Fuzzy Logic & Knowledge Based Systems - Assessment Report

Introduction -

This assessment focuses on the development of a Fuzzy Inference System, the system will have a minimum of 3 input variables and 1 output variable.
This form is dedicated to the study of the fuzzy logic of “Game difficulty”, specifically concurring the overall difficulty based on the adjustable features that alter the gameplay experience.
The physical system that maps this will be implemented on MATLAB, I’ll include screenshots and text descriptions to illustrate my findings and conclusions.

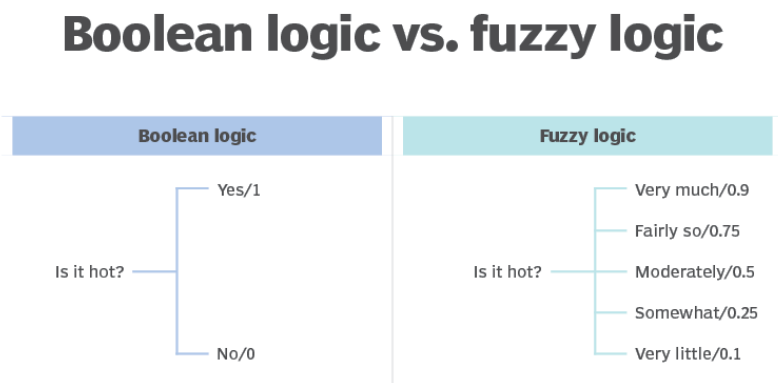
I’ve decided to present “Game difficulty” as it's not an entirely definitive system, mostly settled case by case with individual developers (think Dark Souls / Elden Ring compared to Minecraft), there's a significant degree of fuzzy logic in quantifying the level of difficulty.
I’ll focus my research on the difficulty system of a typical survival horror game, evidentially having several factors that contribute differently to the impact of gameplay.
I’ll research and evaluate these aspects and how overall they dictate the difficulty of a game, mainly by understanding how modifying these settings can produce differing difficulty levels.

I’ll then produce a system that will determine the difficulty level, consider the numerous factors (Player mechanics, enemy aggression, resource availability etc), quantify their variables and align them by the rules to produce individual ranks / levels, these will dictate how the game will proceed and impose a comprehensive system that ranks the skill of players by the level of challenge.

Literature Review -

Fuzzy logic / Fuzzy logic for Game difficulty -

Paper 1 - What is Fuzzy Logic [1]



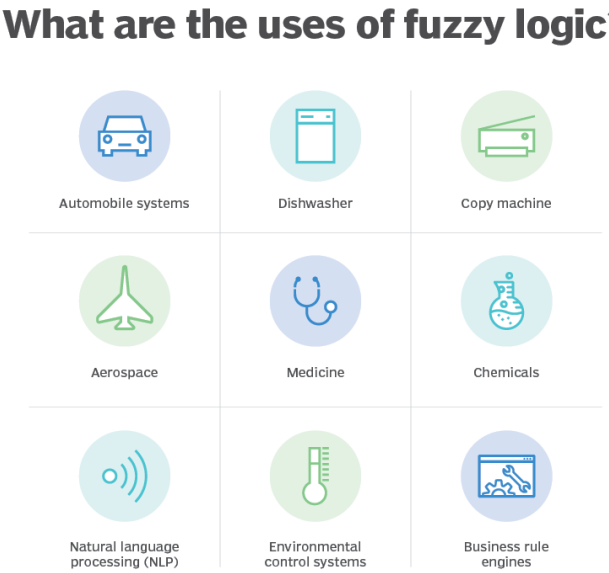
This paper emphasises on the definition and implementation of Fuzzy “degrees of truth” in computational based logic depictions.
Fuzzy logic was first advanced in the 60’s by Lotfi Zadeh of the University of California, studying the concept of “Natural language” in a computational based context, considering that many decision making processes aren’t always as simple as 0 or 1. Fuzzy logic then adds reasoning and variable consideration to the formally boolean processes to determine a more intricate detailing for an output.

The paper continues to say that the Fuzzy process more closely resembles human cognitive processing and reasoning, depicting boolean logic as a case of “extreme truth”, so to speak crisp values where the data can fundamentally be understood as either one way or another.

As a result, Fuzzy logic has various suitable applications, like making decisions without clear certainties and imprecise data (Language processing tech), also regulation / control of machine variables (e.g. Climate control).

Fuzzy logic has an enormous amount of applications, as stated below:

- Automobiles: gear selection (engine load, road conditions, driving style).
- Dishwashers: washing strategy & power required (number of dishes, level of food residue).
- Copy machines: drum voltage (humidity, picture density, temperature).
- Aerospace: manage altitude control for satellites + spacecraft (environmental factors)
- Medicine: computer-aided diagnosis (symptoms, medical history).
- Chemical distillation: control pH and temperature variables.
- Natural language processing: determines semantic relations between concepts represented by words and other linguistic variables.



- Environmental control systems: air conditioners / heaters (current temperature, target temperature).
- Business rules engine: streamlines decision-making according to predetermined criteria.

Paper 2 - Fuzzy Logic: Definitions, Meaning, Examples & History [2]

Continuing on the definition of what Fuzzy logic is and how it can be applied to computational logic, this paper also explores other examples and history of Fuzzy logic.
Defining it as a “heuristic approach that allows for more advanced decision-tree processing and better integration with rule-based programming”.
In other words, it allows for a more detailed approach with a wider-span of factoring and variables for a more comprehensive, logical outlook on a decision with open / imprecise spectrums of data.

Fuzzy logic is a generalisation from standard logic, in which all statements have a truth value of one or zero. In Fuzzy logic, statements can have a value of partial truth, such as 0.9 or 0.5.
Theoretically, this gives the approach more opportunity to mimic real-life circumstances, where statements of absolute truth or falsehood are rare.
Fuzzy logic may be used by quantitative analysts to improve the execution of their algorithms.
Because of the similarities with ordinary language, fuzzy algorithms are comparatively simple to code, but they may require thorough verification and testing.

Fuzzy logic as such flaunts a variety of logical capabilities that far outweighs boolean logic, for instance:

- Fuzzy logic is more likely to reflect real-world problems than classical logic.
- Fuzzy logic algorithms have lower hardware requirements than classical boolean logic.
- Fuzzy algorithms can produce accurate results with imprecise or inaccurate data.

Nothing is without its drawbacks however:

- Fuzzy algorithms require broad validation and verification.
- Fuzzy control systems are dependent on human expertise and knowledge.
- Fuzzification. The process of converting specific input values into some degree of membership of fuzzy sets based on how well they fit.
- Fuzzy rules / knowledge base. These are the If-Then rules to follow, often derived from expert opinions or via more quantitative approaches.
- Inference method. The way of obtaining the final fuzzy conclusion, according to the degree of membership of input variables to fuzzy sets and the detailed fuzzy rules
- Defuzzification. The process of converting the fuzzy conclusions into detailed output values.

