

# Inhalt

## Inhalt I

|                                    |            |
|------------------------------------|------------|
| <b>Abbildungsverzeichnis .....</b> | <b>III</b> |
| <b>1 Einleitung.....</b>           | <b>1</b>   |
| <b>2 Konzeption.....</b>           | <b>3</b>   |
| 2.1 Kurzbeschreibung .....         | 3          |
| 2.2 Features.....                  | 3          |
| 2.3 Technische Daten .....         | 3          |
| <b>3 Der Spieler .....</b>         | <b>5</b>   |
| 3.1 Blick in die Welt.....         | 5          |
| 3.2 Bewegung .....                 | 5          |
| 3.3 Drehung .....                  | 6          |
| 3.4 Schießen .....                 | 6          |
| <b>4 Aussehen .....</b>            | <b>7</b>   |
| 4.1 Begrenzung.....                | 7          |
| 4.2 Levelaufbau.....               | 7          |
| 4.3 Undurchdringbarkeit.....       | 8          |
| 4.4 Texturen und Licht.....        | 8          |
| 4.5 User Interface.....            | 8          |
| 4.6 Arme .....                     | 9          |
| <b>5 Gegner .....</b>              | <b>11</b>  |
| 5.1 Modell .....                   | 11         |
| 5.2 Bewegung .....                 | 12         |
| 5.3 Drehung .....                  | 12         |
| 5.4 Beschossen werden.....         | 13         |
| 5.5 Schießen .....                 | 13         |
| 5.6 Startposition .....            | 13         |

---

|  |   |           |
|--|---|-----------|
| 5.7                                      | <i>Sinne .....</i>                            | 14        |
| 5.8                                      | <i>Waffenhöhe .....</i>                       | 14        |
| <b>6</b>                                 | <b>Strategence .....</b>                      | <b>15</b> |
| 6.1                                      | <i>Dominante Strategien eliminieren .....</i> | 15        |
| 6.2                                      | <i>Exkurs: Neuronales Netz .....</i>          | 15        |
| 6.3                                      | <i>Aufbau des Neuronalen Netzwerks .....</i>  | 16        |
| 6.4                                      | <i>Implementierung .....</i>                  | 17        |
| 6.5                                      | <i>Erfolge .....</i>                          | 18        |
| 6.6                                      | <i>Auslese .....</i>                          | 18        |
| <b>7</b>                                 | <b>Zusammenfassung und Ausblick .....</b>     | <b>19</b> |
| <b>Literatur</b>                         | <b>.....</b>                                  | <b>20</b> |
| <i>Anhang: Trainingsdurchläufe .....</i> |   | 23        |
| <i>Anhang: Zeiterfassung .....</i>       |   | 24        |
| <i>Anlagen: Netzwerke .....</i>          |   | 24        |
| <b>Selbstständigkeitserklärung .....</b> |   | <b>25</b> |

# Abbildungsverzeichnis

|   |    |
|---|----|
| Abbildung 1: Die leere Welt .....                         | 5  |
| Abbildung 2: In die Ecke bewegt .....                     | 5  |
| Abbildung 3: Ein anderer Blickwinkel.....                 | 6  |
| Abbildung 4: Das Level von oben .....                     | 7  |
| Abbildung 5: Das Level in 3D .....                        | 7  |
| Abbildung 6: Collision Detection .....                    | 8  |
| Abbildung 7: Das Level mit Texturen.....                  | 8  |
| Abbildung 8: Das UI .....                                 | 9  |
| Abbildung 9: Die Hände sind sichtbar.....                 | 9  |
| Abbildung 10: Ein High-Poly-Modell .....                  | 11 |
| Abbildung 11: Die endgültigen Gegner-Roboter .....        | 11 |
| Abbildung 12: Ein großer Roboter .....                    | 11 |
| Abbildung 13: Sich bewegende Roboter.....                 | 12 |
| Abbildung 14: Viele sich drehende Roboter.....            | 12 |
| Abbildung 15: Eine fliegende Kugel.....                   | 13 |
| Abbildung 16: Trefferzone .....                           | 13 |
| Abbildung 17: Laserstrahlen.....                          | 14 |
| Abbildung 18: Vom Roboter entdeckt .....                  | 14 |
| Abbildung 19: Schematische Darstellung eines Neurons..... | 15 |
| Abbildung 20: Darstellung der Schichten.....              | 16 |
| Abbildung 21: Ein Netzwerk-gesteuerter Roboter .....      | 17 |
| Abbildung 22: Das Trainingslevel .....                    | 18 |



# 1 Einleitung

Künstlichen Intelligenz ist überall. Sei es bei Sprachsteuerung von Smartphones <sup>[Mei2017]</sup>, der Verbesserung von Zeichnungen <sup>[Koe2017]</sup> oder bei der Heilung von Krankheiten <sup>[Dor2015]</sup>, in vielen Bereichen des menschlichen Lebens kommen Algorithmen zum Einsatz, die bei verschiedensten Problemen helfen sollen. Sogar für den Untergang der Menschheit sollen sie verantwortlich sein können. <sup>[Sst2016]</sup> Fest steht: Die Anwendungen von KIs sind längst Teil des täglichen Lebens geworden.

Auch und gerade in Computerspielen kommen Algorithmen zum Einsatz, die intelligent genannt werden. Bereits 1952 erschien das erste Einzelspieler-Computerspiel namens OXO, in dem der Spieler gegen eine KI antritt um sie in „Drei gewinnt“ zu schlagen. <sup>[Win2013]</sup> Seitdem haben sich die Möglichkeiten, die Computerspiele bieten, drastisch erhöht – und damit auch die Anforderungen an die KI. <sup>[Pfa2012]</sup> Mittlerweile sind die Algorithmen so komplex, dass Forscher diese sich selbst überlassen und beobachten. <sup>[Plu2017]</sup> Der Mensch erforscht also seine eigene Kreation.

Für die Erforschung eines Phänomens ist es natürlich von Vorteil, eine Umgebung zu schaffen, in der möglich wenige Störfaktoren vorliegen um den Forschungsgegenstand ungestört betrachten zu können. Nur unter solchen Laborbedingungen kann gewährleistet werden, dass die Ergebnisse auch wirklich aus den vermuteten Einflüssen resultiert.

Diese Arbeit stellt eine solche Umgebung vor. In dieser ist es möglich, gegen von der KI gesteuerte Avatare in einer dreidimensionalen Umgebung anzutreten, aber auch, diese gegen sich selbst kämpfen zu lassen. Auf diese Art soll der Grundstein für eine Erforschung und Optimierung des Verhaltens einer KI gelegt werden.



## 2 Konzeption

Für eine nutzbare Umgebung wird zunächst ein Spiel programmiert, in dem die Avatare der KI entstehen.

### 2.1 Kurzbeschreibung

Vier Avatare befinden sich in einer kleinen 3D-Umgebung und bekämpfen sich gegenseitig mit Pistolen, wobei jeder Treffer tödlich ist. Ziel ist es, als letzter übrig zu bleiben. Optional werden folgende Features implementiert: Der Spieler kann entweder seinen eigenen Avatar steuern oder dem Kampf als Beobachter beiwohnen. Das Verhalten der KI-gesteuerten Avatare wird mithilfe eines selbstlernenden Algorithmus beschrieben und verbessert sich durch die Durchführung von Kämpfen.

