Fuzzy Logic

By Karl Boghossian

N.B: The FuzzyLogic engine is located in "FuzzyLogic.h", it is a common engine used for all the projects provided in the folder

In the "FuzzySet" struct:

- The fuzzy set type that can have any of the four set types
- Four floats to represent the set's boundaries
- The constant that belongs to this set
- The Degree of membership of a value with this set (used for the mamdani)
- The Masked value for this set that will limit the set with a certain value to obtain the centroid.

- Create_FuzzySet is a function that creates the set by setting its type, constant value and the boundaries
- DegreeOfMembership takes the current value and the degree of membership that it must fill if the value belongs to the set. First it checks for the type of the set and according to the set type, it does the checks. It returns true if there is a degree of membership for this set.

In the "FuzzyRule" struct:

- Contains the fuzzy sets of this fuzzy rule.
- Sets_Count to know the number of sets in this fuzzy rule.
- Two indices to know the two sets contributing to the current value.
- Two degree of membership of the two sets contributing to the current value.

- Create_FuzzyRule is a function that allocates storage for the sets of this fuzzy rule.

- FuzzyRule destructor that deletes the allocated space of the sets
- AllSets_DegreeOfMembership takes a current value, loops through all the sets, checks if the current set contributes for this value, if yes we store the degree of membership in the first variable else we store in the second variable, else we just reset the member in the current fuzzy set to 0
- Compute_Centroid is a function that loops through all the sets, takes the left lower bound of the current set with the left lower bound of the next set in order to have a range. Then using this range it computes the centroid of the rule by computing the denomination along with the numerator then dividing the nume/deno, finally normalizing the centroid.

<u>In the "FuzzyLogic" struct:</u>

This is the class that is the base class for any fuzzylogic problem.
- It contains the array of fuzzy rules.
- The number of fuzzy rules
- The type of logic operator that can an OR, AND or NOT.
- The constructor takes the number of rules to allocate space for them.
- Fuzzy logic destructor that deletes the memory allocated for the rules.
- The fuzzy control function that takes the rule index to work on, the degree of membership of the rule along with the index of the set to take the constant from it. It defuzzifies the value.
- Get_FuzzyFinalDegreeOfMembership takes the rule index to work on and the logic operator to execute. It checks if we have the two degrees of membership set so we must do an AND or an OR and pick the right fuzzy value. Else we can have only one degree of membership so we execute a NOT operation if required.
- Then we defuzzify the value and return it.
- AND takes two values and does the minimum between them.
- OR takes two values and does the maximum between them.
- NOT takes one value and returns its opposite (1-value).

I will talk about the camera fuzzy logic (FuzzyCamera.h):

Here because we want to solve a fuzzy logic problem, we create a class and derive it from the FuzzyLogic base class, so we have all the properties of the engine class.

- In the constructor of the derived class (Camera_Fuzzy here) we set the number of rules that we have in the constructor, this number of rules will be sent to the fuzzy logic constructor to allocate the rules. Then we create all the sets in each rule.
- Get_ObjectAngle takes the camera direction along with the camera position and takes the target pos and direction. Gets the angle between the camera direction and target direction in order to align them using the fuzzy logic, and computes the distance between the camera and the target.
- GetAdjustment_AngleSpeed takes the source and target properties and fills the newangleadjustment with the newspeedadjustment. It gets the angle and distance between the camera and the target and checks all the sets contributions with these two values. Does the fuzzy logic operations, computes the follow percentage using the memdani centroid method and returns a percentage to be used from the new adjusting values.
- These values are then used in the Test.cpp to update the camera position and angle.