

Datum	Zeit in Stunden	Beschreibung		Gesamtzeit
20.04.22	1	Besprechung mit Herr Kretschmer		241
	1	Nachbereitung Besprechung		
26.04.22	1	Besprechung mit Herr Kretschmer		
28.04.22	1	Interne Besprechung 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	0,5	Besprechung mit Herr Kretschmer Mängel an Lastenheft und Protokoll hervorgehoben Sammlung von Meilensteinen		
11.05.22	0,5	Nachbereitung der vorangegangenen Besprechung Erfassen der bis zum nächsten Treffen zu erledigenden Aufgaben Vereinbaren des nächsten Treffens: 13.05.22, 15:00 (verschoben auf 14.05.22 10:00)		
14.05.22	3	Überarbeitung und Erweiterung des Pflichtenheft Aufteilen in Muss-, Kann- und Abgrenzungskriterien		
17.05.22	2	Überarbeitung des Pflichtenhefts entsprechend des Feedbacks von Herrn Kretschmer		
18.05.22	1	Besprechung mit Herrn Kretschmer Anpassung des Pflichtenhefts Überarbeitung der Meilensteine		
22.05.22	1	Initialisieren des git Repositories Aufsetzen der Ordnerstruktur und erstellen der Maven Konfiguration (pom.xml) Testen, ob korrekte Java Version genutzt wird und Junit verwendbar ist		
27.05.22	3	Arbeitspakete er- und überarbeitet zeitliche Schätzungen für Meilensteine		
28.05.22	4	Abwägen und Festlegung effizienter Datenstrukturen (Pieces als Enumtyp, Board als 2-dimensionales Array von Pieces, speichern zusätzlicher Werte: Schach und aktueller Spieler) Erarbeitung Klassendiagramm (Zuggenerator, Stellungsbewertung) Skizzierung von UCIService und GameTree		
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 ShothandMove-Board, UCI-Service)		
31.05.22	3	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) Anpassungen am Klassendiagramm (unter anderem neue Klasse Conductor und Threadhandler)		
01.06.22	1	Absolute theoretische Grundlagen des Multithreadings (Anwendungsfälle, Vor- und Nachteile)		
09.06.22	1	Ermitteln von Abhängigkeiten bei der Implementierung des Zuggenerators, um Aufteilung auf Gruppenmitglieder zu ermöglichen		
10.06.22	2	Beschäftigung mit Minimax, Negamax, Alpha-Beta-Pruning (noch nicht abgeschlossen)		
10.06.22	3	Implementierung von PieceType, Piece und Board		

Datum	Zeit in Stunden	Beschreibung		Gesamtzeit
20.04.22	1	Besprechung mit Herr Kretschmer		241
11.06.22	2	teilweise Implementierung des Fen-Parsers (Rochaderechte, En-Passant-Target, Zugzahl)		
11.06.22	2	Änderungen an Board Abschließen der Fen-Parser-Implementierung (Stellungs-Kodierung aneignen, Übersetzen von Buchstaben in Figuren der richtigen Farbe, Parsen von Stellungen)		
12.06.22	2	geringfügiges Refactoring der FenParser Klasse		
15.06.22	2	Arbeiten am Zuggenerator Codereview (mit Lux) Neue Fristen für Meilensteine		
18.06.22	0,5	Überlegungen zu Schacherkennung (Attack Maps) Code Review Attack Maps		
19.06.22	5	Implementierung geradliniger Zuggenerierung Implementierung einer Methode um ausgehend von einer Board-Instanz ein nach einem Zug darauf folgendes Board zu erzeugen Implementierung der Zuggenerierung für Läufer, Turm und Dame Aktuellen Meilenstein Zuggenerator in Github übertragen Überlegungen zum Testen des Zuggenerators		
