
Sind (trainierte) general-purpose RL Agenten im Brettspiel Othello stärker als (untrainierte) General-Game-Playing Agenten?

Informatikprojekt im Studiengang Informatik
an der Fakultät für Informatik und Ingenieurwissenschaften
der Technischen Hochschule Köln

vorgelegt von: Johannes Scheiermann
Matrikel-Nr.: 11110001
Adresse: Dieringhauser Str. 129
51645 Gummersbach
johannes.scheiermann@live.de

eingereicht bei: Prof. Dr. rer. nat. Wolfgang Konen

Gummersbach, 17.08.2020

Erklärung

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer oder des Verfassers selbst entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Schleiden, 17.08.2020

Ort, Datum

Johannes Scheiermann

Rechtsverbindliche Unterschrift

Inhaltsverzeichnis

Erklärung	I
Tabellenverzeichnis	III
Abbildungsverzeichnis	IV
Abkürzungsverzeichnis	V
1 Einleitung	1
1.1 GBG	1
1.2 Ludii	2
1.3 Projektablauf	2
2 Implementierung der GBG-Ludii-Schnittstelle	3
2.1 Planung	3
2.2 Transform-Schicht	4
2.2.1 Spielbrett	4
2.2.2 Spieler	6
2.3 Use-Cases-Schicht	7
2.3.1 Spielzüge	7
2.3.2 Spielzustand	9
2.4 Gateway-Schicht	11
2.5 SDK- und Frameworkabhängigkeiten	12
3 Datenerhebung und -evaluation	13
4 Hürden und Fallstricke	15
5 Nachtrag	16
5.1 Software-Erweiterungen	16
5.2 Erneute Datenerhebung und -evaluation	16
5.3 Anleitung zur Verwendung der GBG-Ludii-Schnittstelle	17
5.3.1 Verwendung in der grafischen Ludii-Oberfläche	17
5.3.2 Programmgesteuerte Verwendung	18
5.4 Fazit	19
Literaturverzeichnis	20
Anhang	21

Tabellenverzeichnis

1	Indizierung der Spielerfarben	6
---	---	---

Abbildungsverzeichnis

1	Architekturspezifikation der GBG-Ludii-Schnittstelle	3
2	Indizierung des Othello-Spielbretts	4
3	Klassen zur Transformation der Spielbrettindizes als UML-Diagramm	5
4	Klassen zur Transformation der Spielerindizes als UML-Diagramm	6
5	Framework-spezifische Spielzüge-Klassen als UML-Diagramm	7
6	Nutzung eines Ludii-Kontextes in einer GBG-StateObservation als UML-Diagramm	9
7	Relevanter Teil der Ludii-Schnittstelle für eigene Agenten als UML-Diagramm	11
8	Implementierung der initAI-Methode als Java-Code	11
9	Implementierung der selectAction-Methode als Java-Code	12
10	Implementierung der gbgAction-Methode als Java-Code	12
11	Erhobene Spieldaten zwischen dem TD-3-Tupel- (GBG) und dem LudiiAI-Agenten (Version 0.9.3)	13
12	Erhobene Spieldaten zwischen dem TD-3-Tupel- (GBG) und dem LudiiAI-Agenten (Version 1.0.2)	17
13	Screenshot der IntelliJ IDEA	18
14	Auswahl eines beliebigen exportierten GBG-Agenten	19

Abkürzungsverzeichnis

GBG general board game playing and learning

GDL Game Description Language

GGP General Game Playing

RL Reinforcement Learning

1 Einleitung

In dieser Ausarbeitung wird untersucht, ob ein Agent, welcher sich durch einen Reinforcement Learning (RL) Ansatz eigenständig Spielstrategien aneignet, einem trainingsunfähigen Allzweck-Agenten überlegen ist. Die Untersuchung beschränkt sich auf das Brettspiel Othello und zieht zu diesem Zweck die beiden Frameworks general board game playing and learning (GBG) und Ludii hinzu, welche im Folgenden näher erläutert werden.

1.1 GBG

Das GBG-Framework wird von Prof. Dr. Wolfgang Konen und seiner Forschungsgruppe an der Technischen Hochschule Köln entwickelt und befasst sich mit Game Learning. Wie der Name vermuten lässt, beschränkt es sich auf Brettspiele. Diese kennzeichnen sich vor allem dadurch, dass die Spieler ihre Züge in abwechselnder Reihenfolge vornehmen. Unterstützt werden jegliche Brettspiele mit beliebigen Spieleranzahlen unabhängig davon, ob sie deterministisch sind oder nicht.

Mit dem GBG-Framework steht somit, eine in der Programmiersprache Java geschriebene, Open-Source-Software zur Verfügung¹, welche die Benutzeroberflächen mitbringt, um die bereitgestellten Spiele spielen zu können. An einem Spiel können dabei beliebige Kombinationen aus menschlichen oder verschiedenen KI-Spielern teilnehmen, solange das Spielerlimit des entsprechenden Spiels nicht überschritten wird. Um Forschungsvorhaben in diesem Bereich zu unterstützen, stellt das Framework diverse Trainings-, Turnier- und Evaluationsumgebungen zur Verfügung. Außerdem werden Interfaces bereitgestellt, welche nicht nur die Entwicklung eigener Spiele, sondern auch die Implementierung weiterer KI-Agenten ermöglichen.

Anders als man es aus dem General Game Playing (GGP) kennt, können weitere Spiele nicht mit Game Description Language (GDL) beschrieben und zur Laufzeit hinzugefügt werden. Stattdessen müssen diese zur Compilezeit im Java-Quellcode implementiert werden. Für Agenten steht dagegen eine Import-Möglichkeit zur Laufzeit bereit. Das liegt daran, dass das GBG-Framework darauf abzielt eine Umgebung bereitzustellen, in der KI-Agenten ohne vorherige Spielkenntnisse die Möglichkeit haben, sich solche in beliebigen Spielen durch eigenständiges und effizientes Training anzueignen, sodass sie anschließend eine gute Performance in diesen aufweisen.

So beschreibt Świechowski im Jahr 2015 [1, p. 19], dass im GGP Methoden der künstlichen Intelligenz durch GDL erschwert werden, wodurch sie in kompetitiven Vergleichen nicht viel Erfolg haben.

Im GBG-Framework müssen Agenten nicht speziell auf bestimmte Spiele abgestimmt werden und können, sobald sie einmal entwickelt sind, in jedem beliebigen Spiel gegen weitere Agenten oder Menschen antreten.

¹Abrufbar unter <https://github.com/WolfgangKonen/GBG>

1.2 Ludii

Als sogenanntes General Game System wird Ludii im Rahmen des Digital Ludeme Projekts der Universität Maastricht, welches vom Europäischen Forschungsrat finanziert und von Cameron Browne geleitet wird, bereitgestellt². Obwohl Ludii im Bereich des GGP einzuordnen ist, macht die Plattform keinen Gebrauch von GDL [2, p. 2], worauf zugleich dessen Alleinstellungsmerkmal basiert. Denn mit Ludii wird ebenso eine eigene Sprache bereitgestellt, mithilfe dieser sich Spiele durch sogenannte Ludeme deklarativ und für Menschen verständlich beschreiben lassen. Ludeme sind als einzelne Spielkonzepte zu verstehen, die sich wie Bausteine zu jedem traditionellen Strategiespiel zusammensetzen lassen sollen. Eigenen Vergleichen der Forschungsgruppe zufolge benötigt die Beschreibung von Spielen durch Ludeme deutlich weniger Token als mit GDL. Beispielsweise sind für die Beschreibung des Brettspiels Othello 894 GDL-Token notwendig, in Ludii dagegen nur 78 [2, p. 4]. Diese Ludeme-basierte Spielebeschreibungssprache öffnete dem Digital Ludeme Projekt die Tore für ein neues Forschungsfeld mit dem Namen *Digital Archaeoludology*, welches sich mit antiken Brettspielen der Geschichte beschäftigt. Da durch Ludeme konkrete Spielkonzepte beschrieben werden, lassen sich durch diese nicht nur Zusammenhänge zwischen verschiedenen Spielen besser erkennen oder Erkenntnisse zur historischen Entwicklung dieser gewinnen, sondern es können auch Informationslücken in historischen und unvollständigen Spielbeschreibungen geschlossen werden [3, p. 1].

