

**TECHNISCHE
HOCHSCHULE
LÜBECK**

TECHNISCHE HOCHSCHULE LÜBECK
FACHBEREICH INFORMATIK UND SOFTWARETECHNIK
Informatik / Softwaretechnik, B.Sc.

Intelligente Systeme

Autor	Matr.-Nr.
Tim Lüneburg	321226
Denis Alipkina	326771

Wintersemester 2021 & 2022
Lübeck - 19. Dezember 2021

Inhaltsverzeichnis

1. Bird of Prey	4
1.1. Ausgangssituation und Zielsetzung	4
1.2. Aufgabenstellung	4
2. Aufgabe 1	5
3. Aufgabe 2	10
3.1. Schritt 1: Observation	10
3.2. Schritt 2: Aufstellung der Häufigkeitsverteilungen	10
3.3. Schritt 3: Häufigkeitsverteilung in Wahrscheinlichkeiten umwandeln	10
3.4. Schritt 4: Anwenden der Bayes'sche Klassifikation	10
3.5. Schritt 5: Argmax - Auswählen des Optimalsten	10
4. Aufgabe 3	12
5. Implementierung zum Durchlaufen aller 10000 Zeitschritte	14
A. Sonstige Überlegungen	16
A.1. Mindestanzahl der Beobachtungen	16
A.2. Bekannte Beutetiere essen	16
A.3. Nach dem Essen lernen	16
A.4. Merken des Schlechtesten	16
A.5. Gute Beutetiere merken	16
A.6. Laufzeit optimieren	17

Abbildungsverzeichnis

2.1. Häufigkeitsverteilung der Richtungsänderungen der guten Beutetiere	5
2.2. Häufigkeitsverteilung der Richtungsänderungen der giftigen Beutetiere	6
2.3. Häufigkeitsverteilung der Richtungsänderungen der neutralen Beutetiere	6
2.4. Häufigkeitsverteilung der Laufrichtung der Beutetiere	8
2.5. Häufigkeitsverteilung der Schrittlänge der Beutetiere	9
4.1. Gesammelte Punkte	12
4.2. Überlebte Zeit	13
4.3. Energie bei dem Ende oder Tot in der Welt	13
5.1. Berechnung in der zweiten Implementation	14
5.2. Gesammelte Punkte und der Durchschnitt pro Level	15

1. Bird of Prey

1.1. Ausgangssituation und Zielsetzung

In dieser Aufgabe geht es darum einen Agenten zu entwickeln, welcher in einer Welt möglichst lange überlebt. In der Welt existieren drei Arten von Beutetiere:

- Gut ist nahrhaft (+5 Energiepunkte)
- Giftig bringt den Agenten um
- Neutral ist ungefährlich und liefert keine Energie

Um in der Welt zu überleben muss der Agent Gute Beutetiere essen, denn bei jedem Zeitschritt, verliert der Agent einen Energiepunkt.

Die Beutetiere sind äußerlich sehr ähnlich und lassen sich nicht unterscheiden. Deswegen gibt es eine 300 Zeitschritte am Anfang einer Spielrunde zum beobachten der Beutetiere.

1.2. Aufgabenstellung

1. Untersuchen Sie die Beobachtungsdaten, die aus der Anfangsphase der Spielrunden gewonnen werden können und entwickeln Sie einen Ansatz zur Klassifikation der drei Arten von Beutetieren.
2. Entwickeln Sie Programmcode für einen Agenten, der in der oben beschriebenen Welt so viel Energie wie möglich sammelt. Ermitteln Sie die Leistungsfähigkeit Ihres Agenten für alle verfügbaren Spielrunden (d.h. mit allen 100 Dateien in 2021-isys3-data.zip). Achtung: Die Spielrunden müssen unabhängig voneinander durchgeführt werden, d.h. für eine Spielrunde dürfen keine Informationen aus vorangegangenen Spielrunden verwendet werden. Der Agent darf also nur diejenigen Informationen über die Beutetiere verwenden, die er aus den Beobachtungen am Anfang der jeweiligen Runde gewonnen hat.
3. Stellen Sie die von Ihrem Agenten erzielten Ergebnisse in einer geeigneten Form dar. Im Endergebnis sollten im Mittel über alle 100 Spiele mindestens 15000 Energiepunkte gesammelt werden.

2. Aufgabe 1

Hinweis: Bei den Untersuchungen haben wir uns nur auf Welt 1 bezogen. Bei dem Untersuchen der Beobachtungsdaten, haben wir uns zuerst darauf fokussiert die Richtungen in die ein Beutetier läuft zu untersuchen. Dazu haben wir uns angeschaut wie oft ein Beutetier die Richtung wechselt. Bei betrachten der [Abbildung 2.1 Häufigkeitsverteilung der Richtungsänderungen der guten Beutetiere](#) ist zu sehen, dass die Guten Beutetiere sich im Bereich von 0 bis 11 Richtungswechsel befinden und ab 12 Richtungswechsel die Häufigkeit stärker abfällt.

