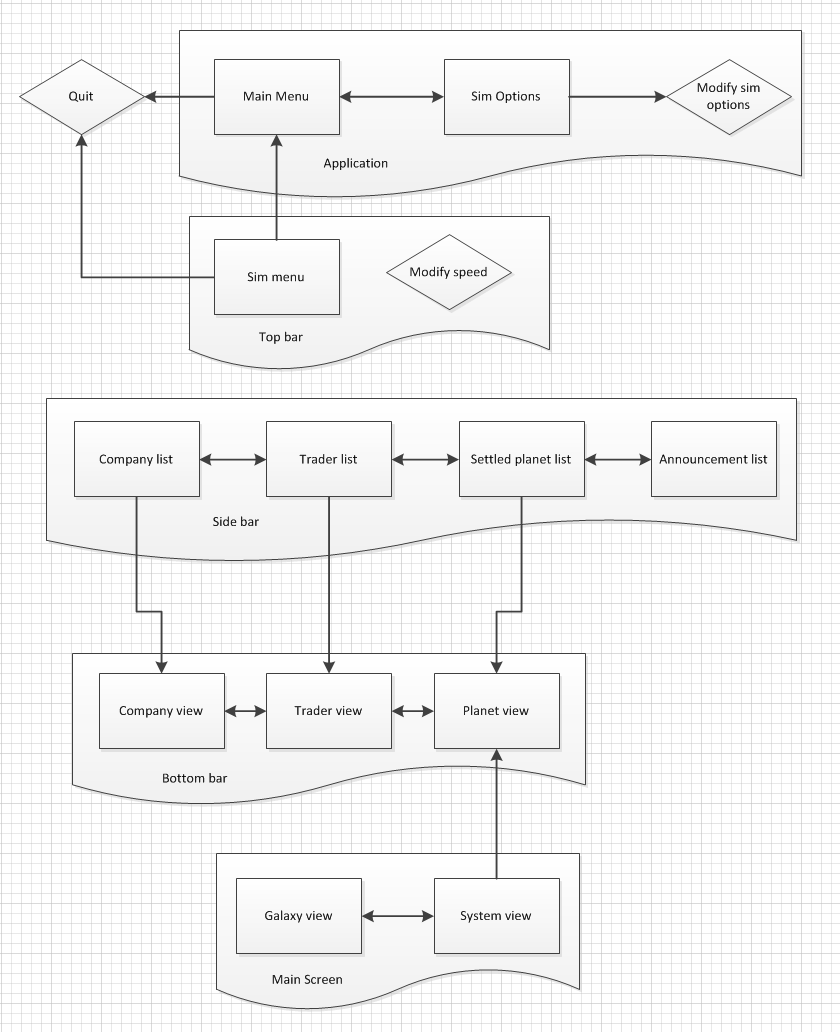


Figure 4 - GUI flow diagram

1. See ‘Trader Characteristics’ (p.4) for a detailed breakdown. [↑](#footnote-ref-1)
2. Additional documentation detailing the algorithms that control colony development, decay and resource consumption is available on request. [↑](#footnote-ref-2)
3. See ‘Trader Decisions’ (p.4) for an algorithmic breakdown of individual calculations. [↑](#footnote-ref-3)
4. Crossover [recombination]: combining aspects from one or more parents; mutation: randomly varying data. [↑](#footnote-ref-4)
5. See ‘Trader Characteristics’ (p.4) for a detailed breakdown. [↑](#footnote-ref-5)
6. Additional documentation detailing the algorithms that control dynamic resource pricing is available on request. [↑](#footnote-ref-6)
7. See figures 3 and 4. [↑](#footnote-ref-7)