Another point to mention, Fuzzy logic has its drawbacks as we have seen, however, one solution to these drawbacks of Fuzzy logic.. Is Fuzzy logic! Specifically Type-2 Fuzzy logic, which takes another approach to the Fuzzy system, evaluating it in a 3D stance that unlocks a new variety of rules and logical outlooks.
Considering this system could be beneficial for future systems.

Utilising Fuzzy Logic for Adaptive Game difficulty -

Paper 3 - Dynamic Difficulty Adjustment through an Adaptive AI [3]

This paper addresses the production of a dynamic game system that needs to autonomously adapt a computer controlled enemies actions based on the players performance throughout, this performance needs to be evaluated based on a metric containing the controlled game mechanics that can be altered to determine an overall difficulty level.
The enemy AI would be adjusted based on this development dynamically by testing the players capabilities against the difficulty metric, favouring the players skill.

With all of this in mind, it's explicit that this process cannot be completed by standard boolean logic, it accounts for a variety of data types that consider how well a player engages in the system against a competitive AI controlled under a difficulty metric.
This calls for the use of Fuzzy logic, considering the data is imprecise, it'll need a logical overview that mimics human decision making, since the metric also considers the players skill more relevant to the outcome, Fuzzy logic can weight the variables to ensure the output accounts for this.

Paper 4 - AI for Dynamic Difficulty Adjustment in Games [4]

Continuing on the lines of adaptive difficulty systems, this document continues to argue that static difficulty settings do not challenge a player with the development of skill based

gameplay, as such more players would likely benefit more from a more flexible approach to a game that links the players overall ability to the overall difficulty.

Using resources as an example, the paper continues on by linking the direct proportionality of game difficulty to the quantitative access to resources, I'd say that this topic can be widely explored in survival based genre's, with resource management being one of the key elements that dictates how well a player can thrive in the challenging environment, showing intuition with both resource collection and resource use.
This concept is still applicable to other game types however, usually with the factoring of resource management dictating what kind of game it is / what it's gameplay focus is.

Overview of Literature -

From the considered literature, I have reached a conclusion on what I intend to develop for this project, initially my "Game Difficulty" Fuzzy Inference System idea was a static concept that would determine how a player proceeded through certain area of the game, considering the gameplay factors, then would store and compare this data with the overall outcome, this outcome would then be revealed to the player at the end to show at what level they were playing at throughout.
I've decided that for a better application of Fuzzy logic to my project, I'll be producing a dynamic difficulty system, much like the ones i've mentioned above, utilising the capabilities of Fuzzy logic to determine a linking algorithm between player action / capabilities to the set difficulty levels.

Solution Overview -

(Develop an idea for what the project is initially designed to be)
A games' difficulty will dictate the player's experience, configuring the multiple variables that build the game system.

Technical Project Development -

Initial Concept -

The initial idea was essentially a difficulty calculator that'd allow a player to adjust gameplay mechanics to fit their individual requirements, then produce a difficulty rating that details and justifies the rank based on their customisation.
This is quite a simplified system considering its mostly sorted by preset values, the complexity could be improved by giving more freedom towards the variable adjustment, however, this would possibly be inefficient for the project scope, evaluating the concept it's clear this system may be more fit towards a ranking system as opposed to difficulty.
However the system can definitely work and has practicality for the project's target market.

Reconsidered Concept -

I'm considering another approach that creates a more elaborate system, producing a dynamic difficulty calculation system that will begin the game at a preset (everything at its

median unless configured to do otherwise), the difficulty would evolve during gameplay, fixing itself based on the data gathered to configure itself to fixate on player ability.

This would most likely include practical user input (no. of enemies dispatched, time taken throughout areas, no. of supplies collected, damage taken etc).
This variable data will quantify the problem and allow us to use fuzzy logic to specify the individual difficulty levels as required.
(A good reference to this style is the MGS4 ranking system, it utilised the user play data to dictate the users play style at completion, this only references how the data was used at the end, consider other dynamic options).
This applies better to the project scope, being more practical for the target system while maintaining the initial concepts rank-style outcome.

Concept Analysis -

The reconsideration has proved to have a more intuitive, dynamic approach to the fuzzy logic solution, a system that collects real-time data to process a players capabilities and skill, evaluating it through a justified rule based system to adjust the variables of difficulty to pre-specified levels. This concedes that this should be the selected approach to evaluate, justify and produce for this project.

Technical Review -

The technical description of the Fuzzy Inference System, delving into the variable selection and rulebase to dictate the products of the defined fuzzy system and justifying how they'll satisfy the system requirements.

Variables (*Inputs & Outputs*) -

The details that'll store the values to be used to determine a player's skill, establishing the game difficulty. (Minimum 3 Inputs, 1 Output)
Encounters (The in-game events) can break up the game into computable sections, these can be utilised to register the difficulty level based on the players capability throughout.

Variables will be based on these customisable settings from a game difficulty example [6]:

Player Settings -
- Amount of damage the player takes from enemies
- Frequency of mid-encounter dynamic checkpoints

Enemy Settings -
- Accuracy of enemy gunfire and frequency of projectiles
- Aggression of enemies advancing & flanking
- Complexity of enemy melee combos
- Movement speed of certain high-threat enemies
- Custom tuning to specific combat encounters

Ally Settings -
- Aggressiveness of allies in combat

- Frequency at which allies kill enemies

Stealth Settings -

- Enemies perception through vision, hearing and smell
- Length of grace period before enemies will alert others
- Conditions for grabbing enemies from stealth

Resource Settings -

- Quantity of ammunition and supplies found in the world
- Durability of melee weapons dropped by enemies
- Yield of certain crafting recipes

Input Variables:

Health (Survival capabilities based on “Damage Taken” & “Restart Count”) [0.40]
[Bad, Average, Good, Outstanding]
Damage Taken (No. of health points lost) *[Low, Medium, High]*
Restarts (Counter for the number of player deaths and / or restarts upon failure of an objective) *[Low, Medium, High]*

Resource Management (Overall Resources Used) *[Bad, Average, Good, Outstanding]* [0.35]
Ammunition / Throwables *[Low, Medium, High]*
Explosives *[Low, Medium, High]*
Med Packs *[Low, Medium, High]*

Enemy Dispatches (Enemy Kills in each encounter “Enemy”, “Stealth”, “Ally Takedowns”) *[Low, Medium, High]* [0.25]
No. of Enemy Takedowns (Enemies Dispatched in combat state)
[Low, Medium, High]
No. of Stealth Takedowns (Enemies dispatched in stealth state)
[Low, Medium, High]
Ally Takedowns (Enemies dispatched via ally interaction)
[Low, Medium, High]

Output variables:

Player Skill (Determines the players capabilities based on their “Health”, “Enemy DIspatches” and “Resource Management”).
[Story Mode, Basic, Moderate, Hardcore, Survivor, Unstoppable]
Current Difficulty (Difficulty level currently set to play at, assigned by “Difficulty Level”)
[Very Light, Light, Moderate, Hard, Survivor, Grounded] [10, 100]
Difficulty Level (The overall outcome for gameplay, input for “Current Difficulty”)
[Very Light, Light, Moderate, Hard, Survivor, Grounded]

Rulebase -

Consider the rules for the rule-based system based on the difficulty levels,
The rules are applied after encounters to designate the difficulty throughout,