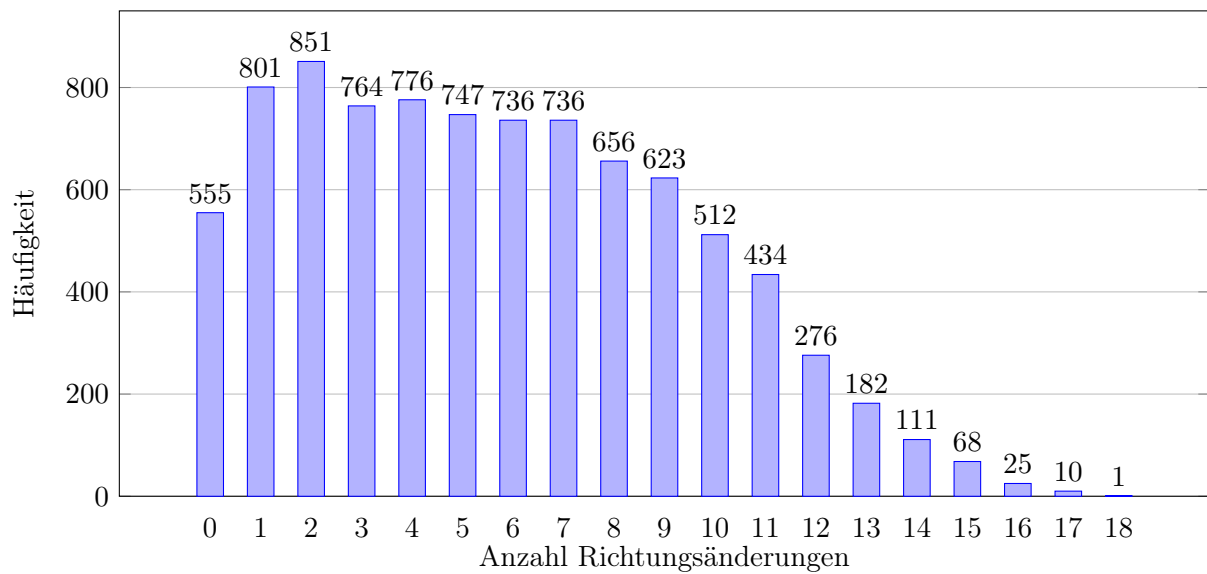


Abbildung 2.1.: Häufigkeitsverteilung der Richtungsänderungen der guten Beutetiere

Wenn wir die [Abbildung 2.2 Häufigkeitsverteilung der Richtungsänderungen der giftigen Beutetiere](#) und die [Abbildung 2.3 Häufigkeitsverteilung der Richtungsänderungen der neutralen Beutetiere](#) betrachten, lassen sich zwischen den giftigen und neutralen Beutetieren kaum Unterschiede erkennen.

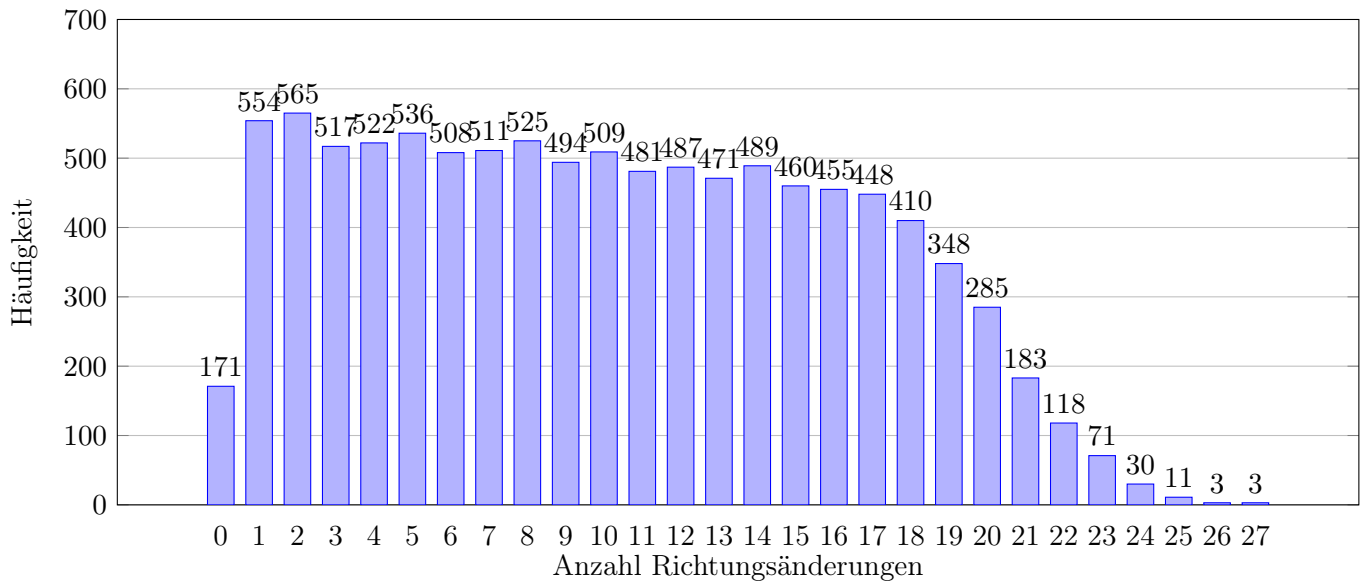


Abbildung 2.2.: Häufigkeitsverteilung der Richtungsänderungen der giftigen Beutetiere

