A logo for college computing

Description automatically generated

**Assessment Cover Page**

|  |  |
| --- | --- |
| *Student Full Name* | Matthew Healy |
| *Student Number* | sba24213 |
| *Module Title* | AI Concepts to Implementation |
| *Assessment Title* | CA1 |
| *Assessment Due Date* | 10 / 11 / 2024 |
| *Date of Submission* | 09 / 11 / 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

[**Question 1** 1](#_Toc182143447)

[Deep Blue, Minimax and Alpha-Beta Pruning 3](#_Toc182143448)

[AlphaZero, Reinforcement Learning and the Monte Carlo Tree Search (MCTS) 5](#_Toc182143449)

[**Question 2** 7](#_Toc182143450)

[**Note: Please refer to the IPYNB file submitted for the solutions to part (a)** 7](#_Toc182143451)

[AI use cases within computer games 7](#_Toc182143452)

[*NPC Pathfinding with the A\* Search Algorithm* 7](#_Toc182143453)

[*Finite State Machines* 7](#_Toc182143454)

[**Question 3** 8](#_Toc182143455)

[AI Characteristics Exhibited by AIBO 8](#_Toc182143456)

[AI Characteristics Not Exhibited by AIBO 8](#_Toc182143457)

[Bibliography 9](#_Toc182143458)

# **Question 1**

Chess, an ancient and deeply complex game of strategy originating in India from the 6th century CE (Encyclopaedia Britannica, 2024), has proven to be one of the greatest tests of one’s ability to intuitively solve problems. The ability to formulate a winning strategy through solid opening moves, and spontaneously adapt said strategy based on insightful positional analyses, devise and implement creative tactics, and the inevitable and effective conversion of advantage to victory during the endgame stage of a given play (Soltis, 2024) is a testament to one’s intelligence in multiple respects.

As such, the performance of AI chess engines throughout the history of the discipline has been highly reflective of the sophistication of the technology. Computer scientists have been attempting to tackle the problem of ‘solving’ the game of chess since the 1930’s; with an estimated 1040 number of possible legal plays (Kiernan, 2024), this problem is still yet to be solved by computers or humans alike. However, since IBM’s 1997 defeat of chess grand master Garry Kasparov with Deep Blue (Perier, 2021), AI has overtaken humans in this field, and only continues to develop and improve. Birthing an important sentiment; if AI can overtake humans in a highly skilled and specialised field such as chess, could the technology also be capable of outperforming humans in any respect (Thompson, 2022)?

Before delving into the specific algorithms that power chess engines, the fundamental concepts of a chess AI must be explored:

Machines are incapable of directly comprehending natural language and alphabetic characters, and as such, key data points within this context must be converted to numerical values. In this instance, the different chess pieces used within a play and their respective numerical values of which represent the potential each piece has to introduce tactical advantage (Perier, 2021).

*The hierarchy of pieces represented by numerical values in this context is as follows:*

* **Queen : 9**
* **Rook : 5**
* **Bishop : 3**
* **Knight : 3**
* **Pawn : 1**

*With this, fundamental concepts of chess theory are integrated within the design of such chess engines notably including but not limited to (Perier, 2021):*

* **King Safety : How defended / exposed the king is**
* **Centre Control : Which player has the most tactical advantage within the 4 center rows**
* **Piece Development : How many pieces have been moved from starting position**
* **Material Imbalances : Which player holds the highest cumulative chess piece value**
* **Pawn Formation : The degree of complexity in which pawns are structured**

Evaluation functions are then used within chess engines to output an optimal move based on the computation of these variables with regards to the current state of the board, the current stage of play, and the optimal long-term strategies that can be followed from this given point (Chessify, 2023). While all chess engines feature similar modes of evaluating and computing future moves, the methods of searching for and computing such strategies have varied and evolved over the years.

## Deep Blue, Minimax and Alpha-Beta Pruning

The first chess engine to successfully defeat a grandmaster. Developed by IBM, Deep Blue was incomprehensibly powerful for its time. Utilising 32 processors, it was capable of evaluating 200 million chess positions per second and boasted 11.38 billion flops of processing speed (IBM, 2024).