### 2.2 Features

- Einfacher Singleplay-Shooter
- Mögliche Aktionen: Avatar drehen, Avatar bewegen, mit Avatar schießen
- Optional: Start- und Endscreen mit Statistiken
- Optional: selbstlernende KI
- Optional: Spiel- und Beobachtermodus

### 2.3 Technische Daten

- Programmierung mit Processing
- Ausschließlich vorhandene Assets nutzen





## 3 Der Spieler

Der Spieler steht in der Mitte einer dreidimensionalen Umgebung und kann sich durch diese bewegen. Zunächst soll er mit den Tasten w, a, s und d eine lineare Bewegung durchführen können, anschließend soll auch eine Drehung möglich sein.

### 3.1 Blick in die Welt

Processing startet im Vollbildmodus in einer 3D-Umgebung (wie für Egoshooter üblich). Zunächst instanziiert es eine Kamera, welche der Bewegung des Spielers folgen soll. Diese besteht im Wesentlichen aus einem Vektor, der die Position der Kamera angibt, einem Vektor, der die Position des Punktes angibt, auf den die Kamera gerichtet ist und einen Vektor, der die Richtung angibt, in der oben ist. <sup>[Pro017]</sup> Für diese Entwicklung ist es erforderlich, das Kamera-Objekt auf mehrere Arten zu manipulieren, weswegen eine eigene Klasse existiert.

Bei jedem Aufruf der Draw-Methode legt das Programm die Kameraposition und -ausrichtung neu fest. Zwischen den Aufrufen ist es also möglich, die Kamera nach Belieben zu manipulieren. Dies geschieht hauptsächlich durch die Player-Klasse, wenn sich also der Spieler bewegt.

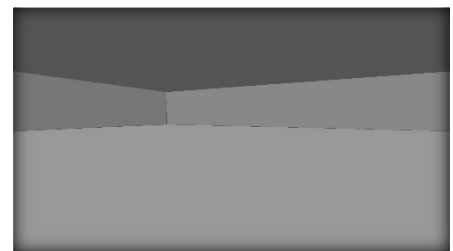


Abbildung 1: Die leere Welt

### 3.2 Bewegung

Als erste Funktionalität soll es möglich sein, den Spieler zu bewegen. Dafür registriert das Spiel das Drücken und Loslassen der Tasten w, a, s und d und leitet das jeweilige Ereignis an das Spielerobjekt weiter, das die entsprechende Bewegung vorbereitet. Der Aufruf der draw-Methode führt zu einer Abfrage, ob sich der Spieler in Bewegung befindet. Wenn dies der Fall ist, berechnet die Spielerklasse die Bewegungsrichtung. Dafür erhält sie von der Kamera die aktuelle Blickrichtung und dreht diese um den von den aktuell gedrückten Bewegungstasten vorgegebenen Winkel. Falls also nur die w-Taste gedrückt wird, muss keine Drehung stattfinden, bei einer Kombination aus s und a bewegt sich der Spieler schräg nach links hinten, was einem Winkel von 225° entspricht.

Dabei verändern sich bis zu vier Werte. Die x- und z-Koordinaten der Kamera passen sich der neu berechneten Position des Spielers an und die x- und z-Koordinaten des Punktes, auf den die Kamera gerichtet ist, passt sich ebenfalls um dieselbe Strecke an. So wird gewährleistet, dass Kamera- und Blickposition niemals übereinstimmen.

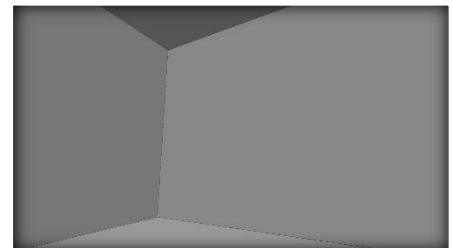


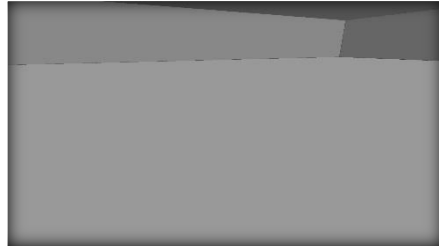
Abbildung 2: In die Ecke bewegt

gewährleistet, dass Kamera- und

### 3.3 Drehung

Neben der Bewegung des Spielers muss es auch möglich sein, die Richtung, in die der Spieler sieht, verändern zu können. Einerseits, um die Bewegungsrichtung exakt anzugeben, andererseits aber auch, um zu bestimmen, in welche Richtung die Pistole zielt. Dafür erfolgt eine Abfrage der Veränderung der Maus-Position, welche an die Kamera übergeben wird. Die Veränderung der x-Position bestimmt die Rotation um die y-Achse, die der y-Position bestimmt die Änderung der Blickrichtung in y-Richtung.

Durch eine Drehung verändert sich lediglich die Position des Zentrums, also des Punktes, auf den die Kamera gerichtet ist. Die Information wird als normalisierter Vektor gespeichert und bei jedem Zugriff auf das Zentrum zu der aktuellen Kamera-Position addiert. Auf diese Weise sind sowohl die Rotation als auch der Zugriff auf die Zentrums-Position möglichst einfach.



**Abbildung 3: Ein anderer Blickwinkel**

### 3.4 Schießen

Für die Bearbeitung der Schießen-Aktion ist es erforderlich, dass andere lebendige Objekte zu existieren. Sie wird deshalb im Kapitel 5.4 beschrieben.

## 4 Aussehen

Die Umgebung besteht aus einer Reihe von nicht belebten Objekten, die weder durchschritten noch zerstört werden können. Allerdings schränken sie die Sicht der Avatare ein und verhindern ein Weiterfliegen der von diesen abgefeuerten Kugeln.

### 4.1 Begrenzung

Das Level ist auf eine quaderförmige Form begrenzt. Sämtliche Interaktionen finden innerhalb dieses Quaders statt. Jedes unbelebte Objekt besteht aus einem PShape-Objekt, das je nach Konfiguration erzeugt, gefärbt und platziert wird. Aus der quaderförmigen Form folgt, dass für die Begrenzung des Levels sechs Objekte benötigt werden: vier Wände, eine Decke und ein Fußboden.

### 4.2 Levelaufbau

Für verschiedene Algorithmen, die zu einer KI gerechnet werden, ist es notwendig, dass sich Hindernisse in dem zu untersuchenden Raum befinden. Einen Wegfindungsalgorithmus in einem leeren Raum zu entwickeln, ist beispielsweise trivial. Allerdings erschwert es die Arbeit an gewissen Algorithmen, wenn sich die Avatare kaum begegnen, schließlich ist die direkte Interaktion ein entscheidender Punkt, an der sich künstliche Intelligenzen messen lassen müssen.

Um die Vergleichbarkeit der verschiedenen Strategien zu gewährleisten, ist der Levelaufbau symmetrisch. So liegt es wirklich an der verwendeten Strategie und nicht an verfälschenden Einflüssen der Umgebung, welches Ergebnis eine gewisse Strategie erzielt.

