

Alpen-Planung

A	E	L
Umgebung Programmieren	1	4h
Spielzüge programmieren	1.1	1h
Brett programmieren	1.0	20min
Ungültige Züge nicht erlauben	1.2	1h
Auf den Richtigen Ablauf des Spiels achten	1.3	1h
Zurückgeben, wer am Gewinnen ist und wer wie viele Steine wo hat (zwei in einer Reihe, drei in einer Reihe usw.)	1.4	20min
Grafische Darstellung mit dem Spiel und dem Agent programmieren	3	4,5h
Bilder herstellen/Zeichnen	3.0	30min
Grundgerüst der Grafischen Klasse erstellen	3.1	30min
Umgebung einbauen	3.2	1h
Agent einbauen	3.4	30min
Verschiedene Etappen des Spieles richtig zum Laufen bringen	3.3	2h
Random-Agent erstellen	2	1h
Modelle definieren	5	30min
Replay-Memory erstellen	4	30min
Train-Controller programmieren	6	3h
Rewards festlegen	8	6h
Agent so umbauen, dass das Model verwendet wird	7	10min
evtl. Netzwerke trainieren	15	2 - 5 d
Monte Carlo Tree Search implementieren	9	4 h
MCTS verbessern	11	11 h
Performance verbessern	12	5h
Netzwerke in MCTS einbauen (AlphaZero implementieren)	13	30 h
MCTS testen	10	5h
Theorie von Machine Learning und Reinforcement learning einfach erklären	16	10 h
Dokumentation schreiben	17	10 h
Computer mit Graphikkarte organisieren	14	4 h
Total	100h - 300h	

Projektjournal

Protokoll			Planung	
Datum + Zeit	Gemachte Arbeiten	Einschätzung / Erkenntnisse / Reflexion	Geplante Arbeiten	Zeitaufwand
3.11.2020 14:10 - 15:40	Ich habe heute geschaut, ob ich die Vorbereitungen für das Projekt schon gemacht hatte. Ich werde manchmal in der Schule und manchmal zuhause schaffen. Danach habe ich das Inhaltsverzeichnis schnell gemacht. Als ich auf die nächsten Schritte warten musste, habe ich an meinem Testprogramm, dass TicTacToe spielt, bei dem ich versucht habe, so viele Probleme wie möglich schon vor der Projektarbeit zu beheben, um Zeit zu sparen. Als es dann zur Alpen-Planung kam habe ich viel Zeit verloren und wurde nicht fertig, weil ich zu fest damit beschäftigt war.	Ich sehe jetzt schon, dass ich viel zu viel Zeit brauchen werde, um diese Arbeit durchzuführen. Ich habe mich zu fest von anderen Sachen ablenken lassen, darum wurde ich nicht mit dem fertig, dass ich machen sollte. Ich sollte mich also besser konzentrieren.	Alpen-Methode fertig planen. Blatt zu wann wird was gemacht ausfüllen GitHub Repository erstellen	20 min 40 min 20 min
5.11.2020 17:00 - 18:30	Ich habe die Alpen-Methode fertig geplant und mit Herrn Haldimann besprochen, wie ich es machen werde, da ich viel zu viel Stunden habe. Ich werde die Arbeit jetzt einfach so gut machen, wie es für mich geht und es ist dann nicht schlimm, wenn es nicht perfekt ist und zum Beispiel nicht gegen Menschen gewinnt. Danach habe ich das GitHub Repository erstellt. Dort werde ich den Code immer hochladen das GitHub-Repository findet man unter https://github.com/42kangaroo/Muele_AI . Im Repository ist die Code Datei mit Mill.ipynb betitelt. Auch im Repository sind eine Open-Source-Lizenz, die besagt, dass jeder meinen Code ohne weiteres kopieren und weiterentwickeln darf und eine Datei, die als Willkommensseite des Repositories dient. Ich	Den Code zu schreiben geht schneller als gedacht, es könnte also doch noch gut funktionieren mit der Zeit. Das liegt an der Vorbereitung, die ich schon vor der Projektarbeit gemacht hatte. Mit PyCharm, der Entwicklungsumgebung, die ich gebrauche, sind Repositories und andere Codeentwicklungsnotwendigkeiten sehr einfach und schnell	Blatt zu wann wird was gemacht ausfüllen Restliche Spielzüge programmieren/nur die erlaubten machen lassen.	40 min 1h