Deep Blue pioneered the use of AI Minimax algorithms with Alpha-Beta Pruning to optimise chess strategies derived by machines.

To thoroughly understand Alpha-Beta Pruning, the Minimax algorithm must first be explored. The Minimax algorithm within the context of chess engine AI, is used to adopt a flexible style of play which prioritizes defence and maintaining material advantage in the endgame stage. The algorithm explores potential long-term strategies, and computes these to identify which tactics ‘minimise the maximum potential loss from a particular move’ (Piech, 2012).

*For instance, a practical implementation of this algorithm would resemble the following (Stanford, 2009):*

* **The algorithm analyses every move-set to a specified depth level (in the context of chess engines, the algorithm will determine every position possible within a given number of moves, as analysing every position from opening to endgame within chess would not be possible due to there being more potential plays within chess than there are atoms in the known universe (Kiernan, 2024.))**
* **Material imbalance could be used to evaluate which player holds an advantage at a given point or depth level. For this example, the AI player will be represented by black.**
* **To determine whether white (human) is currently at an advantage or disadvantage, the material value of black (AI) at a given point will be subtracted by that of white (human).**
* **A positive result would indicate an advantage towards white (human), with a negative indicating advantage towards black (AI), and the distance from 0 indicating the extent of such advantage.**
* **The algorithm assumes at any given point white (human) opting for the move resulting in the highest possible value, whereas black (AI) will opt for that resulting in the lowest possible value.**
* **This results in the algorithm returning an optimal move wherein the best possible response from the opponent (in most instances), would still leave the AI player in a more advantageous position materially speaking.**

One clear drawback to this method, however, is that as the number of neighbouring nodes within the decision tree grows exponentially at each depth, evaluating each move-set becomes more and more computationally expensive (Kang, et al., 2019), limiting how far ahead the chess engine AI can strategise. This is where Alpha-Beta Pruning is utilised by Deep Blue to further optimise the Minimax algorithm. Alpha-Beta Pruning incorporates heuristics alpha and beta which contain the maximum and minimum values respectively to avoid searching through branches containing sub-optimal move-sets.

The algorithm utilises these heuristics by storing the most current minimum and maximum values to guide the search, if a branch within a minimum depth is found to be greater than or equal to the current minimum value stored in alpha, it will be ‘pruned’ and disregarded by the search. Similarly, the maximum value within a given depth will be stored in beta and if the currently examined node within a maximum depth is deemed to be less than or equal to beta, it will be ‘pruned’, and that particular path will not be explored further (Mendoza, 2022).

## AlphaZero, Reinforcement Learning and the Monte Carlo Tree Search (MCTS)

AlphaZero differs from Deep Blue in terms of how it utilises the MCTS to simulate random plays from a given position rather than exhaustively processing every possible move. AlphaZero uses MCTS in conjunction with neural networks to behave more humanly and perform in a far more efficient and adaptive manner (Degni, 2023)...

