Functional Specification Document - Game Theory: Modelling a Simple Society

Jamie Hyland - 17108152

Alex Thornberry - 15516603

Date Completed - 22/11/2019

1.1 Overview

Provides a brief overview of the system / product to be developed. It should include a description of the need for the system, briefly describe its functions and explain how it will work with other systems (if appropriate).

The project to be developed is a 3d generated simulation which will portray game theory. The title of the project is 'Game Theory: Modelling a simple society". It will simulate an ecosystem consisting of at least two species of animals (a predator and a prey), a food source for the prey and a water source for both animals. There will be many factors which will directly influence behaviour. These factors may consist of different attributes like thirst, hunger, gestation period and fatigue which then depend on different individual attributes like age, speed, selfishness, fitness and size.

We would like to start by modelling the landscape to cement the basis for the simulation. Upon full development of the landscape we will then begin releasing animals into the landscape and analysing their behaviour. From there we can create multiple scenes in which we can alter many factors. We will alter the amount of animals/plants, the strength of attributes, the availability of water sources etc..

At a base level the system is being built to observe the kind of situations that need to be in place for elements of cooperation or uncooperation to occur in a society, will an agent act in its own self interest? Or will it act in a way that benefits the long term survival of its species as a whole and what elements in a society force an agent to grow and adapt to its surroundings..There will be two (*minimum with the hope to add more*) species inhabiting our society, one which finds its food from the environment itself (grass, fruit, etc..) and one which relies upon the other species of animal as its food source. We hope to use this as a driving force for cooperative and uncooperative behavior, as the more docile species needs to gather its food, it may share food with other animals of the same species. As its being hunted by another species the more docile species may try to signal to the group as a whole about the presence of a nearby predator and incorporate a sort of flocking behavior. As the predator must source food that actively tries to elude it, it might inherently be more selfish, unwilling to share its food source with other agents. However, as the predator species starts reproducing it may inhibit signs of cooperation opting to share food with offspring or protecting its offspring from other agents of different species.

1.1 Business Context

As this system is a research focused application, it does not have many real business applications, however, doing research on this topic we found that most articles based on this topic rarely had a visual component. We feel having a visual system that evolves over time that can be tweaked by the user not only poses interesting research questions, which only increases the systems long term viability and its ability to be tweaked to different kinds of research in the same field.

2. General Description

2.1 Product / System Functions

Describes the general functionality of the system / product.

The system is a visualization of simple society that evolves over time. Agents in the society will have different traits, which govern how those agents act. Some agents will be more cooperative, favoring the overall success of the group with other agents being more selfish favoring quick short term gains as opposed to long term success. The idea of this system is to watch these agents evolve and record our findings. With this data we will analyze what worked and what failed. This system is intended to give researchers a way to view Game theory ideas and Algorithms in a 3D space with the ability to view the data of the society in real time with the ability to output that data to a file for further research and study. Researchers will also have the ability to interact with the system, tweaking different aspects of the world to watch how the agents react.

2.2 User Characteristics and Objectives

Users should be familiar with the glossary of terms associated with game theory. Basic knowledge of statistics and understanding the various graphs and how to read them properly. The system should give users a useful visualisation of how different game theory algorithms work. This system is intended to be aimed towards researchers in the field of Game Theory, as such, user’s might prefer to have a live view of the data being gathered in real time. Some researchers might want to only see the data in graph form updating in real time as opposed to the 3D simulation. Other researchers might want to hone in on a specific agent in the world and follow it to study its outcome.

2.3 Operational Scenarios

Upon start up of the system users will be given a splash screen giving a brief description of the project, and goal it is trying to achieve. Once a user progresses past this splash screen, they are treated to the visualization of the society, users with mouse or keyboard can zoom in and out with a camera and observe the society from different views. Users will be able to hover over individual agents and see a popup of real time stats of that individual agent. Users will also have access to a menu which will allow the user to speed of the speed of the society.