Abbildung 5: Das Level in 3D

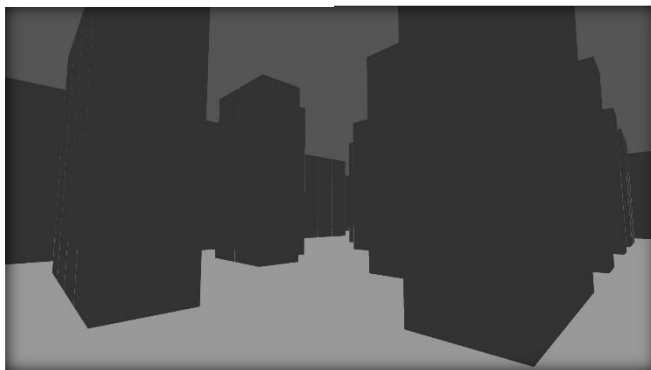
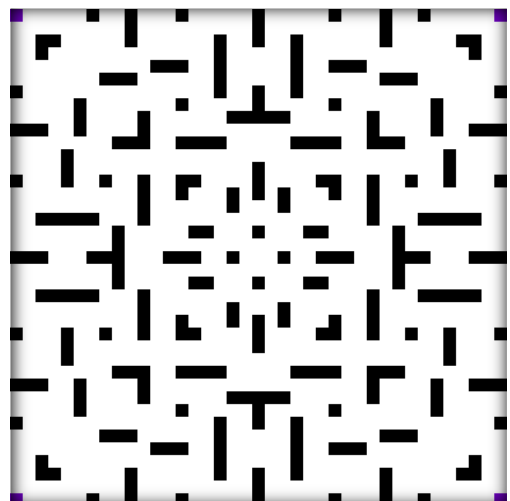


Abbildung 4: Das Level von oben



### 4.3 Undurchdringbarkeit

Um das Durchschreiten von Wänden zu realisieren, muss eine Kollisionsabfrage stattfinden. Für Shape-Objekte steht keine Kollisionsabfrage zur Verfügung, weswegen das Level in ein Gitter von  $39 \times 39$  Einheiten eingeteilt wird. Für eine Kollisionsabfrage ist so nur noch die Bestimmung der Position des Spielers erforderlich. Ein danach erfolgreicher Abgleich mit der Levelbeschreibung liefert die Information, ob der Spieler dieses Gebiet betreten darf. Dazu berechnet das Spiel die neue Position des Spielers. Wenn diese nicht in einem Gebiet mit einem Hindernis ist, scheint die Bewegung durchführbar zu sein. Allerdings hat der Spieler ebenfalls einen gewissen Platzbedarf. Um diesen zu beachten, testet das Spiel drei weitere Punkte. Der eine in Bewegungsrichtung des Spielers, die anderen beiden eine Achtdrehung um den neuen Punkt entfernt. Erst wenn alle drei Punkte frei sind (und die Bewegung damit nicht in einer Mauer endet), erhält die Kamera ihre neue Position.

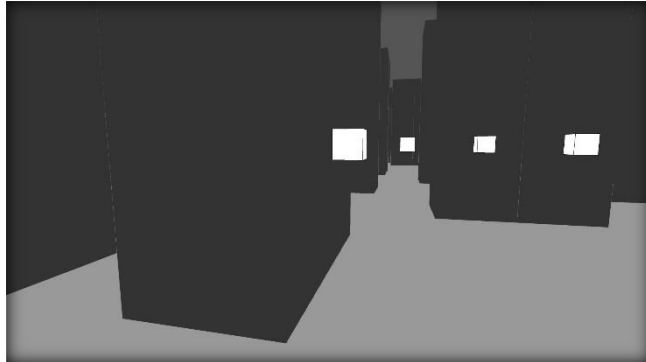


Abbildung 6: Collision Detection

### 4.4 Texturen und Licht

Die Klötze, die als Hindernisse dienen, sind zwar funktional, aber nicht besonders schön anzusehen. Um die Immersion zumindest ein bisschen zu steigern, bekommen diese eine Textur. Nachdem die grafische Qualität nicht entscheidend ist, und um rechtlichen Schwierigkeiten vorzubeugen, werden alle Texturen generiert. Der Boden, die Wände und die Decke bekommen jeweils eine eigene Textur sowie eine Beleuchtung mittels des Processing-Standardlichts.

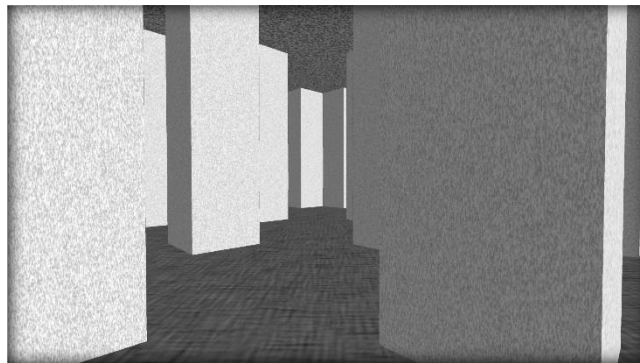
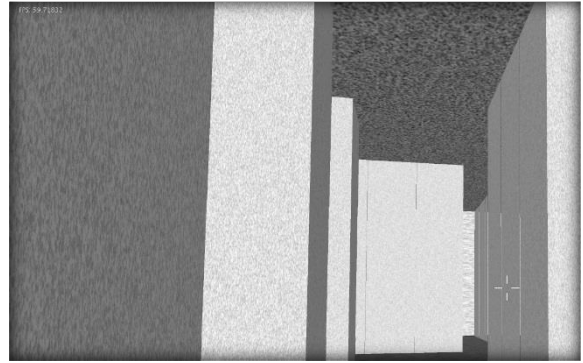


Abbildung 7: Das Level mit Texturen

### 4.5 User Interface

Um Objekte in die Bildelebene zu schreiben, muss zuerst die Kamera auf die Standardposition zurückgesetzt werden. Deshalb gibt es zwei Schleifen für die nicht belebten

Objekte. Zuerst zeichnet das Spiel alle Objekte der 3D-Welt, anschließend erhält die Kamera ihre Standardposition und am Ende zeichnet es alle Objekte, die ständig sichtbar sein sollen. Zu Beginn sind das lediglich ein Fadenkreuz, das angibt, in welche Richtung der Spieler zielt sowie die Angabe der aktuellen FPS-Anzahl. Später sind durchaus andere Informationen denkbar, zum Beispiel, welche Strategien die KI-Gegner verwenden.



**Abbildung 8: Das UI**

## 4.6 Arme

Um die Immersion zu steigern, sieht der Spieler stets die Arme seines Avatars. Somit verstärkt sich der Eindruck, er würde tatsächlich durch die Augen des Avatars in die virtuelle Welt sehen. Für eine einfache Illusion reicht es, ein Bild von einer Pistole zu zeigen, die der Avatar scheinbar in der Hand hält. Auch dieses Bild ist Teil des UI.



**Abbildung 9: Die Hände sind sichtbar.**



## 5 Gegner

Kein Shooter ist ohne Gegner interessant und ohne Avatar, der die Vorgaben der KI ausführt, kann sich diese nicht optimieren. Deshalb besitzt KI-kruten Gegenspieler.

### 5.1 Modell

Die Gegner benötigen ein echtes 3d-Modell, schließlich sollen sie aus allen Richtungen betrachtet werden. Eine Recherche über die Möglichkeiten, 3D-Animationen bei Processing durchzuführen, blieb ergebnislos, weswegen auf Modell-Animationen verzichtet wird. Als Konsequenz scheint ein nicht-menschlicher Avatar sinnvoll zu sein.