	habe auch noch das Brett der Umgebung und 2 der 5 möglichen Spielzüge implementiert. Die Änderungen sind im Repository unter dem Commit " board + 2 from 5 moves " zu finden.	zu handhaben, was den Zeitaufwand viel kleiner macht.		
06.11.2020 17:00 - 18:30	Ich habe am Code weitergearbeitet und habe den "Stein vom Gegner wegnehmen" Spielzug programmiert. Während dem habe ich mit überlegen herausgefunden, dass ich keine 5 verschiedene Spielzüge brauche, sondern nur 4. Das wird beim Trainieren von grossem Vorteil sein. Weil ich es für die Spielzüge gerade brauchte, habe ich noch die Zusammenfassung über die Spiellage programmiert. Ein bisschen Dokumentieren musste ich auch noch, damit ich selbst drauskomme. Der Commit auf GitHub für diese Änderungen heisst "delete move + get free fields for moving + Summary + Type Security"	Es ist gut, dass ich einen Spielzug weniger programmieren und Trainieren muss, denn dann kann ich viel Zeit sparen. Es gehen manche Sachen immer noch schneller als erwartet, aber ich will mich nicht täuschen lassen, denn die Probleme werden schon noch auftauchen.	Blatt zu wann wird was gemacht ausfüllen Letzten Spielzug programmieren R a n d o m - A g e n t programmieren	40 min 30 min 1h
07.11.2020 11:00-12:00	Ich habe den letzten Spielzug, das Bewegen eines Steins von einer zuvor ausgewählten Position, entwickelt. Somit ist die Umgebung fertig. Sie wurde aber noch nicht vollständig ausgetestet und enthält wahrscheinlich noch Fehler. Um deren Anzahl zu minimieren habe ich alle Variablen Typsicher gemacht, das heisst man kann nur noch zum Beispiel eine Zahl reinschreiben und wenn man versucht einen Text reinzuschreiben gibt es ein Error.	Es ist sehr schwierig, nur so theoretisch nach der Programmierung zu sagen, ob alles funktioniert. Denn es können sich sehr viele kleine, aber schwer zu findende Fehler eingeschlichen haben. Darum muss ich auch ausgiebig testen.	Blatt zu wann wird was gemacht ausfüllen R a n d o m - A g e n t programmieren	40 min 1h
07.11.2020 18:00 - 18:30	Das Blatt zu wann ich was mache habe ich ausgefüllt, damit ich wirklich offiziell anfangen darf. Ich habe auch ein Buch ausgesucht, das ich kaufen werde. Es heisst <i>Deep Reinforcement Learning with Python: Master classic RL, deep RL, distributional RL, inverse RL, and more with OpenAI Gym and TensorFlow</i> . Um es zu finanzieren werde ich einen Kurs im Phaenovum geben.	Mit der Zeitplanung sollte es aufgehen, da ich jetzt schon angefangen habe. Das Buch sieht so aus, als wäre es auch für mein Niveau verstehbar.	R a n d o m - A g e n t programmieren Visualisierung: Bilder erstellen	1h 1h
08.11.2020 14:00-17:00	Ich hatte, während dem Leiten des Videochats des RoboRave Cyberspace viel Zeit, um am Projekt zu arbeiten, weil die Teilnehmer keine Fragen stellten. Ich konnte also den Random-Agent austesten und erstellen, die Bilder für die Visualisierung mit GIMP Zeichnen und das Grundgerüst der	Der Agent schafft keinen Durchlauf des Spiels: entweder ist er mit dem Zufälligen auswählen von Aktionen zu langsam oder es	Visualisierung: Umgebung einbauen Visualisierung: Verschiedene Teile	1h 2h

