GALWAY-MAYO INSTITUTE OF TECHNOLOGY

Department of Computing & Mathematics

B.Sc. Software Development – Artificial Intelligence (2016)

ASSIGNMENT DESCRIPTION & SCHEDULE

A Heuristically Informed Fuzzy Maze Challenge

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

1. Overview

The release of the ZX81 in 1981 was a milestone in home computing in Ireland and Britain. Designed by Sinclair Research as an inexpensive home computer for the general public, it provided over 1.5 million purchasers with the capability of both using and programming a computer with 64KB of memory. While some users, through curiosity or necessity, focused on the programming opportunities offered by such devices, the vast majority of people used their home computers as a gaming platform. Indeed, the early 1980s saw the home computer used for a number of pioneering techniques in gaming, including isometric projection (*Knight Lore*, 1984) and parallax scrolling (*Tir Na Nog*, 1984). Even hobbyist programmers could get in on the act, as home computers enabled their owners to program simple games. One such hobbyist-turned-professional was Don Priestley who programmed the 1982 game Mazogs for the ZX81. The game takes place in a randomly generated, scrolling maze, in which the player must find a gold bar, placed a minimum of 200 moves from the start position. Once the gold bar has been located, the player must then find a way out of the maze and avoid or kill the mazogs (spider-like creatures) that randomly patrol the maze. A web based version of the game is available from http://www.zx-gaming.co.uk/games/mazogs/default.htm. ***You should familiarise yourself with the gameplay of Mazogs before starting the assignment.***

2. Minimum Requirements

You are required to create a game based on the Mazogs theme with the set of features listed below. You a free to adapt or extend the theme in any way you like. For example, the story of Theseus and the Minotaur can easily be overlaid on the game. Please note that the whole objective of this assignment is to reinforce your understanding of search algorithms and fuzzy logic. As such, the assignment will be an excellent revision and preparation for your written exam. You will be rewarded for using a cocktail of different algorithms in the game.

- The game should **generate a random 60x60 maze** with one or more exits. The player character should be placed in the maze as far from the exit(s) as possible. The objective of the game should be to find a way out of the maze.

- The game should be **graphical** and use arrow keys and other options to move around.

- The maze should be patrolled by spiders or other **characters that should be threaded**, i.e. the creatures should move independently through the maze. Each character should use one or more traversal / search algorithms to control its movements.

- A number of **weapons** should be available in the maze. Users should be able to select from different types of weapons, e.g. a sword, grenade or even an A-bomb (the latter two could use a depth-limited DFS to destroy everything within a limit of *n* moves). You could even use a depth-limited DFS as a "radar" to show where "life forms" exist nearby.

- A number of prisoners or something similar should also be available in the maze. Interacting with one of these should fire a **heuristically informed algorithm** that displays a path through the maze for a limited amount of time or steps.

This assignment requires that you use the JFuzzyLogic API to handle how the characters in the game interact. You should specify a set of linguistic variables, terms and membership functions in Fuzzy Control Language (FCL) that are capable of fuzzifying the input variables provided by methods in your code. For example, when you encounter a creature in the maze, the following method may execute (in this case, when the players "life force" reaches zero, they are dead):

```
public boolean fight(Weapon weapon, Opponent opponent){
  fis.setVariable("weapon", weapon.getPower());
  fis.setVariable("opponent", opponent.getStrength());
  fis.evaluate();
  float victory = fis.getVariable("victory").getValue());
  this.getLifeForce() = this.getLifeForce() * victory;
}
```

The statement fis.evaluate() might cause the following fuzzy rule to execute: **IF weapon IS powerful AND opponent IS weak THEN victory IS high**. Pay particular attention to the membership functions and terms used. The de-fuzzification of values should also be specified in the FCL document, along with the set of fuzzy rules. Use the JFuzzyLogic API to compute a result based on the fuzzified inputs and rules. NB:You are required to document and explain your choices of terms and membership functions for each linguistic variable as comments in the FCL document. You are also required to comment each fuzzy rule.

Note: you are free to asset-strip any of the resources, including labs and source code, available on the Moodle page for this module.

3. Deployment and Delivery

The project must be submitted by midnight on ***Sunday 3rd April 2016***. The project must be submitted in the form of a single README.txt file using the Moodle upload utility. You can find the area to upload the project under the heading *"A Heuristically Informed Fuzzy Maze Challenge"* in the "Notices and Assignments" section of Moodle. The README.txt file should contain the GitHub URL of the project, the main features you want to draw my attention to and a set of instructions for running the programme. ***Note: it is your responsibility to ensure that the instructions for running the programme are succinct, clear, platform-agnostic and correct.***

4. Marking Scheme

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

Marks	Category
A (40%)	Robustness (GitHub, project compiles, resources deploy with no manual intervention)
B (20%)	Fuzzy Logic
C (20%)	Heuristics and Search Algorithms
D (20%)	Additional Features

Each of the categories above will be scored using the following criteria:

- 0–30%: Not delivering on basic expectation
- 40-50%: Meeting basic expectation
- 60–70%: Tending to exceed expectation
- 80-90%: Exceeding expectations
- 90-100%: Exemplary