NEA 01 – Project Proposal

AQA Guidance: <https://goo.gl/jR6psY>

1. **Who is going to be your end user?**

Greg is a puzzle website owner and has asked for a proof of concept for a maze generation system for a section of his website.

1. **Define the problem you are going to solve.**

The system must be a client-server standalone network for now to see how the network function could be added to the website. A windows forms client and a console server should work. The client should be able to select a type of maze and choose whether it be solved, or they can solve it, before it is sent to the server and generated/solved.

1. **How will you research the problem?**

C#’s networking documentation provides ample examples for creating a client-server connection.

Many websites online compile a list of maze types with simple yet vague instructions on how to generate them, as well as descriptions of pathfinding algorithms.

1. **Provide background information so that a third party can understand what you are doing.**

The proposed system will primarily focus on the generation and solving of mazes. Users will be able to select from various maze types, such as perfect mazes, labyrinth-style mazes, and binary tree mazes. They will also have the option to choose whether they want to generate a solved maze or solve it themselves. The size of the mazes can be customized. Furthermore, users can choose the destination of the maze: whether it is a side exit or the centre.

To achieve this functionality, a client-server network architecture will be utilized. The client component will feature a user-friendly interface, enabling users to interact with the system, select maze parameters, and view the generated mazes. On the other hand, the server component will function as a console-based application. It will facilitate communication with multiple clients, handle maze generation requests, and maintain a statistics database to monitor user preferences and trends.

1. **Provide a list of appropriate, numbered, measurable objectives which cover all aspects of the final product and its functionality.**
2. Client frontend

* Login/registration system connected to a database. (potential)
* Auto search for server + manual port select.
* Maze selection interface
  + Must include perfect, labyrinth, binary tree as a minimum.
* Solved/unsolved selector.
* Size selection. (must overcome odd number restriction of square mazes)
* Destination selection. (side exit or centre)
* Display which shows generated maze to user.
* Functionality to play unsolved maze post-generation.
  + Buttons on app as well as keyboard controls to increase accessibility.
* Nice graphics (potential)

1. Client backend

* Parse settings to JSON/XML/other format for carrying data/objects.
* Send/receive data across the network.

1. Server

* Console-based server.
* Asks which port to operate across on start.
* Queue for multiple clients accessing webservices.
* Send/receive data from client.
* Parameterised webservices which generate user-defined mazes.
* Statistics database for viewing data trends. (maybe with separate client section with windows forms tables)

1. **List the technical skills from the guidance that your project requires.**

* Database (for potential login system)
* Parameterised SQL
* List operations
* Linked list maintenance
* Stack/Queue operations
* Recursive algorithms
* Complex user-defined algorithms
* Complex user-defined use of object-orientated programming
* Complex client-server model