Rules -

- If (Restarts *[High]*) **THEN** (Health *[Bad]*)
- If (Damage Taken *[High]*) **&** (Restarts *[Medium]*) **THEN** (Health *[Bad]*)
- If (Damage Taken *[Low]*) **&** (Restarts *[High]*) **THEN** (Health *[Average]*)
- If (Damage Taken *[Medium]*) **&** (Restarts *[Medium]*) **THEN** (Health *[Average]*)
- If (Damage Taken *[High]*) **&** (Restarts *[Low]*) **THEN** (Health *[Average]*)
- If (Damage Taken *[Low]*) **&** (Restarts *[Medium]*) **THEN** (Health *[Good]*)
- If (Damage Taken *[Medium]*) **&** (Restarts *[Low]*) **THEN** (Health *[Good]*)
- If (Restarts *[Low]*) **THEN** (Health *[Outstanding]*)

- If (Ammo/Throwables Used *[High]*) **||** (Explosives Used *[High]*) **||** (Med Kits Used *[High]*) **THEN** (Resource Management *[Bad]*)
- If (Ammo/Throwables Used *[Medium]*) **||** (Explosives Used *[Medium]*) **||** (Med Kits Used *[Medium]*) **THEN** (Resource Management *[Average]*)
- If (Ammo/Throwables Used *[Low]*) **||** (Explosives Used *[Low]*) **||** (Med Kits Used *[Low]*) **THEN** (Resource Management *[Outstanding]*)

- If (Enemies Takedowns *[Low]*) || (Stealth Takedowns *[Low]*) **||** (Ally Takedowns *[Low]*) **THEN** (Enemy Dispatches *[Low]*)
- If (Enemies Takedowns *[Medium]*) **||** (Stealth Takedowns *[Medium]*) **||** (Ally Takedowns *[Medium]*) **THEN** (Enemy Dispatches *[Medium]*)
- If (Enemies Takedowns *[High]*) **||** (Stealth Takedowns *[High]*) **||** (Ally Takedowns *[High]*) **THEN** (Enemy Dispatches *[High]*)

- If (Health *[Bad]*) **&** (Resource Management *[Bad]*) **THEN** (Player Skill *[Story Mode]*)
- If (Health *[Bad]*) **&** (Resource Management *[Bad]*) & (Enemy Dispatches *[High]*) **THEN** (Player Skill *[Basic]*)
- If (Health *[Average]*) **&** (Resource Management *[Average]*) **THEN** (Player Skill *[Moderate]*)
- If (Health *[Average]*) **&** (Resource Management *[Average]*) & (Enemy Dispatches *[High]*) **THEN** (Player Skill *[Hardcore]*)
- If (Health *[Good]*) **&** (Resource Management *[Good]*) **THEN** (Player Skill *[Survivor]*)
- If (Health *[Outstanding]*) **&** (Resource Management *[Outstanding]*) **THEN** (Player Skill *[Unstoppable]*)

- If (Player Skill *[Story Mode]*) **&** (Current Difficulty *[Very Light]*) **THEN** (Difficulty *[Very Light]*)
- If (Player Skill *[Story Mode]*) **&** (Current Difficulty *[Light]*) **THEN** (Difficulty *[Very Light]*)
- If (Player Skill *[Story Mode]*) **&** (Current Difficulty *[Moderate]*) **THEN** (Difficulty *[Light]*)
- If (Player Skill *[Story Mode]*) **&** (Current Difficulty *[Hard]*) **THEN** (Difficulty *[Light]*)
- If (Player Skill *[Story Mode]*) **&** (Current Difficulty *[Survivor]*) **THEN** (Difficulty *[Moderate]*)
- If (Player Skill *[Story Mode]*) **&** (Current Difficulty *[Grounded]*) **THEN** (Difficulty *[Moderate]*)

- If (Player Skill *[Basic]*) **&** (Current Difficulty *[Very Light]*) **THEN** (Difficulty *[Light]*)
- If (Player Skill *[Basic]*) **&** (Current Difficulty *[Light]*) **THEN** (Difficulty *[Light]*)
- If (Player Skill *[Basic]*) **&** (Current Difficulty *[Moderate]*) **THEN** (Difficulty *[Light]*)
- If (Player Skill *[Basic]*) **&** (Current Difficulty *[Hard]*) **THEN** (Difficulty *[Moderate]*)
- If (Player Skill *[Basic]*) **&** (Current Difficulty *[Survivor]*) **THEN** (Difficulty *[Moderate]*)
- If (Player Skill *[Basic]*) **&** (Current Difficulty *[Grounded]*) **THEN** (Difficulty *[Hard]*)

- If (Player Skill *[Moderate]*) **&** (Current Difficulty *[Very Light]*) **THEN** (Difficulty *[Light]*)
- If (Player Skill *[Moderate]*) **&** (Current Difficulty *[Light]*) **THEN** (Difficulty *[Moderate]*)
- If (Player Skill *[Moderate]*) **&** (Current Difficulty *[Moderate]*) **THEN** (Difficulty *[Moderate]*)
- If (Player Skill *[Moderate]*) **&** (Current Difficulty *[Hard]*) **THEN** (Difficulty *[Hard]*)

- If (Player Skill *[Moderate]*) **&** (Current Difficulty *[Survivor]*) **THEN** (Difficulty *[Hard]*)
- If (Player Skill *[Moderate]*) **&** (Current Difficulty *[Grounded]*) **THEN** (Difficulty *[Survivor]*)

- If (Player Skill *[Hardcore]*) **&** (Current Difficulty *[Very Light]*) **THEN** (Difficulty *[Moderate]*)
- If (Player Skill *[Hardcore]*) **&** (Current Difficulty *[Light]*) **THEN** (Difficulty *[Moderate]*)
- If (Player Skill *[Hardcore]*) **&** (Current Difficulty *[Moderate]*) **THEN** (Difficulty *[Hard]*)
- If (Player Skill *[Hardcore]*) **&** (Current Difficulty *[Hard]*) **THEN** (Difficulty *[Hard]*)
- If (Player Skill *[Hardcore]*) **&** (Current Difficulty *[Survivor]*) **THEN** (Difficulty *[Survivor]*)
- If (Player Skill *[Hardcore]*) **&** (Current Difficulty *[Grounded]*) **THEN** (Difficulty *[Survivor]*)

- If (Player Skill *[Survivor]*) **&** (Current Difficulty *[Very Light]*) **THEN** (Difficulty *[Moderate]*)
- If (Player Skill *[Survivor]*) **&** (Current Difficulty *[Light]*) **THEN** (Difficulty *[Hard]*)
- If (Player Skill *[Survivor]*) **&** (Current Difficulty *[Moderate]*) **THEN** (Difficulty *[Hard]*)
- If (Player Skill *[Survivor]*) **&** (Current Difficulty *[Hard]*) **THEN** (Difficulty *[Survivor]*)
- If (Player Skill *[Survivor]*) **&** (Current Difficulty *[Survivor]*) **THEN** (Difficulty *[Survivor]*)
- If (Player Skill *[Survivor]*) **&** (Current Difficulty *[Grounded]*) **THEN** (Difficulty *[Grounded]*)

