A logo for college computing

Description automatically generated

**Assessment Cover Page**

|  |  |
| --- | --- |
| *Student Full Name:* | Kevin Scully |
| *Student Number* | 24275 |
| *Module Title* | AI Concepts to Implementation |
| *Assessment Title* | CA1 |
| *Assessment Due Date* | 15th November 2024 |
| *Date of Submission* | 10th November 2024 |
|  |  |

**Declaration**

By submitting this assessment, I confirm that I have read the CCT policy on academic misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source.

I declare it to be my own work and that all material from third parties has been appropriately referenced.

I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Contents

[Introduction 3](#_Toc182330553)

[Task 1 – AI and Chess (500words) 3](#_Toc182330554)

[Task 2(b) – Computer Games and AI (500words) 4](#_Toc182330555)

[Task 3 – AIBO (500words) 4](#_Toc182330556)

[References 5](#_Toc182330557)

# Introduction

# Task 1 – AI and Chess

Artificial Intelligence (AI) has been inspired by chess for decades, sparked by the IBM’s Deep Blue programs one point victory over then chess world champion Garry Kasparov in 1997. (Campbell, et al., 2002) This section will analyse AI strategies that have been applied to chess gameplay by two chess engines, Stockfish and AlphaZero.

The first chess engine that will be analysed is Stockfish. This is regarded as a traditional chess engine in that it considers factors such as control of the centre and king safety before making a move. (Degni, 2023) Stockfish uses the strategy of an alpha-beta pruning algorithm to search for the best move and then Efficiently Updatable Neural Network (NNUE) to evaluate the move following the tree search.

The alpha-beta pruning search algorithm improves upon the minimax search by reducing the computational power and time to make a move. It does this by applying forward pruning and reduction which avoids variations that will not occur due to the game be re-directed by either player. (Sadmine, et al., 2023) Forward pruning removes scenarios that are unlikely to be used in optimal play. This does make it possible that the algorithm will mistakenly prune a line of optimal play early. Reduction is when the algorithm searches certain gameplay to a lower depth. The depth reaches increases when promising variations are found, but a short depth is reached when less promising variations are discovered. Additionally, as the game proceeds and the number of possibilities increases, the depth is shortened due to computational restrictions and time. (Maharaj, et al., 2022)

When the alpha-beta pruning search algorithm reaches a node, an evaluation is applied to decided whether to choose a White or Black position. NNUE is used to make this decision. This is trained to predict the output of the move by evaluating the piece position, piece activity and the game phase. This evaluation slows the result but makes up for it in a better outcome. It uses the position of the white and black king in comparison to the other pieces. It then considers the board position before coming to a final evaluation decision. (Eisma, et al., 2024)

The second chess engine that will be analysed is AlphaZero. This is an AI based chess engine that uses the strategy of the Monte Carlo Tree Search and Convolutional Neural Networks (CNNs) to play chess. It uses CNN’s to appraise positions and predict the best moves. The Monte Carlo Tree Search is used to predict the best next move by using its repeated learnings from previous games. Both the neural network and the Monte Carlo Tree Search are trained together, which means both the ability to examine positions, and its search capabilities are improved continuously. By using this strategy, AlphaZero continuously becomes a better chess engine. (Degni, 2023)

AlphaZero uses the Monte Carlo Tree Search algorithm to search for the best moves by using repeated sampling. The algorithm performs random sampling in the form of simulations and then remembers the statistics of moves made to make more educated choices in future games. The search tree starts at the root and works through the branches of the trees according to the node values. The tree is expanded on each move with the node value of the highest probability. Then the algorithm goes back through the tree, updating the probability values of previous nodes passed. After many samples are gathered, the child node with the highest number of samples is selected. (Świechowski, et al., 2023)

Deep CNNs are used by AlphaZero for evaluation of moves and to assign a value to nodes in the tree search. Statistics from millions of self-play chess matches, where AlphaZero plays both sides, is generated. The self-play removes the reliance on human experts and tendencies, letting AlphaZero correct its mistakes and develop game strategies. Following the outcome of a self-play game, the CNN is trained to minimise the loss. During the game the CNN aims to predict the game result and look ahead at future moves to learn probabilities for the tree search. (Maharaj, et al., 2022)

After only four hours of self-play, AlphaZero defeated Stockfish over 100 games, winning 28 and losing zero. (Degni, 2023)

*Research two Artificial Intelligence (AI) strategies that have been used to play chess. Describe the theory and concepts used within these strategies and how they relate to AI.*

# Task 2(b) – Computer Games and AI

AI is essential in modern gaming. The goal for any game is for the characters of the game to be human-like. To achieve this, characters need to go beyond the scripted interactions, and instead be responsive, adaptive and intelligent. They must learn about players during the game, then use their learnings to adapt their behaviours to interact with the player to provide a richer, more realistic experience. (Ram, et al., 2007) This report will focus on two AI strategies used within modern computer games and the discuss the algorithm used to achieve this strategy.