Offenbar kommt Processing nicht besonders gut mit Texturen auf einer High-Poly-Modell zurecht, denn die Bildwiederholungsrate sinkt bei Verwendung einer solchen Textur in einen einstelligen Bereich. Selbst bei einer deutlichen Reduzierung der Vertex-Anzahl reduziert sich auch bei lediglich einem Gegner die FPS-Zahl auf unter die Hälfte. Deshalb erhalten die Modelle nur eine Farbe und keine Textur. Das Ergebnis sind drei Flugdrohnen mit eingebauter Waffe in den einfach zu unterscheidenden Farben rot, grün und blau.

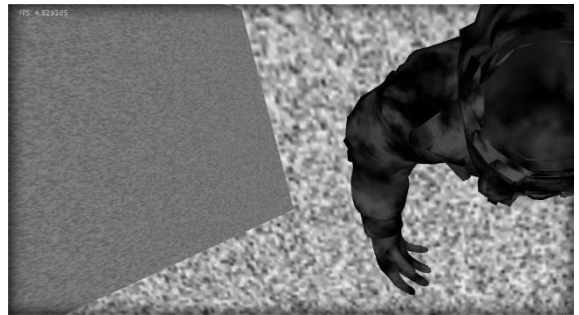


Abbildung 10: Ein High-Poly-Modell

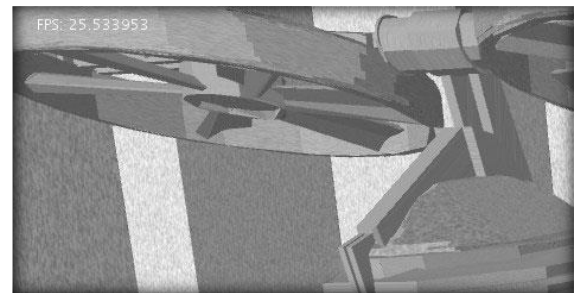


Abbildung 12: Ein großer Roboter



Abbildung 11: Die endgültigen Gegner-Roboter

## 5.2 Bewegung

Die Gegner erhalten zunächst eine sehr simple KI. Der erste Schritt ist es, zu erreichen, dass sie sich durch das Level bewegen ohne an Wänden hängen zu bleiben. Dafür erhalten sie zunächst lediglich die Information, ob eine Bewegung in die bisherige Richtung zu einem Stillstand führen würde. Die dafür benötigte Methode findet bereits beim Spieler Verwendung. Allerdings muss, anders als beim Spieler, bei einem erfolgreichen Test ebenfalls die Position des PShape-Objekts angepasst werden.



Abbildung 13: Sich bewegende Roboter

## 5.3 Drehung

Wie auf Abbildung 13 ersichtlich ist, müssen sich die Avatare drehen können, da die Bewegung ansonsten sehr schnell ein Ende findet. Die erste Strategie sieht vor, dass er sich nur dann dreht, wenn eine Vorwärtsbewegung nicht möglich ist. So kann jeder Gegner Hindernissen ausweichen, wenn auch nicht sonderlich elegant. Allerdings findet jede neue Drehung jeweils um den Ursprung statt, weswegen sich die grafische Position falsch verhält.

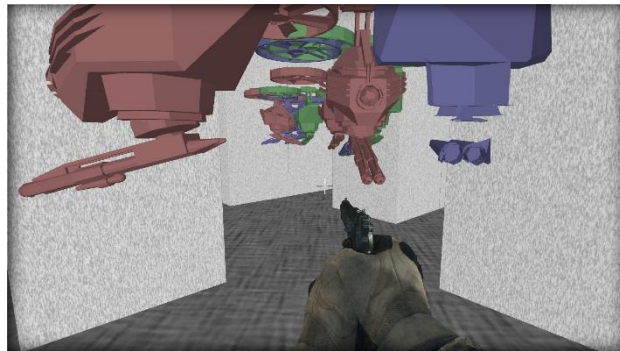


Abbildung 14: Viele sich drehende Roboter

Möglichkeit eins ist, die Transformationsmatrix der PShapes zurück zu setzen, allerdings sinkt die FPS-Rate dadurch dramatisch. Zweitens kann vor der Rotation die Transformation rückgängig gemacht werden, dann rotiert und anschließend wieder transformiert. Dadurch bleibt die Bildwiederholungsrate stabil, da die Translation nicht bei jedem Frame, sondern nur bei einer tatsächlich erfolgenden Rotation erfolgt.

Um ein kleines bisschen weniger berechenbar zu sein, würfeln die Gegner alle fünf Sekunden, in welche Richtung sie sich drehen, sobald sie auf ein Hindernis stoßen.



## 5.4 Beschossen werden

Bisher findet keinerlei Interaktion zwischen den Avataren und dem Spieler statt. Zunächst soll es dem Spieler möglich sein, seine Gegner zu zerstören. Dafür kann er per Mausklick Pistolenkugeln erschaffen, die sich linear fortbewegen, bei Kontakt mit Mauern sich selbst zerstören und bei Kontakt mit einem gegnerischen Avatar diesen. Dafür entsteht eine neue Klasse `Projectile`, die, ebenso wie `Player` und `KIObjekt`, von `MoveableObject` erbt. Außerdem implementieren alle Klassen, die schießen können, ein Interface, `ShootingObject`, um sie von anderen Objekten, wie zum Beispiel Projektilen, unterscheiden zu können.

Die Trefferzone besteht bei den fliegenden Robotern aus einer Kugel, die sich weitgehend dem Körper anschmiegt. Dadurch sind die Rotoren und die Waffe kein Teil davon. Für die geplante Implementierung reicht diese Genauigkeit jedoch.

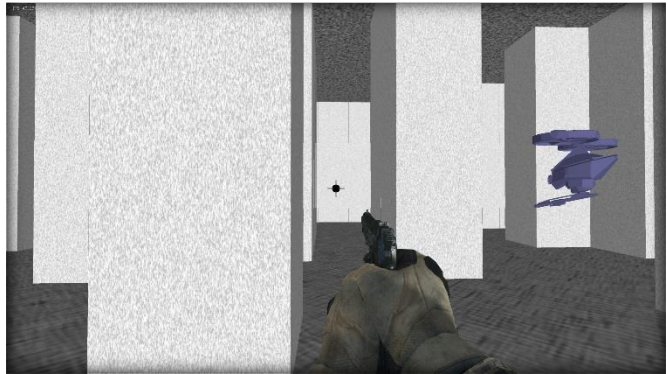


Abbildung 15: Eine fliegende Kugel



Abbildung 16: Trefferzone

## 5.5 Schießen

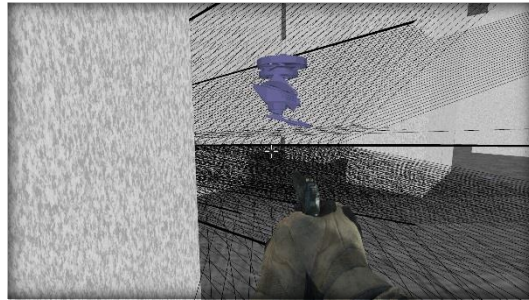
Bisher sind die Roboter wehrlos. Zwar kann der Spieler sie zerstören, aber sie sind nicht selbst in der Lage, Projektilen zu erzeugen. Um das zu ermöglichen, erhalten sie eine eigene Klasse, `FlyingRobots`, die von der `KIObjekt`-Klasse erbt und außerdem das `ShootingObject`-Interface implementiert. Die erste Strategie, „Always move and shoot“ entsteht. Gemäß dieser bewegen sich die Roboter ständig vorwärts bis sie auf ein Hindernis stoßen. Dann drehen sie sich so lange bis sie sich wieder bewegen können. Währenddessen schießen sie ununterbrochen.