Die Ludii Plattform bietet die Möglichkeit, eigens definierte Spiele zur Laufzeit einzubinden, wobei deren Beschreibung auch aus der Anwendung heraus manipuliert werden kann. Unterstützt werden Spiele, bei denen Züge abwechselnd, simultan oder in Echtzeit stattfinden, wobei Letzteres in eingeschränktem Umfang zutrifft. An diesen können neben menschlichen Spielern auch diverse Agenten partizipieren. Es gilt zu beachten, dass nicht jeder von Ludii bereitgestellte KI-Agent auch für alle Spiele zur Verfügung steht. Allerdings wird eine abstrakte Java-Klasse bereitgestellt, mithilfe dieser eigene Agenten implementiert und als Java Archive zur Laufzeit importiert werden können.

1.3 Projektablauf

Um eine Antwort auf die zentrale Forschungsfrage dieser Arbeit (Ist ein Agent, welcher sich durch einen Reinforcement Learning Ansatz eigenständig Spielstrategien aneignet, einem trainingsunfähigen Allzweck-Agenten überlegen?) zu finden, wird eine Möglichkeit benötigt, RL-Agenten aus dem GBG-Framework gegen Allzweck-Agenten aus dem Ludii-Framework antreten zu lassen. Da für diesen Anwendungsfall noch keine technische Lösung vorliegt, wird der erste Schritt des Projekts darin bestehen, einen entsprechenden Software-Rahmen zu implementieren, mit dem im nächsten Schritt automatisierte Partien zwischen GBG- und Ludii-Agenten ausgetragen werden können. Die dabei erhobenen Daten werden anschließend evaluiert, um Informationen zur Beantwortung der Forschungsfrage zu gewinnen.

²Abrufbar unter <https://ludii.games>

2 Implementierung der GBG-Ludii-Schnittstelle

2.1 Planung

Das Ziel der GBG-Ludii-Schnittstelle ist es, einen GBG- gegen einen Ludii-Agenten im Brettspiel Othello antreten zu lassen. Da die Ludii-Umgebung die Möglichkeit bietet, einen eigenen Agenten zu importieren und spielen zu lassen, sieht das Implementierungskonzept vor, einen Pseudo-Ludii-Agenten zu entwickeln, welcher einen GBG-Agenten ummantelt und zugleich die entsprechende Schnittstelle implementiert, um in der Ludii-Umgebung importiert werden zu können. Dieser erhält jede Runde eine Anfrage des Ludii-Frameworks nach seinem nächsten Spielzug, welche er an den gespeicherten GBG-Agenten weiterleitet. Dieser gibt den zu spielenden Zug über den Pseudo-Ludii-Agenten an die Ludii-Laufzeitumgebung zurück. Zu Beginn reicht es aus, wenn die Partien durch einen manuellen Import des Agenten in der Ludii-Oberfläche gestartet werden können. Jedoch wird eine Möglichkeit zur vollständig programmgesteuerten Simulation solcher Partien spätestens bei der Erhebung und Evaluation repräsentativer Spieldaten benötigt. Abbildung 1 spezifiziert grob die für die Implementierung geplante Architektur, welche im Folgenden erläutert wird.

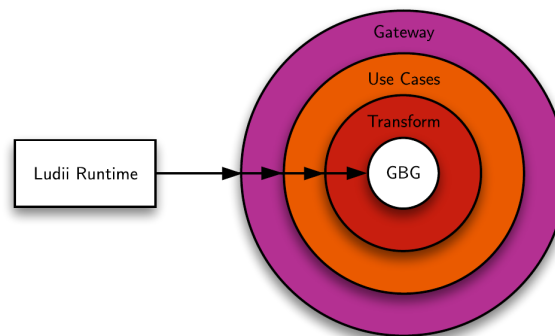


Abbildung 1: Architekturspezifikation der GBG-Ludii-Schnittstelle

Um eine saubere Architektur zu gewährleisten, wird die Anwendung als sogenannte Onion Architecture³ in die vier Schichten Gateway, Use Cases, Transform und GBG aufgeteilt. Letzteres bildet dabei den Anwendungskern, wodurch nicht nur sichergestellt ist, dass die Ludii-Umgebung keinen direkten Zugriff auf GBG-Funktionalitäten hat, sondern auch, dass es eine strikte und domänenbasierte Trennung zwischen GBG- und Ludii-Komponenten gibt. Somit muss kein Quellcode des GBG-Kerns um Ludii-spezifische Logik erweitert werden. Die Pfeile in der Abbildung repräsentieren die Abhängigkeiten zwischen den Komponenten. Damit ist die Ludii-Laufzeitumgebung während einer Othello-Partie lediglich von der Gateway-Schicht abhängig, welche nach außen hin die von Ludii gewünschte Schnittstelle bereitstellt. Die Use-Cases-Schicht stellt dem Gateway diverse Funktionalitäten zur Verfügung. Zu diesen gehört beispielsweise die Nutzung des Ludii-Zustands in GBG-Kontexten, sodass dieser korrekt vom GBG-Kern verarbeitet werden kann. Jedoch geschieht die Umwandlung solcher Ludii-spezifischer in GBG-spezifische Zustände letztendlich in der Transform-Schicht.

³Ein von Jeffrey Palermo beschriebenes Software-Architekturmuster

2.2 Transform-Schicht

Die Implementierungsreihenfolge einzelner Anwendungsschichten lässt sich aus Abbildung 1 herleiten. Da Abhängigkeiten lediglich von außen nach innen bestehen, findet der Entwicklungsprozess umgekehrt von innen nach außen statt, wodurch Abhängigkeiten zu jedem Zeitpunkt aufgelöst werden können. Der GBG-Kern wird durch das GBG-Framework bereitgestellt, weshalb die Entwicklung mit den folgenden Komponenten der Transform-Schicht beginnt.

2.2.1 Spielbrett

Dieser Abschnitt befasst sich mit der Spielbrettindizierung, denn sowohl das GBG-, als auch das Ludii-Framework indizieren die Felder des Othello-Spielbretts mit der Zahlenfolge (0, 1, ..., 63). Es besteht dennoch ein signifikanter Unterschied, denn in GBG beginnt die Indizierung oben links, wohingegen sie in Ludii unten links beginnt. Abbildung 2 visualisiert diesen Unterschied und zeigt zudem ein daraus resultierendes Problem auf. Da die Indexwerte bei gleicher Ausrichtung des Spielbretts vertikal gespiegelt sind, stimmen die Indizes der Spielfiguren nicht überein. So lässt sich beispielsweise erkennen, dass sich die weißen Startfiguren beim GBG-Framework auf den Feldern 27 und 36 befinden, beim Ludii-Framework hingegen auf 35 und 28. Um im letzten Schritt belastbare Spieldaten erheben zu können, muss bei der Umwandlung eines Ludii-Zustandes zu einem GBG-Zustand die fehlerfreie Transformation der Spielbrettindizes sichergestellt sein.