- If (Player Skill *[Unstoppable]*) **&** (Current Difficulty *[Very Light]*) **THEN** (Difficulty *[Hard]*)
- If (Player Skill *[Unstoppable]*) **&** (Current Difficulty *[Light]*) **THEN** (Difficulty *[]Hard*)
- If (Player Skill *[Unstoppable]*) **&** (Current Difficulty *[Moderate]*) **THEN** (Difficulty *[Survivor]*)
- If (Player Skill *[Unstoppable]*) **&** (Current Difficulty *[Hard]*) **THEN** (Difficulty *[Survivor]*)
- If (Player Skill *[Unstoppable]*) **&** (Current Difficulty *[Survivor]*) **THEN** (Difficulty *[Grounded]*)
- If (Player Skill *[Unstoppable]*) **&** (Current Difficulty *[Grounded]*) **THEN** (Difficulty *[Grounded]*)

Testing -

Rulebase Evolution, Variation of Membership Functions, Distribution of Membership Functions, Defuzzification, Implication & Aggregation, Operator Methods, Configurations

FIS Performance Testing -

(Experiment with the rule-based structure by testing the multiple possibilities of the rules and variables.
Consider all the possibilities attainable by the rules and variables, then develop the rulebase and evolve the rules as best done, consider the importance of the rule or if it is combinable with other rules, condensing the rulebase to be dynamic and intuitive.)

The initial performance design considered all the possible variables that came with the rulebase concept as shown in **Appendix A1**, variables and sub-variables of the relative types were produced to follow this concept and classify a rulebase by individually configuring each variable assigning rules and declaring functions, as is shown in **Appendix B1**.

However, this ruling system is very inconvenient, pitting several individual concepts against one another to produce an overall ranking system is unsuitable for the required product, it also produces an awfully convoluted rulebase that doesn't make use of the features of fuzzy logic designs.

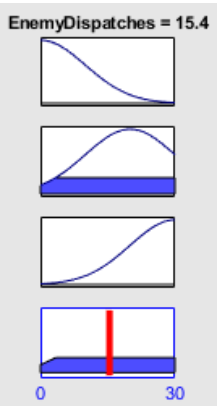
As such an alternative approach has been considered:

Each of these concepts will be evaluated, but rather by assigning them to sets that'll compact the details, make them easier to both read and process.

For instance, if we consider "Health" as a set weighing the values "Damage Taken" & "Restarts", the value of "Health" will take the values as input (weighting 30:70 respectively) and calculate an overall value dictating the players survival capabilities, this summarises the details by assigning them to a common variable, which can now be utilised in the "Player Skill" FIS with more ease.

Also, rather than making "Player Skill" and umbrella variable to store the results of all the combines values through rules and statements, it has become the harbour of the player capability data, as mentioned storing "Health", but also "Resource Management" and "Enemy Dispatches" to dictate the players gameplay abilities.

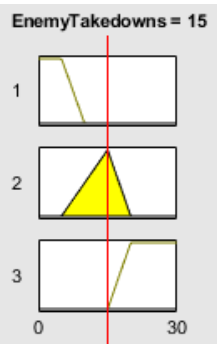
Combining this data with an input specifying the "Current Difficulty" tests the players capabilities against the inferred difficulty setting, dictating any change required to the level output.



Design Testing -

Produce results of the experimentation and present the Fuzzy Inference System design implemented in MATLAB.

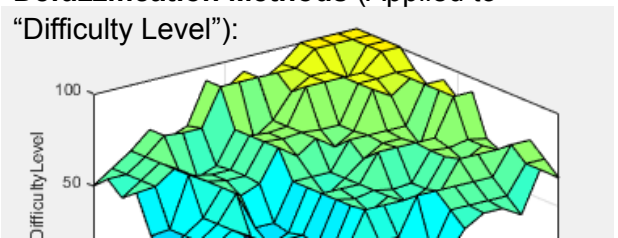
One of the main tasks of this testing concept is to find out what design features work the best for presenting the visual data for the Fuzzy Inference Systems, Ruleviews & Surface views.
The first order being to determine the membership functions suitable for the fuzzy sets, I considered first and foremost that the output variable for the player skill inputs would be Gaussian considering the data was intricate and variant.

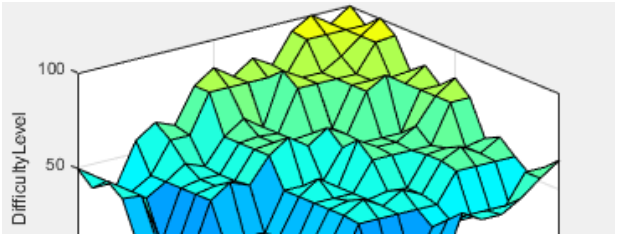


I found that Triangular worked best for median values as it would direct attention to the mid points, Trapezoidal I felt worked best for start and end values as it would link the relation best visually between the functions and the overall outcome.

Using my rule set for "Player Skill" you can see the use of the outcomes of the player inputs to determine the player overall capabilities, clearly showing the use of Gaussian membership functions to depict the intricate user data and produce fair depictions of Player skill.
The other Rule sets can be found in the **Appendix D** set showing how their functions have produced their overall outcomes.

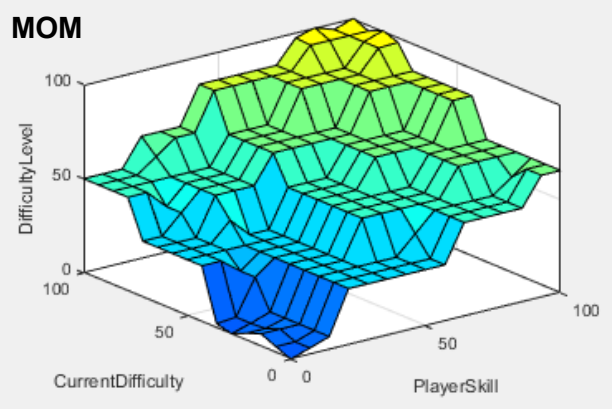
Defuzzification Methods (Applied to "Difficulty Level"):



