Annotated Bibliography

1. Stack Overflow. "What Is the Difference Between Iteration and Recursion?" ***Stack Overflow*****,**

https://stackoverflow.com/questions/19794739/what-is-the-difference-between-iteration-and-recursion. Accessed 16 Jan. 2025.

This source explains the differences between iterative and recursive methods in programming, using examples to illustrate how each approach functions. I selected this source because understanding recursion is fundamental to developing chess engines, particularly for implementing algorithms like minimax and alpha-beta pruning. This source is helpful because it breaks down the concepts in simple terms and provides relatable examples. One limitation of this source is its reliance on technical jargon, which may not be accessible to beginner programmers.

2. "Learn Recursion in 5 Minutes." ***YouTube*****, uploaded by Tech With Tim,**

https://youtu.be/ivl5-snqul8. Accessed 16 Jan. 2025.

This video concisely explains recursion by visually demonstrating how recursive models solve problems through base cases and breaking problems into simpler forms. I selected this source because it offers a visual, practical explanation that complements textual resources. It was particularly helpful for clarifying the recursive thought process. One limitation of this source is that it focuses solely on recursion without directly comparing it to iterative approaches.

3. Healey, Max. "Building My Own Chess Engine." ***Healey Codes*****,**

https://healeycodes.com/building-my-own-chess-engine. Accessed 16 Jan. 2025.

This blog details the author's journey of building a chess engine, including technical challenges and design decisions. This source provides a practical perspective on creating a chess engine, including algorithms like minimax and alpha-beta pruning. This source is helpful because it highlights potential pitfalls and strategies to overcome them. One limitation is that the technical explanations assume moderate programming knowledge.

4. Wikipedia. "Mechanical Turk." ***Wikipedia: The Free Encyclopedia*****,**

https://en.wikipedia.org/wiki/Mechanical_Turk. Accessed 16 Jan. 2025.

This source explains the history of the Mechanical Turk, an 18th-century chess-playing automaton, and its cultural significance. I selected this source because it provides historical context for my capstone, which the Turk inspires. I found this valuable source for understanding the Turk’s design and societal impact. One limitation is that Wikipedia is a tertiary source, and its accuracy depends on the citations provided.

5. Gonfalonieri, Alexandre. "How Does Amazon Alexa Work? Your Guide to Natural Language Processing (AI)." ***Towards Data Science*****, 21 Nov. 2018,**

https://towardsdatascience.com/how-amazon-alexa-works-your-guide-to-natural-language-processing-ai-7506004709d3.

This article explains how Amazon Alexa uses natural language processing to interpret voice commands and respond appropriately. I selected this source because it provides insights into AI's role in creating interactive systems, which parallels my interest in integrating interactivity into Chess Turk. I found it helpful in understanding the underlying technologies that make voice recognition and response possible. One limitation is its lack of technical depth, which may require further research to apply the concepts thoroughly.

6. "Minimax Algorithm." ***YouTube*****, uploaded by Computerphile,**

https://youtu.be/l-hh51ncgDI. Accessed 16 Jan. 2025.

This video explains the minimax algorithm and how it applies to two-player games like chess, emphasizing how players maximize or minimize outcomes. I selected this source because minimax is a foundational algorithm for chess engines. It was helpful for its visual representation of decision trees and practical examples. One limitation is that it doesn't delve deeply into optimization techniques like alpha-beta pruning.

7. "Alpha-Beta Pruning." ***YouTube*****, uploaded by Computerphile,**

https://youtu.be/l-hh51ncgDI. Accessed 16 Jan. 2025.

This video extends the minimax explanation by introducing alpha-beta pruning, which improves efficiency by eliminating unnecessary branches in the decision tree. I selected this source because it focuses on optimizing chess engine performance. I found it helpful for its clear diagrams and explanations. One limitation is that it assumes prior knowledge of the minimax algorithm.

8. "Alpha-Beta Pruning in Artificial Intelligence." ***GeeksforGeeks*****,**

https://www.geeksforgeeks.org/alpha-beta-pruning-in-artificial-intelligence/. Accessed 16 Jan. 2025.

This article provides an in-depth explanation of alpha-beta pruning, including its benefits and implementation. I selected this source for its detailed breakdown of the algorithm and example pseudocode. I found it helpful to understand better how to implement chess engine pruning. One limitation is that it doesn't provide real-world coding examples.

9. Witten, Ian H., Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, 2011.

This book provides foundational knowledge of machine learning algorithms and their applications. I selected this source because it explains concepts like decision trees and optimization, which are relevant to chess engine design. I found it helpful for its detailed theoretical explanations. One limitation is that it is not explicitly focused on chess or game algorithms.

10. Mitchell, Melanie. *Artificial Intelligence: A Guide to Intelligent Systems.* Pearson, 2009.

This book explores AI systems, including natural language processing, decision-making, and machine learning. I selected this source because it offers a comprehensive overview of AI, helping connect the chess engine's algorithmic design with broader AI applications. I found it helpful in understanding how chess engines fit into the larger AI landscape. One limitation is that it is an older publication, and some examples may be outdated.