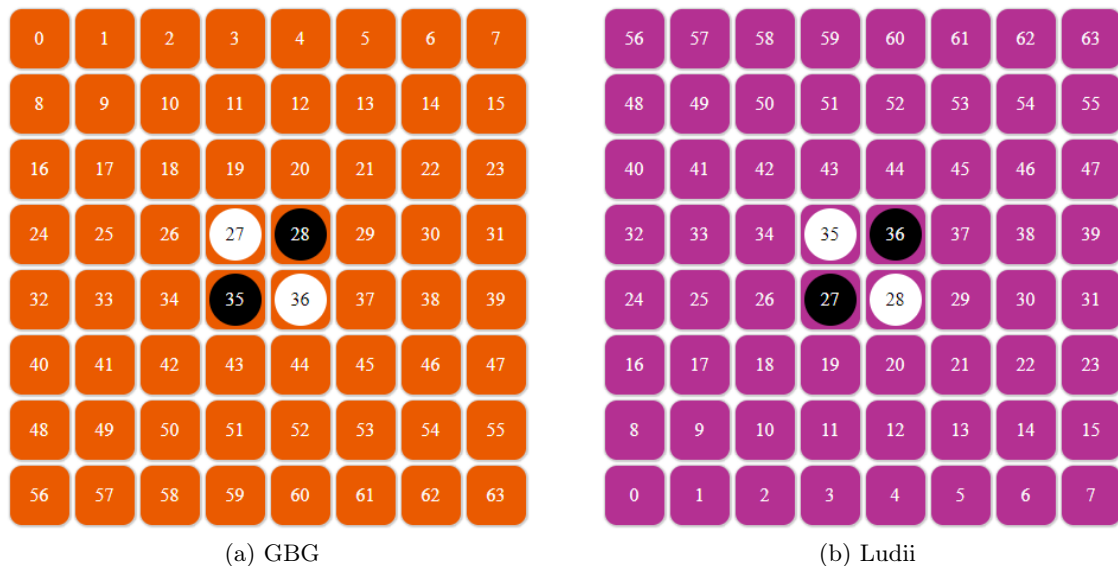


Abbildung 2: Indizierung des Othello-Spielbretts

Um dieses Problem zu lösen und so eine angemessene Transformation der Indexwerte zu gewährleisten, werden die Java-Klassen `BoardIndex`, `GbgBoardIndex` und `LudiiBoardIndex` implementiert (vgl. Abbildung 3). `BoardIndex` ist dabei eine abstrakte Superklasse, die ihren Subklassen `GbgBoardIndex` und `LudiiBoardIndex` vorschreibt, dass sie eine `toInt`-Methode bereitstellen müssen, welche einen entsprechenden Indexwert zurückgeben soll. Außerdem implementiert `BoardIndex` eine `isValid`-Methode, welche prüft, ob es sich bei einer Instanz um einen validen Index handelt. In der weiteren Entwicklung sollen alle GBG-

spezifischen Codestellen mit dem GbgBoardIndex und alle Ludii-spezifischen Codestellen mit dem LudiiBoardIndex arbeiten. Da Ludii nach dem Event-Sourcing-Verfahren arbeitet und somit den Spielzustand in Form einer Sequenz vergangener Spielzüge bereitstellt, können aus diesen, im weiteren Verlauf der Entwicklung, jeweils LudiiBoardIndex-Objekte instanziiert werden, welche sich wiederum mithilfe der toGbgIndex-Methode, unter Sicherstellung einer korrekten Transformation des Indexwertes, zu einem GbgBoardIndex umwandeln lassen. GBG-spezifische Logik wird den Typ LudiiBoardIndex nicht akzeptieren und ausschließlich mit GbgBoardIndex arbeiten. Somit wird bereits auf Compilerbene ausgeschlossen, dass ein Indexwert aus dem Ludii-System ohne vorherige Normalisierung ins GBG-System gelangen und damit Berechnungen verfälschen kann. Außerdem müssen weitere Komponenten dadurch nicht wissen, auf welcher algorithmischen Basis die Umwandlung zwischen den beiden verschiedenen Indextypen stattfindet. Diese müssen nur definieren, ob sie einen Ludii- oder GBG-Index erwarten.

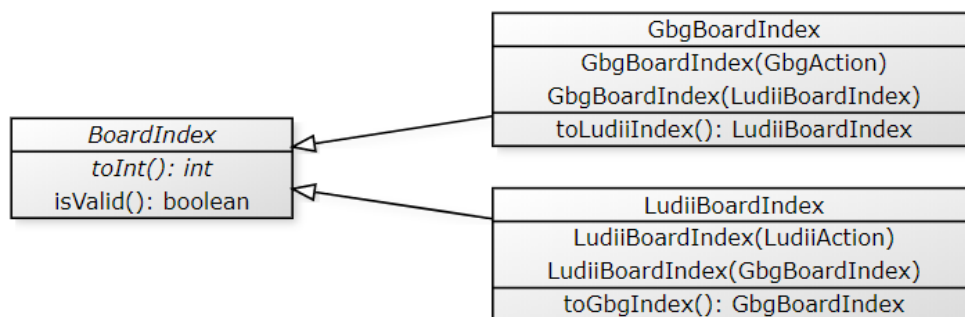


Abbildung 3: Klassen zur Transformation der Spielbrettindizes als UML-Diagramm

Um das korrekte Verhalten der drei implementierten BoardIndex-Klassen zu jedem Zeitpunkt überprüfen zu können, wurden zusätzliche Komponententests auf Basis der folgenden Klauseln entwickelt:

1. Ein BoardIndex ist nur valide, wenn sich der Rückgabewert seiner toInt-Methode in der Zahlenfolge (0, 1, ..., 63) befindet.
2. Wird ein GbgBoardIndex durch eine Gbg-Aktion erzeugt, muss der Rückgabewert seiner toInt-Methode gleich dem Integer-Wert der entsprechenden Gbg-Aktion sein.
3. Wird ein GbgBoardIndex durch einen LudiiBoardIndex erzeugt oder in einen solchen transformiert, so muss in jedem Fall der Index korrekt umgerechnet werden.
4. Wird ein LudiiBoardIndex durch eine Ludii-Aktion erzeugt, muss der Rückgabewert seiner toInt-Methode gleich dem Integer-Wert der entsprechenden Ludii-Aktion sein.
5. Wird ein LudiiBoardIndex durch einen GbgBoardIndex erzeugt oder in einen solchen transformiert, so muss in jedem Fall der Index korrekt umgerechnet werden.

2.2.2 Spieler

Wie sich in Tabelle 1 erkennen lässt, unterscheidet sich die Indizierung der Spielerfarben Schwarz und Weiß in den beiden Frameworks. Auch in diesem Fall gilt, dass nur dann aussagekräftige Spieldaten erfasst werden können, wenn die einwandfreie Transformation der Spielerindizes sichergestellt ist.