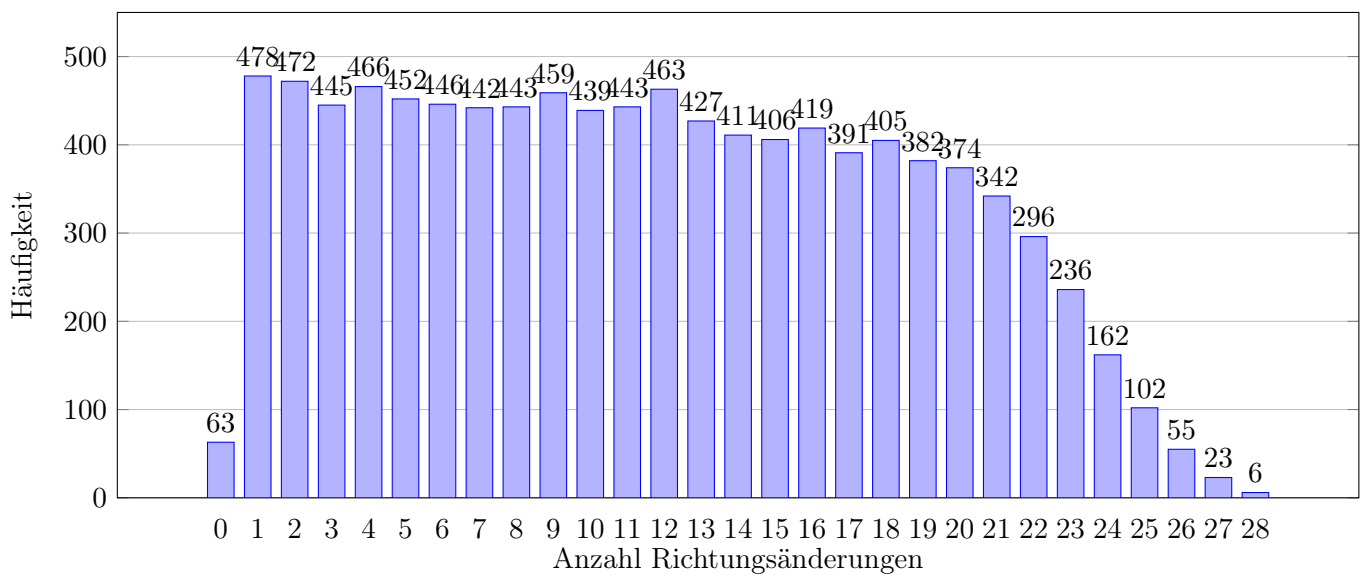


Abbildung 2.3.: Häufigkeitsverteilung der Richtungsänderungen der neutralen Beutetiere

Auf Grund dessen haben wir uns dazu entschieden zu beobachten in welche Richtung ein Beutetier läuft. Dabei haben wir festgelegt, dass es diese Möglichkeiten gibt:

- VOR
- RECHTS
- RUNTER
- LINKS

In welche Richtung ein Beutetier läuft ist folgender Tabelle zu entnehmen:

Letzte Ausrichtung	Aktuelle Ausrichtung	Laufrichtung
0	0	VOR
0	90	RECHTS
0	180	RUNTER
0	270	LINKS
90	0	LINKS
90	90	VOR
90	180	RECHTS
90	270	RUNTER
180	0	RUNTER
180	90	LINKS
180	180	VOR
180	270	RECHTS
270	0	RECHTS
270	90	RUNTER
270	180	LINKS
270	270	VOR

Wenn man nun die Daten in der [Abbildung 2.4 Häufigkeitsverteilung der Laufrichtung der Beutetiere](#) betrachtet, ist zu erkennen dass die guten Beutetiere viel öfter **VOR** laufen als die giftigen und neutralen. Die neutralen Beutetiere laufen öfter nach **LINKS** und nach **RECHTS** und die Giftigen laufen auch oft **VOR** aber auch sehr oft nach **LINKS** im Vergleich zu den guten Beutetiere. Anhand dieser Häufigkeitsverteilung haben wir uns entschieden diese Merkmal für die Klassifikation zu verwenden.

