



Bachelor Thesis 2013

Fachbereich Informatik

Decision Making in Starcraft

Pflichtenheft

Studierende: Hulliger Moritz
Lauener Benjamin

Professoren: Dr. Jürgen Eckerle

Experte: Dr. Federico Flückiger

Datum: 6. April 2013

Dieses Dokument dient als Wegweiser für die zu erledigenden Arbeiten und beinhaltet die Ziele und Anforderungen, welche im Rahmen der Bachelor Thesis umgesetzt werden.



Änderungsverzeichnis

Version	Datum	Autor	Änderung
0.1	18.02.2013	laueb2	Aufsetzen der Initialversion
0.2	19.02.2013	laueb2, hullm2	Beschreibung der Scouting und Buildmanager Anforderung
0.3	23.02.2013	laueb2, hullm2	Beschreibung der Pathfinding und Decision Making Anforderung
0.4	27.02.2013	laueb2, hullm2	Beschreibung der Kommando und Micromanager Anforderung
0.5	10.03.2013	laueb2, hullm2	Einleitung erweitert, allgemeine Korrekturen
0.6	06.04.2013	laueb2	Überarbeitung des Dokuments

Tabelle 0.1: Änderungsverzeichnis

Referenzierte Dokumente

Nummer	Dokument
1	Projekt 2 - Abschlussdokument - 2013-01-13.pdf

Tabelle 0.2: Referenzierte Dokumente





Inhaltsverzeichnis

1	Einleitung	1
1.1	Starcraft	1
1.2	Ausgangslage	3
2	Anforderungen	5
2.1	Pathfinding	5
2.2	Build Manager	5
2.3	Scouting	6
2.4	Building Decision Making	6
2.5	Kommando Manager	7
2.6	Micro Manager	7
2.7	Systemanforderungen	8
3	Priorisierung	9
	Glossar	11





Abbildungsverzeichnis

1.1	Kampf zwischen Terraner und Protoss	2
1.2	BWAPI Logo	2





Tabellenverzeichnis

0.1	Änderungsverzeichnis	i
0.2	Referenzierte Dokumente	i
2.1	Anforderungen: Pathfinding	5
2.2	Anforderungen: Build Manager	5
2.3	Anforderungen: Scouting	6
2.4	Anforderungen: Building Decision Making	6
2.5	Anforderungen: Kommando Manager	7
2.6	Anforderungen: Micro Manager	7
2.7	Anforderungen: Systemanforderungen	8
3.1	Priorisierung	9





1 Einleitung

Das vorliegende Dokument beschreibt die Anforderungen und Ziele, welche im Frühjahrssemester 2013 im Rahmen der Bachelor Thesis umgesetzt werden. Kurz zusammengefasst ist das Ziel unserer Arbeit einen Bot zu erstellen, welcher in der Lage ist ein Starcraft Spiel zu absolvieren. Dabei geht es uns vorwiegend darum, anhand von bekannten Informationen mittels Decision Making Entscheidungen zu treffen. Ausserdem wäre es interessant mit unserem Bot an der AIIDE¹ (Artificial Intelligence and Interactive Digital Entertainment) teilzunehmen. Die AIIDE veranstaltet seit 2010 jährlich einen Wettbewerb, an welchem selbst entwickelte Bots gegeneinander antreten.

1.1 Starcraft

Starcraft ist ein sehr populäres Echtzeit-Strategiespiel, welches sich, aufgrund seiner Ausgewogenheit zwischen den verschiedenen Rassen, vorallem im E-Sport überaus grosser Beliebtheit erfreut. Das Spiel Starcraft wurde im Jahr 1998 von Blizzard Entertainment (heute Activision Blizzard²) entwickelt. Unterdessen wurde eine Erweiterung für Starcraft, genannt „Starcraft Broodwar“, und das Spiel „Starcraft 2“ von Blizzard herausgegeben. Am 12.03.2013 erscheint die erste Erweiterung „Heart Of The Swarm“ für Starcraft 2.