Spielerfarbe	GBG	Ludii
Schwarz	0	1
Weiß	1	2

Tabelle 1: Indizierung der Spielerfarben

Die Transformation wird auch hier bereits auf Compilerbene durch die beiden Klassen `LudiiPlayer` und `GbgPlayer` (vgl. Abbildung 4) gewährleistet, indem Ludii-spezifische Komponenten mit dem Typ `LudiiPlayer` und GBG-spezifische Komponenten mit `GbgPlayer` arbeiten, wodurch der Transformationsalgorithmus, wie schon zuvor bei der Spielbrettindizierung, in diesen Klassen der Transform-Schicht gekapselt ist. Die wichtigsten Funktionalitäten der beiden Klassen sind deren `toInt`-Methoden, welche jeweils einen Spielerindex basierend auf Tabelle 1 zurückgeben. Ansonsten spezifiziert das UML-Diagramm neben der Möglichkeit einer Validitätsprüfung von `LudiiPlayer`-Instanzen diverse Lösungen zur Transformation solcher `LudiiPlayer`- in `GbgPlayer`-Instanzen. Eine Umwandlung in umgekehrter Richtung ist nicht notwendig und damit auch nicht spezifiziert, da eine Anfrage der Ludii-Laufzeitumgebung ohnehin die entsprechenden Parameter liefert, aus denen direkt ein `LudiiPlayer`-Objekt instanziiert werden kann.

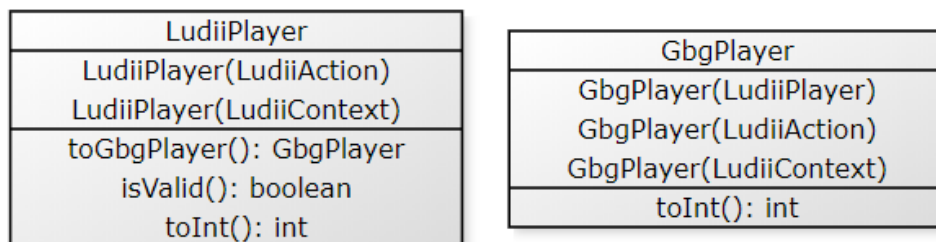


Abbildung 4: Klassen zur Transformation der Spielerindizes als UML-Diagramm

Das gewünschte Verhalten der entwickelten Komponenten wird durch folgende Komponententests gewahrt:

1. Wird ein `GbgPlayer`-Objekt durch eine schwarze `LudiiPlayer`-Instanz konstruiert, gibt seine `toInt`-Methode den Zahlenwert 0 zurück.
2. Wird ein `GbgPlayer`-Objekt durch eine weiße `LudiiPlayer`-Instanz konstruiert, gibt seine `toInt`-Methode den Zahlenwert 1 zurück.
3. Wird ein `LudiiPlayer`-Objekt durch eine Ludii-Aktion des Spielers mit der Farbe Schwarz instanziiert, gibt seine `toInt`-Methode den Zahlenwert 1 zurück.

4. Wird ein LudiiPlayer-Objekt durch eine Ludii-Aktion des Spielers mit der Farbe Weiß instanziiert, gibt seine toInt-Methode den Zahlenwert 2 zurück.
5. Wird ein LudiiPlayer-Objekt durch einen Ludii-Kontext, in dem der Spieler mit der Farbe Schwarz an der Reihe ist, instanziiert, gibt seine toInt-Methode den Zahlenwert 1 zurück.
6. Wird ein LudiiPlayer-Objekt durch einen Ludii-Kontext, in dem der Spieler mit der Farbe Weiß an der Reihe ist, instanziiert, gibt seine toInt-Methode den Zahlenwert 2 zurück.

2.3 Use-Cases-Schicht

Nach der nun abgeschlossenen Entwicklung der Transform-Schicht wird als Nächstes die davon abhängige Use-Cases-Schicht implementiert. Durch die Transform-Schicht wird die Indizierung des Spielbretts und der Spielerfarben von dessen konkreter Umsetzung in beiden Frameworks abstrahiert, wodurch es für abhängige Schichten keine Rolle spielt, wie diese technisch realisiert ist. Es muss lediglich für beide Frameworks zu jedem Zeitpunkt einen gemeinsamen Konsens geben, wie das Spielbrett aussieht und wie die entsprechenden weißen und schwarzen Spielfiguren auf diesem positioniert sind. Da diese Basis nun gegeben ist, können in diesem Schritt weitere Komponenten entwickelt werden, die zwar auch relevant für die Kommunikation zwischen den Frameworks, aber losgelöst von der dafür notwendigen Transformation primitiver Datentypen sind.

2.3.1 Spielzüge

Wie bereits erwähnt wurde, arbeitet das Ludii-Framework nach dem Event-Sourcing-Verfahren, wodurch der Spielzustand lediglich in Form einer Sequenz vergangener Spielzüge bereitgestellt wird. Damit eine Anfrage der Ludii-Umgebung nach dem zu spielenden Zug überhaupt an einen GBG-Agenten weitergereicht werden kann, muss dieser auf die Informationen zum Spielzustand zurückgreifen können. Diese existieren zwar in Form der von Ludii bereitgestellten Sequenz, basieren aber auf den Spielbrett- und Spielerindizes von Ludii. Deshalb soll aus dieser Ludii-spezifischen Sequenz, mithilfe entwickelter Komponenten der Transform-Schicht, eine GBG-konforme Sequenz vergangener Spielzüge erstellt werden, damit aus dieser in einem weiteren Schritt ein GBG-Spielzustand erzeugt werden kann. Die hierfür angestrebte Problemlösung wird in Abbildung 5 spezifiziert.



Abbildung 5: Framework-spezifische Spielzüge-Klassen als UML-Diagramm

Ein Objekt der Klasse LudiiMoves lässt sich durch einen Ludii-Kontext erzeugen und hat damit Zugriff auf die vergangenen Spielzüge einer Partie, welche es über die played-Methode als Zuordnungstabelle bereitstellt. Diese Tabelle weist, in chronologischer Reihenfolge, je

einem Spielbrettindex, auf dem ein Spielzug stattgefunden hat, einen entsprechenden Spieler zu, welcher anschließend auf diesem Feld steht. Analoges gilt für die GbgMoves-Klasse, wobei diese darauf ausgelegt ist, die Spielzughistorie der LudiiMoves-Klasse in eine GBG-konforme Repräsentation umzuwandeln. Dies wird intern durch die Verwendung von Klassen der Transform-Schicht realisiert, welche die notwendigen Methoden bereitstellen, um den Typ LudiiBoardIndex in GbgBoardIndex oder analog LudiiPlayer in GbgPlayer zu transformieren. In Abbildung 5 lässt sich auch erkennen, dass die LudiiMoves-Klasse eine weitere Methode mit dem Namen `availableMoveBy` spezifiziert. Diese ist im Hinblick darauf entstanden, dass ein Spielzug der von einem GBG-Agenten berechnet wird, letztendlich zurück in eine Ludii-konforme Datenstruktur umgewandelt werden muss, damit die Ludii-Laufzeitumgebung diesen verwenden kann. Somit akzeptiert die `availableMoveBy`-Methode einen GBG-Spielzug und gibt den dazu äquivalenten Ludii-Spielzug zurück, sofern er in dieser Runde spielbar ist. Auch für diesen Zweck sind intern die Klassen der Transform-Schicht notwendig.

