Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Projektbericht

Gaming AI

vorgelegt von

Friedrich Braun
Valentin Krön

Fakultät für Mathematik, Informatik und Naturwissenschaften
Fachbereich Informatik
Arbeitsbereich Wissenschaftliches Rechnen

Studiengang:	Wirtschaftsinformatik	Informatik
Matrikelnummer:	6252218	6700970
Betreuer:	Eugen Betke	Julian Kunkel

Hamburg, 2017-01-01

Abstract

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Contents

1. Einleitung

In this chapter, ...

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.[1]

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet:

- Lorem
- Ipsum

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. “Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.” [**?**]

Reference example: For more information, see

- Chapter **??**,
- **??**,
- **??** on page ??.

[1] I am a footnote.

2. Projektziel

In diesem Kapitel gehen wir auf das Projektziel näher ein und erläutern warum wir das Projektziel ändern mussten.

2.1. Ursprüngliches Projektziel

Unser ursprüngliches Projektziel bestand darin eine AI zu entwickeln und diese mit Hilfe eines genetischen Algorithmus zu trainieren. Diese AI sollte auf die Spring-RTS-Engine [Spring] angewandt werden. Die Absicht bestand darin zu zeigen, dass eine AI auf diese Weise lernen kann RTS-Spiele gut, eventuell sogar besser als menschliche Spieler, zu spielen.

2.2. Gründe Für die Änderung des Projektziels

Bei der Realisierung dieser Zielsetzung trafen wir allerdings auf unlösbare Probleme. Eines der Probleme lag im unübersichtlichen Aufbau der Spring-Engine [Spring] selbst. Zwar bietet die Spring-Engine eine Lobby an, jedoch lassen sich nur wenige Spiele tatsächlich starten und selbst bei diesen lassen sich keine externen AI's einbinden. Dadurch fehlte uns eine Vorlage, auf der wir unsere AI hätten aufbauen können; insbesondere fehlte uns Wissen zu den Schnittstellen zwischen unserer AI und der Spring-Engine [Spring].
Einige Spiele haben im Code für das Spiel direkt eingebundene AI's, die auch lauffähig sind. Diese AI's halfen uns jedoch auch nicht weiter, gerade da sie direkt ins Spiel hineinkompiliert sind. Dies bedeutet nämlich nicht nur, dass die AI selber kaum zu Debuggen ist, man muss auch direkt am Spiel selber mit der Sprache lua arbeiten. Da die Auseinandersetzung mit dem Aufbau eines Spieles für die Spring-Engine und mit der Sprache lua den Projektrahmen gesprengt hätte, entschieden wir uns für eine Änderung der Zielsetzung.

2.3. Neues Projektziel

Da eine AI für die Spring-Engine selber nicht möglich war, beschlossen wir eine Simulation, also eine vereinfachte Version eines Spiels auf Basis der Spring-Engine, zu bauen. Wichtig ist bei unserem neuen Ziel vor allem ein "proof-of-concept", sprich: Wir wollen zeigen, dass es möglich ist eine AI für ein RTS zu bauen und diese mit Hilfe eines genetischen Algorithmus zu trainieren.

3. Aufbau

In diesem Kapitel beschreiben wir kurz den Aufbau der wichtigsten Bausteine des Projekts.

3.1. Aufbau der Simulation

Die Simulation ist ein sehr einfaches RTS. Sie ist ausgelegt auf genau zwei Spieler. Jeder der Spieler hat ein Hauptgebäude, dass in regelmäßigen Zeitabständen Einheiten erstellt. Diese Einheiten können zu Positionen auf dem Spielfeld laufen und, wenn sie dicht genug dran sind, Schaden an gegnerischen Einheiten verursachen. Fallen die Lebenspunkte einer Einheit auf null, so wird diese gelöscht.
Es gibt genau einen Einheitentyp. Die Simulation beinhaltet keinerlei Mechaniken, die Ressourcen in irgendeiner Form verwenden.
Ein Match dauert zehn Minuten.

3.2. Aufbau der AI

Da bei einem RTS die Anzahl der Einheiten, und damit die Anzahl der Möglichkeiten, variabel ist, entschieden wir uns, auch durch Anraten der Betreuer, dafür, die AI als Schwarmintelligenz zu designen.
Jede Einheit hat einen Sichtradius und gibt diese Information als Input an die AI. Diese wiederum generiert daraus über ein neuronales Netz eine Entscheidung. Damit verhalten sich alle Einheiten gleich bei gleicher Eingabe.
Durch den Aufbau als Schwarmintelligenz ist die AI problemlos skalierbar. Zudem benötigen wir keinerlei Koordination der Einheiten, da diese sich aus der Schwarmintelligenz von alleine ergibt.

3.3. Aufbau des Trainingsalgorithmus

Das Training der AI erfolgt dadurch, dass das neuronale Netz trainiert wird. Im Detail heißt das: Wir verwenden die Gewichte des neuronalen Netzes als Gene in einem genetischen Algorithmus. Dabei müssen wir nicht zwischen Geno- und Phänotyp unterscheiden. Die Rekombination erfolgt über einen zufälligen Schnitt. Die Mutation findet mit einer gewissen Wahrscheinlichkeit statt, wenn sie stattfindet so wird genau ein Gen durch ein zufälliges neues ersetzt. Die Fitness-Funktion basiert auf der Anzahl lebender Einheiten, eigener wie gegnerischer, bei Spielende. Die genaue Umsetzung haben

wir zu Auswertungszwecken variiert, weswegen wir in der Auswertung auch nocheinmal genauer darauf eingehen.