1.1.1 Echtzeit-Strategiespiel

In einem Echtzeit-Strategiespiel (engl. Real Time Strategy Game) agieren alle Gegner gleichzeitig auf dem Spielfeld. Das Abbauen von Rohstoffen, Erbauen von Gebäuden und Einheiten, sowie das Erforschen von neuen Technologien sind in Echtzeit simuliert. Ziel ist das Besiegen der Gegner durch das taktisch geschickte Ausbauen der eigenen Basis (Build Manager 2.2), sowie das Ausbilden von Kampfeinheiten, um alle gegnerischen Einheiten und Gebäude in einer oder mehreren Schlachten zu zerstören. Oftmals treffen eigene Einheiten auch rein zufällig auf gegnerische, da die Spielkarte meistens mit dem „Fog Of War“ verdunkelt ist, so dass die gegnerischen Aktivitäten nicht ersichtlich sind. In diesen Situationen gilt es zu ermitteln, ob gekämpft oder die Flucht ergriffen werden soll (Kommando Manager 2.5) und welche Strategie man in einem Kampf verfolgen will (Micro Manager 2.6). Um aber den Gegner im Vorhinein zu lokalisieren und mehr über sein Vorgehen in Erfahrung zu bringen, kann mittels einer Strategie, welche Scouting genannt wird, die Karte mit einer eigenen Einheit abgesucht werden. Da beim Scouting nicht das Bekämpfen von Gegnern verfolgt wird, ist die Pfadsuche (engl. Pathfinding) relevant, so dass man beim Erkunden der Karte einen Pfad verwendet, welcher gesichtete feindliche Einheiten umgeht. Aus den gewonnenen Informationen, welche das Scouting liefert, kann oft auf eine gegnerische Strategie geschlossen werden. Ziel ist es dann, seine eigene Strategie so anzupassen, dass der Gegner eine möglichst geringe Gefahr darstellt (Decision Making 2.4).

1.1.2 Starcraft Broodwar

Das erste Spiel aus der Starcraft Reihe mit der Broodwar Erweiterung ist die Umgebung, in welcher der von uns entwickelte Bot laufen wird. In dem Spiel gibt es drei verschiedene Rassen, die Terraner, Zerg und Protoss, welche unterschiedliche Einheiten und Gebäude ausbilden können und somit unterschiedliche Funktionen und Fähigkeiten bieten. Jede Rasse hat seine Stärken und Schwächen und es gibt verschiedene Strategien die Schwächen einer gegnerischen Rasse auszunutzen um sich einen Vorteil zu verschaffen. Damit eine Zivilisation erweitert werden kann, müssen sogenannte Mineralien abgebaut werden. Stärkere Einheiten und Gebäude benötigen auch noch Gas, welches über ein Gasdepot gewonnen wird.

¹<http://www.aiide.org> 10.03.2013

²<http://www.activisionblizzard.com> 10.03.2013



Abbildung 1.1: Kampf zwischen Terraner und Protoss

Es existiert eine API (BWAPI 1.1.3), welche es Entwicklern ermöglicht, eigene AI Module für das Spiel zu entwickeln.

1.1.3 BWAPI

Die BWAPI³ (BroodWar Application Programming Interface) ist ein Opensource C++ Framework zur Erstellung von AI Modulen für das Spiel Starcraft Broodwar. Seit 2008 wird an der API aktiv weiterentwickelt und sie ermöglicht es, Informationen über Einheiten und Spieler zu bekommen sowie eine Vielzahl von Befehlen zu erteilen. BWAPI erlaubt es nur Informationen zu bekommen die auch für den Spieler sichtbar sind. Das bedeutet, dass wenn

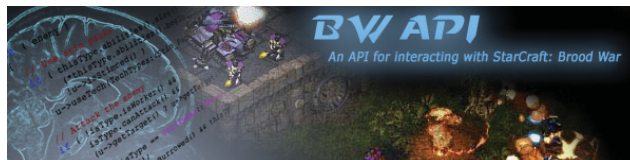


Abbildung 1.2: BWAPI Logo <https://code.google.com/p/bwapi> 10.03.2013

eine gegnerische Einheit zurück in den „Fog Of War“ geht, über diese Einheit keine neuen Informationen abgeholt werden können. Dies erlaubt es mittels der BWAPI faire AI's zu entwickeln die unter den gleichen Umständen wie menschliche Spieler funktionieren, was unter anderem eine der Anforderungen für die Teilnahme an der AIIDE ist.

1.1.4 BWSAL

BWSAL⁴ (BroodWar Standart Addon Library) ist eine Opensource C++ Library die auf der BWAPI aufbaut. Die BWSAL implementiert mittels der BWAPI einige Grundlegende Funktionen die benötigt werden um mit einen Starcraft Bot zu erstellen. Dabei ist die nützlichste Funktionalität, welche verfügbar ist, das koordinierte Bauen von Gebäuden nach Priorität.

