

	Zeit in Stunden	Beschreibung		Gesamtzeit:	201	
20.04.22	1	Besprechung mit Herr Kretschmer				
	1	Nachbereitung Besprechung				
26.04.22	1	Besprechung mit Herr Kretschmer				
28.04.22	1	Interne Besprechung Hr. Schönenberger, Fr. Neu, Hr. Lux Klärung Pflichtenheft UCI-Protokoll unterstützt in Java geschrieben Als Command Line Interface aufgerufen Spielbaumtiefe selbst einstellbar? Bekommt eine Stellung als Eingabe und gibt als Ausgabe den als optimal festgestellten Spielzug zurück Nutzung eines Mini-Max-Algorithmus zur Berechnung des optimalen Zugs? Einigung auf Architektur mit Service-Interfaces für die Kommunikation mit UCI				
29.04.22	2	gemeinsames Aufarbeiten des UCI-Protokolls gemeinsame Recherche UCI, FEN-Notation, tieferes Verständnis der Schachregeln weitere Besprechung des Pflichtenheft				
04.05.22	1	Anfertigung des Pflichtenhefts				
11.05.22	1	Besprechung mit Herr Kretschmer Review Pflichtenheft Gemeinsames Überlegen von Meilensteinen Nachbesprechung mit Hr Schönenberger u. Fr Neu Umtragen des Arbeitsprotokolls				
14.05.22	3	Überarbeitung des Pflichtenhefts nach der Struktur der Gruppe "Spielstärkeermittlung"				
17.05.22	2	Überarbeitung des Pflichtenhefts nach Feedback von Herr Kretschmer, Erstellung von Meilensteinen				
18.05.22	1	Besprechung mit Herr Kretschmer, Anpassung Pflichtenheft und Meilensteine				
27.05.22	3	Arbeitspakete er- und überarbeiten, Termine für Meilensteine festlegen				
28.05.22	6	Erarbeitung Programmarchitektur: - Zug-Generator - UCI-Service - Stellungsbewertung - Spielbaum - Beziehungen dazwischen verdeutlicht Festlegung von Datenstrukturen (Board, Piece(s)) - Pieces als Enum mit Wert - Board als 2D-Array von Pieces mit zusätzlichen Informationen über den GameState Erarbeitung Klassendiagramm - verdeutlichung der Datenstrukturen Board und Pieces - Definition der Utility-Klassen MoveGenerator und PositionEvaluator - Skizzierung von UCIService und GameTree				
29.05.22	2	Vorbereitung des Treffens am 30.05.22: Skizzieren des Klassendiagramms für GameTree, GameNode und UCIService als Diskussionsgrundlage				
30.05.22	4	Suchen nach Dokumentation von UCI (nicht erfolgreich, nur bekannte Spezifikationsdatei gefunden) Suchen nach Dokumentation der Arena GUI (.chm Datei in Arena Download) Entfernen der Arbeitspakete zu Remis und Aufgeben der Engine da nicht von UCI unterstützt Erneut Ideen zur Generierung des Spielbaums und Bewertung der entstehenden Stellungen sammeln Klassendiagramm erweitern (Shorthandmove, Übersetzungsmechanismus ShorthandMove-Board, UCI-Service)				

	Zeit in Stunden	Beschreibung		Gesamtzeit:	201	
31.05.22		Festlegung der Lebenszeit von GameTree-Instanzen weitere Überlegungen zu Stellungsbewertung im Spielbaum Eignung einer Implementierung der UCI-Befehle als Enum-Konstanten diskutiert (Implementierung als Enum + Wrapper für dynamische Daten)				
09.06.22	3	Anpassungen am Klassendiagramm (unter anderem neue Klasse Conductor und Threadhandler)				
	1	Überlegungen zur Implementierung (Möglichkeiten zur Parallelisierung) mit Herr Schönenberger				
11.06.22	4	Erstellen einer Klasse für den Move-Generator Implementierung des Iterators, der das ganze Feld durchläuft und sich von jeder Figur die möglichen Moves ausgeben lässt Anlegen von Unterfunktionen für die einzelnen Figurtypen mit TODOs, die noch gefüllt werden müssen Einrichten von JDK 18, Problembehebung in der IDE Vorbereitung Implementierung Bauer				