## 5.6 Startposition

Bisher starten alle Roboter und der Spieler in der Mitte, nicht in den dafür vorgesehenen Ecken. Also erhält die neu geschaffene Liste `startPositions` alle vier Ecken als Positionen und verteilt diese zufällig an die drei Roboter und den Spieler.

## 5.7 Sinne

Die Roboter nehmen lediglich zwei Dinge wahr: Den eigenen Abschuss und ob sie direkt vor einer Wand stehen. Dem soll noch ein dritter Sinn hinzugefügt werden, nämlich eine Art Sehsinn. Dafür sendet jeder Roboter einige Strahlen aus, die den Roboter mit Informationen versorgen, nämlich der Art des Objekts (Wall, Robot oder Player) und der Position des Objektes. Mit der Standardstrategie „Look for opponents“ verhält sich der Roboter wie mit der Strategie, sendet jedoch ununterbrochen Laserstrahlen aus um Gegner zu lokalisieren. Stößt ein Strahl auf einen Gegner, bleibt der Roboter stehen und dreht sich so lange auf der Stelle bis er in die Richtung der Position der Sichtung zeigt.



**Abbildung 17: Laserstrahlen**



**Abbildung 18: Vom Roboter entdeckt**

## 5.8 Waffenhöhe

Die Waffen der Roboter schießen aus einem Winkel, der aus dem verwendeten Modell hervorgeht. Dabei gelingt es ihnen zwar, den menschlichen Spieler zu treffen, sich gegenseitig können sie jedoch nicht beschießen. Also erhalten sie die Möglichkeit, ihre Waffenhöhe zu variieren. Die Strategie „Look for opponents“ passt automatisch die Waffenhöhe an wenn ein Gegner in Sicht ist und wechselt zu einer mittleren Lage wenn sich in letzter Zeit keiner gezeigt hat.

## 6 Strategence

Die Standardstrategien „Always move and shoot“ sowie „Look for opponents“ sind zwar geeignet dafür, die grundlegenden Funktionalitäten der Roboter zu testen, besonders intelligent fühlen sich die Roboter allerdings nicht an. Deshalb soll das Arsenal an auswählbaren Strategien für die Roboter um Strategences (**Stragegy with Intelligence**) erweitert werden.

### 6.1 Dominante Strategien eliminieren

Aktionsmöglichkeiten, die zu verwenden eine dominante Strategie darstellen, machen eine Strategie immer uninteressanter als sie sein könnte. Momentan sind Schießen und mit maximaler Anzahl Laserstrahlen Gegner aufspüren solche dominanten Strategien. Deshalb bekommen Roboter eine Art Währung zugeteilt, die sie ausgeben müssen um bestimmte Aktionen durchzuführen. Diese nennt sich Energy. In der Tabelle rechts ist aufgeführt, welche Aktion wie viel Energy kostet.

| Ereignis              | Energy |
|-----------------------|--------|
| Spielbeginn           | +200   |
| Schießen              | -10    |
| Laserstrahl aussenden | -1     |
| jede Sekunde          | +10    |

### 6.2 Exkurs: Neuronales Netz

Ein Neuronales Netz besteht aus einer Summe von Neuronen, die jeweils eine kleine Verarbeitungseinheit darstellen. Jedes Neuron besitzt eine beliebige Anzahl an Eingängen  $x_n$ , die jeweils ein Gewicht  $w_{nj}$  besitzen und einen konstanten Schwellenwert  $\theta_j$ . Das Neuron addiert üblicherweise die Stärke der Eingangsreize multipliziert mit dem jeweiligen Gewicht und subtrahiert davon den Schwellenwert. Das Ergebnis dieser Berechnung wird zur Bestimmung der Stärke des Ausgangsreizes verwendet. Im einfachsten Fall feuert das Neuron mit voller Stärke, wenn das Ergebnis größer als 0 ist, ansonsten gar nicht. <sup>[HeX2009]</sup> Es können jedoch auch kompliziertere Methoden angewendet werden. <sup>[Lip1987]</sup> Außerdem besitzt jedes Neuron beliebig viele Ausgaben  $y_j$ , die mit dem Empfänger des Reizes verknüpft sind.

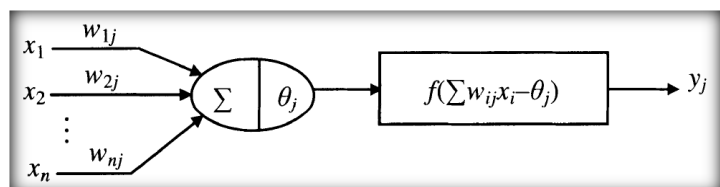


Abbildung 19: Schematische Darstellung eines Neurons

Diese Neuronen sind in funktionelle Gruppen, sogenannte Schichten, eingeteilt, die in einer Reihe angeordnet sind. Dabei besitzen Neuronen einer Schicht in einfachen Netzwerken keine Verbindung zu Neuronen einer nicht benachbarten Schicht. Die Außenwelt interagiert mit dem Neuronalen Netzwerk, indem es Reize an die Neuronen des Input layers sendet und die Signale des Output layers empfängt. [HeX2009]

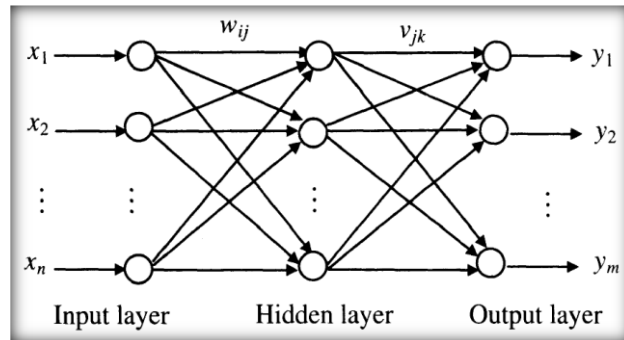


Abbildung 20: Darstellung der Schichten

### 6.3 Aufbau des Neuronalen Netzwerks

Das verwendete Netzwerk soll eine möglichst einfache Struktur beinhalten, dabei aber in der Lage sein, alle Erwartungen zu erfüllen. Die verwendeten Neuronen sind in der Tabelle unten dargestellt.

| Input layer             | Hidden layer        | Output layer  |
|-------------------------|---------------------|---------------|
| Haptic input            |                     | Move          |
| Visual input (type)     |                     | Rotate        |
| Visual input (position) | Process information | Adjust weapon |
| Energy level            |                     | Look          |
| Static input            |                     | Shoot         |

Die Neuronen in der ersten Schicht erhalten die Informationen, die dem Roboter zur Verfügung stehen. In jedem Durchlauf erhält er die Information, ob sich vor ihm eine Wand befindet (Haptic input). Falls er im letzten Durchlauf einen Laserstrahl zur Gegnerentdeckung gestartet hatte, bekommt er jetzt die Ergebnisse (Visual input). Als letzten regulären Input erfährt der Roboter, wie groß sein Energy-Level im Moment ist. Außerdem gibt es einen Static input, der ständig aktiv ist, sodass der Roboter auch Aktionen durchführen kann, wenn er keinen sonstigen Input erhält.