* **Two networks are utilised within AlphaZero, a policy network which decides on the chess engines next move, and a value network which evaluates which player holds the advantage and thus is more likely to win (McGrath, et al., 2021).**
* **The MCTS is used to randomly simulate plays, the tree follows a sequence wherein a player state or depth is followed by an opponent state (McGrath et al., 2021).**
* **The value neural network evaluates and scores these plays on a scale of -1 to 1, with values closer to 1 indicating a player win and values closer to -1 indicating an opponent win (McGrath et al., 2021).**
* **The MCTS is run until a termination point is reached (i.e. a win or a loss), storing each position or node in sequential order, and this particular play is then returned once the termination point is reached and is input to the value network (McGrath et al., 2021).**
* **The value network at this point will assign a score to a given play, within the range -1 to 1 as previously mentioned. This score will be assigned to every state within the given play for both player 1 and player 2, with player 1 states being assigned the positive value and player 2 states being assigned a negative (i.e. 1, -1). These values are reversed at each depth, as the perspective within MCTS shifts at each state. For instance, a player state will be deemed positive at a depth representing the AI players perspective, whereas the player state will be deemed negative at a depth representing the human opponents perspective (McGrath et al., 2021).**
* **A vast quantity of these simulations will be run, after which states from random points within these plays will be input to the value network to test its evaluation capabilities. The value network will return a score which may or may not need to be fine-tuned to the actual score of a given position. For example, if a position taken from a losing play is scored highly by the value network, the value network will at this point be informed of the actual outcome score of this play (McGrath et al., 2021).**
* **This process will be repeated until the value network displays an understanding of what solid positioning resembles on a given board.**
* **At this point, the policy network will be evaluated by inputting random positions from game states, the policy network will suggest which moves result in optimal outcomes and express this suggestion with a percentage value. Ideally, it would suggest moves resulting in an opponent checkmate with a value of 100 percent, and moves resulting in an exposed king with a far lower percentage (McGrath et al., 2021).**
* **The policy network is then trained using the MCTS, after which the MCTS is then guided by any insights discovered by the policy network. This codependency means that as the policy network improves, more optimal paths are found within the MCTS, and as more optimal paths are found within the MCTS, the further the policy network is optimised itself (McGrath et al., 2021).**
* **AlphaZero used this methodology to train itself to play chess without any domain specific knowledge or feature engineering required. It took only 4 hours of training before the chess engine was capable of defeating Stockfish (Degni, 2023), which was at the time, considered the pinnacle of sophistication among AI chess engines (McGrath et al., 2021).**

This approach to designing a chess engine results in far stronger plays output, which resemble that seen from human players. The nature of how AlphaZero learns also makes a far more optimal mode of developing a chess engine to account for slight variations in rules for a given game (Tomašev, et al., 2020). The engine is capable of developing more creative strategies, and adapting better to the playstyle of a particular opponent. While AlphaZero opted to fully and exclusively utilise neural networks and game simulations in its approach, it has been explored that a hybrid approach wherein a MCTS algorithm designed to incorporate traditional Minimax and Alpha-Beta pruning design principles alongside reinforcement learning methodologies, would be the optimal scenario (Gao, Muller and Hayward, 2018). AlphaZero pioneered the use of neural networks within the design of chess engines, with engines like Stockfish (Goucher, 2021) and Leela Chess Zero (LCZero, 2022) now incorporating such practices within their own design. Much in how Deep Blue marked an important milestone wherein the capabilities of AI in general were explored as a result of the technologies successful integration within the field of chess, AlphaZero also raised an important point: if neural networks can be used to create an engine which masters such a highly skill-based discipline such as chess in a matter of hours, it is worth exploring what other capabilities and implementations this AI concept has.

# **Question 2**

## **Note: Please refer to the IPYNB file submitted for the solutions to part (a)**

## AI use cases within computer games

AI has been the backbone of modern videogames, utilised in a myriad of ways, a primary use case seen within the curation of immersive experiences through integration of responsive non-player characters (NPCs) (Tupe, et al., 2016). While machine learning has become a hugely popular subject within the development of AI systems, the videogame industry continues to favour the integration of expert systems. This is largely a result of a few factors:

* **Most triple A titles follow a story and as such, the environments, events and NPCs within a game need to behave with a degree of predictability in order to retain continuity within a given title’s concept, as such, the level of precision in AI design that comes with the integration of expert systems is far more preferable for this application (Council, 2018).**
* **Many industry-standard frameworks and game engines for example Unity (Makinde, 2023), don’t support the integration of machine learning systems within NPCs, and instead support the integration of Finite State Machines and Behaviour Trees for NPC behaviour and search algorithm powered AI pathfinding systems.**
* **Expert systems also typically favour the requirements of game developers, allowing for faster refinement of NPC behaviours, whereas a machine learning approach would require regular prototyping and refinement of training systems to fine tune the performance of a given NPC. In addition, machine learning is a highly specialised field, meaning that the lowered entry barrier for instead working with expert systems is a far more attractive option for game developers (AI and Games, 2023).**
* **With all this in mind, the integration of more traditional expert systems is widely speaking a far more attractive option for game developers.**

Two notable examples of which being the use of AI pathfinding algorithms (the most popular of which being the A\* heuristic search algorithm) to enable realistic and seamless NPC traversal through a game environment (Rafiq, et al., 2021), and the incorporation of Finite State Machines and Behaviour Trees to dictate how NPCs interact with each-other and a human player (Filipović, 2023).

