



# **GALWAY-MAYO INSTITUTE OF TECHNOLOGY**

***Department of Computer Science & Applied Physics***

---

## **B.Sc. Software Development – Artificial Intelligence (2021)** **MAIN ASSIGNMENT**

### ***Autonomous Computer-controlled Game Characters***

*This assignment constitutes 50% of the total marks for this module.*

#### **Overview**

You are required to use **fuzzy logic**, **neural networks** AI technologies to control a set of characters that are moving randomly through a game model. The AI controlled characters should be able to interact both with each other and the single instance of a user-controlled player. A suite of stubs is provided on Moodle that already implements a JavaFX game window with a model, view and controller. A thread pool containing a set of runnable autonomous characters has also been implemented already.

The purpose of this assignment is for you not to programme a game, but to **design and implement fuzzy logic and neural network controllers for the autonomous game characters**. Therefore, much of the programming will be declarative, e.g. fuzzy control language or involve secondary AI design considerations such as developing a neural network topology or preparing, transforming and processing training data. You have been given an [open brief](#) regarding the function and nature of interaction of the different game characters and you are free to implement any reasonable behaviour that you see fit. This extends to the actual goal of the game and how the game should terminate. Consequently, will have to **carefully document your rationale** for each of the design decisions that you make and justify the various approaches that you have employed.

#### **Minimum Requirements**

- Clearly **document** in a README file the overall goal for the game and your rationale for the set of controllers that you have implemented. You must document all aspects of the fuzzy controller(s) that you design, as well as the architecture, activation functions and input vectors to your neural network. Any hyperparameters should be clearly reasoned and presented.
- Only use the **JFuzzyLogic** and **Encog 3.4** libraries. Do not use any other 3<sup>rd</sup> party software. You can asset-strip, repackage and re-use any of the code from the AI labs, including the 3-layer neural network.
- Each character should be **controlled by some intelligence**, either fuzzy logic, a neural network, a heuristically informed search algorithm or even a cocktail of all three.
- The fuzzy logic should be encapsulated in one of more FCL files and placed in a **.resources/fuzzy** directory. You should consider carefully the following when designing the fuzzy logic components of the system:
  - **Linguistic variables** and Universe of Discourse
  - **Fuzzy Sets** and Membership Functions

- **Defuzzifiers**, e.g. COG, COGS, LM, MM, RM.
- Membership Functions for output variables, e.g. **Mamdani** or **Suegno**.
- **Fuzzy Rules**. All fuzzy sets must be covered by at least one fuzzy rule.

You must ***comment the FCL with your rationale*** for each of the design decisions that you make. Note that you can include more than one function block per FCL file.

- The neural network should be trained, in a reasonable amount of time (< 1 minute), when the game starts up and have a **fully documented topology** presented in the README. Do not use any other 3<sup>rd</sup> party implementation. All the training and validation resources for the neural network should be placed in the ***.resources/neural*** directory.
- You can assume that the **JavaFX** library and the dependencies for **Encog 3.4** and the **JBuzzyLogic** libraries are already available on the ***module-path***. These have already been added to the ***module-info.java*** module descriptor. Do not include any of these dependencies with your submission or change how they are declared in the module descriptor.

### Deployment and Delivery

Please **read the following carefully** and pay particular attention to the files that you are required to submit and those that you should not include with your assignment:

- ***The project must be submitted by midnight on Sunday 4<sup>th</sup> April 2021.*** Before submitting the assignment, you should review and test it from a command prompt on ***a different computer*** to the one that you used to program the project.
- The project must be submitted as a Zip archive (***not a 7z, rar or WinRar file***) using the Moodle upload utility. You can find the area to upload the project under the “Autonomous Computer-controlled Game Characters - (50%) Assignment Upload” link on Moodle. Only the contents of the submitted Zip will be considered. ***Do not add comments to the Moodle assignment upload form.***
- The name of the Zip archive should be ***<id>.zip*** where ***<id>*** is your GMIT student number.
- The Zip archive should have the structure shown below. Do NOT submit the assignment as an Eclipse project.

Marks	Category
game.jar	<p>A Java archive containing your API and runner class with a <b><i>main()</i></b> method. You can create the JAR file using the following command from inside the “<b><i>bin</i></b>” folder of the Eclipse project:</p> <p style="text-align: center;"><b><i>jar -cf game.jar *</i></b></p> <p>The application should be executable from a command line using the <b><i>module-path</i></b> as follows:</p> <ul style="list-style-type: none"><li>• <b><i>Linux/Mac:</i></b> <b><i>java --module-path ./path/to/your/javafx-sdk-11.0.2/lib:/path/to/your/encog-3.4:/path/to/your/jfuzzylogic --module gmit.software/ie.gmit.sw.ai.Runner</i></b></li><li>• <b><i>Windows:</i></b> <b><i>java --module-path ./path/to/your/javafx-sdk-11.0.2/lib:/path/to/your/encog-3.4:/path/to/your/jfuzzylogic --module gmit.software/ie.gmit.sw.ai.Runner</i></b></li></ul> <p>You only need to use the module-path syntax above to check your application after creating the JAR. Change the parameter <b><i>/path/to/your/</i></b> to the real system path on your computer.</p>
resources	<p>A directory that contains <b><i>all the resources</i></b> required by the application. You can add to the existing set of subdirectories (<b><i>images, fuzzy, neural</i></b>) if necessary. All fuzzy</p>

	definitions should be stored in the <i>fuzzy</i> directory and neural network training and validation data in the <i>neural</i> directory.
<b>src</b>	A directory that contains the packaged <b>source code</b> for your application.
<b>README.pdf</b>	A <b>PDF</b> file detailing the main features of your application in <b>no more than 300 words</b> . All <b>features and their design rationale</b> must be fully documented.

### Marking Scheme

Marks for the project will be applied using the following criteria:

Element	Marks	Description
<i>Structure</i>	10	<b>All-or-nothing</b> . The packaging and deployment are both correct. The JAR file executes perfectly without any manual intervention when executed with the command <code>java --module-path ./path/to/your/javafx-sdk-11.0.2/lib:/path/to/your/encog-3.4:/path/to/your/jfuzzylogic --module gmit.software/ie.gmit.sw.ai.Runner</code> .
<i>README.pdf</i>	25	A <b>PDF</b> file detailing the purpose, design and implementation of the AI components of your application in <b>no more than 300 words</b> . All <b>features and their design rationale</b> must be fully documented.
<i>Fuzzy Logic</i>	25	Fuzzy logic controller(s) fully <b>implemented, integrated and documented</b> .
<i>Neural Network</i>	25	Neural network controller(s) fully <b>implemented, integrated and documented</b> .
<i>Extras</i>	15	Only <b>relevant AI extras</b> that have been fully documented in the README.

You should treat this assignment as a project specification. Any deviation from the requirements will result in some or all marks not being earned for a category. Each of the categories above will be scored using the following criteria:

Range	Expectation
<b>0-39%</b>	<b>Not delivering</b> on basic expectations. The submission does not meet the minimum requirements.
<b>40-60%</b>	Meets <b>basic</b> expectations. The minimum requirements are met in whole or in part.
<b>61-74%</b>	Tending to exceed expectations. A <b>high-quality</b> assignment with additional functionality added.
<b>75-89%</b>	Exceeding expectations and <b>very high-quality</b> . Any mark over 70% requires evidence of <b>independent learning</b> and cross-fertilization of ideas, i.e. your project must implement a set of features using advanced concepts not covered in depth during the module.
<b>90-100%</b>	<b>Exemplary</b> . Your assignment can be used as a teaching aid with little or no editing.