24.06.22	3	Anpassungen an MoveGenerator Test für Läufer, Turm und Königin-Bewegung geschrieben Tests für Bewegungsberechnung in Ecken geschrieben am Mergen von Branches gescheitert		
25.06.22	5	weiter am Mergen von 2 Branches scheitern (+ fertigen Merge verlieren) Debugging (MoveGeneration) Github aufräumen Github Action zum automatischen Testen mittels Maven aufsetzen Test für copySpaces() schreiben		
28.06.22	1	Tests für Board schreiben		
30.06.22	4	Tests für Board schreiben Fehler in Turm beheben (Rochaderechte wurden nicht entfernt)		
11.07.22	3	Refactoring (Interfaces, Wrapper)		
12.07.22	1	Schlechten Code vom Vortag verbessern		
12.07.22	1	Besprechung mit Herrn Kretschmer Nachbesprechung mit Lux		
14.07.22	1	Erstellen der Interfaces GameTree und GameNode gemäß Klassendiagramm		
21.07.22	2	GameNode implementiert Teilweise GameTree implementiert		
28.07.22	4	mit Lux: Aufräumen im Git-Repo Überarbeitung Interfaces GameNode und GameTree Code Review GameNode und GameTree Struktur der Hauptschleife im GameTree festlegen Umbenennen von Klassen (Board zu Position)		
02.08.22	4	Lesen von <a href="https://www.chessengines.org/">https://www.chessengines.org/</a> bis einschließlich Beschreibung Minimax Versuchswises Implementieren von MiniMax Mit Alpha-Beta-Pruning beschäftigt (Chess-Engine Blog, Chess-Programming-Wiki, Wikipedia, Video von Sebastian Lague)		
03.08.22	0,5	einige Unklarheiten mit Alpha-Beta-Pruning beseitigen		
03.08.22	1	Besprechung mit Herrn Kretschmer & Herrn Lux Nachbesprechung mit Lux		
03.08.22	3	Mini-Max beschreiben (zur Dokumentation) Alpha-Beta-Pruning beschreiben (Beschreibung muss noch überarbeitet werden)		
04.08.22	7	Alpha-Beta-Pruning implementieren Refactoring: Generische Interfaces und Klassen für Tree, Node, Evaluator, Evaluables (WIP)		

Datum	Zeit in Stunden	Beschreibung		Gesamtzeit
20.04.22	1	Besprechung mit Herr Kretschmer		241
05.08.22	5	weiter an Generics arbeiten: Interfaces aufräumen AlphaBeta und MiniMax als Generische Klassen mit nicht-generischen Subklassen queryChildren(), getParent() etc typen korrekt weitere Interfaces hinzugefügt Dokumentation		
06.08.22	3	Interface Board hinzugefügt Teile der Klasse Position in Board verlagert Code-Redundanz von Alpha-Beta-Pruning reduziert Evaluable Interface erweitert Dokumentation		
08.08.22	5	TreeEvaluator arbeiten auf Bäumen statt Nodes Dokumentation Node Klassenhierarchie refactored Klassendiagramm für GameTree, Node, Evaluator		
10.08.22	3	zwei Fehler in Alpha-Beta-Pruning beheben Node testen: Mockklasse IntNode geschrieben, Testbaum erstellt einige Fehler behoben		
12.08.22	2	Code mergen (Generische Klassen -> Main)		
17.08.22	5	Generische Node Implementierung testen Fehler beheben Dokumentation GameTree-Klassendiagramm aktualisieren Alpha-Beta-Pruning debuggen		
17.08.22	2	Besprechung mit Lux Aktuellen Stand besprechen weiteres Vorgehen planen		
18.08.22	3	mehrere Fehler in Alpha-Beta-Pruning beheben Besprechung mit Herrn Kretschmer und Lux Nachbesprechung mit Lux		
19.08.22	3	weiteren Test für MoveGeneration schreiben GameNode testen Tests debuggen Fehler erfassen		
20.08.22	2	Testklassen leicht umstrukturiert Test für Alpha-Beta-Pruning geschrieben		