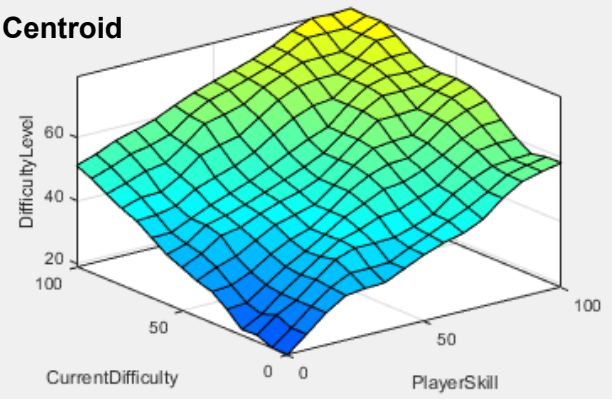


LOM **SOM**

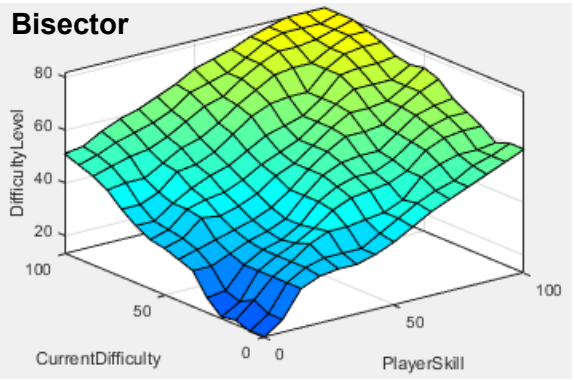
MOM



Centroid



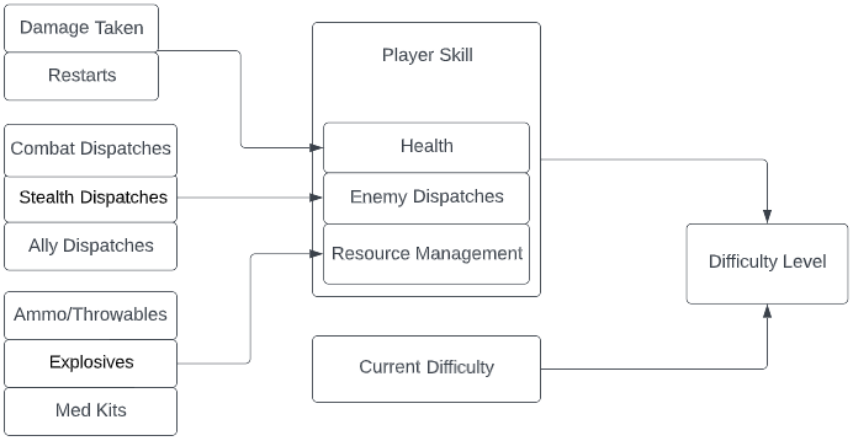
Bisector



Clear to see that **Centroid** produces the best uniform example of difficulty.

System Design -

I've decided to go with this design, considering all of the variables and their links, all of the player skill input would work appropriately to affect the 3 main aspects of player skill, this alongside the currently active difficulty would stand to produce a difficulty level that stands to the levels of the player capabilities and alter the gameplay as such.



Critical Evaluation -

I believe this project has produced viable results depicting how a Fuzzy system could be used to create a dynamically evolving difficulty system to an otherwise standard system to add a level of challenge into a new area of game development.
This concept could mean a potential evolution of mechanics in game play, in fact being a topic that has been explored by games in the past (Hello Neighbour specifically flaunted an adaptive enemy AI system that would learn the players approach to the game and act to keep the player out of the target area), this is a highly anticipated concept for players, even being a focus point of the future of gameplay due to its adaptive technology.
Considering this approach via Fuzzy logic can broaden the possibility of this technology being implemented in future games and AI's.

I also feel that this topic was a good consideration to evaluate the use of Fuzzy logic in a modern computational approach, considering it would have to take otherwise "Natural Logic" to dictate exactly how skilled a player could be or how difficult a game can be.
The dynamic approach to this also made the use of Fuzzy logic more applicable in a visual and non-visual role, considering the difficulty level more accurately through each encounter and evaluating it against the currently applied difficulty.

I do feel there is room for improvement for my depiction of the topic.
For instance, I believe there could have been more use of the several features of Fuzzy logic, like exploring more with the membership functions to dictate the best use for each set, being more creative with the sets, depicting more logical rules etc.
There's more to work with with fuzzy logic, so if I were to redo this topic, I would ensure I take full advantage of the scope of Fuzzy logic.

Conclusion -

I believe this project was handled well, the topic matter and approach I believe was a good choice for evaluating the use of Fuzzy logic in an applicable subject.
Without a doubt though, I would broaden the scope of the project with more time, consider more rules and variables that could make the outcome more detailed. The test cases could have been better handled and reviewed, the overall system design could have also been considered and updated with more time to add further depth to the overall project.

Overall though, I believe I have met the requirements for producing a good documented logical overview of Fuzzy logic in an applied example.
With some more reflection on the existing points, data and tests, the project could have been made much more substantial.

References -

[1] SearchEnterpriseAI. (n.d.). *What is Fuzzy Logic? - Definition from SearchEnterpriseAI*. [online] Available at: https://www.techtarget.com/searchenterpriseai/definition/fuzzy-logic#:~:text=Fuzzy%20logic%20is%20an%20approach.

[2] Scott, G. (2021). *Fuzzy Logic Definition*. [online] Investopedia. Available at: https://www.investopedia.com/terms/f/fuzzy-logic.asp.

[3] Research Gate. (n.d.). *Dynamic Difficulty Adjustment through an Adaptive AI* [online] Available at: https://www.researchgate.net/publication/312241128_Dynamic_Difficulty_Adjustment_through_an_Adaptive_AI

[4] Research Gate. (n.d.). *AI for dynamic difficulty adjustment in games* [online] Available at: https://www.researchgate.net/publication/228889029_AI_for_dynamic_difficulty_adjustment_in_games

[5] Research Gate. (n.d.). *Design and implementation of the rhythm action game difficulty based on fuzzy logic* [online] Available at: https://www.researchgate.net/publication/292793847_Design_and_implementation_of_the_rhythm_action_game_difficulty_based_on_fuzzy_logic

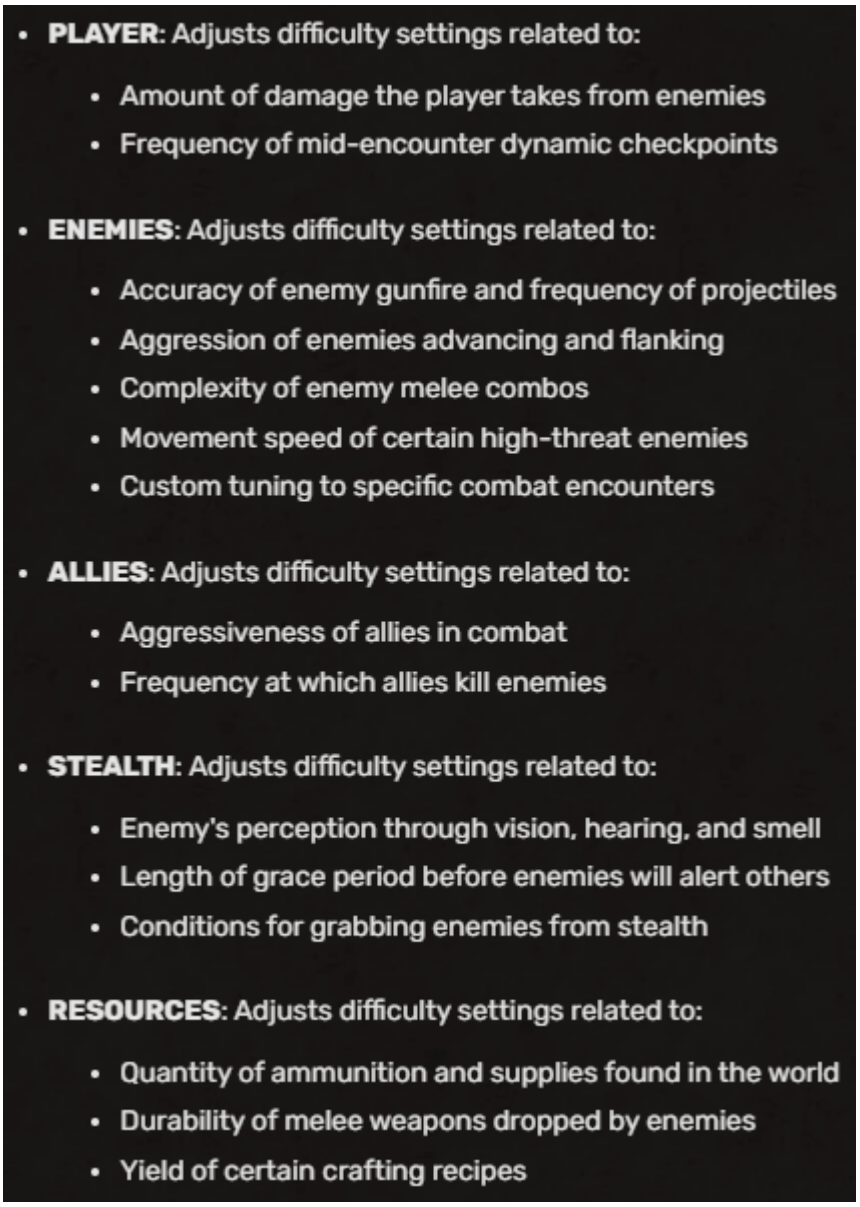
[6] The Last of Us Wiki. (n.d.). *Difficulty*. [online] Available at: https://thelastofus.fandom.com/wiki/Difficulty#Custom_Difficulty