Die Neuronen des mittleren, Hidden layers erfüllen keine konkrete Funktion hinsichtlich des Spiels, allerdings sind sie notwendig um die Informationen der input Neuronen zu verarbeiten, denn erst mit einer dritten Schicht ist die Lösung von XOR-Problemen möglich. <sup>[HeX2009]</sup>

In der letzten Schicht befinden sich die output Neuronen, welche tatsächlich Aktionen auslösen. Wenn diese Neuronen aktiviert sind, bewegt sich der Roboter, dreht sich, richtet den Winkel seines Waffensystems neu aus, inspiziert seine Umgebung oder schießt.

Für den Anfang sollen alle Neuronen mit jeweils allen Neuronen voll vermascht sein. Sollte sich im Laufe der Entwicklung herausstellen, dass gewisse Kanten überhaupt nicht benötigt werden, so kann die Vermaschung im Nachhinein angepasst werden.

## 6.4 Implementierung

Die gesamte Funktionalität benötigt lediglich drei neue Klassen und eine neue Strategie, „Neural network“, für die Roboter. Die erste Klasse, Network, legt den Aufbau des Netzes fest und bietet Schnittstellen nach außen. Dabei instanziiert sie eine Reihe von Neurons, welche ihre jeweiligen Inputs verarbeiten und entsprechend durch ihre Outputs feuern. Diese werden durch die Klasse Link dargestellt, die sowohl als Verbindung zwischen den Neuronen des Netzes als auch zur Speicherung der Informationen von und nach außen dienen.

Jegliche Strategie kann ihr jeweiliges Netzwerk folgendermaßen benutzen:

1. Netzwerk neu berechnen. Dabei berechnet jedes Neuron seinen Output neu und modifiziert die Links entsprechend.
2. Die Outputs der Output-Neuronen abholen und die Werte auf eine Weise interpretieren, dass sie zu gewissen Aktionen führen können. Bei der Standardstrategie führt zum Beispiel ein Wert von größer als 0,5 bei dem Neuron Shoot dazu, dass eine Kugel abgefeuert wird.
3. Die Inputs neu setzen. Dafür müssen die relevanten Werte zu Floats konvertiert werden, am besten zwischen -1 und 1, da dies auch die Werte sind, die zwischen den Neuronen ausgetauscht werden.

Es gibt zwei Prototypen für die Generierung eines Neuronalen Netzwerkes. „Default“ bedeutet, dass alle Gewichte und Schwellwerte auf die jeweiligen Standardwerte gestellt sind. Daraus resultiert ein Roboter, der bei jedem Durchlauf versucht, alle Aktionen auszuführen. „Random“ setzt alle genannten Werte auf jeweils einen Zufallswert innerhalb des jeweils gültigen Bereichs.



**Abbildung 21: Ein Netzwerk-gesteuerter Roboter**

## 6.5 Erfolge

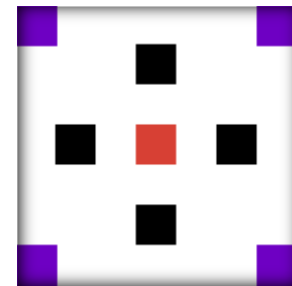
Um den Erfolg eines Netzwerks zu messen, wird die Definition eines Erfolgskriteriums benötigt. Ziel des Spiels ist es, als letzter Überlebender übrig zu bleiben, das ist also sicherlich als Erfolg anzusehen. Außerdem hilft es, einen Opponenten auszuschalten, das sollte also

ebenfalls als Erfolg gelten. Handlungen, die notwendig dafür sind, dass einer der genannten Erfolge erzielt werden kann, sind Sehen und Schießen. Deshalb werden auch diese Handlungen entsprechend belohnt. Daraus ergibt sich die Verteilung der Erfolgspunkte in der Tabelle rechts.

| Erfolg               | Punkte |
|----------------------|--------|
| Sehen                | 2      |
| Schießen             | 5      |
| Opponent ausschalten | 100    |
| Gewinnen             | 500    |

## 6.6 Auslese

Ein sehr großer Teil der zufällig ausgewählten Netzwerke wird nicht besonders funktionstüchtig sein. Der bereits vorgestellte Typ „Default“ fliegt beispielsweise ständig nur im Kreis und wird so aus eigenem Antrieb niemals auf einen Gegner treffen. Deshalb sollen in einem ersten Durchlauf zuerst alle Strategien, die nicht funktionstüchtig erscheinen, ausgesiebt werden. Dazu entsteht ein zweites Level, das vom Aufbau her wesentlich einfacher und dazu auch noch kleiner ist. Abbildung 22 zeigt das neu geschaffene Level.



**Abbildung 22: Das Trainingslevel**

Im ersten Durchgang findet eine Selektion statt, die alle Netzwerke, deren Roboter innerhalb von 10 Sekunden keinen einzigen Punkt erzielt, entfernt. Die übrigen Netzwerke speichert das Spiel in einer JSON-Datei zur späteren Verwendung. Von den 212 getesteten Netzwerken überstehen 20 die Selektion.

Der zweite Durchgang findet unter denselben Bedingungen statt, allerdings muss eine Punktzahl von 10 erreicht werden und es starten nur Netzwerke, welche den ersten Durchgang überstanden haben. Lediglich acht Netzwerke erreichen eine andere Punktzahl als 0.

Beim dritten Durchgang finden die gleichen Parameter Einsatz, jedoch benötigen die Netzwerke einen Erfolg von 100 um zu bestehen. Das erreichen lediglich vier Netzwerke, alle anderen erreichen eine Punktzahl vom 0.

## 7 Zusammenfassung und Ausblick

Im Zuge dieser Arbeit ist ein Programm entstanden, das eine interessante Möglichkeit darstellt, intelligente Strategien zu entwerfen und zu testen. Sowohl der Spieler als auch die Roboter haben eine ausreichende Anzahl von Interaktionsmöglichkeiten, sodass sie nicht-triviale Strategien anwenden können um zu gewinnen.

Bei der Auslese der Netzwerke fällt auf, dass ein sehr großer Teil der Netzwerke während des zweiten Durchgangs keine Punkte erzielte. Das könnte auf einen Fehler zurückzuführen sein, der beim Speichern bzw. Laden der Netzwerke auftritt.

Eine Fortführung der Arbeit könnte folgende Aspekte einschließen:

- Einfügen eines Start- und eines Endbildschirms, auf denen Optionen und Statistiken dargestellt sind.
- Automatisierte Tests. Bisher muss jeder Versuch manuell gestartet und vor jedem Durchgang müssen alle Netzwerk-Dateien manuell umbenannt werden.
- Trainieren der bestehenden Netzwerke. Vielversprechende Netzwerke könnten variiert werden um eine größere Anzahl an interessanten Netzwerken zu erhalten ohne allzu viele Netzwerke blind generieren zu müssen.
- Debugging. Gelegentlich bleiben Roboter in Wänden hängen, das Speichern und Laden der Netzwerke könnte fehlerhaft sein und der Spieler blockiert auch im Training-Modus Sichtlinien.
- Performance-Optimierung. Die Methoden der Roboter verwenden einige aufwendige Algorithmen, zum Beispiel 3d-Transformationen. Diese zu optimieren, müsste die Performance deutlich erhöhen.