21.08.22	4	weiteren Test für Alpha-Beta-Pruning geschrieben Klasse BaseTree angelegt Testklassen restrukturiert Tests für Schachmatt und Patt geschrieben		
25.08.22	2	Hilfsmethode zum Erzeugen von Bäumen (Testdaten) geschrieben Mängel an Alpha-Beta festgehalten (TODOs)		
26.08.22	7	Alpha-Beta-Pruning testen Dokumentation einiger Methoden verbessern Terminal-basierte Demo-Anwendung geschrieben (Fen in-Algebraic out) unter Verwendung des Lichess-Analyse-Bretts und der Demo-Anwendung gegen die Engine gespielt und auftretende Fehler dokumentiert		
27.08.22	2	Debugging von #72 mit Lux Testen für #72		
28.08.22	1	Von Lux entdeckten Fehler untersuchen und dokumentieren		

Datum	Zeit in Stunder	Beschreibung		Gesamtzeit
20.04.22	1	Besprechung mit Herr Kretschmer		241
01.09.22	1	Besprechung mit Herrn Kretschmer und Lux Problem beim automatischen Mergen eines Branches beheben		
04.09.22	4	Test als Fehlerbeispiel für #79 equals für Move und Coordinate implementiert (gleichzeitig auch von Lux implementiert, mein Fehler) equals getestet nicht-deterministisches Verhalten von #79 untersuchen		
05.09.22	3	inkonsistentes Verhalten weiter untersucht und einige Tests geschrieben Terminal-basiere Demo-Anwendung erweitert: Berechnungstiefe für jede Eingabe einstellbar Meeting mit Lux		
06.09.22	1	Überlegungen mit Lux zur Matt- und Patt-Erkennung und zur weiteren Verbesserung der Speichereffizienz		
07.09.22	2	einige Tests für Alpha-Beta geschrieben Verifiziert, dass Ersetzung von Hashsets durch Listen das inkonsistente Verhalten beseitigt hat		
08.09.22	3	Checkout eines alten Commits um sicherzustellen, dass trotz (damals) inkonsistentem Verhalten immer Stellungen mit gleicher Bewertung ausgewählt wurden Dies bestätigt die vermutete Ursache und Korrektheit des "Bugfixes" (weniger ein tatsächlicher Bug, eher unerwartetes Verhalten). - statt Position speichert jetzt Node die von AlphaBeta benutzen Werte - dadurch unnötig gewordene Klassen (EvaluableInteger) entfernt und entsprechende Tests angepasst - einige Typparameter angepasst		
09.09.22	2	Code vom Vortag geprüft Tiefe von Alphabeta in einigen Tests reduziert, um diese zu beschleunigen Integer-basierte Alpha-Beta-Tests umstrukturier, sodass sie auf verschiedene MiniMax/AlphaBeta-Versionen angewandt werden können Zähler für Anzahl der durch AlphaBeta ausgewerteten Knoten implementiert		
10.09.22	5	Fen-Parser strenger gemacht und getestet Terminal-Demo-Anwendung komplett überarbeitet und ausgebaut Tests für Minimax/Alpha-Beta umstrukturier, damit man sie einfach auf verschiedene Implementierungen anwenden kann		
13.09.22	4	Vorgehen zum Anwenden von Minimax/AlphaBeta-Tests geändert: Verwendung von Lambdas statt abgeleiteten Klassen, die eine abstrakte Methode implementieren Code gemergt Detaching implementier: Position-Objekte werden aus Nodes gelöscht, nachdem sie zur Erzeugung von Kindknoten genutzt wurden		
14.09.22	5	Alpha-Beta angepasst, sodass Kinder ausgewerteter Knoten gelöscht werden Besprechung mit Lux: Überlegungen für zukünftige Aufgaben (Multithreading, Dokumentation, Tracken der 30-Sekunden-Zeitvorgabe) Bugfix mit Rochaderechten selbstzerstörendes Pruning in Verbindung mit Arena ausprobiert		