[7] Lucidchart (2022). *Online diagram software & visual solution | Lucidchart*. [online] Lucidchart. Available at: https://www.lucidchart.com/pages/.

Appendices -

(Include relevant images and visual data that accentuates the reported details, include all here and consider its location here in comparison to the report
Incl. drop down appendices list to document relevance to report literature)

Appendix A1: (Variable and set concept idea (from “The Last of Us Part 2” difficulty system)



Appendix B1: (Initial Rulebase considering all initial values against each other)
Inputs: *Damage Taken, Restart Count, Frequency of Resources Used, No. of Enemy Takedowns, No. of Stealth Takedowns, Ally Reliance.*
Outputs: *Player Skill, Resource Use, Overall Difficulty Level.*

If (Damage Taken *[High]*) **&** (Restart Count *[None]*) **THEN** (Player Skill *[Survivor]*)

- If (Damage Taken *[Low]*) **&** (Restart Count *[Low])* **THEN** (Player Skill *[Survivor]*)
- If (Damage Taken *[Medium]*) **&** (Restart Count *[Low]*) **THEN** (Player Skill *[Hardcore]*)
- If (Damage Taken *[Low]*) **&** (Restart Count *[Medium]*) **THEN** (Player Skill *[Hardcore]*)

- If (Damage Taken *[High]*) **&** (Restart Count *[Low]*) **THEN** (Player Skill *[Moderate]*)
- If (Damage Taken *[Medium]*) **&** (Restart Count *[Medium]*) **THEN** (Player Skill *[Moderate]*)
- If (Damage Taken *[Low]*) **&** (Restart Count *[High]*) **THEN** (Player Skill *[Moderate]*)
- If (Damage Taken *[High]*) **&** (Restart Count *[Medium]*) **THEN** (Player Skill *[Basic]*)
- If (Damage Taken *[Medium]*) **&** (Restart Count *[High]*) **THEN** (Player Skill *[Basic]*)

- If (Restart Count *[None]*) **THEN** (Player Skill *[Unstoppable]*)
- If (Restart Count *[High]*) **THEN** (Player Skill *[Story Mode]*)

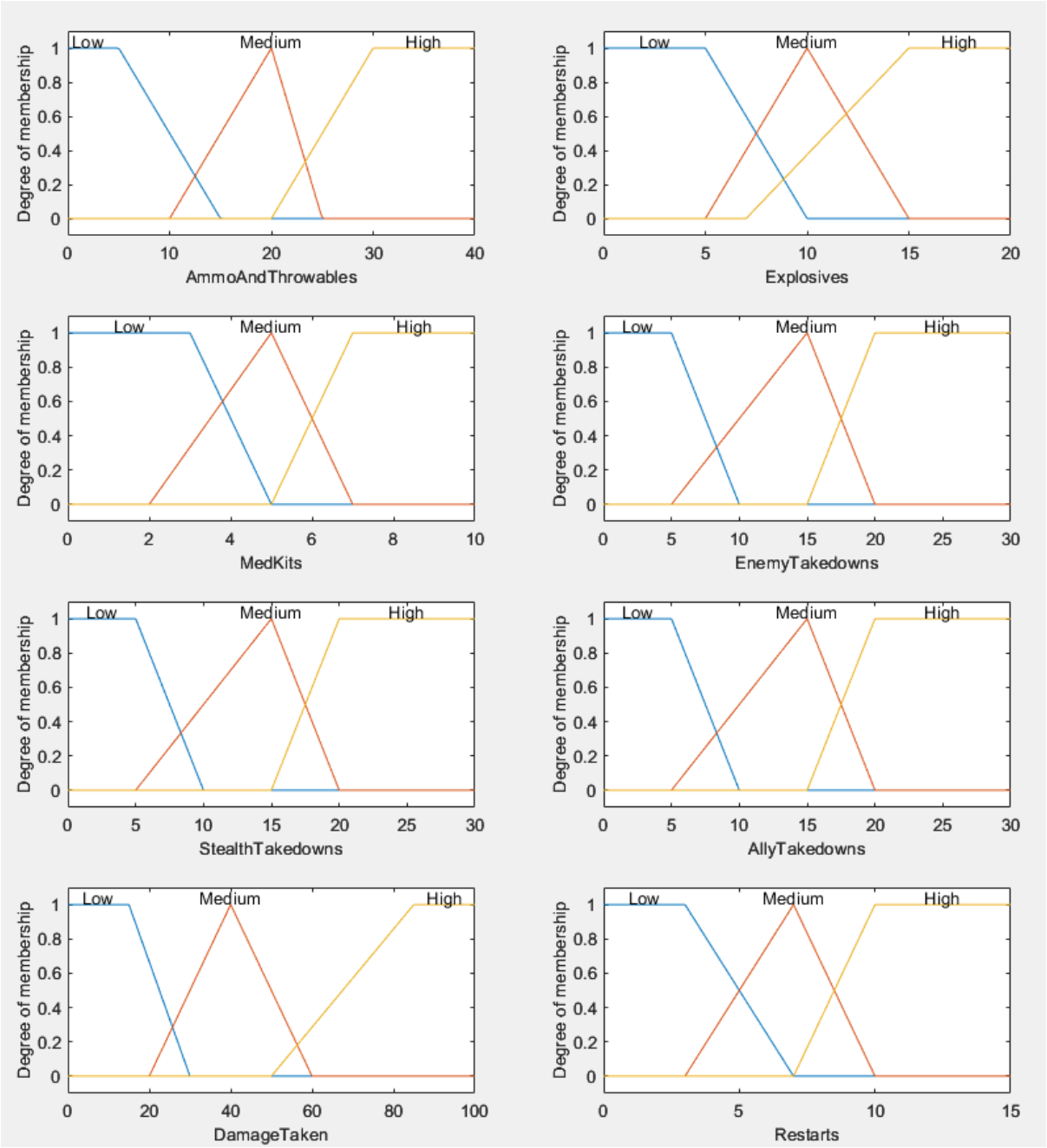
- If (Ammo/Throwables Used *[None]*) **&** (Explosives Used *[None]*) **&** (Med Kits Used *[None]*) **THEN** (Resource Use *[None]*)
- If (Ammo/Throwables Used *[Low]*) **&** (Explosives Used *[Low]*) **&** (Med Kits Used *[Low]*) **THEN** (Resource Use *[Low]*)
- If (Ammo/Throwables Used *[Medium]*) **&** (Explosives Used *[Medium]*) **&** (Med Kits Used *[Medium]*) **THEN** (Resource Use *[Medium]*)
- If (Ammo/Throwables Used *[High]*) **&** (Explosives Used *[High]*) **&** (Med Kits Used *[High]*) **THEN** (Resource Use *[High]*)