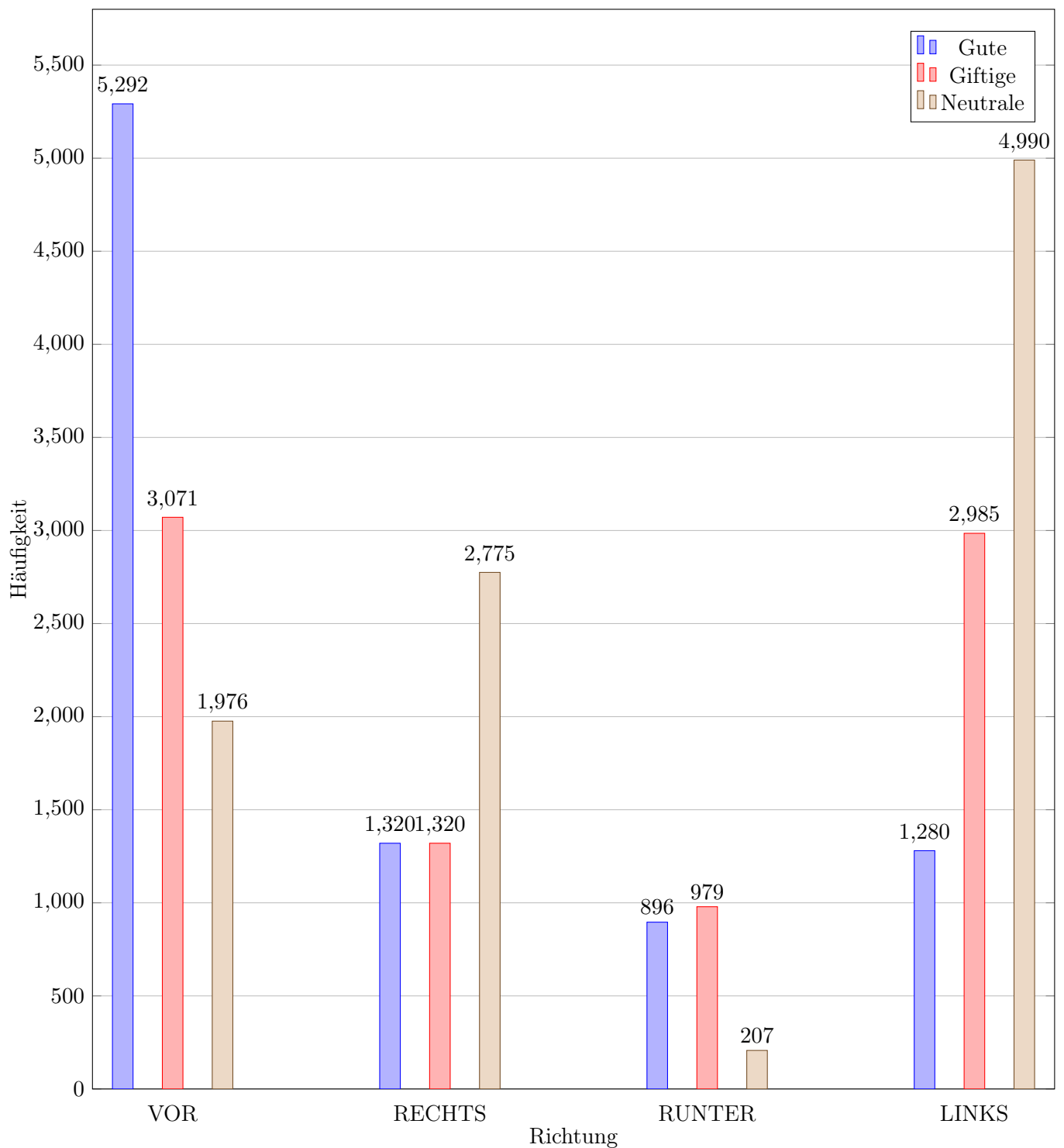


Abbildung 2.4.: Häufigkeitsverteilung der Laufrichtung der Beutetiere

Außerdem haben wir noch die Schrittlänge der Beutetiere beobachtet. Und wenn man die [Abbildung 2.5 Häufigkeitsverteilung der Schrittlänge der Beutetiere](#) betrachtet, ist zu erkennen, dass die Guten und Neutralen Beutetiere viel öfter die Schrittlänge 1 im Vergleich zu den Giftigen haben. Die Neutralen machen ebenfalls wesentlich öfter einen Schritt der Länge 1 als die Guten, jedoch haben die

Neutralen kaum zweier und dreier Schritte. Die Guten dagegen haben öfter auch mal eine Schrittlänge von 2. Bei den Giftigen ist es relativ gleichverteilt. Anhand dieser Häufigkeitsverteilung haben wir uns ebenfalls dazu entschieden dieses Merkmal für unsere Klassifikation zu verwenden, da man klare bzw. eindeutige Grenzen/Schwellwerte erkennen kann.