	Visualisierungs-klasse programmieren.	gibt Bugs in der Umgebung. Diese kann ich aber nur beheben, wenn ich die Visualisierung habe und dann mit eigenen Augen sehen kann, was vorgeht.	des Spieles zum Laufen bringen	
09.11.2020 16:00-18:00	Man kann jetzt ein ganzes Spiel gegen sich selbst spielen, da ich die Visualisierung gemacht habe. Ich musste recht lange Debuggen und schauen, dass alle Spielzüge gut und richtig durchlaufen werden. Auch musste ich darauf achten, dass es immer den Richtigen Spielzug vorgibt, damit man im Richtigen Moment nur schieben, wegnehmen oder setzen kann. Die beiden Commits: "displayer and bugfixes" und "Debuging von Environment" sind aus diesen Änderungen entstanden	Der Random-Agent kann jetzt das ganze Spiel in ca. 0,1 s durchlaufen, und schafft es auch in 1000 von 1000 malen. Das heisst, dass es keine Situationen mehr gibt, wo die Umgebung einen Spielzug vorgibt, der gar nicht ausführbar ist.	Visualisierung: Agent einbauen Replay-Memory erstellen Modelle definieren	1h 1h 2h
10.11.2020 14:10 - 15:30	Ich habe heute den Agent in die Visualisierung eingebaut. Das heisst dass man jetzt gegen einen Agent, der einfach per Zufall seine Steine legt, spielen kann und zuschauen kann, wie er gegen sich selbst spielt. Danach habe ich dann das Modell und den Replay-Speicher definiert. Das meiste konnte ich glücklicherweise von meinem Vorbereitungsprojekt, dem TicTacToe Spieler, übernehmen. Danach habe ich den Trainings-Controller angefangen. Die Commits sind " moderated graphics ", " SumTree + DQModel + Memory " und " Controller started + processState "	Es ist recht lustig, zu sehen, wie der Agent gegen sich selbst spielt und sehr dumme Züge macht. Für mich ist es gut, dass ich bald schon mit dem Trainieren anfangen kann, denn ich muss mit dem Buch, dass am Freitag oder Montag ankommt, schauen, wie ich die Netzwerke besser trainieren kann, denn beim TicTacToe war mein trainiertes Netzwerk sehr schlecht.	Controller fertig machen Rewards festlegen	3h 6h
11.11.2020 16:00 - 18:00	Ich wurde mit dem Controller fertig und habe auch für den ersten Zug schon die Rewards festgelegt. So konnte ich dann mit dem Trainieren anfangen. Auf meinem Eigenen Computer war es sehr langsam. Darum habe ich nach alternativen gesucht. Ich habe Kaggle gefunden, ein Web-Machine-	Die Netzwerke, die ich bis jetzt entworfen habe, sind sehr schlecht. Sie lernen nicht mal, wie man richtig spielt. Ich warte also	Netzwerke trainieren	300 h

	Learning Dienst von Google, auf dem das Trainieren um ein Vielfaches schneller geht. Ich kann nur 42 Stunden pro Woche dann ein Netzwerk trainieren, aber das ist für mich schon von weitem genug. Commit: " Controller finished for move 0 started training "	gespannt auf das Buch, dass bessere Methoden enthalten sollte.		
12.11.2020 16:00 -18:00	Ich habe heute die wenigen Rewards, die es noch brauchte, eingestellt und festgelegt. Währenddessen habe ich das Netzwerk für den Spielzug "Setzen" trainiert und einen Fehler festgestellt. Ich hatte immer gleich viele Punkte verteilt, und so konnte er nichts lernen. Jetzt funktioniert es ein wenig besser.	Es ist mühsam, den Code jedes Mal auf Kaggle hochladen zu müssen, um ihn ausführen zu können. Es gibt auch eine Möglichkeit, dass ich es von meinem Computer aus hochlade, aber das müsste ich testen.	Netzwerke trainieren	300 h
14.11.2020 9:00 - 13:00	Ich war heute beim ICT-Campus, wo ich Leute habe, denen ich Fragen stellen kann. Ich habe anhand der Trainings der Netzwerke herausgefunden, dass es mit einer so Naiven Herangehensweise nicht funktionieren wird. Also habe ich mich im Internet umgeschaut und bin auf Monte Carlo Tree Search gekommen. Dort wird von einem Spielstand alle möglichen Züge ausprobiert und der, der am besten ist dann ausgeführt. Ich habe also heute versucht, es zu implementieren. Das schaffte ich, testen konnte ich das ganze aber noch nicht.	Die absolut besten Spiele-KIs, die es gibt (z.B AlphaZero) benutzen die Methode, die ich jetzt versuchen werde zu implementieren, kombiniert mit anderen Sachen. Durch diese Methode kann ich viel Zeit sparen, weil man sie nicht trainieren muss.	MCTS testen	5h
16.11.2020 14:10 -15:30	Ich habe versucht, das Monte Carlo Tree Search zum Laufen zu bringen. Dafür musste ich zuerst das MCTS in die Grafikumgebung einbauen, um es testen zu können. Danach habe ich noch ein paarmal das Laufen lassen und geschaut, welche Fehler entstehen. So konnte ich einen Teil davon beheben. Alle Fehler sind aber noch nicht weg, ich muss noch weitertesten.	Ich glaube trotz den vielen Fehlern daran, dass es am Ende ganz gut funktionieren wird, denn diese Methode hat keine Nachteile, ausser dass sie Zeitaufwendig ist.	MCTS testen	3h
18.11.2020 16:00 - 18:00	Ich habe das MCTS jetzt so weit geflickt, dass es läuft und man dagegen spielen kann. Das Problem ist jetzt einfach, dass es für einen Spielzug sehr lange braucht und auch sehr schlecht Spielzüge macht. Ich muss noch schauen wieso, aber dafür habe ich ja noch Zeit.	Mein Ziel ist es jetzt, die Architektur von AlphaZero nachzubauen, dann das ist der beste Spielalghorismus der Welt	MCTS testen	1h
19.11.2020	Das MCTS fängt allmählich an, gut zu spielen. Es schafft	Es freut mich, dass es	MCTS verbessern	10h

