

## Aamana Afzal

aamanaafzal@outlook.com | +447951034128 | github.com/aamanaafz

### EDUCATION

<b>The University of Manchester: BSc (Hons) Computer Science</b> <b>Predicted:</b> Upper Second class (2:1)	<b>Sept 2021 – June 2025</b>
<b>Loreto Sixth Form College</b> <b>A-Levels:</b> Mathematics (A*), Computer Science (A), Physics (B)	<b>Sept 2019 – June 2021</b>
<b>Manchester Islamic High School for Girls</b> <b>GCSEs:</b> Nine 9-4 including English (6), Mathematics (7), and Science (7-7)	<b>Sept 2014 – June 2019</b>

### SOFTWARE PROJECTS

#### Stendhal MORPG Java Team Project:

- In a team of 7, we assigned issues in the Stendhal open-source codebase. Fixed bugs created robust tests, and utilised Git techniques for version control. Developed JUnit tests in Eclipse IDE, efficiently fixing issues in dedicated feature branches, ensuring smooth codebase integration.

#### Matrix Implementation in C

- Developed a matrix processing library in C, implementing a wide range of matrix operations. Incorporated file handling techniques to allow matrices to be loaded and saved, as well as testing my code against a test suite.

#### Hypothesis Testing and Statistical Analysis using Python:

- Leveraged the Panda's library and Jupyter Notebook to conduct comprehensive data analysis and exploration. Proficiently read data from CSV files using the Pandas library and utilised SciPy and Matplotlib for data visualisation. Applied the Monte Carlo bootstrapping method to test my hypothesis and draw a meaningful conclusion.

#### Spam Filter using the Naïve Bayes classifier in Python:

- Implemented a Naïve Bayes spam filter for emails, using Panda's library and Jupyter Notebook. Conducted thorough data cleaning, exploration, and employed a train-test split to optimise model performance. Applied machine learning to identify spam accurately.

#### K-Maps Website

- As a group of five, we created a route finder for the Kilburn Building using HTML, CSS, and JavaScript. As part of the database team, I used PHP and SQL to store instructions and locations in a database.

#### Python Tkinter Game:

- Leveraged Tkinter in Python to craft an engaging Christmas-themed game, skilfully integrating elements like collision detection, a dynamic scoring system, an interactive leaderboard, and the functionality to pause, resume, save, and load game progress, enhancing both user experience and gameplay depth.

#### Java Selfish Game:

- Utilised Java and a provided class diagram to develop the "Selfish" game, incorporating packages, OOP, and interfaces.
- Documented code thoroughly using Javadoc, created 8 classes, and executed 599 JUnit tests to validate functionality, structure, and documentation compliance.

#### Python Caching Implementation:

- Applied my knowledge on caching strategies to implement diverse caching mechanisms using Python, encompassing methodologies like random, cyclic, and least recently used.

### VERILOG AND ASSEMBLY HARDWARE PROJECTS

#### Stump ALU (Simplified 16-bit Processor)

- Computed the result of ALU operations and the flags outputted to the condition code register using Questa. Implemented a function to check the overflow for the operations, using Verilog code.

#### Display Decoder

- Implemented a combinatorial circuit for a seven-segment display decoder that displayed numbers 0-9 and letters A-F.

#### Toucan Crossing

- Created a sequential circuit for controlling the light sequence at a pedestrian/cycle crossing on a circuit board.

#### MU0 (Manchester University 0) Datapath

- Used D-type Flip Flops to create registers for the accumulator, PC, and IR. Designed multiplexers for the Datapath schematic.