³<https://code.google.com/p/bwapi> 10.03.2013

⁴<https://code.google.com/p/bwsal> 10.03.2013



1.2 Ausgangslage

1.2.1 Projekt 2: Ergebnisse

Als Grundlage der Zieldefinition dienen die Resultate aus dem Projekt 2 und es wird somit auf diesem Codestand weitergearbeitet. Darin wurde eine eigene Pfadsuche implementiert mit welcher Gebiete vermieden werden, in welchen eine bestimmte Zeit zuvor feindliche Einheiten gesichtet wurden. Ausserdem wurde durch die Erstellung von Hilfsklassen (Logger, ArgosMap) eine Grundlage für ein einfacheres Weiterarbeiten gelegt. Eine technisch detaillierte Beschreibung der Ergebnisse kann im referenzierten Dokument 1 (Projekt 2 - Abschlussdokument - 2013-01-13.pdf) eingesehen werden.

1.2.2 Datamining

Eine Anforderung in diesem Dokument ist das Treffen von Entscheidungen aufgrund von bereits vorhandenen Informationen. Für das Treffen der Entscheidungen sammelten wir bereits ARFF⁵ Dateien aus über 5000 Aufzeichnungen von Kämpfen zwischen professionellen Starcraft Spielern. Für dieses Datamining⁶ werden wir das Programm WEKA⁷ einsetzen.

⁵<http://weka.wikispaces.com/ARFF>

⁶http://eis.ucsc.edu/StarCraft_Data_Mining

⁷<http://www.cs.waikato.ac.nz/ml/weka/>





2 Anforderungen

2.1 Pathfinding

Die Pfadsuche war eine Anforderung, welche im Rahmen des Projekts 2 umgesetzt wurde. Jedoch hat die Suche noch Performance Probleme, welche im Rahmen dieser Arbeit behoben werden sollen.

Nummer	Anforderung
1	Berechnung des Pfades erfolgt innerhalb eines Frames

Tabelle 2.1: Anforderungen: Pathfinding

2.2 Build Manager

Der Build Manager ist verantwortlich für das Ausbilden von Einheiten und Erbauen von Gebäuden. Ausserdem wird auch das Abbauen von Ressourcen vom Build Manager organisiert. Zusammengefasst sind folgende Anforderungen dazu notwendig:

Nummer	Zeitpunkt	Anforderung
1	In zeitlichen Intervallen	Ausbilden von Arbeitern und Kampfeinheiten
2	In zeitlichen Intervallen	Erbauen von Gebäuden
3	Konstant	Abbauen und managen von Ressourcen
4	In zeitlichen Intervallen	Abarbeiten der Aufträge nach Priorität
5	Zu Beginn des Spiels	Die Vorgegebene Build Order einlesen und umsetzen

Tabelle 2.2: Anforderungen: Build Manager

- Nummer 4 ist verantwortlich für die Abarbeitung von Nummer 1-3 nach Priorität. Das heisst, der Build Manager priorisiert die eingehenden Aufträge und lässt sie ausführen. Die Priorität für das Erbauen von Gebäuden ist durch Nummer 5 gegeben.
- Ziel von Anforderung Nummer 5 ist das Einlesen einer vorgegebenen Build Order, welche dem Bot von uns angegeben wird. In welcher Form die Build Order vorgegeben wird, muss noch ermittelt werden. Dabei könnte es sich um eine Xml- oder Textdatei handeln. Ausserdem muss von uns noch analysiert werden, welche Reihenfolge für das erfolgreiche Vorgehen in einem Spiel notwendig ist.



2.3 Scouting

Ziel des Scoutings ist das Auskundschaften der Landkarte und Lokalisieren der gegnerischen Einheiten. Für diese Funktionalität sind folgende Anforderungen umzusetzen:

Nummer	Zeitpunkt	Anforderung
1	Beginn des Spiels	Drohne sucht alle möglichen Startpositionen auf gegnerische Einheiten und Gebäude ab
2	In zeitlichen Intervallen	Drohne durchkämt die Karte und lokalisiert gegnerische Einheiten
3	Sichtkontakt mit gegnerischen Einheit	Speichern von Einheitstyp, Stärke und Position der gegnerischen Einheit

Tabelle 2.3: Anforderungen: Scouting

- Durch Nummer 1 wird gleich zu Beginn die Basis des Gegners lokalisiert und der Bot weiss, wo sich der Gegner hauptsächlich aufhält
- Nummer 2 entspricht einer Überwachung, mit welcher laufend die Positionen der gegnerischen Einheiten ermittelt werden
- Nummer 3 stellt sicher, dass die Informationen für eigene, spätere Aktivitäten im Spiel für den Bot verfügbar sind. Diese Anforderung bezieht sich nicht nur auf die Drohne, sondern auch auf alle anderen eigenen Einheiten.