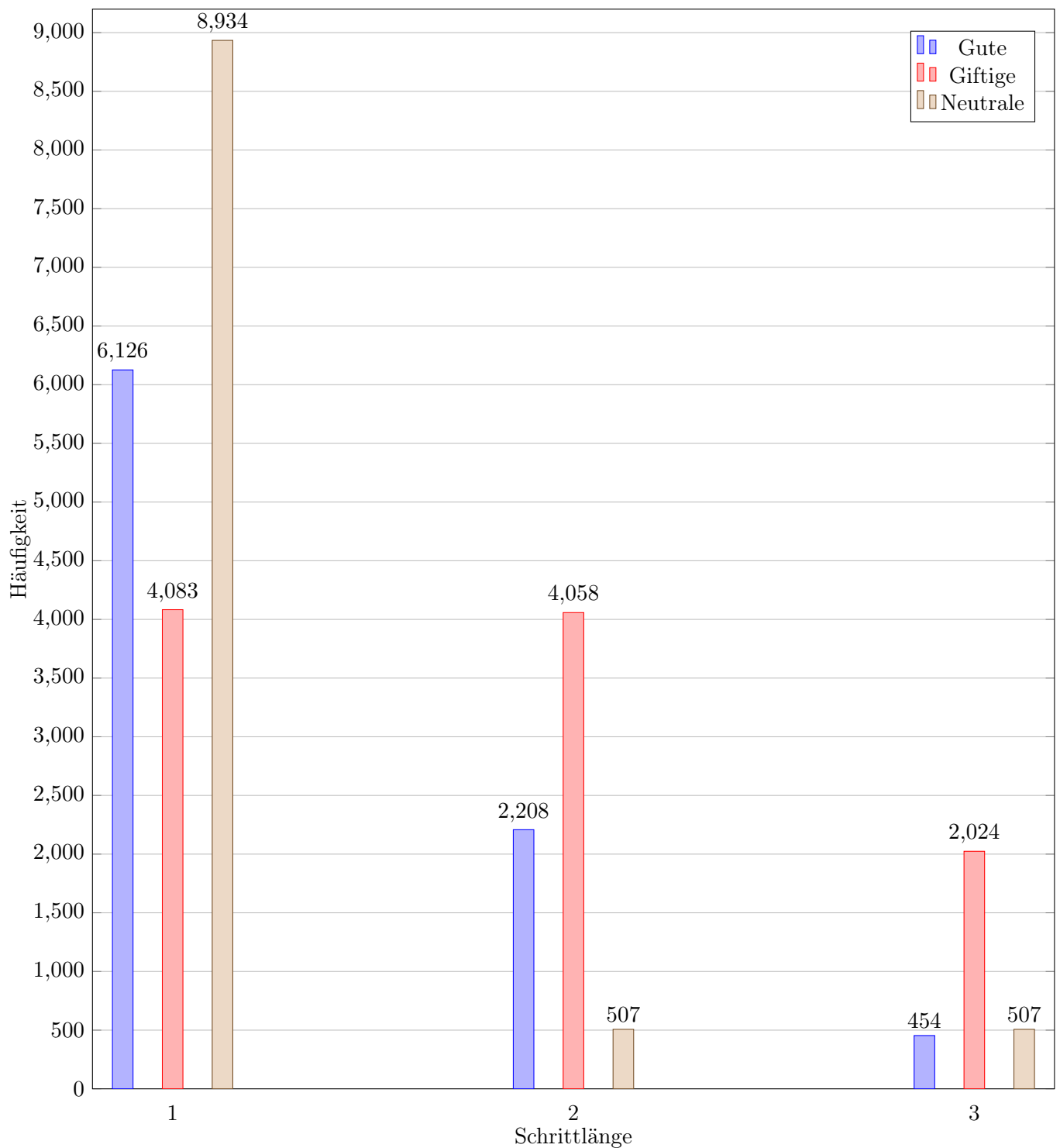


Abbildung 2.5.: Häufigkeitsverteilung der Schrittlänge der Beutetiere

3. Aufgabe 2

3.1. Schritt 1: Observation

In der Observation werden alle Daten aus den ersten 300 Zeitschritten in einer Map gespeichert.

3.2. Schritt 2: Aufstellung der Häufigkeitsverteilungen

Es findet eine Analyse der Laufrichtung der Beutetiere statt und wird als Häufigkeitsverteilung in drei verschiedenen Maps: Gute, Giftige und Neutrale Beutetiere gespeichert.

Ebenfalls findet eine Analyse für die Schrittlänge statt und wird als Häufigkeitsverteilung in drei verschiedenen Maps: Gute, Giftige und Neutrale Beutetiere gespeichert.

3.3. Schritt 3: Häufigkeitsverteilung in Wahrscheinlichkeiten umwandeln

Die aus [Schritt 3: Häufigkeitsverteilung in Wahrscheinlichkeiten umwandeln](#) gewonnene Häufigkeitsverteilungen werden in Wahrscheinlichkeiten umgewandelt und in drei verschiedenen Maps: Gute, Giftige und Neutrale Beutetiere gespeichert. Zeitgleich um etwas Rechenzeit zu sparen werden für diese Wahrscheinlichkeiten gleich der Logarithmus gespeichert, da wir im nächsten [Schritt 4: Anwenden der Bayes'sche Klassifikation](#) die Wahrscheinlichkeiten miteinander multiplizieren würden und dann sehr kleine Werte bekommen würde. Dies würde unsere Rechengenauigkeit gefährden und wir addieren stattdessen die Logarithmen der Wahrscheinlichkeiten.

3.4. Schritt 4: Anwenden der Bayes'sche Klassifikation

Da hier eine Markov-Kette der 0-Ordnung vorliegt und dadurch die Werte als unabhängig von einander betrachtet werden, ist es möglich die Werte miteinander zu multiplizieren. Wie schon in [Schritt 3: Häufigkeitsverteilung in Wahrscheinlichkeiten umwandeln](#) angedeutet, rechnen wir hier die Wahrscheinlichkeiten für ein Beutetier zusammen, dabei multiplizieren wir nicht mehr die Wahrscheinlichkeiten, sondern addieren die Logarithmen der Wahrscheinlichkeiten zusammen.

3.5. Schritt 5: Argmax - Auswählen des Optimalsten

Hier wird auf Basis der Werte aus [Schritt 4: Anwenden der Bayes'sche Klassifikation](#) ermittelten Werte entschieden ob ein Beutetier gefressen werden soll oder nicht. Und welches das optimalste Beutetier zum Verzehr sei.

Betrachten tun wir nur Beutetiere die eine bestimmte Anzahl an Zeitschritten überlebt haben. Bei diesen Beutetieren muss die Wahrscheinlichkeit, dass es ein gutes Beutetier ist größer sein als ein

neutrales Beutetier. Ebenso muss die Wahrscheinlichkeit, dass es ein gutes Beutetier ist mindesten um 10 größer sein als ein giftiges Beutetier. Dabei haben wir den Wert von 10 durch einfaches Probieren herausgefunden. Damit haben wir die durchschnittlich gesammelte Punktzahl optimiert.

4. Aufgabe 3

Bei Durchlaufen aller 100 Welten erhalten wir einen Durchschnitt von **30407** gesammelten Punkten. Wenn man die [Abbildung 4.1 Gesammelte Punkte](#) betrachtet sieht man die orangene Linie die den Durchschnitt darstellt, nicht unter die 30000 Punkte Marke geht. Ebenfalls ist zu erkennen dass wir sehr oft Punkte in dem Bereich von 30000 bis 35000 Punkte in einer Welt erreichen. Aber es gibt auch leider viele Welten in denen wir im Vergleich sehr wenig Punkte erreichen. Dies zieht den Durchschnitt der gesammelten Punkte noch sehr stark runter.