12.06.22	3	Implementierung Bauer - Einzelschritt - Doppelschritt - Umwandlung - Schlagen (ohne En-Passant)				
13.06.22	3	Implementierung En-Passant Aufräumen Code (z.B. Umwandlung von 1 Funktion auf 2 aufteilen) Hinzufügen Dokumentation für den Bauern im Move-Generator				
15.06.22	4	toString-Methode für Pieces und das Board verfasst Debuggen des FEN-Parsers Code Review mit Herr Schönenberger Abschätzung neue Meilenstein-Fristen Überlegungen zu Schach-Erkennung, Einigung darauf, Attack Maps zur Schach-Erkennung probieren				
16.06.22	2	Erfolgloser Versuch, Attack Maps zu implementieren				
17.06.22	4	Erfolgreicher Versuch, Attack Maps zu implementieren (Läufer, Turm, Dame + Tests)				
18.06.22	4	Attack Maps Bauer, König, Springer implementiert + Tests Tests für den Bauern-Move-Generator				
19.06.22	3	Fixed bug in enPassant-Parsing Board implementiert jetzt Comparable für Tests MoveGenerator für Bauer fertig und getestet				
20.06.22	2	MoveGenerator für Springer implementiert und getestet				
21.06.22	4	MoveGenerator für König implementiert und getestet Statische Stellungsbewertung implementiert (Werte aus PieceType, Läuferpaar gibt Bonus von 2)				
25.06.22	3	Vorbereitung Merge von Läufer, Dame und Turm auf einen Branch mit den Neuerungen von Main Auflösung von Merge Conflicts Durchführen des Merges				
26.06.22	1	Schreiben des Demo-Programms, das einen FEN-String über Standard-In bekommt und auf Standard-Out die Follow-Up-Boards ausgibt				
08.07.22	2	Schreiben der Move-Klasse, die den Spielzug enthält, der zu einem Boardstate geführt hat. Verändern der MoveGeneration, sodass in jedem FollowUp-Board geschrieben steht, durch welchen Spielzug es entstanden ist				

	Zeit in Stunden	Beschreibung		Gesamtzeit:	201	
09.07.22	5	Konvertierung der Bauernwerte zu Centipawns Anpassung des Bonus für das Läuferpaar Lektüre des Blogs https://www.chessengines.org/ Tests anpassen an die neuen Figurenwerte Restrukturisieren der Tests (aufteilen auf mehrere Testklassen) Erste Überlegungen zur Implementierung eines UCI-Parsers, Gedanken zum Tokenizer				
11.07.22	2	Beschäftigen mit UCI Definition der notwendigen Tokens für den Tokenizer Auflisten der notwendigen Sendebefehle für den UCI-Operator herausfinden, wie man auf StdIn schreibt in Java				
12.07.22	1	Besprechung mit Herrn Kretschmer Nachbesprechung mit Schönenberger				
14.07.22	2	Aktualisierung Klassendiagramm				
28.07.22	4	Aufräumen im Git-Repo Überarbeitung Interfaces GameNode und GameTree Code Review GameNode und GameTree Überlegungen über die Implementierung von Iterative Deepening Struktur der Hauptschleife im GameTree festlegen Umbenennen von Klassen (Board zu Position)				
03.08.22	1	Besprechung mit Herrn Kretschmer Nachbesprechung mit Schönenberger				
15.08.22	3	Implementierung des UCI-Tokenizers				
16.08.22	2	Anfertigen von Test für Tokenizer Schreiben des UCI-Operators				
17.08.22	2	Besprechung mit Schönenberger Aktuellen Stand besprechen weiteres Vorgehen planen				
