# Complex Game Systems – Design

Fuzzy Logic State Machine Unity Proposal

By Jacob Cooper

# System

The fuzzy state machine is an extension to standard finite-state machines where it uses fuzzy logic to determine behaviours. Instead of processing crisp values, both the inputs and outputs are fuzzy values. This is advantageous for AI, as its intelligence comes from the feeling of the ai making choices based on the graphical formulars used to determine the desired desire level.

The goal is to implement a modular Fuzzy State Machine into unity where the user can create and use custom states in order to manipulate and give the outcome of the AI a more realistic taste.

* Look into the likelihood/confidence formular for Fuzzy Logic

# Mathematical Operations

The mathematical operations involved in this system include:

* Converting a crisp set to a fuzzy set.
* Calculating a variables current desirability.
* Using the desirability in logic to output the correct state.

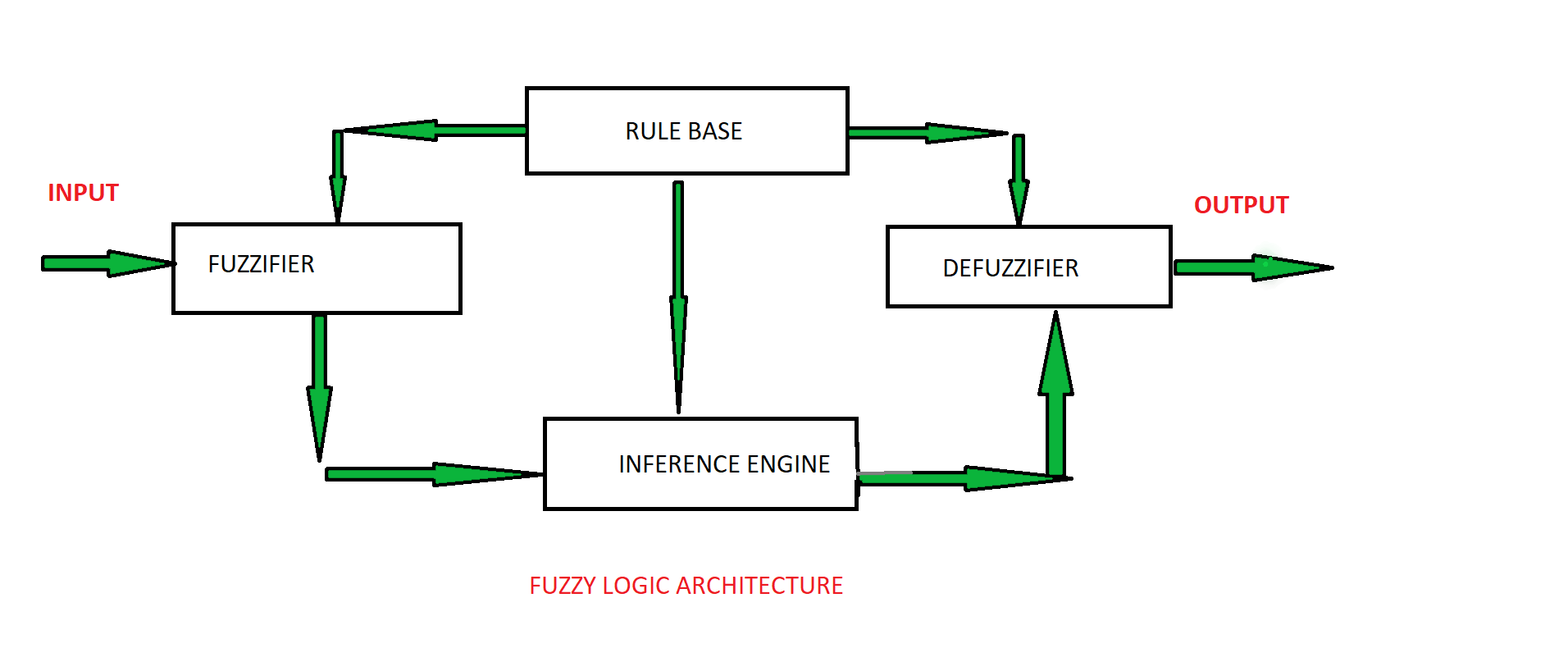
Logical Operators:

* (x and y) = min(x, y)
* (x or y) = max(x, y)
* Not x = 1 – x

# Advanced Algorithms

A fuzzy state machine algorithm will have to be included which calculates fuzziness and states all under a set of rules and then outputs the correct desirable state to the AI to execute.

* Provided an in-depth explanation on how crisp data becomes fuzzy and how to undo the fuzzy data to make choices



# Modularity

To make my system modular, I will be creating a Fuzzy State Machine object which can be edited using a custom node graph editor which will then be implemented onto objects.

The user will be able to inherit from two node classes to manipulate the graph editor/AI which include:

* FuzzyLogic – Create your own fuzzy logic script which allows manipulation of states and variables inputted into the script.
* FuzzyState – Fuzzy state which will be manipulated by the fuzzy logic script, which then will be passed off to an AI script which will execute this state during runtime.

Neither of these classes will inherit mono behaviour as their executions exist solely through an AI script.

The graph editor will give the user the ability to create states, a fuzzy logic calculator, and variable inputs which will all output necessary information which can be accessed later by another script which uses this component. Allowing for a customizable fuzzy state machine which can be implemented anywhere and used by any AI.

The graph editor is created using *Graph View* a unity experimental backend library which assisted in the development of the animator graph editor. The editor will be opened up through Windows/Fuzzy State Machine Editor or by simply double clicking the *.fsm* (Fuzzy State Machine) object.

The graphs object data will be mounted onto a loader component which can be accessed by getting the fuzzy state machine loader component on the game object,where you can grab/manipulate fuzzy data, in order to simulate fuzzy logic on an AI.

Hopefully later in development if there is time left, there will be a Fuzzy Logic Graph visualiser which will during runtime output a desirability graph chart in the component inspector to visualise the next most likely fuzzy logic state outcome.

This system will be exported into a unity package and can be loaded in any unity application by simply importing it.

# Integration

Integration of my system will be intricate, and will include:

**Importing**. To import the *“Fuzzy State Machine.unitypackage”*, you can double click the package or in unity, go to Assets/Import Package and import the package.

Machine Object, open it in the Fuzzy State Machine Editor, create your states and assign variables and actions to them, then assign it to the AI object you wish it to run with and then you are done.

Also included is the ability to create your own State Machine nodes with code.

Instructions included in a readme and a demo inside of the project to help the user.

* create the Fuzzy State Machine Object, How?
* open it in the Fuzzy State Machine Editor, How?
* create your states, How? Also can you reuse states?
* assign variables and actions to them, How?
* then assign it to the AI object you wish it to run with. Much more detail required, mostly how this will work
* Also included is the ability to create your own State Machine nodes with code. , How?

# Additional Libraries

The project will be using the Unity Version 2020.1.10f. It will require the use of the experimental backend library of Graph View which is already included in unity which will be used to create the fuzzy state machine node graph.