16:00 -17:00	<p>schon, einfach 2er-Grüppchen meinerseits zu unterbrechen und so zu verhindern, dass ich 3 in einer Reihe schaffe. Auch macht es eigene 2-er Grüppchen. Das grosse Problem ist aber, dass er diese nicht verwertet und schliesst, dass würde heissen dass er mir einen Stein wegnehmen könnte. Ich vermute, dass es da einen Fehler gibt. Diesen muss ich noch suchen. Commit: Monte Carlo Tree Search works but bad moves</p>	<p>langsam zu spielen lernt. Ich glaube, dass ich es bis nächsten Dienstag bei der PA-Stunde schaffe, einen passabel spielenden Agent zu haben. Aber es ist dann noch ein weiter Weg, bis er erstens gut spielt und er schneller werden kann. Auch wenn ich dann noch ein Neuronales Netzwerk einfügen will habe ich noch sehr viel zu tun. Denn AlphaZero ist hat die beiden Komponenten: Netzwerk und MCTS.</p>		
20.11.2020 8:00 - 9:00	<p>Ich hatte heute im InL nichts mehr zu tun und so habe ich an meiner PA weitergearbeitet. Ich hatte durch überlegen einen Fundamentalen Fehler im Code entdeckt: Ich wählte die Züge immer so, dass in Zukunft für beide die besten Züge möglich waren, nicht nur für den gerade spielenden Spieler. So hat mein Programm dann besser gespielt, wollte aber immer noch nicht selbst eine Mühle machen.</p>	<p>Es gibt immer noch Fehler im Code und in der Denkweise, den die Punkte, die ich dem Programm gebe, sind so ausgelegt, dass das Programm versuchen soll, die Mühle zu machen, und weniger das Abwehren des Gegners.</p>	MCTS verbessern	9h
21.11.2020 10:00 - 12:00	<p>Der Fehler, den ich gestern erwähnt habe, ist, dass die Zukünftigen Punkte viel zu stark gewichtet werden, und dass er so immer eine Mühle vorbereitet, aber nie schliesst. Ich habe mehrere Lösungsversuche programmiert, aber keiner davon hat funktioniert. darum werde ich eine Skizze machen, wie die Punkte vergeben werden müssen, bevor ich einen erneuten Versuch mache. Commit: Monte Carlo tree search not jet fully optimised</p>	<p>Wenn ich das geschafft habe und das Programm passabel spielt werde ich machen, dass das Programm gleichzeitig mehrere Spiele spielt und so das "Denken" vor dem Zug massiv schneller geht</p>	MCTS verbessern	7h
24.11.2020 14:10 - 15:30	<p>Die Skizze, die ich gemacht habe, hat mir sehr geholfen, aber alles, was darauf zu sehen ist, ist nicht sehr verständlich, da ich es halb im Kopf und halb auf dem Papier gemacht habe.</p>	<p>Ich habe jetzt sehr viele verschiedene Punkte vergeben, je nachdem, was</p>	MCTS verbessern	5h

	Das hat mir beim Implementieren sehr geholfen und es funktionierte dann so, wie ich es wollte. Das heisst, dass zukünftige Züge nicht übergewichtet werden und er auch wirklich versucht, zum jetzigen Zeitpunkt den besten Zug zu spielen. Danach habe ich einmal gegen die KI gespielt und dabei habe ich verloren, weil ich ganz am Anfang absichtlich schlecht gespielt habe. Ich habe auch geschaut, wie man machen kann, dass er parallel mehrere Spiele am Laufen hat.	er für Züge machte. Ich werde versuchen, so wenig Input wie möglich zu geben, damit es die besten Züge komplett selbstständig herausfinden kann.		
25.11.2020 9:00 -9:30	Ich habe dem Programm jetzt nur noch punkte gegeben, wenn es einen Stein vom Gegner wegnehmen konnte. So hat das Programm auch bessere eigenständige Taktiken entwickelt, brauchte aber auch sehr viel länger dafür. Weil Frau Önal krank war, hat uns die Vertretungslehrerin gesagt, wir könnten das machen, was wir wollen. So haben mehre Leute gegen meine Künstliche Intelligenz gespielt.	Das Programm hat gegen Oskar gewonnen. Somit war das der erste Sieg, bei dem ich nicht selbst dagegen gespielt habe und keine absichtlich schlechten Spielzüge gemacht wurden. Gegen Timo hat mein Programm aber verloren. Das heisst, dass ich die Punkte noch ein wenig verbessern muss.	MCTS verbessern	4h
28.11.2020 15:30 - 17:30	Ich habe sehr viele verschiedene Punktekombinationen ausprobiert, um die beste zu finden, bin aber noch nicht ganz fertig. Auch habe ich die Performance einzelner Funktionen verbessert. Ich habe auch erste Funktionen für das Multithreading implementiert, dh. das mehrere Spiele gleichzeitig ausgeführt werden.	Ich habe sogar eine Taktik von meinem Agent gelernt, die ich nicht kannte und mit der man eine Mühle erzwingen kann. Das heisst, dass sie allmählich sehr gut wird.	MCTS verbessern	2h
01.12.2020 14:10 - 15:40	Um die verschiedenen parallellaufenden Spiele zusammenzuführen musste ich viel testen und schauen, dass nicht immer das gleiche rauskommt, weil es nur pseudo-random spielt. Danach musste ich sie auch parallel laufen lassen, und testen wie viel schneller es so wird.	ca. 4. mal schneller ist das Programm, wenn es auf mehreren Prozessorkernen läuft. Das ist schon recht gut. Aber ich werde wahrscheinlich dem Programm nicht weniger Zeit lassen, um zu spielen, denn dann ist das Programm	Performance verbessern	5h