18.08.22	4	- Doppelschritt-Generierung für Bauern fixen - Weitere Tests für die Doppelschritt-Generierung schreiben um sicherzugehen, dass die Fixes funktionieren - Überarbeitung des Tokenizers und der Command-Klasse um eine Baumstruktur zu nutzen - Besprechung mit Herr Kretschmer - Nachbesprechung mit Herr Schönenberger				
27.08.22	5	- Bugfix Bauern (Ticket #71, Algebraische Notation nach Schlagen wurde nicht richtig generiert) - Fehlersuche Ticket #72 (Bauer gerät nach Schlagen in letzter Reihe out of bounds) alleine, konnte Fehlverhalten aber nicht reproduzieren - Fehlersuche Ticket #72 zusammen mit Herr Schönenberger, mutmaßliche Quelle gefunden - Ticket #72 gelöst, Problem war fehlerhafter Code in der Umwandlungsgenerierung - Ticket #67 (Schacherkennung fehlerhaft) gelöst, fehlerhafter Code in AttackMapGeneration behoben				
28.08.22	2	- Testen der Demo-App FenToAlgebraic - Ticket #78 (Refaktorisierung damit die konkrete Implementation von Board reibungslos ausgetauscht werden kann)				
02.09.22	6	- Tokenizer kann nun dynamisch viele Kinder einem Argument hinzufügen - Angefangen, Parser zu schreiben - Recherche von Details über die Funktionsweise von UCI - Arena GUI aufsetzen - Recherche darüber, wie man das Projekt so kompiliert dass Arena es ausführen kann - Testprogramm schreiben um Engine mit GUI zu verbinden - Engine kann jetzt den Handshake zur Engine abwickeln, Positionen initiieren und Züge übersenden - angefangen mit der Demo-App um zufällige Züge zu spielen				

	Zeit in Stunden	Beschreibung		Gesamtzeit:	201	
03.09.22	4	<ul style="list-style-type: none"> - Demo-App weiterarbeiten - Bug fixen, wo Bauernumwandlungen nicht berücksichtigt wurden - Bug fixen, wo FEN-Strings vom UCITokenizer in einzelne Tokens zerlegt wurden (indem man sie nachträglich wieder zusammenklebt) - Bug fixen, wo STARTPOS-Token vom Tokenizer auch als FEN-Strings erkannt und mit ihren Nachfolgern zusammengeklebt wurden 				
04.09.22	4	Start: 10 Uhr 20 Bugfix: es werden illegale Positionen generiert (König steht im Schach und es werden Züge generiert, die ihn im Schach zurücklassen) - Problem identifizieren (Bei RayAttacks wurde nicht überprüft, ob eine Position den König im Schach zurücklässt bevor sie zu den möglichen Zügen hinzugefügt wurden) - Problem beheben (Überprüfung hinzugefügt bevor eine Position zum Ergebnisset von ComputeRay hinzugefügt wird) Austesten Demo-App Demo-App zufällige Moves durch AB-Pruning ersetzen				
05.09.22	3	<ul style="list-style-type: none"> - Meeting mit Schönenberger wegen nichtdeterministischem Fehler - Änderungen im MoveGenerator (statt Sets werden Listen genutzt) - Erstellen einer ByteBoard-Klasse um Speicher zu sparen 				
06.09.22	4	<ul style="list-style-type: none"> - Ausbessern der Byteboard-Klasse - Gemeinsames Überlegen in Schriftform mit Schönenberger über den besten Weg, die Matt- und Patt-Erkennung zu implementieren - Parser aufräumen 				
09.09.22	6	<ul style="list-style-type: none"> - Conductor schreiben - Kapselung verbessern mit durchgereichten Funktionsaufrufen - Zuglogging nutzen um zu schauen, ob alle übergebenen Moves applied werden müssen. - überprüfen, ob Züge aus demselben Game stammen (mit Test schreiben) - viiiieel Debuggen der Funktion, die nur den letzten Zug applied 				