15.09.22	5	Besprechung mit Herrn Kretschmer Nachbesprechung mit Lux Anteil der Move-Generation an Laufzeit ermitteln Move-Ordering in Alpha-Beta implementiert		
16.09.22	2	Demo/Terminal-Test-Anwendung verbessert		
17.09.22	3	Speichereffizienz von ein und zwei-dimensionalen Arrays untersucht (8*8 byte/boolean Arrays verbrauchen etwa doppelt so viel Speicher wie ein-dimensionale 64 byte/boolean Arrays) mit Auswertungsfunktionen beschäftigt (primär Piece-Square-Tables)		
18.09.22	1	Move-Ordering aufgeräumt, in Test-Anwendung eingebaut und gemergt		
21.09.22	3	Erweiterung des Mechanismus um Tests für verschiedene AlphaBeta-Varianten in einer einzigen Klasse zu schreiben von Integer-Nodes auf GameNodes Code geschrieben, der Stellungen spiegelt um die Move-Generation auf Symmetrie zu testen Spiegelungs-Code getestet		

Datum	Zeit in Stunden	Beschreibung		Gesamtzeit
20.04.22	1	Besprechung mit Herr Kretschmer		241
22.09.22	5	Bug in Spiegelung untersucht und Korrektur des Full-Move Counter in Spiegelung für Folgezüge implementiert einzelne Methode geschrieben, die alle für die Zug-Generierung einer Figur nötigen Tests anwendet (Vergleich erwarteter & tatsächlicher Folgestellungen, Spiegelung der Stellung & der Folgezüge, Vergleich gespiegelter erwarteter und tatsächlicher Folgestellungen der gespiegelten Stellung) anpassen von Move-Generation-Tests um diese Methode zu nutzen (damit Mirroring in den Tests angewandt wird, um jeden Test für weiß auch für schwarz auszuführen und umgekehrt)		
23.09.22	3	Besprechung mit Lux Durchgehen des Multi-threading Move-Generation Codes (von Lux geschrieben) Besprechen des Vorgehens hinsichtlich iterative deepening, Verbesserung der Auswertungsfunktion		
24.09.22	4	Asymmetrischen Testbaum für Alpha-Beta Pruning angelegt Tests für Baum-Auswertung verstärkt (Testbäume werden für beide Spieler ausgewertet, dazu werden die Blattwerte und das erwartete Ergebnis mit -1 multipliziert ) Fehler im Invertierungscode identifiziert und behoben (Overflow durch Zweierkomplementdarstellung beim Multiplizieren von Integer.Min_Value mit -1, sodass $\text{Min\_Value} * -1 = \text{Min\_Value}$ )		
25.09.22	4	Klasse geschrieben um Bäume aufs Terminal auszugeben um Testen von alpha-beta zu erleichtern Tests für Alpha-Beta auf 2 asymmetrischen Bäumen geschrieben		
26.09.22	4	Evaluable überarbeitet Alpha-Beta-Pruning getestet		
27.09.22	10	Besprechung mit Lux Evaluable für Iterative Deepening überarbeitet und getestet mit Lux an Zugweiderholungserkennung gearbeitet Move-Generator mit Stockfish getestet Debuggen von #135 (erster Zug anders)		
28.09.22	8	Iterative Deepening weiter vorbereitet Testen MoveGen Debuggen von #135 (erster Zug anders)		
29.09.22	10	Iterative Deepening implementiert TreeEvaluator Interface verändert Diverse kleine Issues abgearbeitet iterative Deepening getestet weitere Tests für Node implementiert Byte-Attackmaps mit Lux entworfen		
30.09.22	10	kleine Issues abgearbeitet aufgeräumt weitere Test für Iterative Deepening, AlphaBeta, Move-Generation und mehr geschrieben		