User begins simulation

- Current System State: Splash screen greets the user outlining the purpose of the project and loads simulation
- Informal Scenario: User has launched the simulation and is waiting for it to begin.
- Next Scenario: User observes simulation

User observes simulation

- Current System State: User is viewing ongoing simulation
- Informal Scenario: User is viewing simulation and is given various camera options
- Next Scenario: User selects alternate camera

User selects alternate camera (Agent Lock):

- Current System State: The user is looking at the simulation in any of the previous camera angles
- Informal Scenario: The user then clicks on an individual agent locking the camera in the view of that agent
- Next Scenario: The user is able to stop focusing on said individual agent

User selects alternate camera (Isometric view):

- Current System State: User is viewing the simulation
- Informal Scenario: User presses a button to change the camera to isometric
- Next Scenario: User is viewing the simulation in an isometric view

User speeds up simulation:

- Current System State: User is viewing the simulation and given the option to speed up simulation
- Informal Scenario: The user presses the button to speed up the simulation
- Next Scenario: User wants to return simulation to regular speed

User views live simulation statistics:

- Current System State: User is viewing the simulation and given the option to view the live statistics being recorded
- Informal Scenario: The user presses the button to view the live statistics
- Next Scenario: The user is taken to the live statistics screen

2.4 Constraints

One of the team members is limited to a machine without a dedicated graphics card.

As the system is heavily research leaning a large focus on data collection is required. The system will be full of dozens of agents all moving and acting independently of one another, each frame data will have to be collected on each individual agents state, things like its hunger, thirst, when it reproduces, when it dies will all need to be recorded. The aim for the project is to run a visualization at a max of 60fps and at a lowest of 30fps. Constantly recording data on a per frame basis can lead to heavy performance dips.

3. Functional Requirements

- **Description -** The system must run at a minimum of 30fps. The system should be able to handle many instances of autonomous agents on screen acting independently of one another while recording data on the state of each agent on screen without any dropped frames.
- **Criticality -** Having a stable frame rate for our visualization is critical for our system as the system will be collecting data on a *per frame* basis. Any dropped frames could lead to inaccurate data which will hamper our progress.

- **Technical issues -** Properly handling the many processes running at once in the system will be the key technical issue with this requirement.

- **Description -** The system must contain at *least* two types of agents, cooperative and non cooperative. These agents must be able to adapt to their surroundings and evolve over generations. They must also be able to have some sort of interactions with each other (They must be aware they both exist in the world and act accordingly).
- **Criticality -** This piece of functionality is crucial to the success of our project. Especially the interaction between species, this is the backbone of our project and as such is our top priority.
- **Technical issues -** The key technical issues involved with this requirement is handling the interaction between the species.

- **Description -** The system must be able to be altered by the user in some way (such as limiting available food, decreasing agents radius of search etc) to watch how the agents respond to the change in the world
- **Criticality -** This requirement is critical to the long term success of the system as the ability to affect different aspects of the world allows research to conduct research in finer detail
- **Technical issues -** The key technical issues involved with this requirement is handling the interaction between the species.

