



GALWAY-MAYO INSTITUTE OF TECHNOLOGY

Department of Computer Science & Applied Physics

Artificial Intelligence **ASSIGNMENT DESCRIPTION & SCHEDULE**

Controlling Game Characters with Neural Networks and Fuzzy Logic



Note: *This assignment will constitute 50% of the total marks for this module.*

You are required to create a simple maze-based game in which the characters are controlled by fuzzy logic or neural networks and incorporates a subset of the path finding algorithms covered in this module. Specifically, you are required to:

- **Work in pairs** to create a 2D maze game. The stubs for a game, along with a “Spartan Warrior” main character and 8 spider sprites {Black, Blue, Brown, Green, Grey, Orange, Red, Yellow} are available on Moodle under the link *Partial Maze Game*. A suite of maze generation algorithms will also be available on Moodle. These include the *Binary Tree*, *Recursive Backtracker*, *Random Depth-First* and *Hunt and Kill* algorithms.
- **Use fuzzy logic and neural networks** to control the enemy spider sprites. The sprites should be configurable with one (or more...) path finding algorithms and should be fully threaded, i.e. each sprite should have it's own independent form of movement. Use the JFuzzyLogic API and the A 3-Layer Back-propagation Neural Network (available on Moodle) to control how the enemy sprites interact with the main character. Note that a FCL function block equates to a method and that the inputs and output from a neural network can also be related to methods parameters and return types. You can also opt to allow the game to play in “automatic” mode by adding a fuzzy logic or neural net controller to the “Spartan Warrior”.

You are free to asset-strip any of the software that we have used in labs (fuzzy logic, neural networks, maze generation and path finding algorithms. The emphasis should be on the creative application of these resources to a new problem. In addition, as the maze “game” is substantially complete, you should concentrate your efforts on the features listed under the marking scheme below. You must document your rationale for the choice of fuzzy sets / rules / network topology / algorithm etc. used in your application and you will be required to demonstrate your application in a lab after the Easter break.

Deployment and Submission

- **The project must be submitted by midnight on Friday 21st April 2017** as a Zip archive (**not a rar or WinRar file**) using the Moodle upload utility. You can find the area to upload the project under the "Controlling Game Characters with Neural Networks and Fuzzy Logic - (50%) Assignment Upload" heading in the "Main Assignment - Description and Resources" section of Moodle.
- The name of the Zip archive should be *{id}.zip* where *{id}* is your student number.
- You must use the package name **ie.gmit.sw.ai** for the assignment. **Do not include the JFuzzyLogic JAR with your submission.**
- The Zip archive should have the following structure (do NOT submit the assignment as an Eclipse project):

| Name | Description |
|------------|--|
| src | A directory that contains the packaged source code for your application. You must package your application with the namespace ie.gmit.sw.ai . Your code should be well commented and explain the space and time complexity of its classes. |
| README.txt | A text file outlining the main features of your application. |
| game.jar | <p>A Java Application Archive containing the classes in your application. Use the following syntax to create a JAR from a command prompt:</p> <pre>jar -cf game.jar ie/gmit/sw/ai/*.class</pre> <p>Your application should be launched using the following syntax from a command prompt:</p> <pre>java -cp ./game.jar ie.gmit.sw.ai.GameRunner</pre> |

Marking Scheme

Assuming that your assignment is *submitted correctly and executes correctly*, the following rubric will be used when marking the project:

- **Fuzzy Logic – 25%**
 - Linguistic Variables & Membership Functions – 13%
 - Fuzzy Rules – 12%
- **Neural Network – 25%**
 - Network Topography (nodes in input, hidden and output layers) – 10%
 - Network Training (Data Sets & Parameters, e.g. learning rate) – 15%
- **AI Search Algorithms – 20%**
 - Application of algorithms in the project by characters or objects, e.g. help.
- **Threaded Characters - 10%**
 - Use a thread pool (*ExecutorService*) to thread the characters. The application is either properly threaded or not, i.e. the 10% is all or nothing...
- **Relevant Extra Features - 20%**
 - The word “relevant” in this case implies that the extra functionality is AI related and belongs to the following domains: fuzzy logic, neural networks, semantic networks, heuristics, path finding. Don’t waste your time on the GUI look and feel, as this is not a graphics course.