## Literatur

- [Plu2017] Pluta, Werner: Wenn künstliche Intelligenzen zocken, <https://www.golem.de/news/deepmind-forscher-untersuche-wie-ki-systeme-ticken-1702-126105.html>, verfügbar am 15.04.2017 um 10:10
- [Koe2017] Kölzsch, Tobias: Googles neues Malprogramm verbessert klägliche Zeichnungen, <https://www.golem.de/news/autodraw-googles-neues-malprogramm-verbessert-klagliche-zeichnungen-1704-127272.html>, verfügbar am 15.04.2017 um 10:10
- [Mei2017] Meineck, Sebastian: Funktioniert Sprechen so gut wie Tippen?, <http://www.spiegel.de/netzwelt/apps/spracherkennung-fuer-ios-und-android-im-test-wie-gut-funktioniert-das-a-1134324.html>, verfügbar am 15.04.2017 um 10:15
- [SSt2016] Schmitt, Stefan: Dieser Mann denkt über den Untergang der Menschheit nach, <http://www.zeit.de/2016/21/nick-bostrom-oxford-philosoph-kuenstliche-intelligenz>, verfügbar am 15.04. um 10:20
- [Dor2015] Dorrier, Jason: Exponential Medicine: Deep Learning AI Better Than Your Doctor at Finding Cancer, <https://singularityhub.com/2015/11/11/exponential-medicine-deep-learning-ai-better-than-your-doctor-at-finding-cancer/>, verfügbar am 15.04. um 10:35
- [Win2013] Winter, David: Noughts And Crosses - The oldest graphical computer game, <http://www.pong-story.com/1952.htm>, verfügbar am 15.04.2017 um 10:50
- [Pro2017] camera(), [https://processing.org/reference/camera\\_.html](https://processing.org/reference/camera_.html), verfügbar am 15.04. um 15:55



- [HeX2009] He, Xingui, Xu, Shaohua: Process Neural Networks – Theory and Applications, Hangzhou, Zhejiang University Press, Hangzhou and Springer-Verlag Berlin Heidelberg, 2009
- [Lip1987] Lippmann, Richard P.: An Introduction to Computing with Neural Nets, IEEE ASSP MAGAZINE, April 1987
- [Pfa2012] Pfannkuch, Tilo: Künstliche Intelligenz in Computerspielen, Stuttgart, Universität Stuttgart – Institut für Visualisierung und Interaktive Systeme, 2012
- [Rea2014] Reas, Casey, Fry, Ben: Processing: a programming handbook for visual designers and artists, Cambridge, The MIT Press, 2014