### *NPC Pathfinding Through a Navigation Mesh with the A\* Search Algorithm*

There are two primary components required for an NPC to find paths within an environment (Graham, 2003):

1. **Navigation Mesh**

A navigation mesh is a map, usually comprised of polygons, which allow NPCs to interpret their environment. Polygons represent different surfaces within a three-dimensional environment. These surfaces contain information which inform NPC AI as to whether they can cross that particular surface (Graham, 2003), for example, within Nintendo’s ‘The Legend of Zelda: Tears of the Kingdom’, some NPCs can only cross polygons classed as water, others may only be able to cross polygons classed as solid ground, and some NPCs can cross both.

1. **Search Algorithm**

Enter search algorithms for pathfinding: each polygon has nodes and edges associated with it. If an enemy NPC that cannot cross water polygons wishes to travel towards the player who is across said body of water, they will initialise an A\* heuristic search to find the shortest path that satisfies these conditions (Morina and Rafuna, 2023). The algorithm will attempt to find a shortest path which leads to the player, while avoiding untraversable polygons within the navigation mesh. While utilising A\*, it will in this instance, utilise the straight-line distance between itself and the player as a heuristic to find the lowest cost path, ignoring those that cross untraversable polygons.

### *NPC Behaviour and Behaviour Trees*

Behaviour Trees were first popularised and pioneered by Bungie within ‘Halo 2’, they take the form of a tree data structure which functions by utilising composite parent nodes, and child nodes which in turn dictate a given behaviour. These composite nodes are divided into two primary types (Tupe, et al., 2016):

1. **Selector:**

A selector node decides on executing an appropriate behaviour based on a particular input, for example within Halo 2, Grunt enemy types are programmed to flee from combat if high ranking Elite enemy types are killed by the player. In the context of a behaviour tree, if the player character is in sight and an Elite enemy type is present, the selector node within the Grunt’s behaviour tree will select an attack player behaviour, however if the Elite NPC is killed, the behaviour tree will then reevaluate the environment and decide to execute a flee behaviour.

1. **Sequence:**

These nodes execute multiple behaviours in sequence, using friendly NPCs as an example, if the player character enters a vehicle with a vacant turret attached to it, it will execute a sequence node which tells the NPC to run towards the vehicle, then enter the turret, and then target any enemy NPCs they may come across.

Another key functionality within behaviour trees are blackboards, these enable the storage of important data points as they are discovered to eliminate unnecessary backtracking or additional branches within the tree structure. They can also enable communication between NPCs to further optimise their performance. For example, if one NPC in ‘Halo 2’ spots the player character, every other NPC which shares that same blackboard is informed of the players location, and it allows NPC AI to strategise and engage with the player in a more realistic and immersive way (Unreal Engine, 2024).

In conclusion, while there have been countless advancements in the field of machine learning in recent years, it is not an all-encompassing field of study and there are still many instances wherein the application of more traditional AI expert systems is a more attractive and efficient solution to a given problem domain (Council, 2018). That isn’t to say machine learning will never be utilised within these fields however, but rather that machine learning solutions won’t necessarily replace expert systems entirely and will instead be used in conjunction with them to maximise their capabilities. With the videogame industry as an example, machine learning is slowly becoming more integrated within the design of videogames to curate more customised and immersive user experiences through the incorporation of machine learning methodologies within expert systems such as Finite State Machines, Decision Trees and pathfinding techniques (Chen, 2023).

# **Question 3**

The recent continuation of the previously discontinued AIBO line by Sony has made significant waves in the field of artificially intelligent robots. Functioning as a social entertainment AI, AIBO is designed to realistically emulate the characteristics of a living pet puppy. While it features many advanced implementations of AI concepts such as advanced facial recognition and the ability to not only discern between different people, but behave differently around separate individuals depending on how their relationship develops. In addition to this, AIBO is also utilises advanced AI pathfinding techniques and exhibits an ability to recognise and seamlessly adapt to its environment (Sony, 2024).