		besser. Ich habe das Ganze jetzt nur testweise geschrieben und werde das jetzt besser in Funktionen umwandeln		
08.12.2020 14:10 - 15:40	Ich war während den ersten 45 Minuten bei der Einführung für den Schriftlichen Teil der Arbeit. Dort habe ich das Inhaltsverzeichnis und die Verschiedenen Kapitel erstellt. Danach habe ich nachgeschaut, wie man bei einem Programm sehen kann, welche Teile länger brauchen und welche weniger lang. Damit konnte ich dann herausfinden, wo ich schlechten Code geschrieben habe und diesen dann verbessern. Commit: Profiler + parallel , Profiler improvements	von 1 min 20 s zu 20 - 30 s für den ersten Zug ist eine sehr deutliche Verschnellerung und ich bin recht stolz darauf, wie ich die schlechten Code-Stellen verbessert habe. Für die Züge in der Mitte vom Spiel braucht die KI aber noch entschieden weniger lang.	Performance verbessern	3h
12.12.2020 9:00 - 12:30	Ich habe heute den Kurs für das Phänovum durchgeführt, um das Geld für das Beschaffen des Buches zu verdienen. Dieser lief gut, obwohl wir nur 2 Teilnehmer hatten.	Mit dem Lohn und den 15 CHF der Schule werde ich das Buch sehr einfach kompensieren können.	Performance verbessern	3h
13.12.2020 11:00 - 12:30	Ich habe ein paar Parameter geändert, um die KI noch ein wenig zu verbessern, aber mit dieser Herangehensweise wird sie nicht mehr stärker. Ich werde darum noch ein neuronales Netzwerk einbringen, damit es noch besser wird.	Es wird sehr schwer, das zu machen, denn diese Idee wurde von Professionellen Forschern erdacht und zum ersten Mal implementiert. Auch wird es sehr schwer, genug Performance für das Trainieren zu haben.	Netzwerke einbringen	20h
15.12.2020 14:10 - 15:45	Ich habe mein Neuronales Netzwerk so modifiziert, dass es so aussieht wie das von AlphaZero. Das heisst, dass es ein Residual Layers hat, das sind spezielle Neuronen, bei denen anders gelernt wird als bei den normalen. Danach hatte ich noch das Standortgespräch mit Herr Haldimann.	Dank Keras geht es sehr einfach, schwierige Netzwerke zu definieren. Ich habe eine Idee gehabt, wie man bessere Ergebnisse beim MCTS erzielen könnte. Diese werde ich beim nächsten Mal versuchen zu implementieren.	Netzwerke einbringen	19h

