

Computer Games Development CW208

Software Functional Specification

Year IV

|  |  |
| --- | --- |
| [Student Name]  [Student Number] | Richard Fleming  C00241212 |
| [Date of Submission]  28/04/2022 | |

Contents

[Acknowledgements](#_heading=h.l9ckjdn52ht6) **2**

[Functional Specification](#_heading=h.30j0zll) **3**

[References](#_heading=h.3znysh7) **3**

# Acknowledgements

We would like to thank the following people who assisted in completing this project including;

Oisin Cawley who acted as my supervisor throughout the duration of the project, providing advice and guidance when needed.

I would also like to thank my fellow classmates for providing advice when needed.

# Functional Specification

The major functionalities of the software will be the game itself, the ANN and its training mechanisms, as well as player interaction for the Backpropagation training mechanism.

To be more specific, the game itself will be a slice of a very simple Infinite Runner that uses Megaman 3 aesthetics. There will be two obstacle types: An obstacle the player/ANN must jump over, and an obstacle they must pass under. The player/ANN will be on the far left and will only be able to jump. The obstacles will travel from the far right to the left. The obstacles chosen will be random.

Speaking of the ANN, it will be playing through the game using two training mechanisms; Reinforcement Learning and Backpropagation. In the case of Backpropagation, in order to gather the data necessary for the ANN, the game will have functionality in place for the user to play through the game, their inputs being saved as they play. These inputs are then given to the ANN. For Reinforcement Learning I plan to use the Q learning method to train the ANN. This will entail the creation of a Q table that the ANN uses to store which action to take in any given situation. Once the data has been gathered, I will have the ANN play through the game several times using each training method. Once the ANN has gone through several cycles of play, at the end I will compare how the ANN performed. I’ll be looking at how long the ANN manages to survive, as well as the time and resources necessary to reach that result.

As I implied earlier, the software will be written in Python, using the Tensorflow library and Keras API.

# References

**Book**

Sutton, R. and Barto, A. (2020). *Reinforcement Learning.* 2nd Edition. Cambridge, Massachusetts: The MIT Press

Russel, S. and Norvig P. (2010). *Artificial Intelligence: A Modern Approach.* 3rd Edition. Upper Saddle River, New Jersey: Prentice Hall