## AI Characteristics Exhibited by AIBO

AIBO exhibits many advanced AI characteristics, the most notable of which including:

* **Advanced Facial Recognition:**

AIBO utilises sophisticated TOF sensors in tandem with deep neural networks and reinforcement techniques for highly accurate facial recognition (TOF Sensors 2024).

* **Reinforcement Learning and Pathfinding:**

Pathfinding for robotics is a significant hurdle. The ability to map out an unseen environment, identify obstacles and create an optimal path from a start position to a goal requires the implementation of sensors and reinforcement learning techniques. As discussed previously, reinforcement learning utilises neural networks and works in a cycle wherein an action is taken, and a reward is given depending on how effective said action was. The AI then learns from this and identifies said action as optimal. As this process continues the AI in question, in this instance AIBO, gradually becomes more competent within said task, which is in this case navigating its environment. With the example of social robots, AIBO utilises proximity sensors to map out a space and triangulate its own position within such space (Perez et al., 2010).

* **Reinforcement Learning and Personality**

AIBO also utilises reinforcement learning techniques to develop certain unique characteristics depending on how it is nurtured. If AIBO is played with frequently it will become more playful and dependant, however if it is typically ignored it will become more self sufficient in terms of entertaining itself. This is also promoted through integration of neural networks surrounding behaviour. AIBO begins with a small number of basic behaviours and actions. It is also programmed with an intrinsic interest in certain ‘events’, a reward will be granted to its AI should a certain event occur, and the more the event occurs the more AIBO’s typical response behaviour to such an event develops. This is known as an IMRL algorithm (Intrinsically Motivated Reinforcement Learning). This enables AIBO to overtime develop hierarchical skills (Soni and Singh, 2006).

## AI Characteristics Not Exhibited by AIBO

With this in mind, there are still certain AI characteristics AIBO lacks, such as:

* **Natural Language Processing:**

The discipline of combining advanced computational linguistics with machine learning to enable AI systems to process and understand human expression and return responses mimicking that of a human to a high degree of realism (IBM, 2024). AIBO has capabilities to respond to certain prompts, however it lacks advanced natural language processing capabilities and cannot truly understand and organically respond to commands or human (natural) language.

* **Emotional Intelligence:**

AIBO lacks true nuanced emotional intelligence. While it is capable of mimicking the emotions seen in a typical pet animal, these are merely the result of algorithms coupled with reinforcement learning techniques to mimic the development of personality and emotional responses seen among real puppies. While advanced AI methodologies are developing to mimic such emotional traits and responses with a high degree of realism (Power, 2024), AIBO lacks these advanced capabilities.

In conclusion, AIBO is an incredibly advanced social robot marking an important historical milestone primarily within the field of designing robotic agents powered by advanced AI methodologies which have the capability to seamlessly and intuitively traverse and interact with its environment, and the humans within it. AIBO excels at learning traversal and adapting to a physical environment, however it has much room for growth with regards to behaving like an organic puppy with regards to closely and realistically mimicking emotions, and processing and responding to natural languages.

# Bibliography

**AI and Games, 2023.** *How Machine Learning is Transforming the Video Games Industry*. YouTube. Available at: <https://www.youtube.com/watch?v=dm_yY-hddvE> [Accessed 10 November 2024].

**Soni, V. and Singh, S. (2006).** *Reinforcement Learning of Hierarchical Skills on the Sony Aibo robot*. Available at: <https://web.eecs.umich.edu/~baveja/Papers/icdl06.pdf> (Accessed: 10 November 2024).

**Chen, Y., 2023.** *Game designs based on machine learning systems.* Higher Education Studies, [online] Available at: <https://drpress.org/ojs/index.php/HSET/article/view/6652/6447> [Accessed 10 November 2024].

**Chessify, 2023.** *Behind the Numbers: Understanding Chess Engine Evaluations*. Available at: <https://chessify.me/blog/chess-engine-evaluation#:~:text=The%20concept%20of%20evaluation%20numbers,which%20side%20has%20an%20advantage> [Accessed 9 November 2024].