19.12.2020 14:00 - 16:00	Die Idee, die ich am Dienstag hatte, bestand darin, dass man immer nur bis zum nächsten Zug spielt, bei dem man einen Stein wegnehmen darf. Das könnte das ganze verschnellern. Leider funktionierte dieser Ansatz nicht, da es manchmal sehr lange braucht, bis man an einem Punkt ist, wo man einen wegnehmen kann.	Ich habe sehr lange für das gebraucht, aber nebenbei auch ein Artikel über AlphaZero gelesen. Ich bin nicht sicher, ob ich es wirklich noch schaffe, es einzubringen, denn es besteht aus vielen Elementen, die Perfekt zusammenspielen müssen	Netzwerke einbringen	19h
20.12.2020 16:00- 17:30	Ich hatte beim Spiele oft das Gefühl, dass die KI nicht auf die lange Sicht spielt und oft sehr aggressiv immer versuchte, sofort eine Mühle zu machen. Das lag daran, dass es viel öfter durch die Möglichkeiten des Spielens ging, weil es weniger andere Möglichkeiten gibt. Dem habe ich entgegengewirkt, indem ich der KI für diese Züge weniger Punkte gab. Das funktionierte sehr gut. Aber ich bemerkte, dass dann oft eine Patt-Situation entstand, das heisst dass es für beide Seiten unmöglich war, eine Mühle zu machen, solange der Andere Spieler keinen Fehler machte. Also schaute ich mir nach den offiziellen Regeln um, und fand heraus, dass es unentschieden ist, wenn eine Position zweimal auftaucht. Diesen Umstand programmierte ich dann ein. Commit: long-term play + same Position Rule .	Weil bei der neuen Spielregel alle Positionen gespeichert werden müssen, ist das ganze jetzt viel langsamer. Dafür ist es mir aber nicht mehr möglich, gegen meine KI zu gewinnen, auch wenn ich aus meiner Sicht aus perfekt spiele.	Netzwerke einbringen	19h
23.12.2020 16:00-17:30	Ich versuchte das Ergebnis zu wiederholen, aber es funktionierte nicht mehr wie zuvor, und die KI spielte wieder schlecht. Ich versuchte viele verschiedene Methoden, um die KI wieder zum Laufen zu bringen, aber es funktionierte nicht. Also habe ich den Code in verschiedene Dateien aufgeteilt, um eine bessere Übersicht zu haben. Commit: multiple files + long term not working any more	Ich frage mich, was ich geändert habe, denn ich sehe nichts. Ich bin auch auf den Status vom letzten Commit zurückgegangen und es funktionierte auch nicht mehr.	Netzwerke einbringen	19h
26.12.2020 16:00-17:30	Ich habe testweise weniger Exploration zugelassen und das Auswählen als gut angesehene Züge bevorzugt beim Suchen. Das hat gut funktioniert und die KI wieder gut zum Laufen gebracht. Danach bin ich auf dem Repository auf eine andere Branch gewechselt, das heisst dass ich zwei verschiedene	Ich war mir am Anfang bei der Idee, AlphaZero zu implementieren noch nicht bewusst, dass es so kompliziert ist und auch viel	Netzwerke einbringen (AlphaZero implementieren)	29h

	<p>Kopien meines Codes habe: eine die ohne Netzwerk und nur mit MCTS funktioniert und eine, bei der ich an AlphaZero herumexperimentiere. Danach habe ich noch ein paar Artikel über die Funktionsweise von AlphaZero herausgesucht, und diese gelesen, um den Algorithmus ganz zu verstehen und selbst implementieren zu können. Commit: long-term play works again</p>	<p>Zeit in Anspruch nehmen wird, aber ich glaube, dass ich das hinbekommen werde.</p>		
<p>30.12.2020 11:00-12:30</p>	<p>Ich habe das Netzwerk, das Lernen muss, zum Laufen gebracht, denn davor konnte ich keine Daten durchschleusen, ohne das es einen Error gab. Dafür habe ich eine Unittest-Datei erstellt. Unittests testen kleine Portionen Code, meist nur eine Funktion. Dort kann man dann Vergleiche einfach erstellen, die schauen, dass die Funktion wirklich das zurückgibt, was sie soll. Auch das Codieren des Brettes, damit es einfach ist, daraus zu lernen, habe ich gemacht. Commit: Network working + tests</p>	<p>Für einen Durchlauf des Netzwerks braucht mein Computer 0.02 s. Das mag nach recht wenig aussehen, aber das muss dann sehr oft laufen: pro Zug wird es ca. 100-mal durchlaufen, ein Spiel dauert ca. 150 Züge, pro Trainingsdurchlauf werden ca. 150 Spiele gespielt und mindestens 20 Trainingsdurchläufe werden es schon brauchen. Das macht dann schon 10,5 Tage nur für das Generieren von Trainingsdaten. Danach muss das Netzwerk aber auch noch trainiert werden, und das Dauert dann auch ca. gleich lang. Aus diesem Grund wird es nicht möglich sein, das auf meinem Rechner durchzuführen. Rechner mit einer Starken GPU sind beim Trainieren eines Netzwerkes um bis zu 100-mal schneller, und das will ich dann ausnützen. Das Problem ist dann aber, solch einen</p>	<p>Netzwerke einbringen (AlphaZero implementieren)</p>	<p>27h</p>