2.4 Building Decision Making

Das Ziel ist es anhand von bekannten Informationen zu entscheiden welche Gebäude und Einheiten benötigt werden. Dazu werden Entscheidungsalgorithmen benutzt die mit Hilfe von Datamining funktionieren. Zuerst soll mittels des Decision Making entschieden werden welche Strategie der Gegner verfolgt. Weiter muss entschieden werden wie auf diese Strategie am besten reagiert wird.

Nummer	Anforderung
1	Verarbeiten der gesammelten Daten und finden des besten Algorithmus' mittels WEKA
2	Den Algorithmus aus Anforderung 1 im Bot abbilden
3	Gewonnen Informationen aus dem Scouting (2.3) nutzen um die Strategie des Gegners zu klassifizieren
4	Entscheiden wie auf eine ermittelte gegnerische Strategie reagiert wird

Tabelle 2.4: Anforderungen: Building Decision Making

Mithilfe von WEKA soll ein Optimaler Algorithmus gefunden werden, mit dem die Strategie des Gegners möglichst präzise Klassifiziert werden kann. Dabei gilt es zu beachten, dass wir nie die kompletten Informationen besitzen. Zum Beispiel ist der Zeitpunkt an welchem ein bestimmtes Gebäude vom Gegner gebaut wird oftmals ein wichtiger Indikator, welche Strategie von der anderen Partei verfolgt wird. Diese Information ist meistens aber nicht verfügbar, da gegnerische Gebäude erst sichtbar werden, wenn eine eigene Einheit sie entdeckt. Solche und andere Fälle von fehlenden Werten muss unser Algorithmus für die Entscheidungsfindung in Betracht ziehen. Hat man die Strategie klassifiziert gilt es zu entscheiden wie darauf reagiert werden soll.



2.5 Kommando Manager

Der Kommando Manager organisiert die Truppenbewegung und ermittelt bei Sichtkontakt mit dem Gegner ob eine Angriffs- oder Rückzugsstrategie verfolgt werden soll.

Nummer	Zeitpunkt	Anforderung
1	Bei Sichtkontakt mit Gegner	Ermittlung der Chance auf Sieg
2	Gegner ist zu stark	Rückzug der eigenen Truppen
3	Chance auf Sieg	Angriff des Gegners

Tabelle 2.5: Anforderungen: Kommando Manager

Für Anforderung Nummer 1 werden die Informationen verwendet, welche zum Zeitpunkt des Aufeinandertreffens bekannt sind. Dabei wird der Typ, die Stärke, die Anzahl gegnerischer Einheiten und die Lebenspunkte eine Rolle spielen. Wie die verschiedenen Faktoren am besten miteinander kombiniert werden sollen, muss von uns noch ermittelt werden.

2.6 Micro Manager

Der Micro Manager ist dafür zuständig, die Einheiten im Kampf zu bewegen und Angriffe auszuführen. Weiter organisiert der Micro Manager die Einheiten während einer Truppenverschiebung. Er sorgt dafür, dass sich die Einheiten bei der Bewegung nicht gegenseitig behindern.

Nummer	Anforderung
1	Koordinierte Bewegung der Truppen
2	Intelligentes Micro während der Kämpfe

Tabelle 2.6: Anforderungen: Micro Manager

Der Micro Manager soll mittels Potential Field umgesetzt werden.

- In der Anforderung Nummer 1 geht es darum, die im Projekt 2 erarbeitete Pfadsuche auf mehrere Einheiten anzuwenden und dafür zu sorgen, dass sich diese intelligent fortbewegen.
- Ziel von Anforderung Nummer 5 ist das Einlesen einer vorgegebenen Build Order, welche dem Bot von uns angegeben wird. In welcher Form die Build Order vorgegeben wird, muss noch ermittelt werden. Dabei könnte es sich um eine Xml- oder Textdatei handeln. Ausserdem muss von uns noch analysiert werden, welche Reihenfolge für das erfolgreiche Vorgehen in einem Spiel notwendig ist.



2.7 Systemanforderungen

Damit die AI ausgeführt werden kann müssen folgende Anforderungen erfüllt sein.

