# Release Notes - Sprint 1

Sprint Goals: The main goal for this sprint was to research the available frameworks, programming languages, and discuss with my peers if the project was feasible. Additionally, I aimed to determine if Python was a suitable programming language for this project, as well as define my expected end results.

## Sprint Achievements:

* Conducted research on available frameworks and programming languages for building a chess engine
* Discussed the feasibility of the project with peers a couple of times and received feedback
* Chose Python as the programming language for the project, due to its vast libraries for AI and machine learning
* Defined my expected end results for the entire project

## Project-Related Documents:

* Backlog and sprint planning: This document
* Analysis document: -
* Technical design document: -
* Testing strategy document: -
* Code repository: -

## Expected End Results

My expected end results for the entire project are to build a chess engine that can evaluate a position and make a good move based on machine learning algorithms. Additionally, I plan to deploy the application to the cloud and make it available for public use.

## Sprint Planning for Next Sprint

For the next sprint, my goals are to start building the core functionality of the chess engine, such as defining the board state, valid moves, and evaluating a position. I also plan to start integrating machine learning algorithms to improve the engine's move prediction, but that is for another sprint.

Note: As this is the first sprint, the documents are still under development, and their contents may change in the coming sprints.

# Release Notes - Sprint 2

Sprint Goals: The main goal for this sprint was to research available solutions on the internet, research how a chess engine should work, and define the testing criteria for the chess engine.

## Sprint Achievements

• Conducted thorough research on available solutions on the internet, exploring various approaches to building a chess engine

• Discovered dozens of possible solutions, but only one was found to be suitable, meeting all the requirements perfectly

• Researched how a chess engine should work, including how to define the board state, valid moves, and evaluate a position

• Defined the testing criteria for the chess engine, which will be to beat a 1000 ELO player

## Project-Related Documents

• Backlog and sprint planning: This document

• Analysis document: In progress

• Technical design document: In progress

• Testing strategy document: In progress

• Code repository: <https://github.com/YVLC/ChessEngine6>

After conducting extensive research on the internet, I discovered several approaches to building a chess engine. I studied each of these approaches and evaluated them based on their ability to meet the project requirements. Although many solutions seemed promising, only one was found to be a good fit for the project. This solution will be used as the basis for the development of the chess engine.

In addition to researching available solutions, I also researched how a chess engine should work. This included defining the board state, valid moves, and evaluating a position. I found that there are many approaches to evaluating a position, including heuristics, minimax, and Monte Carlo tree search. I will explore each of these approaches in greater detail during future sprints.

Another important aspect of building a chess engine is defining the testing criteria. The primary testing criteria for the chess engine will be to beat a 1000 ELO player. While this may seem like a relatively easy task for an advanced chess engine, it is still a significant achievement and will require careful planning and implementation.

Looking ahead, my goals for the next sprint are to start building the core functionality of the chess engine, including defining the board state, valid moves, and evaluating a position. I will also begin exploring different approaches to move prediction, including heuristics, minimax, and Monte Carlo tree search. Additionally, I will continue to refine the testing criteria and develop a comprehensive testing strategy to ensure that the chess engine meets the project requirements.

Note: As the project progresses, the documents will continue to evolve and change. The content of the documents will be refined based on the insights and knowledge gained during each sprint.

# Release Notes - Sprint 3

The primary goal for Sprint 3 was to research the code and infrastructure of the found repository, and determine the requirements needed for the chess engine to be fully functional. This sprint also involved filling in and adding to the Analysis document, Technical Design document, and Testing Strategy document for the project.

## During this sprint, we achieved the following:

• Conducted thorough research on the code and infrastructure of the found repository

• Identified the requirements needed for the chess engine to be fully functional

• Updated the Analysis document, Technical Design document, and Testing Strategy document to reflect the current state of the project

• Created a working Proof of Concept (POC) of a chess game that is currently available to play multiplayer

To ensure that the chess engine is useful for players to get relevant insights about the game, it is crucial that all the documentation is implemented. Therefore, we focused on completing these features during this sprint.

Moving forward, the primary focus for the next sprint will be to build the core functionality of the chess engine. This includes implementing the board state, valid moves, and evaluating a position. Additionally, we will work on implementing an AI opponent to play against, which will require creating a Chess AI Research Document. By the end of sprint 4, the chess engine should be fully functional, and players should be able to play against the AI as well as against other human players.

## Project-Related Documents

• Backlog and sprint planning: This document

• Analysis document: <https://tinyurl.com/2p8p634n>

• Technical design document: <https://tinyurl.com/dws2d5jn>

• Testing strategy document: <https://tinyurl.com/3mtsc6at>

• Code repository: <https://github.com/YVLC/ChessEngine6>

Note that as the project progresses, the documents will continue to evolve and change. The content of the documents will be refined based on the insights and knowledge gained during each sprint.

## Release Notes - Sprint 4

The primary goal for Sprint 4 was to build the core functionality of the chess engine. This involved implementing the board state, valid moves, and evaluating a position, as well as creating an AI opponent to play against.

## During this sprint, we must achieve the following:

• Implemented the board state, valid moves, and evaluating a position

• Created an AI opponent to play against using the minimax algorithm

• Created the Chess AI Research Document, outlining the AI approach and algorithm used

• Integrated the AI opponent into the chess game, allowing players to play against the computer

By the end of this sprint, the chess engine has to be fully functional, with all features implemented and tested. Players could play against both other human players and the AI opponent, and the engine provided relevant insights about the game.

Moving forward, the primary focus for the next sprint will be to improve the AI opponent's performance and make it more challenging for players to beat. We will also continue to refine and improve the codebase, as well as maybe work on additional features such as game history and analysis.

## Project-Related Documents

• Backlog and sprint planning: Mostly on paper

• Analysis document: Updated to reflect the current state of the project

• Technical design document: Updated to reflect the current state of the project

• Testing strategy document: Updated to reflect the current state of the project

• Code repositories: <https://github.com/YVLC/ChessEngine6>