- If **NOT** (Resource Use *[None]*) **&** (Enemies Dispatched *[None]*) **OR** (Stealth Dispatches *[None]*) **THEN** (Player Skill *[Story Mode]*)
- If (Resource Use *[High]*) **&** (Enemies Dispatched *[Low]*) **OR** (Stealth Dispatches *[Low]*) **THEN** (Player Skill *[Basic]*)
- If (Resource Use *[x]*) **&** (Enemies Dispatched *[x]*) **OR** (Stealth Dispatches *[x]*) **THEN** (Player Skill *[Moderate]*)
- If (Resource Use *[Medium]*) **&** (Enemies Dispatched *[Low]*) **OR** (Stealth Dispatches *[Low]*) **THEN** (Player Skill *[Moderate]*)
- If (Resource Use *[High]*) **&** (Enemies Dispatched *[Medium]*) **OR** (Stealth Dispatches *[Medium]*) **THEN** (Player Skill *[Moderate]*)
- If (Resource Use *[Low]*) **&** (Enemies Dispatched *[Medium]*) **OR** (Stealth Dispatches *[Medium]*) **THEN** (Player Skill *[Hardcore]*)
- If (Resource Use *[Medium]*) **&** (Enemies Dispatched *[High]*) **OR** (Stealth Dispatches *[High]*) **THEN** (Player Skill *[Hardcore]*)
- If (Resource Use *[None]*) **& NOT** (Enemies Dispatched *[None]*) **OR NOT** (Stealth Dispatches *[None]*) **THEN** (Player Skill *[Survivor]*)
- If (Resource Use *[Low]*) **&** (Enemies Dispatched *[High]*) **OR** (Stealth Dispatches *[High]*) **THEN** (Player Skill *[Survivor]*)
- If (Resource Use *[None]*) **&** (Enemies Dispatched *[None]*) **OR** (Stealth Dispatches *[None]*) **THEN** (Player Skill *[Unstoppable]*)
- If (Resource Use *[None]*) **&** (Enemies Dispatched *[High]*) **OR** (Stealth Dispatches *[High]*) **THEN** (Player Skill *[Unstoppable]*)

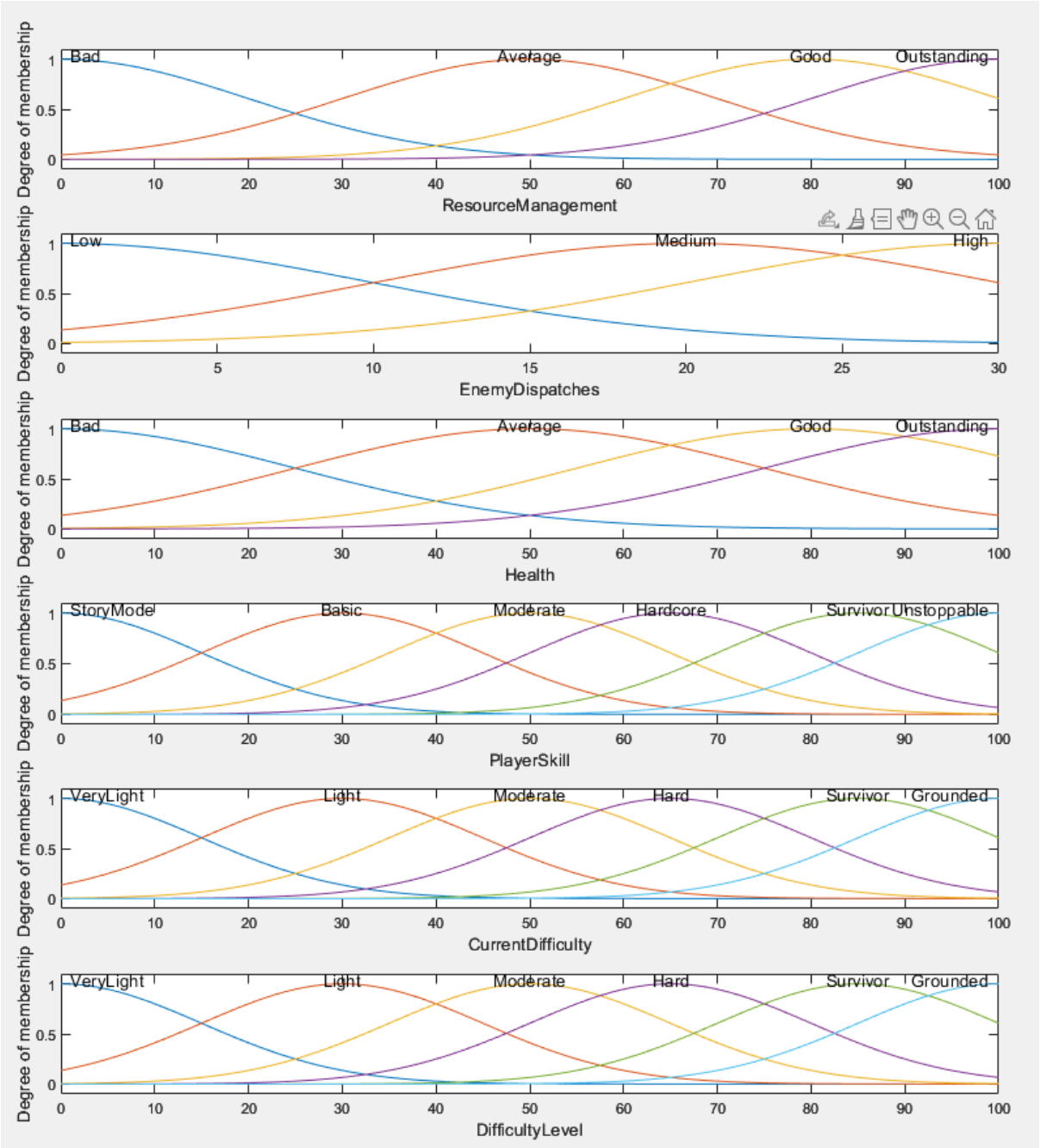
- If (Enemies Dispatched *[None]*) **OR NOT** (Stealth Dispatches *[None]*) **&** (Damage Taken *[None]*) **&** (Resource Use *[None]*) **THEN** (Player Skill *[Unstoppable]*)