**Council, G., 2018.** *Machine learning AI vs expert systems AI*. Parascript. Available at: <https://www.parascript.com/blog/machine-learning-ai-vs-expert-systems-ai/#:~:text=The%20reason%20why%20results%20will,machine%20learning%20to%20be%20removed> [Accessed 10 November 2024].

**Degni, R., 2023.** *The ultimate checkmate: AI and chess engines*. Codemotion. Available at: <https://www.codemotion.com/magazine/ai-ml/the-ultimate-checkmate-ai-and-chess-engines/> [Accessed 9 November 2024].

**Encyclopaedia Britannica, 2024.** *Where did chess originate?.* Available at: https://www.britannica.com/question/Where-did-chess-originate#:~:text=Chess%20first%20appeared%20in%20India,a%20singular%20piece%20(king) [Accessed 9 November 2024].

**Filipović, A., 2023*.*** *The role of artificial intelligence in video game development*. Kultura polisa. Available at: <https://www.researchgate.net/publication/375488989_The_Role_of_Artificial_Intelligence_in_Video_Game_Development> [Accessed 10 November 2024].

**Gao, C., Muller, M. and Hayward, R., 2018.** *Three-Head Neural Network Architecture for Monte Carlo Tree Search*. Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI), pp. 523-529. Available at: <https://www.ijcai.org/proceedings/2018/0523.pdf> [Accessed 10 November 2024].

**Goucher, A.P., 2021**. *The Neural Network of the Stockfish Chess Engine*. Available at: <https://cp4space.hatsya.com/2021/01/08/the-neural-network-of-the-stockfish-chess-engine/> [Accessed 10 November 2024].

**Graham, R., 2003***. Pathfinding and artificial intelligence*. The ITB Journal, 4(2). Available at: <https://arrow.tudublin.ie/cgi/viewcontent.cgi?article=1063&context=itbj> [Accessed 10 November 2024].

**IBM, 2024.** *What is NLP (natural language processing)?* Available at: <https://www.ibm.com/topics/natural-language-processing> (Accessed: 10 November 2024).

**IBM, 2024.** *Deep Blue: IBM’s computer checkmated a human chess champion in a computing tour de force*. Available at: <https://www.ibm.com/history/deep-blue> [Accessed 9 November 2024].

**Kang, X., Wang, Y. and Hu, Y., 2019.** *Minimax search algorithm with alpha-beta pruning: A comprehensive study and applications in artificial intelligence*. Journal of Intelligent Learning Systems and Applications, 11(2), pp. 109-118. Available at: <https://www.scirp.org/journal/paperinformation?paperid=90972> [Accessed 10 November 2024].

**Kiernan, P., 2024.** *Which is the greater number: the number of atoms in the universe or the number of chess moves?* Liverpool Museums. Available at: <https://www.liverpoolmuseums.org.uk/stories/which-greater-number-of-atoms-universe-or-number-of-chess-moves> [Accessed 9 November 2024].

**LCZero, 2022.** *Neural Network Topology*. Available at: <https://lczero.org/dev/backend/nn/> [Accessed 10 November 2024].

**Makinde, D.C., 2023.** *UnityEngine’s AI Pathfinding: Exploring the artificial intelligent side of the Unity engine*. Medium. Available at: <https://medium.com/our-internship-journey/unityengines-ai-pathfinding-exploring-the-artificial-intelligent-side-of-the-unity-engine-1404a3ab76f5#:~:text=Pathfinding%20Algorithms%3A,and%20then%20discards%20the%20rest> [Accessed 10 November 2024].

**McGrath, T., Kapishnikov, A., Tomašev, N., Pearce, A., Hassabis, D., Kim, B., Paquet, U. and Kramnik, V., 2021.** *Acquisition of chess knowledge in AlphaZero*. DeepMind. Available at: <https://arxiv.org/pdf/2111.09259> [Accessed 9 November 2024].

**Mendoza, P., 2022.** *Alpha-Beta Pruning Algorithm: The Intelligence Behind Strategy Games*. School of Mathematical and Computer Sciences, Yachay Tech University. Available at: <https://www.researchgate.net/publication/360872512_Alpha-Beta_Pruning_Algorithm_The_Intelligence_Behind_Strategy_Games> [Accessed 9 November 2024].