10.09.22	4	<ul style="list-style-type: none"> - Verhackten Merge vom Abend vorher aufräumen - Bug in der Schacherkennung fixen (Ticket #92) 				
11.09.22	4	<ul style="list-style-type: none"> - Bugfix Ticket #92 - Test für den Bugfix schreiben (Matt-Erkennung) - PositionEvaluator nachbessern mit einer Gewichtung für Matts (frühere Matts besser als spätere) - nach Testfällen suchen in denen AB-Pruning ein Patt als bestes Outcome sehen würde - Tests für Patt-Erkennung schreiben - TODOs hinzufügen um darüber mit Schönenberger zu reden 				
13.09.22	2	<ul style="list-style-type: none"> - Merge-Konflikt auflösen mit dem Branch für Ticket #92 - Überlegungen für Speicheroptimierung - Eigenen Branch aufgemacht für Speicheroptimierung, angefangen, da Änderungen vorzunehmen 				
14.09.22	5	<ul style="list-style-type: none"> - Überlegungen für zukünftige Aufgaben (Multithreading, Dokumentation, Tracken der 30-Sekunden-Zeitvorgabe) - Bugfix mit Rochaderechten - selbstzerstörendes Pruning in Verbindung mit Arena ausprobiert 				
15.09.22	2	Besprechung mit Herrn Kretschmer Nachbesprechung mit Schönenberger Anteil der Move-Generation an Laufzeit ermitteln				
16.09.22	2	<ul style="list-style-type: none"> - aufteilen der MoveGenerator-Klasse in mehrere Piece-spezifische Klassen - erster Draft Multithreading 				
17.09.22	1	<ul style="list-style-type: none"> - nachbessern Multithreading, richtige Ergebnisse werden jetzt generiert 				
18.09.22	2	nachbessern Multithreading - Analyse warum bestimmte Test jetzt fehlschlagen - herausfinden, warum Performance schlechter ist als ohne MT				
23.09.22	4	Besprechung mit Schönenberger Durchgehen des Multi-threading Move-Generation Codes Besprechen des Vorgehens hinsichtlich iterative deepening, Verbesserung der Auswertungsfunktion				

	Zeit in Stunden	Beschreibung		Gesamtzeit:	201	
25.09.22	3	<ul style="list-style-type: none"> - Finalisierung Multi-Threading MoveGen - Zusammenführen mit MoveOrdering - Falten der PieceArrays - Implementieren von ByteEncoded Pieces in eigener Klasse mit Konstanten und Hilfsfunktionen 				
26.09.22	2	<ul style="list-style-type: none"> - Implementieren von PieceSquareTables - Anfangen von Remis-Erkennung 				
27.09.22	8	<ul style="list-style-type: none"> - Remis-Erkennung implementiert (Listen in Positions gespeichert) - Remis-Erkennung debuggt - Remis-Erkennung neu implementiert (mit einer zentralen, statischen Liste) - Remis-Erkennung debuggt - Remis-Erkennung mit zentraler Hashmap programmiert - Zurückgekehrt zu zentraler Liste 				
28.09.22	10	<ul style="list-style-type: none"> - Remis-Erkennung mit zentraler Liste finalisiert - Soweit möglich überall Piece durch ByteEncodedPiece ersetzt - TreeEvaluationTests überprüft mit Schönenberger - Anfang Iterative Deepening 				
29.09.22	10	<ul style="list-style-type: none"> - Iterative Deepening mit-implementiert - AttackMaps auf Byte umgestellt - Position speichert jetzt keine komplette AttackMap mehr sondern nur die nötigen Felder - Byte[] in Board halbiert - 2 Stunden BugHunt - Multithreading für UCI 				
30.09.22	10	<ul style="list-style-type: none"> - MultiThreading für AttackMapGeneration implementieren - MultiThreading optimieren - MultiThreading wegschmeißen - Tests für Iterative Deepening + BugFixes - CleanUp - Bugfixes 				