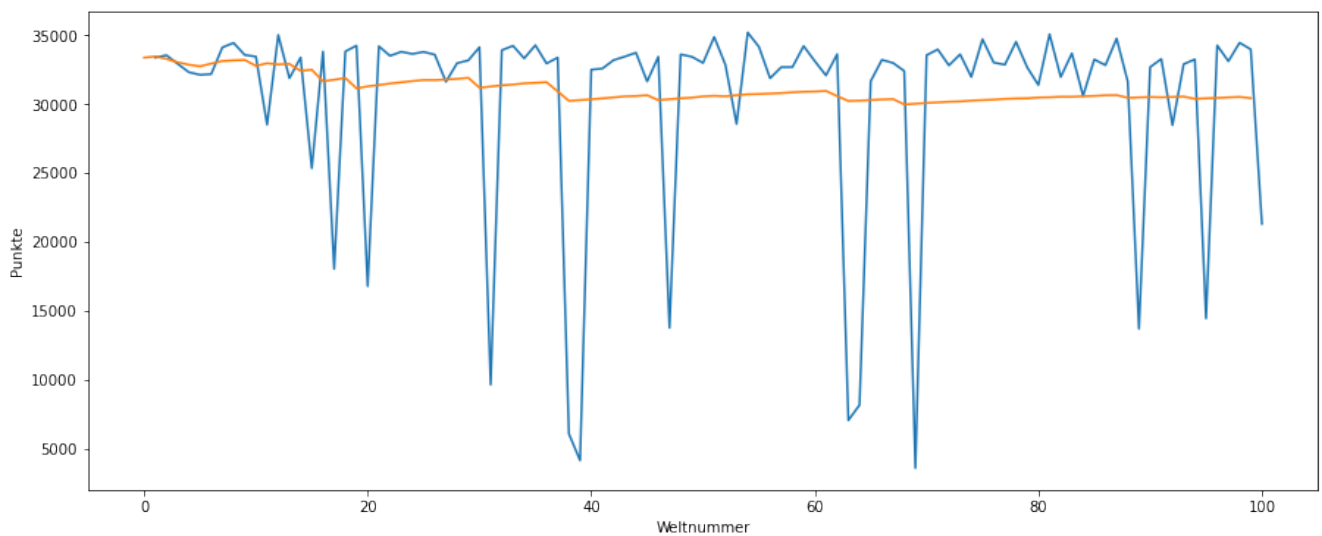


Abbildung 4.1.: Gesammelte Punkte

Wenn man sich die überlebte Zeit des Agenten in der [Abbildung 4.2 Überlebte Zeit](#) anschaut ist zu erkennen, dass es einige Welten gibt in denen der Agent bis zum Ende der Welt überlebt hat und keinen Giftigen gegessen hat. Aber bei den Welten wo wir wenig Punkte erreicht haben, ist in Betrachtnahme der [Abbildung 4.3 Energie bei dem Ende oder Tot in der Welt](#) zu erkennen, dass wir nie gestorben sind weil wir keine Energie mehr haben, sondern immer einen giftigen gegessen haben.