- If (Ally Reliance *[None]*) **THEN** (Player Skill *[Survivor]*)
- If (Ally Reliance *[Low]*) **THEN** (Player Skill *[Hardcore]*)
- If (Ally Reliance *[Medium]*) **THEN** (Player Skill *[Moderate]*)
- If (Ally Reliance *[High]*) **THEN** (Player Skill *[Basic]*)

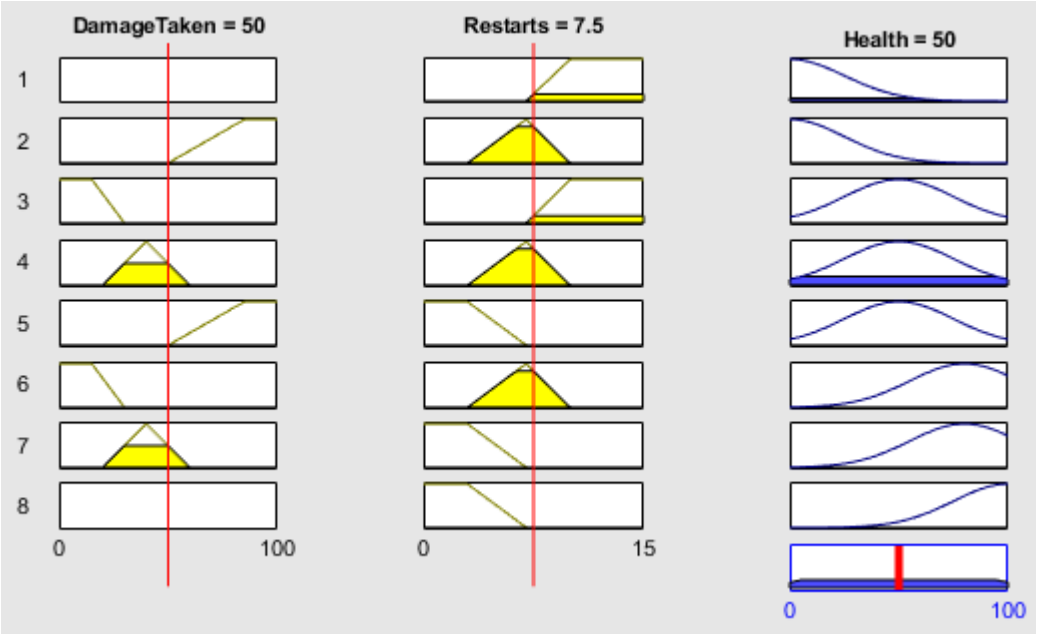
Appendix C1: (Figure 1: Membership Plot Inputs)



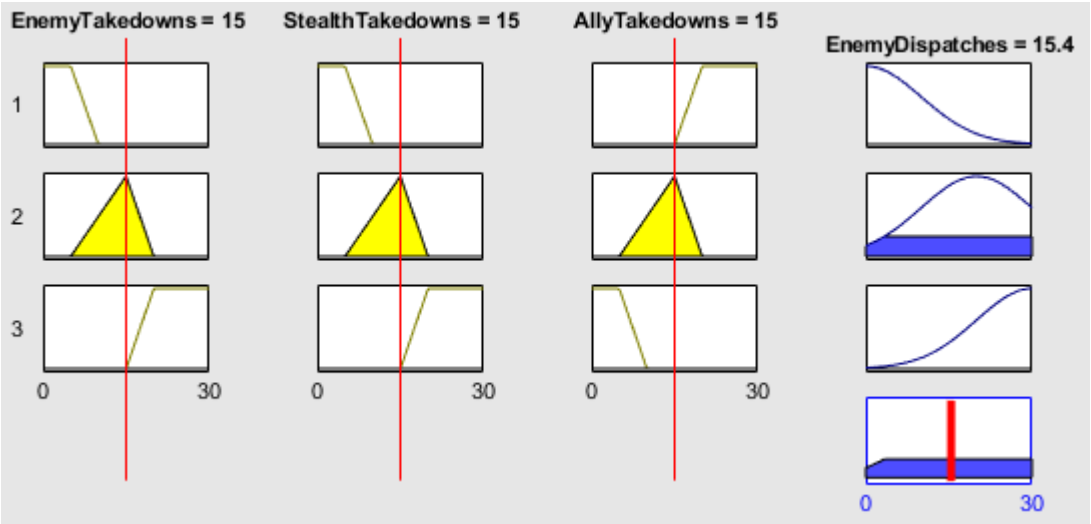
Appendix C2: (Figure 1: Membership Plot Outputs)



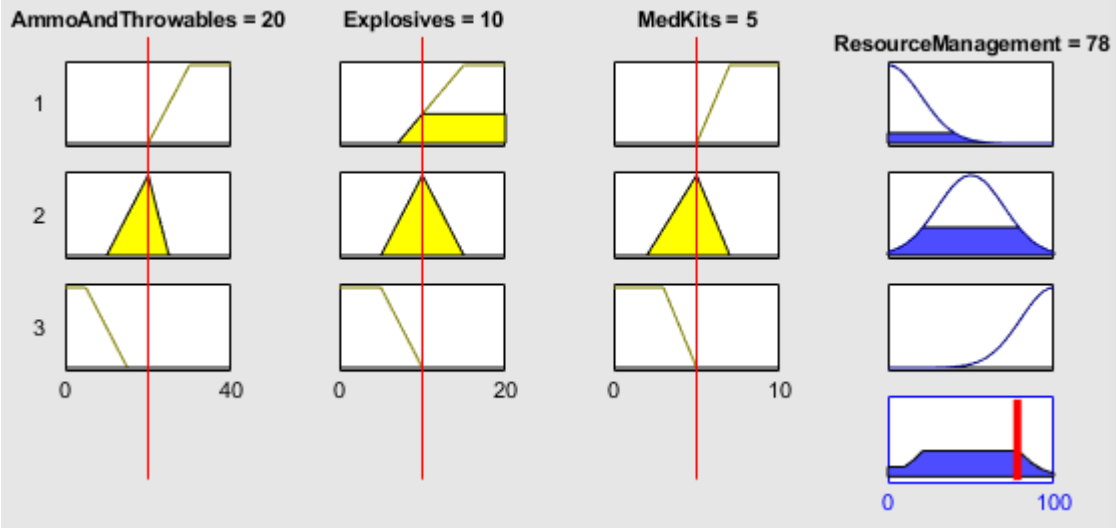
Appendix D1: (Rule view: Health)



Appendix D2: (Rule view: Enemy Dispatches)



Appendix D3: (Rule view: Resource Management)



Appendix D4: (Rule view: Difficulty Level)