		Computer für eine Längere Zeit zur Verfügung zu haben.		
2.01.2021 11:00 - 12:30	Eine Strategie, um schneller mehr Beispiele für das Netzwerk zum Trainieren zu haben, ist, die natürlichen Symmetrien im Spiel auszunützen. Denn wenn man das Spielbrett dreht hat es immer noch den gleichen Wert, das heisst die Wahrscheinlichkeit, von dort aus zu gewinnen. So habe ich die Funktionen, die für das Erstellen dieser Symmetrien gebraucht werden, implementiert. Commit: generate symetries	Die Tests haben sich bewährt, denn ich kann einfach einen der Tests immer schnell ausführen, aber auch alle zusammen, um zu schauen, ob alles noch funktioniert.	Netzwerke einbringen (AlphaZero implementieren)	25h
3.01.2021 0:00- 11:30	Ich habe viele der Hyperparameter, dass heisst die Konstanten, die man für ein gutes Training festlegen muss, herausgesucht und ihnen einen plausiblen Wert gegeben.	Einen guten Wert zu finden, ist sehr schwierig, denn noch niemand hat vor mir AlphaZero für Mühle implementiert. Auch sind nicht alle Webseiten gleicher Meinung, wenn es darum geht, wie man den Optimal Wert findet.	Netzwerke einbringen (AlphaZero implementieren)	23 h
5.01.2021 14:10 -15:40	Ich habe die Klasse, die einen bestimmten Spielstatus darstellt, umprogrammiert, damit man damit die Methoden von AlphaZero anwenden kann, den davor konnte man sie nur für das MCTS benutzen.	Es ist manchmal schwer, die Methoden zu ändern, wenn man sie schon mal programmiert hat, denn es gibt immer noch neue Randfälle zu beachten, die in der Implementierung eingeschlossen werden müssen.	Netzwerke einbringen (AlphaZero implementieren)	21 h
07.01.2021 16:00 - 17:30	Ich habe eine einzelne Suche, dass heisst das wiederholte Simulieren eines Spieles, um den Besten Zug zu finden. Programmiert und getestet. Dabei hat sich im Vergleich zum normalen MCTS geändert, dass man nicht mehr bis zum Schluss des Spieles spielt, sondern nur bis zum Nächsten Zug, der noch nicht simuliert wurde. Dieser wird dann durch das Neuronale Netzwerk bewertet und deren Wert berechnet. So geht es schneller und genauer.	Für der ersten Zug des Spieles braucht es jetzt 9 Sekunden, um sich für einen Zug zu entscheiden, aber die benötigte Zeit sollte mit der Zeit rapide sinken.	Netzwerke einbringen (AlphaZero implementieren)	19 h
11.01.2021	Ein Spiel ganz durchzuspielen, bis es unentschieden steht	Das einmalige Spielen mit	Netzwerke	17 h

16:00 - 17:30	oder einer gewonnen hat habe ich programmiert. Das wurde von mir auch getestet. Es funktioniert auch, denn ich habe es getestet. Ein Spiel geht jetzt so ab. er simuliert mehrere Male einen Zug, wie ich am 07.01.2020 programmiert hatte. Dann nimmt es den besten Zug und Speichert die Daten von der Simulation, um diese dem Netzwerk zu lernen. Dabei werden die Symetrien ausgewertet und auch mitgespeichert.	den Einstellungen, die ich geplant habe, dauert 3-5 minuten. Das macht es noch wichtiger, einen Computer zu finden, auf dem ich es laufen lassen kann.	einbringen (AlphaZero implementieren)	
12.01.2021 14:10 - 15:40	Das Netzwerk konnte man nicht trainieren, denn es hat zwei outputs: den für die wahrscheinlichkeit des wählen jedes Spielzuges von einer bestimmten position aus und den des Werts, also ob man wahrscheinlich gewinnen oder verlieren wird. Um das mit den gewonnen Daten zu trainieren, musste ich anders programmieren.	Ich kann ihn mit Daten trainieren, die nicht von den Simulationen kommen, aber die Daten der Simualtionen sind irgendwie nicht benutzbar.	Netzwerke einbringen (AlphaZero implementieren)	15 h
14.01.2021 16:00 - 17:30	Ich habe versucht das Generieren von Spielen zu verschnellern, indem ich es wie das MCTS parallel laufen lies, aber wegen dem Netzwerk geht das nicht. Auch das Problem mit dem formatieren der Daten für das Trainieren des Netzwerkes habe ich versucht zu lösen, aber keinen Erfolg erzielt.	Weil das nicht funktioniert bin ich gerade in einem Stimmungstief, denn ich habe wieder Zweifel, ob es wirklich funktionieren wird. Aber ich werde mich dranhängen und hoffe, es zu schaffen	Netzwerke einbringen (AlphaZero implementieren)	13 h
16.01.2021 10:00-11:30	Ich habe das trainieren mit den Daten für das Netzwerk jetzt doch noch geschafft. Das Problem war gestern, das die Daten für das Netzwerk beim generieren der Spiele anders formatiert war als beim trainieren des Netzwerkes. Jetzt muss ich noch das gegeneinander Spielen zweier Netzwerkversionen programmieren und alle drei Schritte, also gegen sich selbst spielen, netzwerk trainieren und gegen ältere version spielen, zusammenführen. Vielleicht schaue ich noch, ob es funktioniert, es wieder parallel zu gestalten. Commit: generate play + fit	Da es jetzt funktioniert hat ist es besser, aber die zwei schritte zusammen dauern ca. 8 minuten. Darum ist das Parallele spielen eben so wichtig.	Netzwerke einbringen (AlphaZero implementieren)	11 h