**Morina, V. and Rafuna, R., 2023.** *A comparative analysis of pathfinding algorithms in NPC movement systems for computer games*. UBT – Higher Education Institution, Prishtine, Kosovo. Available at: <https://www.researchgate.net/publication/375895663_A_Comparative_Analysis_of_Pathfinding_Algorithms_in_NPC_Movement_Systems_for_Computer_Games> [Accessed 10 November 2024].

**Perez, X., Angulo C., Escalera S., Pardo D., 2010**. *Vision-based Navigation and Reinforcement Learning Path Finding for Social Robots*. Available at: <http://www.maia.ub.es/~sergio/linked/cvcrd2010xavi.pdf> (Accessed: 10 November 2024).

**Perier, T., 2021.** *Analysis of artificial intelligence in chess*. Astrakhan Innovation Management. Available at: <https://astrakhan.consulting/blog/analysis-of-artificial-intelligence-in-chess/> [Accessed 9 November 2024].

**Piech, C., 2012.** *Deep Blue*. Stanford University. Available at: <https://stanford.edu/~cpiech/cs221/apps/deepBlue.html> [Accessed 9 November 2024].

**Power, R. 2024**. *The Human Element: Integrating Emotional Intelligence into AI*. Available at: <https://www.forbes.com/sites/rhettpower/2024/05/29/the-human-element-integrating-emotional-intelligence-into-ai/> (Accessed: 10 November 2024).

**Rafiq, A., Kadir, T.A. and Ihsan, S.N., 2021.** *Pathfinding algorithms in game development*. IOP Conference Series: Materials Science and Engineering, 769(1), p.012021. Available at: <https://iopscience.iop.org/article/10.1088/1757-899X/769/1/012021/pdf> [Accessed 10 November 2024].

**Soltis, A.E., 2024.** *Chess: Development of theory*. Encyclopaedia Britannica. Available at: <https://www.britannica.com/topic/chess/Development-of-theory> [Accessed 9 November 2024].

**Sony 2024.** *AIBO Features: Discover the AI Behind AIBO*. Available at: <https://us.aibo.com/feature/feature3.html> (Accessed: 10 November 2024).

**Stanford University, 2009.** *Minimax algorithm. Strategies and Tactics for Intelligent Search*. Available at: <https://cs.stanford.edu/people/eroberts/courses/soco/projects/2003-04/intelligent-search/minimax.html> [Accessed 9 November 2024].

**Thompson, C., 2022.** *What the history of AI tells us about its future*. MIT Technology Review. Available at: <https://www.technologyreview.com/2022/02/18/1044709/ibm-deep-blue-ai-history/> [Accessed 9 November 2024].

**Tomašev, N., Paquet, U., Hassabis, D. and Kramnik, V., 2020.** *Assessing game balance with AlphaZero: Exploring alternative rule sets in chess.* DeepMind. Available at: <https://arxiv.org/pdf/2009.04374> [Accessed 9 November 2024].

**TOF Sensors 2024.** *TOF Camera Technology's Crucial Role in AIBO AI Robot*. Available at: <https://tofsensors.com/blogs/tof-sensor-knowledge/tof-camera-technologys-crucial-role-in-aibo-ai-robot?srsltid=AfmBOooCjZpSWi02FZn8hVP-P6vjes0ddcoJT4v9a6jbOdyI66-_9Bgp> (Accessed: 10 November 2024).

**Tupe, K., Singh, P., Parmar, S. and Pandey, T., 2016*.*** *AI & NPC in games*. International Journal of Computer Engineering and Research, [online] Available at: <https://ijcert.org/ems/ijcert_papers/V3I307.pdf> [Accessed 10 November 2024].

**Unreal Engine, 2024**. *Behavior Tree in Unreal Engine – User Guide*. Epic Games. Available at: <https://dev.epicgames.com/documentation/en-us/unreal-engine/behavior-tree-in-unreal-engine---user-guide> [Accessed 10 November 2024].