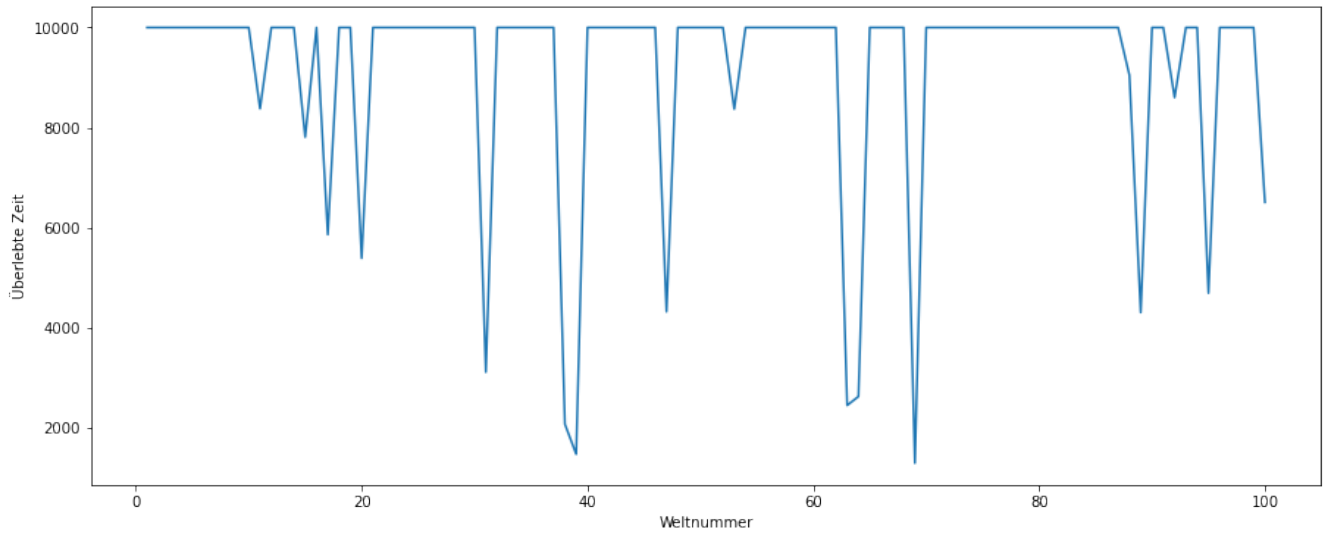


Abbildung 4.2.: Überlebte Zeit

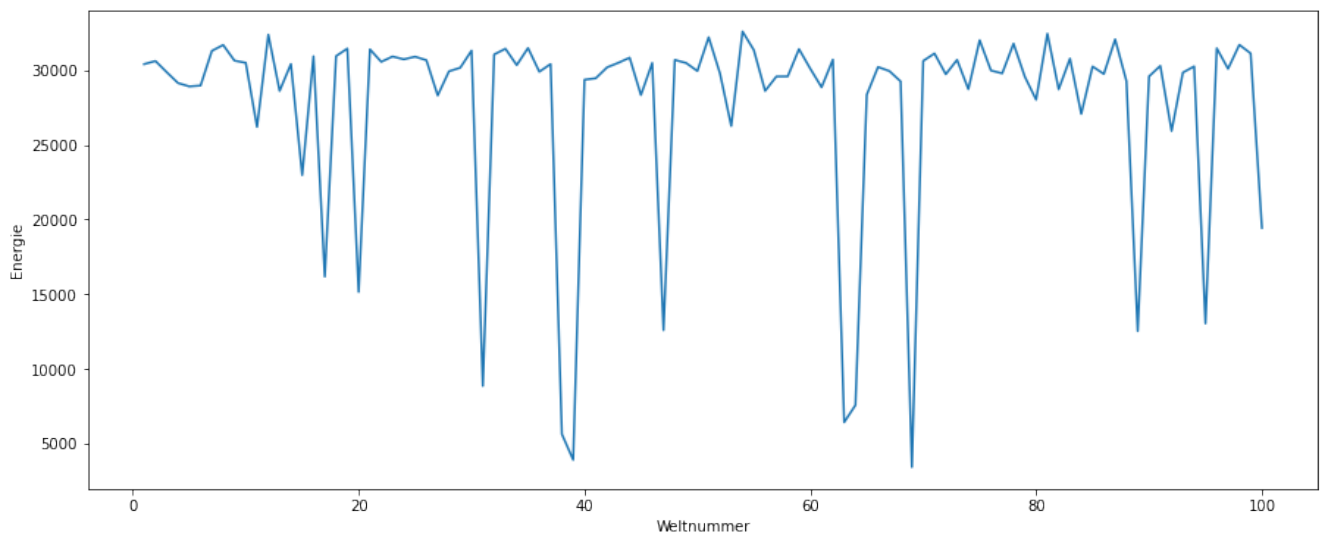


Abbildung 4.3.: Energie bei dem Ende oder Tot in der Welt

5. Implementierung zum Durchlaufen aller 10000 Zeitschritte

Im Vergleich zur Implementierung aus Aufgabe 2, gibt es nur kleine Unterschiede wie z.B. die Klassifikation. Wenn der Agent eine Abfolge von Schritten beobachtet, wendet er für jedes Merkmal (Schrittlänge und Richtung) eine Rechnung durch. Die Rechnung ist der [Abbildung 5.1 Berechnung in der zweiten Implementation](#) abgebildet. Der Wert von SL muss nicht wie im Beispiel den Wert 1 annehmen. Alle möglichen Werte sind dem Diagramm zu entnehmen (siehe Schrittlänge).

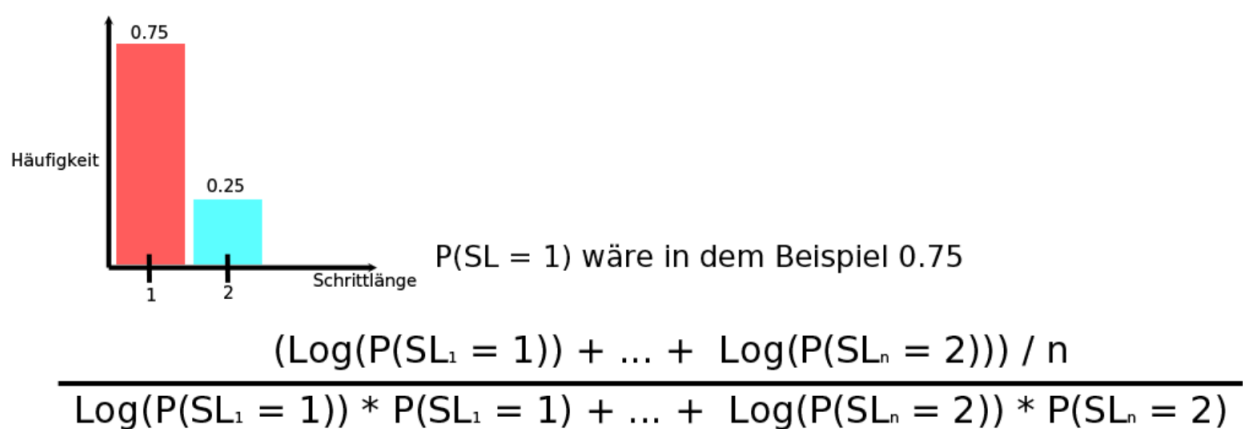


Abbildung 5.1.: Berechnung in der zweiten Implementation

Wenn die betrachteten Schritte dem Model sehr ähnlich ist, nähert sich die Lösung der Rechnung dem Wert 1. Abhängig von der Differenz zur 1, wird eine Wahrscheinlichkeit dass es z.B. ein gutes Beutetier ist berechnet. Somit erhielten wir Werte wie: Schrittlänge(Gut = 0.47, Giftig = 0.2, Neutral = 0.1) und Richtung(Gut = 0.37, Giftig = 0.2, Neutral = 0.4). Durch einfaches Multiplizieren der Wahrscheinlichkeiten kann man die Wahrscheinlichkeiten miteinander vergleichen. Ist die Wahrscheinlichkeit für ein Guten höher als für einen Giftigen, so hat man dann einen potenziell gutes Beutetier. Nur noch aus den Potenziellen den Besten finden und schon kann der Agent zuschnappen. Es gibt in der Implementierung nur einen einstellbaren Schwellwert, welcher angibt, ab welcher Distanz zur 1 sich der Agent zu 0 Prozent sicher sein soll. Der Agent gewinnt jedes Level und sein Punkte befinden sich ca. im Interval 17000 - 24500. Dies ist auch an [Abbildung 5.2 Gesammelte Punkte und der Durchschnitt pro Level](#) zu erkennen. Die orangene Linie zeigt den Durchschnitt bis zu der jeweiligen Welt.