Damit in zukünftigen Iterationen des Softwareentwicklungsprozesses weiterhin sichergestellt bleibt, dass die in diesem Kapitel entwickelten Features jederzeit wie gewünscht funktionieren, werden für folgende Klauseln Komponententests implementiert:

1. Eine LudiiMoves-Instanz, die aus dem LudiiContext eines neuen Spiels erzeugt wird, besitzt initial eine Historie von folgenden vier Spielzügen:
 - (a) Der Ludii-Spieler mit dem Index 1 steht auf dem Spielfeld mit dem Ludii-basierten Index 36.
 - (b) Der Ludii-Spieler mit dem Index 1 steht auf dem Spielfeld mit dem Ludii-basierten Index 27.
 - (c) Der Ludii-Spieler mit dem Index 2 steht auf dem Spielfeld mit dem Ludii-basierten Index 35.
 - (d) Der Ludii-Spieler mit dem Index 2 steht auf dem Spielfeld mit dem Ludii-basierten Index 28.
2. Eine GbgMoves-Instanz, die aus dem LudiiContext eines neuen Spiels erzeugt wird, besitzt initial eine Historie von folgenden vier Spielzügen:
 - (a) Der GBG-Spieler mit dem Index 0 steht auf dem Spielfeld mit dem GBG-basierten Index 28.
 - (b) Der GBG-Spieler mit dem Index 0 steht auf dem Spielfeld mit dem GBG-basierten Index 35.
 - (c) Der GBG-Spieler mit dem Index 1 steht auf dem Spielfeld mit dem GBG-basierten Index 27.
 - (d) Der GBG-Spieler mit dem Index 1 steht auf dem Spielfeld mit dem GBG-basierten Index 36.
3. Die Historie der vergangenen Spielzüge ist in der LudiiMoves-, als auch in der GbgMoves-Klasse chronologisch sortiert.

- Die Methode `availableMoveBy` der `LudiiMoves`-Klasse liefert nur dann einen verfügbaren Ludii-Spielzug, wenn ein solcher, der äquivalent zum übergebenen GBG-Spielzug ist, existiert und in der aktuellen Runde regelkonform ausgespielt werden darf.

2.3.2 Spielzustand

Der aktuelle Entwicklungsfortschritt kann so zusammengefasst werden, dass es nun mithilfe der Transform-Schicht-Klassen möglich ist, bei einer eingehenden Anfrage der Ludii-Umgebung eine Historie vergangener Spielzüge der aktuellen Partie in eine GBG-konforme Spielzughistorie zu transformieren. Im nächsten Schritt soll aus dieser der Spielzustand durch einen sogenannten `Replay`⁴ abgeleitet werden, sodass daraus ein `StateObservation`-Typ implementiert werden kann, der von GBG-Agenten als Eingabe für die Berechnung eines Spielzugs erwartet wird. Zur Lösung dieser Anforderung gibt Abbildung 6 einen groben Überblick, über die benötigten Datenstrukturen und deren Beziehungen untereinander.

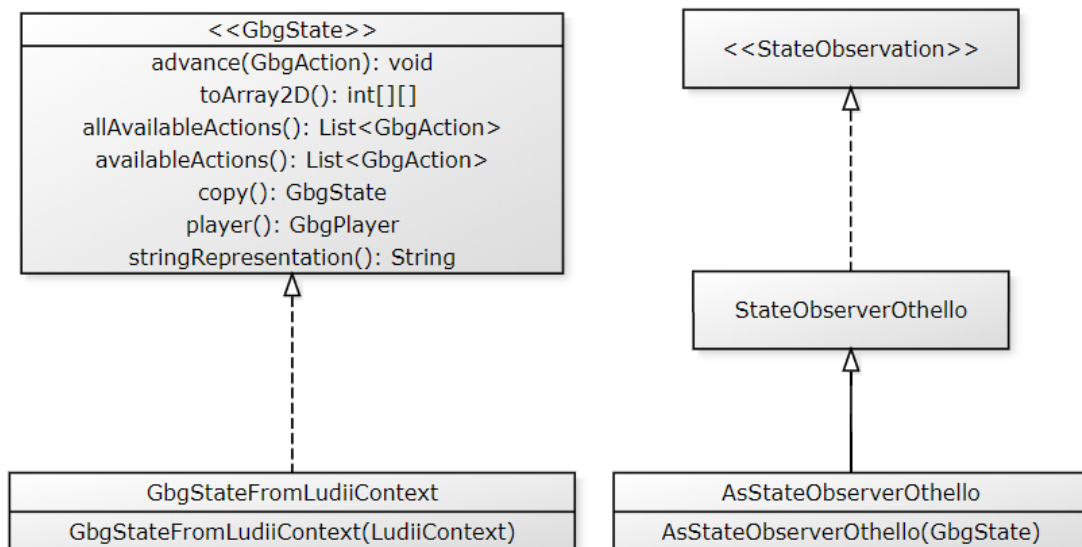


Abbildung 6: Nutzung eines Ludii-Kontextes in einer GBG-StateObservation als UML-Diagramm

Das Interface `GbgState` abstrahiert diverse Spielzustand-spezifische Methoden, welche allgemeingültig und somit nicht auf das Spiel Othello beschränkt sind. Eine Implementation dieser Schnittstelle ist durch die Klasse `GbgStateFromLudiiContext` gegeben, welche zusätzlich einen Konstruktor, der einen Ludii-Spielkontext akzeptiert, bereitstellt. Die Implementation der vom Interface definierten Methoden erfolgt in Abhängigkeit von diesem Ludii-Kontext und den in den vorherigen Schritten entwickelten Klassen. Da GBG-Agenten zur Berechnung des nächsten Spielzugs eine `StateObservation`-Instanz benötigen, erbt die Klasse `AsStateObserverOthello` von `StateObserverOthello`, welche wiederum den Typ `StateObservation` implementiert, und bietet die Funktionalität ein solches `StateObserverOthello`-Objekt durch eine `GbgState`-Instanz zu erzeugen. Da Letztere durch einen Ludii-Kontext erzeugt werden kann, schließt sich somit der Kreis, denn es ist letztend-

⁴Bezeichnet im Event-Sourcing das, zu einem Zustand führende, "Abspielen" gespeicherter Events

lich möglich aus einem Ludii-Kontext, unter Berücksichtigung der korrekten Transformationen primitiver Datentypen in der Transform-Schicht, eine für GBG-Agenten kompatible StateObservation-Instanz zu konstruieren.

Dieses implementierte Verhalten wird durch folgende Komponententest sichergestellt:

1. Die Indizes der Spielfelder aller möglichen Züge eines Spiels beschränken sich auf die Zahlenfolge (0, 1, ..., 63) ohne die Zahlen 27, 28, 35 und 36, da es sich bei Letzteren um Startpositionen handelt.
2. Eine aus einem Ludii-Kontext erzeugte StateObservation-Instanz berechnet in einer Spielrunde die erlaubten Züge, den aktuellen Spieler und den Zustand des Spielbretts mit demselben Ergebnis wie die bereits vorhandene StateObserverOthello-Klasse,
 - (a) wenn Erstere
 - i. aus dem Ludii-Kontext eines neuen Spiels instanziiert wurde.
 - ii. aus dem Ludii-Kontext eines Spiels mit einem bereits getätigten Zug instanziiert wurde.
 - iii. aus dem Ludii-Kontext eines Spiels mit mehreren bereits getätigten Zügen instanziiert wurde.
 - iv. aus dem Ludii-Kontext eines bereits entschiedenen Spiels instanziiert wurde.
 - (b) wenn nach der Instanziierung
 - i. ein zusätzlicher Spielzug auf der StateObservation-Instanz ausgeführt wurde.
 - ii. mehrere zusätzliche Spielzüge auf der StateObservation-Instanz ausgeführt wurden.
 - iii. soviel zusätzliche Spielzüge auf der StateObservation-Instanz ausgeführt wurden, dass das Spiel dadurch entschieden wurde.
3. Ein Spiel ist
 - (a) nicht entschieden,
 - i. wenn noch kein Spielzug ausgeführt wurde.
 - ii. wenn erst ein einziger Spielzug ausgeführt wurde.
 - iii. solange der Spieler, der an der Reihe ist, noch mindestens einen möglichen Spielzug ausspielen darf.
 - (b) entschieden,
 - i. wenn es aus dem Ludii-Kontext eines bereits entschiedenen Spiels instanziiert wurde.
 - ii. wenn alle Felder des Spielbretts belegt sind.
 - iii. wenn ein Spieler keine Figuren mehr auf dem Spielbrett besitzt.