Nummer	Anforderung
1	Windows 7: Als Betriebssystem wird Windows 7 eingesetzt.
2	Starcraft BW 1.16.1: Starcraft Broodwar muss in der Version 1.16.1 installiert sein.
3	Chaoslauncher 0.5.4.4: Der Chaoslauncher wird benötigt um die entwickelte AI ins Spiel zu injecten.
4	BWAPI 3.7.4: Damit die AI läuft muss die BWAPI v3.7.4 installiert sein
5	Eigene Rasse: Als eigene Rasse werden die Terraner eingesetzt
5	Gegnerische Rasse: Der Bot ist fähig gegen alle möglichen Rassen zu spielen.

Tabelle 2.7: Anforderungen: Systemanforderungen

- Der Chaoslauncher¹ dient dazu die generierte DLL in das Spiel einzuspeisen, so dass dann die AI als Spieler agiert.

¹<http://wiki.teamliquid.net/starcraft/Chaoslauncher> 10.03.2013



3 Priorisierung

In dieser Arbeit legen wir unser Augenmerk hauptsächlich auf die Entscheidungsfindung mittels Data Mining. Ausserdem sind wir interessiert an der Umsetzung der Funktionen, welche erforderlich sind, damit ein Bot selbstständig ein vollständiges Spiel spielen kann. Damit wir im Falle von unerwarteten Zeitverzögerungen trotzdem noch unsere Hauptziele erreichen können, werden die Anforderungen in Prioritäten *hoch*, *mittel* und *tief* eingeteilt und sortiert.

hoch: die Anforderung muss zwingend für eine erfolgreiche Bachelorthesis umgesetzt werden

mittel: Ziel ist die Umsetzung der Anforderung, kann jedoch bei Zeitmangel vernachlässigt werden

tief: die Funktion ist wünschenswert und wird nur bei Zeitüberschuss umgesetzt

Kapitel	Nummer	Anforderung	Priorität
2.1 Pathfinding	1	Berechnung des Pfades erfolgt innerhalb eines Frames	hoch
2.3 Scouting	3	Speichern von Einheitstyp, Stärke und Position der gegnerischen Einheit	hoch
2.4 Building Decision Making	1	Verarbeiten der gesammelten Daten und finden des besten Algorithmus' mittels WEKA	hoch
2.4 Building Decision Making	2	Den Algorithmus aus Anforderung 1 im Bot abbilden	hoch
2.4 Building Decision Making	3	Gewonnen Informationen aus dem Scouting (2.3) nutzen um Entscheidungen zu treffen	hoch
2.4 Building Decision Making	4	Entscheiden wie auf eine ermittelte gegnerische Strategie reagiert wird	hoch
2.2 Build Manager	1	Ausbilden von Arbeitern und Kampfeinheiten	mittel
2.2 Build Manager	2	Erbauen von Gebäuden	mittel
2.2 Build Manager	3	Abbauen und managen von Ressourcen	mittel
2.3 Scouting	1	Drohne sucht alle möglichen Startpositionen auf gegnerische Einheiten und Gebäude ab	mittel
2.5 Kommando Manager	3	Angriff des Gegners	mittel
2.2 Build Manager	4	Abarbeiten der Aufträge nach Priorität	tief
2.2 Build Manager	5	Die Vorgegebene Build Order einlesen und umsetzen	tief
2.3 Scouting	2	Drohne durchkämmt die Karte und lokalisiert gegnerische Einheiten	tief
2.5 Kommando Manager	1	Ermittlung der Chance auf Sieg	tief
2.5 Kommando Manager	2	Rückzug der eigenen Truppen	tief
2.6 Micro Manager	1	Koordinierte Bewegung der Truppen	tief
2.6 Micro Manager	2	Intelligentes Micro während der Kämpfe	tief

Tabelle 3.1: Priorisierung





Glossar

Build Order Eine Strategie für die ersten Minuten des Spiels um die Ressourcen und das Timing zu optimieren, sie definiert in welcher Reihenfolge welche Gebäude erstellt werden.

Fog Of War Gebiet, welches noch nicht erkundet wurde. Unerkundete Gebiete werden auf der Karte Schwarz dargestellt. Somit sind die gegnerischen Aktivitäten nicht ersichtlich.

Micro Als Micro wird die Fähigkeit bezeichnet Truppen individuell zu bewegen.

Potential Fields Eine Karte wird in verschiedene Felder unterteilt. Für jedes dieser Felder wird aufgrund von zu definierenden Gegebenheiten ein Potential vergeben. Einheiten bewegen sich nun auf die Felder mit dem grössten Potential.