The first AI strategy for computer games is pathfinding. At a high level, this refers to the shortest route between two points. This is important in role-playing and strategy computer games where characters must over-come obstacles and move from their current location to a pre-determined or played determined destination. (Cui & Shi, 2011)

The A\* Algorithm is a popular algorithm for pathfinding within computer games. It is a best first search algorithm that uses a two-dimensional grid to find the shortest path between two points. Before using A\*, the problem must be abstracted to a two-dimensional grid. It maintains two queues, an open and closed queue. The open queue contains a list of nodes that have not yet been searched but will be searched next. When the node is searched, it is moved to the closed queue as it does not need to be searched again. A heuristic value is applied to all nodes, this is calculated as the cost of traveling from the start node to the current node plus the estimated cost of travel. The path chosen will be based on the lowest heuristic values. (Liu, 2023)

A\* Algorithm is used in the classic real-time strategy game ‘Age of Empires’. It is applied to move the military through a 256x256 grid with 65,536 possible locations. However, problems persist in the game play when obstacles such as forest become more complex, the military can become stuck in them. Cui & Shi (2011) found that more efficient pathfinding solutions are required for solving the increasing complexity of the pathfinding solutions of the modern game.

The second strategy used by video gaming companies to enhance the gaming experience is enhanced dialogue. This is done through a type of AI called Natural Language Processing (NLP). This significantly enhances responses, dialogues and personalities of non-playing characters within the game. This can allow for engaging dialogue, where responses can alter the outcome of the games story. (Picca, et al., 2015)

A common algorithm used for NLP is the Latent Semantic Analysis (LSA). This is an unsupervised AI model that is fully automatic. This model attributes words according to their contextual distributions across large volumes of text. It uses the context in which words appear and do not appear to determine meaning. After analysing the text, LSA creates a matrix of words and documents. In the matrix each row is a word, and each document is a column. It then reduces the number of dimensions in the matrix by applying Singular Value Decomposition (SVD) which determines the similarity between words using a scale of zero to one. (Miaskiewicz, et al., 2008)

A real example of NLP being used in a video-game is ‘Diablo III’ to form an interactive game that can result in different outcomes following dialogues with non-playing characters. (Varaksina & Dyshuk, 2024)

*AI has been essential in computer games since the 1960s. Provide an example of two AI strategies that are used in modern games. You should identify the algorithm being used, provide a high-level description of the algorithm and provide an example of a game where this algorithm has been implemented.*

# Task 3 – AIBO (500words)

*SONY has developed a robot puppy named AIBO. You can find information about this robot at https://us.aibo.com/. a) Discuss the AI characteristics that you think the AIBO exhibits. [15 marks] b) Discuss AI characteristics that you think it still lacks. [15 marks]*

# References

Campbell, M., Hoane Jr, A. & Hsu, F., 2002. Deep Blue. *Artificial Intelligence,* 134(1-2), pp. 57-83.

Degni, R., 2023. *The Ultimate Checkmate: AI and Chess Engines.* [Online]   
Available at: https://www.codemotion.com/magazine/ai-ml/the-ultimate-checkmate-ai-and-chess-engines/#:~:text=It%20uses%20a%20combination%20of,games%20from%20a%20given%20position.  
[Accessed 10 November 2024].

Eisma, Y., Koerts, R. & de Winter, J., 2024. Turing tests in chess: An experiment revealing the role of human subjectivity. *Computers in Human Behavior Reports,* Volume 16, p. 100496.

Maharaj, S., Polson, N. & Turk, A., 2022. Chess AI: competing paradigms for machine intelligence. *Entropy,* 24(4), pp. 1-13.

Ram, A., Ontañón, S. & Mehta, M., 2007. Artificial Intelligence for Adaptive Computer Games. *Cognitive Computing Lab,* pp. 22-29.

Sadmine, Q., Husna, A. & Müller, M., 2023. Stockfish or Leela Chess Zero? A Comparison Against Endgame Tablebases. *In Advances in Computer Games,* pp. 26-35.

Świechowski, M., Godlewski, K., Sawicki, B. & Mańdziuk, J., 2023. Monte Carlo tree search: A review of recent modifications and applications. *Artificial Intelligence Review,* 56(3), pp. 2497-2562.