2.4 Gateway-Schicht

Zum aktuellen Zeitpunkt ist neben der Entwicklung der Transform- auch die der Use-Cases-Schicht abgeschlossen. Um einen GBG- nun gegen einen Ludii-Agenten antreten lassen zu können, wird ein Pseudo-Agent benötigt, welcher die Klasse AI des Ludii-Frameworks implementiert und entsprechende Anfragen an einen GBG-Agenten weiterleitet. Diese abstrakte AI-Klasse stellt dabei diverse Methoden zur Verfügung (vgl. Abbildung 7).

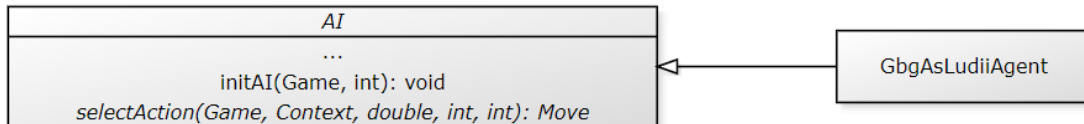


Abbildung 7: Relevanter Teil der Ludii-Schnittstelle für eigene Agenten als UML-Diagramm

Für die vorliegende Arbeit ist neben der Methode `selectAction`, welche als abstrakt gekennzeichnet ist und somit von der `GbgAsLudiiAgent`-Klasse implementiert werden muss, auch die Methode `initAI` von Bedeutung, die in `GbgAsLudiiAgent` überschrieben wird. Während Erstere mehrfach aufgerufen wird, um den nächsten Spielzug anzufragen, wird Letztere zur Initialisierung des Agenten benötigt und dazu lediglich einmal aufgerufen.

```

@Override
public void initAI(final Game game, final int playerId) {
    try {
        gbgAgent = new ArenaTrainOthello(
            "GBG vs. Ludii – Othello Arena",
            false
        )
        .tdAgentIO
        .loadGBGAgent(gbgAgentPath);
    } catch (final Exception e) {
        e.printStackTrace();
    }
}
  
```

Abbildung 8: Implementierung der `initAI`-Methode als Java-Code

Die Implementierung von `initAI` sieht vor, dass ein exportierter GBG-Agent, zum Zeitpunkt des Spielstarts, durch die Angabe eines absoluten Pfads einmalig aus dem Dateisystem geladen und in einem Feld vom Typ `PlayAgent` gespeichert wird, sodass dieser anschließend für Spielzugberechnungen zur Verfügung steht (vgl. Abbildung 8).

Die `selectAction`-Methode (vgl. Abbildung 9) verwendet intern die private `gbgAction`-Methode (vgl. Abbildung 10). Letztere erzeugt aus einem Ludii-Kontext eine `StateObservation`-Instanz und ruft mithilfe dieser die `getNextAction2`-Methode des GBG-Agenten auf, um den von diesem Agenten berechneten Spielzug abzufragen, welcher auch von der `gbgAction`-Methode zurückgegeben wird. Die `selectAction`-Methode nutzt diesen GBG-spezifischen Spielzug um dessen äquivalenten Ludii-spezifischen Spielzug zu ermitteln und an die anfragende Ludii-Laufzeitumgebung zurückzugeben.

```

@Override
public Move selectAction(
    final Game game,
    final Context context,
    final double maxSeconds,
    final int maxIterations,
    final int maxDepth
) {
    return
        new LudiiMoves(context)
            .availableMoveBy(
                gbgAction(context)
            )
            .get();
}

```

Abbildung 9: Implementierung der selectAction-Methode als Java-Code

```

private Types.ACTIONS gbgAction(final Context ludiiContext) {
    return
        gbgAgent
            .getNextAction2(
                new AsStateObserverOthello(
                    new GbgStateFromLudiiContext(ludiiContext)
                ),
                false,
                false
            );
}

```

Abbildung 10: Implementierung der gbgAction-Methode als Java-Code

Die Klasse GbgAsLudiiAgent ist die einzige Komponente der Gateway-Schicht und benötigt keine Komponententests, da sie lediglich auf bereits getestete Logik zurückgreift und das Projekt selbst nicht um relevante Logik erweitert.

2.5 SDK- und Frameworkabhängigkeiten

Im gesamten Verlauf des Entwicklungsprozesses sind folgende SDK- bzw. Frameworkabhängigkeiten hinzugekommen:

1. Die Entwicklung verlief gegen die Ludii-Version 0.9.3. Diese ist als Java Archive "Ludii-0.9.3.jar" im Unterordner lib hinzugefügt worden.
2. Für die Komponententests wurde das Framework JUnit Runners in der Version 1.3 verwendet, um Testklassen schachteln zu können und damit eine bessere Übersicht der Testfälle zu erhalten. Dieses ist als Java Archive "junit-runners-1.3.jar" im Unterordner lib hinzugefügt worden.

3 Datenerhebung und -evaluation

Da die GBG-Ludii-Schnittstelle nun vollständig implementiert und eine hinreichende Testabdeckung vorhanden ist, kann in diesem Kapitel untersucht werden, mit welchen Ergebnissen Partien zwischen GBG- und Ludii-Agenten ausgetragen werden. So sollen Spieldaten durch die programmgesteuerte Simulation von 100 Othello-Partien zwischen einem Ludii- und einem GBG-Agenten erhoben und im Anschluss evaluiert werden, um Informationen zur Beantwortung der Forschungsfrage abzuleiten. Das GBG-Framework wird dabei von dem bereitgestellten "TCL3-fixed6_250k-lam05_P4_H001-diff2-FAM.agt.zip"-Agenten (Stand: 23.07.2020) vertreten, während Ludii durch dessen "LudiiAI"-Agenten in der Version 0.9.3 repräsentiert wird. Jeder Agent tritt jeweils 50 Mal mit der weißen, als auch der schwarzen Farbe an. Da Agenten des GBG-Frameworks mithilfe der entwickelten Schnittstelle in der Ludii-Umgebung verwendet werden können, nicht aber umgekehrt, finden alle Spielsimulationen in der Ludii-Laufzeitumgebung statt.

Abbildung 11 visualisiert die erhobenen Spieldaten, welche im Folgenden erläutert werden.