4. Umsetzung

In diesem Kapitel beschreiben wir kurz die Umsetzung der wichtigsten Bausteine des Projekts.

4.1. Umsetzung der Simulation

Die Simulation entwickelten wir in Unity [Unity]. Wir entschieden uns für Unity, da wir Unity für unser Projekt kostenlos nutzen können und da Unity allgemein sehr bekannt ist, weswegen es viele Onlinetutorials gibt.
Unity lässt für seine Skripte nur C# oder Javascpript, bzw. mitlerweile Unityscript, zu. Wir entschieden uns für C#, da dies näher an den Sprachen ist, mit denen wir uns bereits auskannten.
Da wir beide auf Windows-Rechnern arbeiten und sich Microsoft Visual Studio automatisch öffnete, als wir das erste Mal versuchten ein Skript zu öffnen, blieben wir bei dieser IDE.

4.2. Umsetzung der AI

Wie bereits im Aufbau erwähnt, ist der Kern der AI ein neuronales Netz. Dazu suchten wir uns eine Bibliothek [NN-Bib], sodass wir die Struktur nicht selber bauen mussten.
Eine große Hürde bei der Verwendung eines neuronalen Netzes bestand darin, dass das neuronale Netz als Input Floats erwartet und den Output auch in Form von Floats liefert, was uns dazu zwingt den Input auf Floats herunter zu brechen. Die Umsetzung in der von uns verwendeten Bibliothek verwendet den Wertebereich null bis eins.
Als Input verwenden wir die eigene Position der Einheit auf der Ebene, die wir einfach linear reskalieren. Dazu kommen die Einheiten, inklusive der Hauptgebäude, im Sichtradius der Einheit, sowohl freundliche als auch feindliche. Je ein Inputneuron verwenden wir um zu signalisieren, ob das jeweilige Hauptgebäude (Freund/Feind) zu sehen ist. Ist dies der Fall, so gibt ein weiteres Neuron eine ungefähre Entfernungsabschätzung an. Zusätzlich verwenden wir noch je ein Neuron um die Anzahl der Freunde bzw. Feinde im Sichtbereich zu codieren; dies geschieht indem wir diese Anzahl durch die maximal mögliche Anzahl teilen. Insgesamt kommen wir somit auf acht Inputneuronen.
Um die AI sinnvoll einsetzen zu können, brachen wir zuerst einmal die Entscheidungsmöglichkeiten des Menschen bzw. der AI auf zwei Funktionen herunter:

- SetDestination: Bekommt die Koordinaten eines Punktes in der Ebene und setzt die Zielposition der Einheit auf diesen Punkt. Die Einheit läuft dann zum Zielpunkt.

- SetTarget: Bekommt eine andere Einheit oder ein Hauptgebäude als ”target”. Solange dieses ”target” existiert, läuft die Einheit darauf zu und, sollte es feindlich sein, greift es an.

Daraus leitet sich der Output ab. Das erste Outputneuron entscheidet darüber welche der beiden Funktionen, SetDestination oder SetTarget, ausgeführt wird. Zwei Neuronen bestimmen dann die Position des Zielpunktes; es erfolgt wieder eine lineare Reskalierung. Drei weitere Neuronen dienen dann zur Bestimmung des ”target”. Das erste davon legt fest, ob ein Freund oder Feind zum ”target” wird, das zweite legt fest ob das jeweilige Hauptgebäude zum ”target” wird und das dritte legt fest welche Einheit zum ”target” wird, wenn es nicht das Hauptgebäude ist. Dabei fangen wir natürlich die Fälle ab, bei denen sich ein ”target” ergeben würde, das nicht existiert. Somit kommen wir auf sechs Outputneuronen.
Wir entschieden uns dafür ein Hidden-Layer zu verwenden, da wir es wichtig fanden mindestens ein Hidden-Layer zu haben, allerdings das neuronale Netz nicht unnötig aufblähen wollten. Die Entscheidung zehn Neuronen in diesem Layer einzubauen trafen wir, weil aus unserem, zugegebenermaßen beschränkten, Erfahrungsschatz hervorgeht, dass man die Größenordnung der Layer zu einander nicht variiert.
Dadurch kommen wir auf 140 Gewichte $(8 * 10 + 10 * 6)$.
Die von uns verwendeten Gewichte liegen im Wertebereich von minus eins bis eins.

5. Conclusion

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Appendices

[Spring] https://springrts.com/ [Unity] https://unity3d.com/ [NN-Bib] http://franck.fleurey.free.fr/

A. Chapter

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

List of Figures

List of Listings

List of Tables

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Studiengang XXX selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Ort, Datum	Unterschrift

Veröffentlichung

Ich bin damit einverstanden, dass meine Arbeit in den Bestand der Bibliothek des Fachbereichs Informatik eingestellt wird.

Ort, Datum	Unterschrift