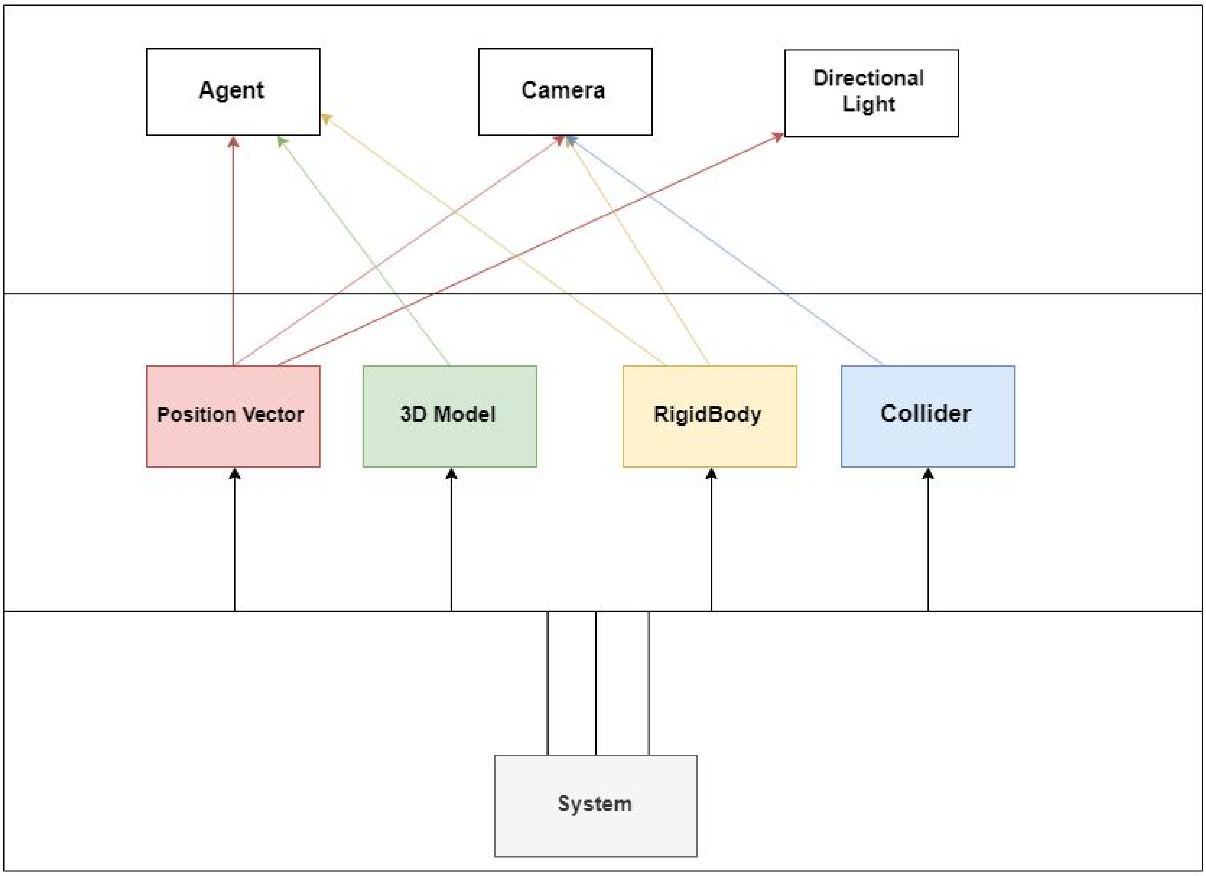
- **Description -** The system must be able to gather data on the different species attributes. Things like hunger level, fatigue level, reproduction must all be recorded over time and preferably output to a JSON file or CSV file for further analysis.
- **Criticality -** This requirement is critical to the further development of the system during its development. As this project is heavily research based being able to analyze the data we're gathering and adjust our algorithms on the fly is key to the long term success of our project.
- **Technical issues -** Constantly recording data and outputting to a file could be heavily cpu intensive. A lot of consideration will be put into how appropriately manage the gathering of data to not incur heavy performance dips.

- **Description -** The system should allow for users to pan, zoom, and rotate the camera around the world space.
- **Criticality -** This requirement although important is not key to the success of the project as a whole.
- **Technical issues -** Careful attention must be thought about the amount of camera freedom we give users, this would be less of a technical issue but a UX issue. Full

range of movement could allow players to get stuck in world geometry and could pan the camera out to a level which completely removes the world from view.