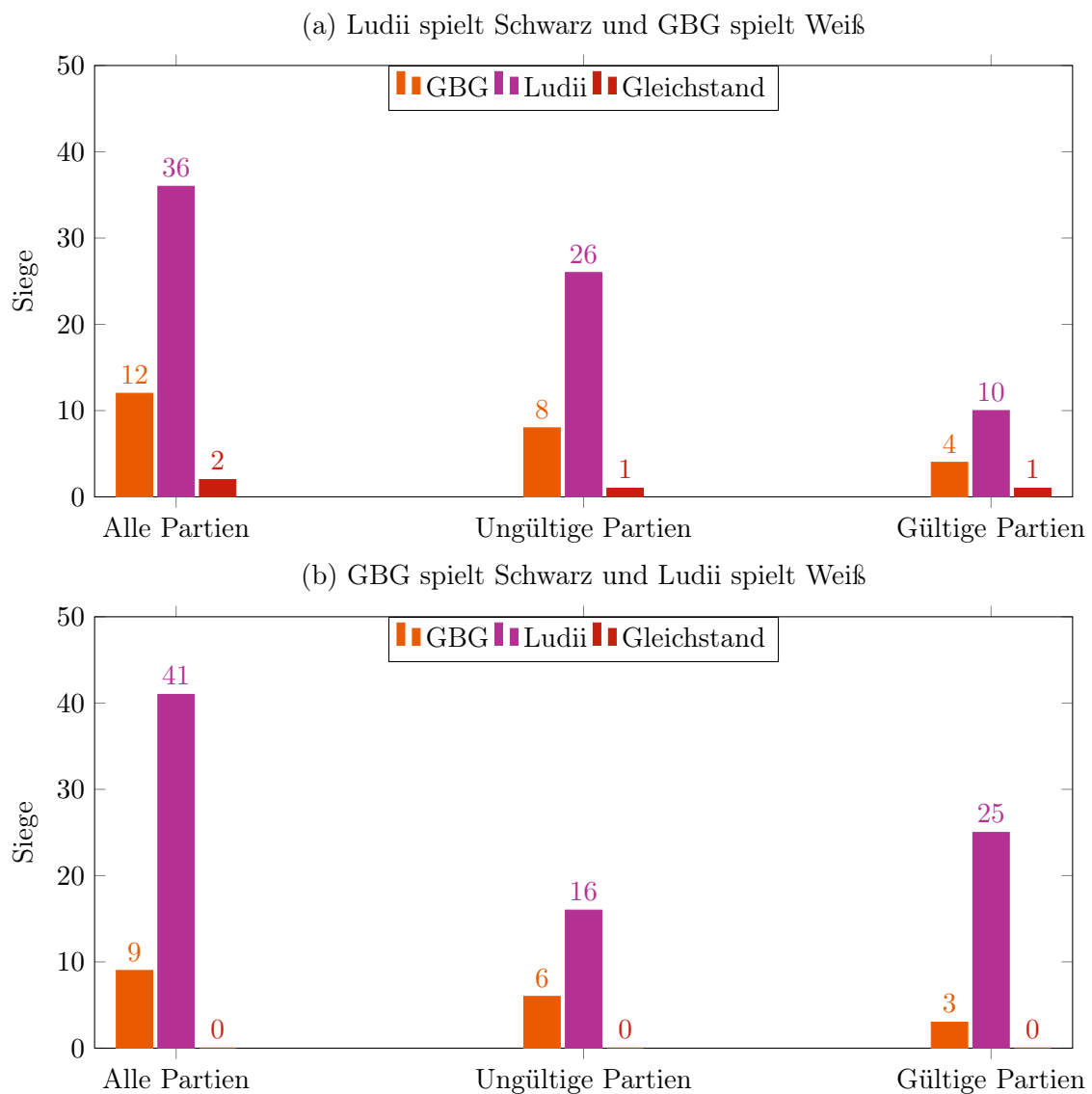


Abbildung 11: Erhobene Spieldaten zwischen dem TD-3-Tupel- (GBG) und dem LudiiAI-Agenten (Version 0.9.3)

Bei den Spielsimulationen ist aufgefallen, dass einige Spiele frühzeitig terminiert wurden. Der Grund dafür findet sich darin, dass die Simulationen in der Ludii-Umgebung stattfanden und Ludii ein Spiel bereits dann für beendet erklärt, wenn ein Spieler in einer Runde keinen legitimen Zug spielen kann. Im GBG-Framework ist in diesem Fall zunächst der andere Spieler an der Reihe und die Partie wird erst dann für beendet erklärt, wenn auch dieser keinen legalen Spielzug durchführen kann und passen muss.

Solche Spiele, die nur von einem der beiden Framework für beendet erklärt werden und damit nicht repräsentativ sind, werden in dem Diagramm in Abbildung 11 als "Ungültige Partien" kategorisiert. In den "Gültigen Partien" ist der Ludii-Agent seinem Gegner aus dem GBG-Framework erkennbar überlegen, was aber nicht repräsentativ sein muss.

Denn weitere experimentelle Simulationen, die lediglich im Rahmen des GBG-Frameworks stattfanden, zeigten, dass der TD-3-Tupel-Agent sich in Partien häufiger "klein" macht und dadurch versucht Spielsituationen herbeizuführen, in denen der Gegner teilweise mehrere Runden hintereinander passen muss, sodass der TD-3-Tupel-Agent einige aufeinanderfolgende Züge ausführen und damit das Blatt schnell wenden kann. Dabei handelt es sich um eine Strategie, die nicht selten zum Sieg führte. Da Partien in solchen Situationen vorzeitig durch die Ludii-Umgebung beendet werden, wodurch der Spieler mit den, zu diesem Zeitpunkt, meisten Spielsteinen gewinnt, kann nicht pauschal geschlussfolgert werden, dass der Ludii-Agent dem GBG-Agenten per se überlegen ist. Denn es lässt sich nicht ausschließen, dass der GBG-Agent einen Großteil der, als ungültig kategorisierten, Spiele für sich entschieden hätte, wenn er diese zu Ende spielen könnte.

4 Hürden und Fallstricke

Im Entwicklungsprozess traten regelmäßig kleinere Fallstricke auf, die nicht sofort durch Kompilierungs- oder Laufzeitfehler deutlich wurden, sondern vielmehr auffielen, weil sie ein unerwartetes Verhalten nach sich zogen. Solche Fehler ließen sich in der Regel identifizieren, indem simulierte Partien zwischen Agenten der beiden Frameworks händisch in dessen grafischen Spieloberflächen nachgestellt wurden. Das sieht praktisch so aus, dass in beiden Frameworks ein Spiel zwischen einem menschlichen Spieler und dem entsprechenden Agenten des Frameworks gestartet wird, wobei der menschliche Spieler in beiden Frameworks eine unterschiedliche Farbe spielt. Wenn beispielsweise der GBG-Agent eine Partie mit dem ersten Spielzug eröffnet, spielt der menschliche Spieler diesen Zug stellvertretend in der Ludii-Umgebung gegen den Ludii-Agenten. Der Zug mit dem der Ludii-Agent in diesem Fall reagiert würde dann anschließend durch den menschlichen Spieler in der GBG-Umgebung durchgeführt werden. So ist es ohne eine entsprechende Software-Schnittstelle bereits möglich Agenten Framework-übergreifend gegen einander antreten zu lassen, wenn auch nicht programmgesteuert. Wenn bei dieser händischen Simulation andere Spielzüge gespielt werden als in einer Simulation, die mithilfe der entwickelten Schnittstelle entstanden ist, obwohl der Spielzustand in beiden Fällen gleich ist, deutet dies auf ein Fehlverhalten der Schnittstelle hin. Für diese Annahme muss jedoch ein Zufallsfaktor in den Berechnungen der Agenten ausgeschlossen werden können. Solche Bugs hatten die verschiedensten Hintergründe und ließen sich durch Debuggen identifizieren. Außerdem konnte im Anschluss stets sichergestellt werden, dass solche Bugs bei Wiederauftreten sofort durch entsprechende Komponententests aufgedeckt werden. So stellte sich beispielsweise heraus, dass die Spielfelder und Spielerfarben der beiden Frameworks unterschiedlich indiziert sind.