19.01.2021 14:10 - 15:45	Zuerst habe ich programmiert, das die neue, trainierte Version der KI gegen seine Ältere Version spielt, um zu schauen, ob das Lernen funktioniert. Das ging sehr schnell, denn ich konnte viel vom gegen sich selbst Spielen übernehmen. Ich habe eine Bibliothek, dass heisst schon vorgeschriebene Programmcodes, gefunden, mit der es funktioniert, die Netzwerke parallel laufen zu lassen. Sie heisst Ray (https://ray.io/). Ich musste dann ganz viele verschiedene Tests machen, um herauszufinden, wie sie wirklich funktioniert und wie ich sie in mein Programm einbauen könnte. Ich hatte aber keine Zeit mehr, es dann effektiv einzubauen, denn ich musste lange mit den Objekten kämpfen, die im gemeinsamen Speicher sind und auf die alle parallel laufenden Prozesse Zugriff haben müssen. Commit: ray_test and pit	Das ist eine sehr gute Nachricht, denn sie wird das ganze voraussichtlich 4 mal Schneller machen. Das ist schon ordentlich viel.	Netzwerke einbringen (AlphaZero implementieren)	9 h
20.01.2021 16:30 - 18:00	Ich habe diese Library in meinen Code eingebracht und kann jetzt alles parallel laufen lassen. Auch mit dem Speicher, der geteilt werden muss, habe ich keine Probleme mehr. Tatsächlich ist es jetzt nur ca 3 mal schneller, weil es manchmal pro Process mehr als einen CPU core braucht, aber das wird sich mit der Grafikkarte ändern. Ich habe auch noch einen Fehler gefunden, der machte, das die Spiele sehr oft unentschieden ausgingen.	Ich muss jetzt nur noch den Teil Programmieren, der alle funktionen in der Richtigen Reihenfolge abruf. Das wird nicht mehr so viel Zeit in anspruch nehmen. Auch habe ich mich für den ICT Campus am Samstag angemeldet, wo ich an meinem Projekt weiterarbeiten können werde. Ich habe auch schon dort angegeben, dass ich einen Computer gerne für 2 Wochen gebrauchen würde. Ich bin zuversichtlich, dass ich dort einen finden werde.	Netzwerke einbringen (AlphaZero implementieren)	3 h
21.01.2021 17:00 - 18:30	Ich habe jetzt die main gemacht, dass heisst die Datei, wo alles ausgeführt wird. Dort wird zuerst das Netzwerk initialisiert, dann Spiele gegen sich selbst gespielt, dann das Netzwerk trainiert und dann wird geschaut, welches Netzwerk besser ist. Das bessere Netzwerk wird dann für die nächsten	Jetzt kommt wirklich der Endspurt und das suchen eines guten Computers, denn ich bin fast fertig. Ich muss für das Programm nur noch	Netzwerke einbringen (AlphaZero implementieren)	1,5 h

	<p>Spiele gebraucht. So geht das immer weiter. Das funktionierte auf meinem Rechner mit sehr abgespeckten Konfigurationen, dass heisst das weniger Spiele und Simulationen ausgeführt werden. Commit: main Working, tensorBoard callback problems</p>	einen Logger programmieren, dass heisst etwas, dass immer mitschreibt, was gerade im Programm passiert und mit dem man Fehler zurückverfolgen kann.		
23.01.2021 9:00 - 10:30	<p>Ich hatte heute Online-Campus, dort habe ich Zeit und Leute, die ich fragen kann, um weiterzukommen. So konnte ich sehr schnell meinen Logger programmieren. So bin ich jetzt mit meinem Programm komplet fertig. Danach habe ich noch die "Chefin" von ihnen gefragt, ob ich einen Computer für das Trainieren ausleihen könnte. Sie hat eingewilligt, mir beim suchen eines Computers zu helfen. Gefunden habe ich aber leider noch keinen. Commit: requirements.txt + logger + own activation</p>	Ich bin in der geplanten Zeit mit dem Programm eigentlich fertig geworden, ich hoffe einfach, das das Trainieren und das finden eines Computers keine Probleme machen wird.	Computer organisieren	4 h