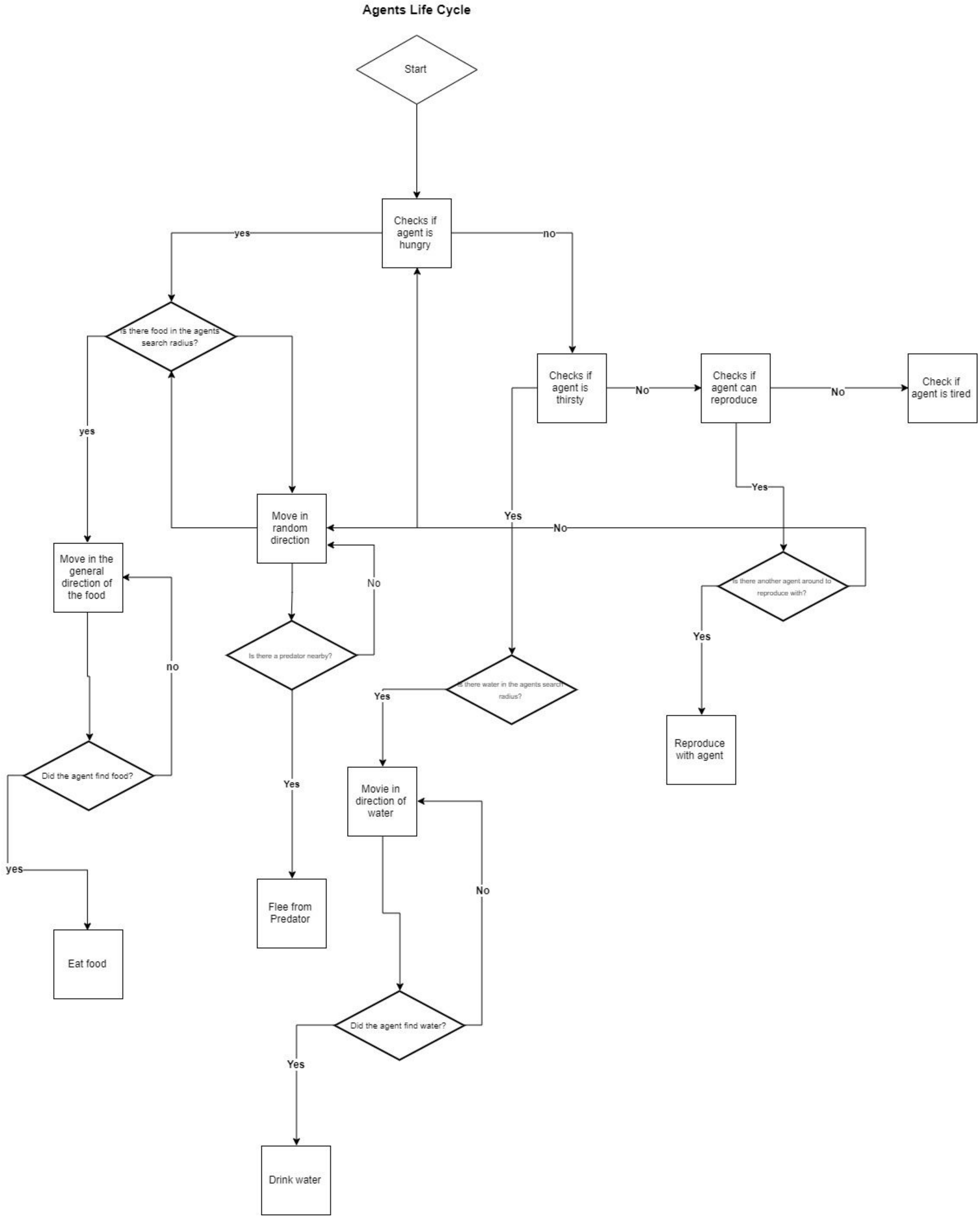
4. System Architecture

As our system is built using the Unity Game engine it will use an Entity Component System. This differs from typical OOP hierarchies as every object in the system is a **GameObject.** Every GameObject in the world is an **Entity,** every entity is made up of multiple **Components.** A Component is a simple piece of data (A collider, a position vector, Sprite, 3D model et cetera…). They are similar to structs in C/C++ as they are only capable of storing data without being able to act on it. The last piece in the systems architecture is the **System** itself. The system acts on Components in a logical fashion, checking for key presses, then checking the collider for collisions then updating the GameObjects position vector if necessary.



5. High-Level Design

As this system is a visualization with very little user input this section will describe the basic structure of an agent's life cycle in the visualization.



6. Preliminary Schedule