## Anhang: Trainingsdurchläufe

| Durchgang                            | 1   | 2     | 3    | 1            | 2   | 3     | 1            | 2            | 3    |
|--------------------------------------|-----|-------|------|--------------|-----|-------|--------------|--------------|------|
| Mindesterfolg                        | 1   | 10    | 100  | Netzwerk 68  | 0   |       | Netzwerk 140 | 0            |      |
| Dauer (s)                            | 10  | 60    | 60   | Netzwerk 69  | 170 | 12480 | 1064         | Netzwerk 141 | 0    |
| Anzahl<br>erfolgreicher<br>Netzwerke | 20  | 8     | 4    | Netzwerk 70  | 0   |       | Netzwerk 142 | 0            |      |
|                                      |     |       |      | Netzwerk 71  | 0   |       | Netzwerk 143 | 0            |      |
| Netzwerk 1                           | 0   |       |      | Netzwerk 72  | 0   |       | Netzwerk 144 | 0            |      |
| Netzwerk 2                           | 0   |       |      | Netzwerk 73  | 0   |       | Netzwerk 145 | 0            |      |
| Netzwerk 3                           | 0   |       |      | Netzwerk 74  | 0   |       | Netzwerk 146 | 0            |      |
| Netzwerk 4                           | 0   |       |      | Netzwerk 75  | 0   |       | Netzwerk 147 | 0            |      |
| Netzwerk 5                           | 0   |       |      | Netzwerk 76  | 0   |       | Netzwerk 148 | 0            |      |
| Netzwerk 6                           | 0   |       |      | Netzwerk 77  | 0   |       | Netzwerk 149 | 65           | 0    |
| Netzwerk 7                           | 0   |       |      | Netzwerk 78  | 0   |       | Netzwerk 150 | 0            |      |
| Netzwerk 8                           | 0   |       |      | Netzwerk 79  | 0   |       | Netzwerk 151 | 0            |      |
| Netzwerk 9                           | 0   |       |      | Netzwerk 80  | 52  | 686   | 5320         | Netzwerk 152 | 0    |
| Netzwerk 10                          | 0   |       |      | Netzwerk 81  | 0   |       | Netzwerk 153 | 0            |      |
| Netzwerk 11                          | 0   |       |      | Netzwerk 82  | 120 | 2566  | 5            | Netzwerk 154 | 0    |
| Netzwerk 12                          | 590 | 995   | 0    | Netzwerk 83  | 0   |       | Netzwerk 155 | 72           | 0    |
| Netzwerk 13                          | 4   | 0     |      | Netzwerk 84  | 80  | 0     | Netzwerk 156 | 0            |      |
| Netzwerk 14                          | 0   |       |      | Netzwerk 85  | 0   |       | Netzwerk 157 | 0            |      |
| Netzwerk 15                          | 0   |       |      | Netzwerk 86  | 0   |       | Netzwerk 158 | 0            |      |
| Netzwerk 16                          | 0   |       |      | Netzwerk 87  | 0   |       | Netzwerk 159 | 40           | 4430 |
| Netzwerk 17                          | 0   |       |      | Netzwerk 88  | 0   |       | Netzwerk 160 | 0            | 3930 |
| Netzwerk 18                          | 0   |       |      | Netzwerk 89  | 0   |       | Netzwerk 161 | 0            |      |
| Netzwerk 19                          | 0   |       |      | Netzwerk 90  | 0   |       | Netzwerk 162 | 0            |      |
| Netzwerk 20                          | 0   |       |      | Netzwerk 91  | 0   |       | Netzwerk 163 | 0            |      |
| Netzwerk 21                          | 0   |       |      | Netzwerk 92  | 0   |       | Netzwerk 164 | 0            |      |
| Netzwerk 22                          | 0   |       |      | Netzwerk 93  | 0   |       | Netzwerk 165 | 0            |      |
| Netzwerk 23                          | 22  | 0     |      | Netzwerk 94  | 0   |       | Netzwerk 166 | 0            |      |
| Netzwerk 24                          | 77  | 15638 | 5500 | Netzwerk 95  | 0   |       | Netzwerk 167 | 0            |      |
| Netzwerk 25                          | 0   |       |      | Netzwerk 96  | 0   |       | Netzwerk 168 | 0            |      |
| Netzwerk 26                          | 0   |       |      | Netzwerk 97  | 0   |       | Netzwerk 169 | 0            |      |
| Netzwerk 27                          | 0   |       |      | Netzwerk 98  | 0   |       | Netzwerk 170 | 0            |      |
| Netzwerk 28                          | 0   |       |      | Netzwerk 99  | 0   |       | Netzwerk 171 | 0            |      |
| Netzwerk 29                          | 0   |       |      | Netzwerk 100 | 0   |       | Netzwerk 172 | 0            |      |
| Netzwerk 30                          | 0   |       |      | Netzwerk 101 | 0   |       | Netzwerk 173 | 0            |      |
| Netzwerk 31                          | 0   |       |      | Netzwerk 102 | 0   |       | Netzwerk 174 | 0            |      |
| Netzwerk 32                          | 0   |       |      | Netzwerk 103 | 0   |       | Netzwerk 175 | 0            |      |
| Netzwerk 33                          | 0   |       |      | Netzwerk 104 | 0   |       | Netzwerk 176 | 0            |      |
| Netzwerk 34                          | 0   |       |      | Netzwerk 105 | 0   |       | Netzwerk 177 | 0            |      |
| Netzwerk 35                          | 0   |       |      | Netzwerk 106 | 0   |       | Netzwerk 178 | 20           | 0    |
| Netzwerk 36                          | 0   |       |      | Netzwerk 107 | 0   |       | Netzwerk 179 | 0            |      |
| Netzwerk 37                          | 0   |       |      | Netzwerk 108 | 0   |       | Netzwerk 180 | 0            |      |
| Netzwerk 38                          | 0   |       |      | Netzwerk 109 | 0   |       | Netzwerk 181 | 0            |      |
| Netzwerk 39                          | 0   |       |      | Netzwerk 110 | 0   |       | Netzwerk 182 | 0            |      |
| Netzwerk 40                          | 96  | 0     |      | Netzwerk 111 | 120 | 356   | 0            | Netzwerk 183 | 0    |
| Netzwerk 41                          | 0   |       |      | Netzwerk 112 | 0   |       | Netzwerk 184 | 0            |      |
| Netzwerk 42                          | 0   |       |      | Netzwerk 113 | 0   |       | Netzwerk 185 | 0            |      |
| Netzwerk 43                          | 0   |       |      | Netzwerk 114 | 0   |       | Netzwerk 186 | 0            |      |
| Netzwerk 44                          | 0   |       |      | Netzwerk 115 | 0   |       | Netzwerk 187 | 0            |      |
| Netzwerk 45                          | 0   |       |      | Netzwerk 116 | 0   |       | Netzwerk 188 | 0            |      |
| Netzwerk 46                          | 0   |       |      | Netzwerk 117 | 0   |       | Netzwerk 189 | 0            |      |
| Netzwerk 47                          | 0   |       |      | Netzwerk 118 | 0   |       | Netzwerk 190 | 255          | 0    |
| Netzwerk 48                          | 0   |       |      | Netzwerk 119 | 0   |       | Netzwerk 191 | 0            |      |
| Netzwerk 49                          | 0   |       |      | Netzwerk 120 | 0   |       | Netzwerk 192 | 0            |      |
| Netzwerk 50                          | 0   |       |      | Netzwerk 121 | 0   |       | Netzwerk 193 | 0            |      |
| Netzwerk 51                          | 78  | 2     |      | Netzwerk 122 | 0   |       | Netzwerk 194 | 0            |      |
| Netzwerk 52                          | 0   |       |      | Netzwerk 123 | 0   |       | Netzwerk 195 | 0            |      |
| Netzwerk 53                          | 0   |       |      | Netzwerk 124 | 0   |       | Netzwerk 196 | 0            |      |
| Netzwerk 54                          | 0   |       |      | Netzwerk 125 | 0   |       | Netzwerk 197 | 0            |      |
| Netzwerk 55                          | 0   |       |      | Netzwerk 126 | 0   |       | Netzwerk 198 | 0            |      |
| Netzwerk 56                          | 0   |       |      | Netzwerk 127 | 0   |       | Netzwerk 199 | 0            |      |
| Netzwerk 57                          | 0   |       |      | Netzwerk 128 | 0   |       | Netzwerk 200 | 0            |      |
| Netzwerk 58                          | 0   |       |      | Netzwerk 129 | 0   |       | Netzwerk 201 | 0            |      |
| Netzwerk 59                          | 0   |       |      | Netzwerk 130 | 5   | 0     | Netzwerk 202 | 0            |      |
| Netzwerk 60                          | 0   |       |      | Netzwerk 131 | 0   |       | Netzwerk 203 | 0            |      |
| Netzwerk 61                          | 2   | 365   | 0    | Netzwerk 132 | 0   |       | Netzwerk 204 | 0            |      |
| Netzwerk 62                          | 0   |       |      | Netzwerk 133 | 0   |       | Netzwerk 205 | 0            |      |
| Netzwerk 63                          | 0   |       |      | Netzwerk 134 | 0   |       | Netzwerk 206 | 0            |      |
| Netzwerk 64                          | 0   |       |      | Netzwerk 135 | 100 | 0     | Netzwerk 207 | 0            |      |
| Netzwerk 65                          | 0   |       |      | Netzwerk 136 | 0   |       | Netzwerk 208 | 0            |      |
| Netzwerk 66                          | 0   |       |      | Netzwerk 137 | 0   |       | Netzwerk 209 | 0            |      |
| Netzwerk 67                          | 0   |       |      | Netzwerk 138 | 0   |       | Netzwerk 210 | 235          | 0    |
| Netzwerk 68                          | 0   |       |      | Netzwerk 139 | 0   |       | Netzwerk 211 | 0            |      |

## Anhang: Zeiterfassung

| Datum      | Dauer | Tätigkeiten  |
|------------|-------|--|
| 15.04.2017 | 10:30 | Kapitel "Einleitung", "Konzeption", "Der Spieler" und "Umgebung" bis "Levelaufbau" schreiben sowie Umsetzung des Geschriebenen   |
| 16.04.2017 | 10:45 | Kapitel "Umgebung" von "Undurchdringbarkeit" bis "User Interface" schreiben und umsetzen   |
| 17.04.2017 | 11:00 | Unterkapitel "Arme" im Kapitel "Aussehen" sowie "Gegner" bis "Beschossen werden" schreiben und umsetzen  |
| 18.04.2017 | 04:30 | Kapitel "Gegner" ab "Beschossen werden (Trefferzone)" bis "Schießen" schreiben und umsetzen  |
| 03.10.2017 | 08:00 | Versuch, Processing wieder zum Laufen zu bringen   |
| 06.10.2017 | 10:45 | Processing zum Laufen bringen, Kapitel "Startposition" und "Sinne" schreiben und implementieren  |
| 07.10.2017 | 09:00 | Kapitel "Waffenhöhe", "Stratagence", "Dominante Strategien eliminieren", "Exkurs: Neuronales Netz", "Aufbau des Neuronalen Netzwerks" und "Implementierung" schreiben und umsetzen |
| 08.10.2017 | 08:00 | Kapitel "Erfolge", "Auslese" und "Zusammenfassung und Ausblick" schreiben, implementieren, Versuche durchführen und Beleg finalisieren   |
| Summe      | 72:30 |  |

## Anlagen: Netzwerke

Die gespeicherten Daten, welche die Netzwerke darstellen, sind in drei Dateien gespeichert:

- „networks\_round\_01“
- „networks\_round\_02“
- „networks\_round\_03“

## Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 08.10.2017

A handwritten signature in black ink, appearing to read 'Sebastian Schleier'. The signature is fluid and cursive, with a long horizontal stroke at the end.

Sebastian Schleier