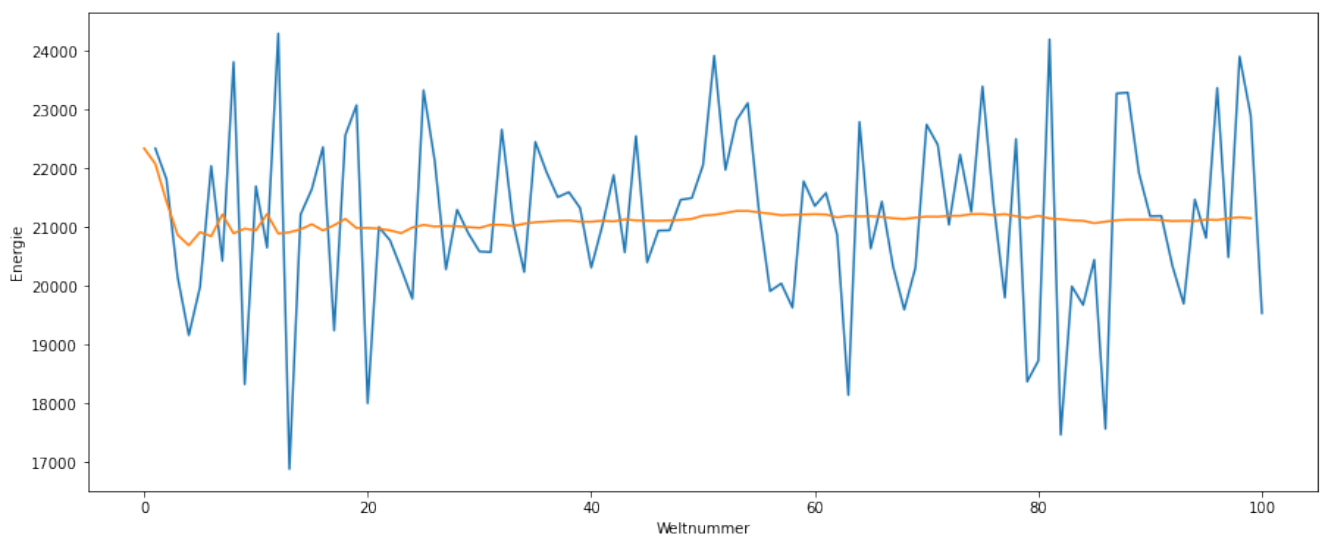


Abbildung 5.2.: Gesammelte Punkte und der Durchschnitt pro Level

A. Sonstige Überlegungen

A.1. Mindestanzahl der Beobachtungen

Bevor ein Agent ein Beutetier klassifiziert, muss er ihn oft genug gesehen haben. Wie wir nach der Implementation bemerkt haben, lebt ein Beutetier nur 30 Runden. Eine geeignete Mindestanzahl der Beobachtungen muss dadrunter liegen, damit der Agent genug Zeit hat gute Beutetiere zu fressen die evt. schon bald verschwinden. Wir haben in unserer Implementierung die durchschnittliche Lebenszeit der Beutetiere in der Beobachtungsphase benutzt. Dies hat funktioniert, da die Beutetiere nicht zum Zeitpunkt 0 erscheinen, sondern schon vorher existiert haben und somit das erste Beutetier mit der ID 0 in Welt 1 schon im vierten Zug verschwindet. Somit ist unsere Mindestanzahl ungefähr 26. Wir hatten uns überlegt, dass wenn der Agent Energie verliert bzw. unter einem bestimmten Wert ist, sich die Mindestanzahl der Beobachtungen senkt. Dies erwies sich nicht als effektiv, da unser Agent nicht unter den Schwellwert von z.B. 40 Energie fiel.

A.2. Bekannte Beutetiere essen

Nach den 300 Zügen weiß der Agent noch, welche Beutetiere aus der Beobachtungsphase gut sind. Diese kann er auch in den ersten Runden fressen, während er sich die neuen Beutetiere anschaut. Diese Strategie wurde in der zweiten Implementation umgesetzt.

A.3. Nach dem Essen lernen

Nach dem der Agent ein Beutetier gegessen hat, kann der Agent feststellen ob es ein gutes oder schlechtes Beutetier ist und somit mit jedem mal die Häufigkeitstabelle und somit die Wahrscheinlichkeiten verbessern. Wir vermuten jedoch, dass dieser Aufwand sich nur minimal lohnen würde.

A.4. Merken des Schlechtesten

Ähnlich wie in [Nach dem Essen lernen](#) kann der Agent das schlechteste Beutetier finden und vermuten, dass es sich um ein schlechten handelt. Dieses Beutetier kann dann genutzt werden, um z.B. die Häufigkeitsverteilung der Schrittlänge zu aktualisieren. Dies haben wir aus dem selben Grund wie in [Nach dem Essen lernen](#) nicht implementiert.

A.5. Gute Beutetiere merken

Da wir erst im Nachhinein herausgefunden haben, dass die durchschnittliche Lebenszeit 30 ist, konnten wir die folgende Idee nicht mehr implementieren. Die Idee besteht darin in [Schritt 5: Argmax - Auswählen des Optimalsten](#) nicht den besten zu wählen, sondern sich alle zu merken die gute Werte

haben. Anschließend wird derjenige aus dem gemerkten Bereich gegessen, welcher am nächsten an der Lebenszeit 30 ist. Damit verhindert man, dass evt. ein gutes Beutetier (welches aber zur bestimmten Zeit nicht das Beste ist) ausgelassen wird.

A.6. Laufzeit optimieren

Sobald man ein besonders schlechtes oder besonders schlechte Beutetiere gefunden hat und [Merken des Schlechtesten](#) angewandt hat, kann man diese aus der Erinnerung löschen. Dadurch wird nicht jedes Mal [Schritt 5: Argmax - Auswählen des Optimalsten](#) auf einem bereits als schlecht erkanntem Beutetier angewandt. Außerdem könnte man in [Schritt 4: Anwenden der Bayes'sche Klassifikation](#) nicht jedes Mal aufs Neue die Logarithmen der Schritte berechnen. Man könnte die Summe in der Erinnerung abspeichern und bei jeder neuen Beobachtung nur den einen Summand aufaddieren.