Eine weitere Hürde äußerte sich darin, dass die Ludii-Umgebung in der Version 0.9.4, welche zum jetzigen Zeitpunkt die Aktuellste ist, einen NullPointerException beim Simulieren von Othello-Partien schmeißt. Zudem funktioniert Othello in dieser Version auch nicht in der grafischen Spieloberfläche. Aus diesem Grund fand die Entwicklung gegen die Version 0.9.3 statt.

Der schwerwiegendste Fallstrick wurde zum Ende des Projekts deutlich. Denn unterschiedliche Othello-Implementierungen in den beiden Frameworks sorgen dafür, dass Othello-Partien in der Ludii-Umgebung bereits terminieren, wenn ein Spieler in einer Runde passen muss, während im GBG-Framework der betroffene Spieler lediglich übersprungen wird. Dadurch verloren einige erhobene Spieldaten und somit das Ergebnis der Evaluation an Repräsentativität.

5 Nachtrag

5.1 Software-Erweiterungen

Kurz nach der Verfassung dieser Ausarbeitung fanden noch interessante Entwicklungen im Projekt statt, die noch eine Erwähnung in dieser Arbeit finden sollen. Mittlerweile ist Ludii in der Version 1.0.2 verfügbar. In dieser Version sind die zuvor beschriebenen Bugs behoben, weshalb eine erneute Datenerhebung und -evaluation stattfinden soll. Zu diesem Zweck wurde die Software zunächst um folgende Funktionalitäten erweitert, welche die Bedienung etwas vereinfachen sollen:

1. Die verwendete Ludii-Version wurde von 0.9.3 auf 1.0.2 aktualisiert. In diesem Rahmen sind zwar einige Kompilierungsfehler entstanden, allerdings konnten diese zügig behoben werden.
2. Der Dateipfad eines GBG-Agenten muss nicht mehr zur Kompilierungszeit im Code hinterlegt sein, sondern kann bei Verwendung der GBG-Ludii-Schnittstelle flexibel zur Laufzeit über ein Dialog-Fenster ausgewählt werden.
3. Auftretende Fehler werden nun in einem entsprechenden Dialog-Fenster angezeigt. Zuvor gab es nur sehr eingeschränkte Möglichkeiten auftretende Fehler zu ermitteln.
4. Dem GitHub-Repository wurde ein Code-Skript mit dem Namen `RunCustomMatch` hinzugefügt, mit dem programmgesteuert Partien zwischen plattformübergreifenden Agenten ausgetragen werden können.

5.2 Erneute Datenerhebung und -evaluation

Da die für diese Arbeit schwerwiegenden Bugs im Ludii-Quellcode nun obsolet sind, ist es sinnvoll, eine erneute Auswertung der Agenten-Spielstärke durchzuführen. Zuvor war ein großer Teil der erhobenen Daten ungültig und damit nicht aussagekräftig. Es wäre auch interessant herauszufinden, ob sich die zuvor aufgestellte Theorie nachweisen lässt, dass der TD-3-Tupel-Agent aus strategischen Gründen Spielsituationen herbeigeführt hat, in denen der Gegenspieler passen muss. Sollte dem so sein, so konnte er seine gesamte Spielstärke zuvor nicht entfalten.

Eine erneute Datenerhebung unter Verwendung der Ludii-Version 1.0.2 führte zu folgenden Ergebnissen:

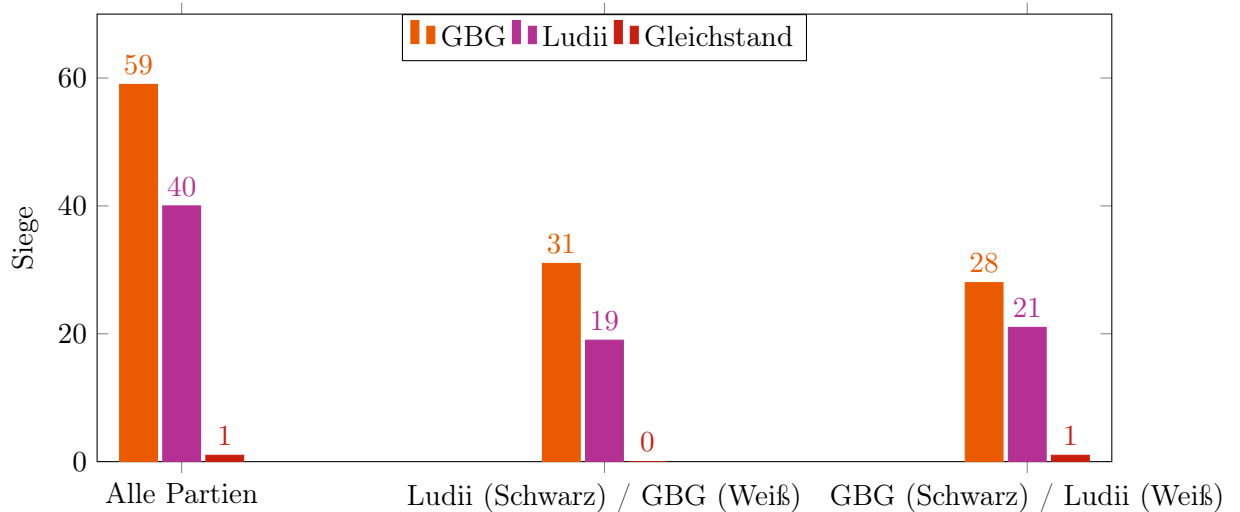


Abbildung 12: Erhobene Spieldaten zwischen dem TD-3-Tupel- (GBG) und dem LudiiAI-Agenten (Version 1.0.2)

Die erneute Erhebung der Daten (vgl. Abbildung 12) bestätigt tatsächlich die zuvor aufgestellte Theorie zur Spielstrategie des TD-3-Tupel-Agenten und zeigt zugleich, dass diese gut performt. Denn der GBG-Agent hat mit beiden Spielfarben durchschnittlich jeweils fast 50% mehr Partien gewonnen als sein Konkurrent aus dem Ludii-Framework.

5.3 Anleitung zur Verwendung der GBG-Ludii-Schnittstelle

5.3.1 Verwendung in der grafischen Ludii-Oberfläche

Erzeugen des Pseudo-Agenten als Java Archive

Bevor die GBG-Ludii-Schnittstelle verwendet werden kann, muss der Pseudo-Agent `GbgAsLudiiAgent` als Java Archive exportiert werden. Für die IDE IntelliJ IDEA wird dieser Vorgang nachfolgend erläutert. Bei geöffnetem Projekt kann in der Anwendung oben rechts ein Button angeklickt werden, welcher ein Fenster mit dem Namen "Project Structure" öffnet (vgl. Abbildung 13). Dort muss unter dem Reiter "Artifacts" lediglich ein neues Artefakt hinzugefügt werden. Durch die Checkbox "Include in project build" wird veranlasst, dass das Artefakt bei jedem Build-Prozess automatisch als Java Archive im angegebenen Verzeichnis erzeugt wird. Wichtig ist, dass die gesamte GBG-Software (inklusive aller Bibliotheken von denen diese abhängig ist) als Output angegeben wird, sodass es zur Laufzeit nicht zu Fehlern kommt. Alternativ kann auch das im GitHub-Repository eingetragene Java Archive `PseudoLudiiAgent.jar` verwendet werden.

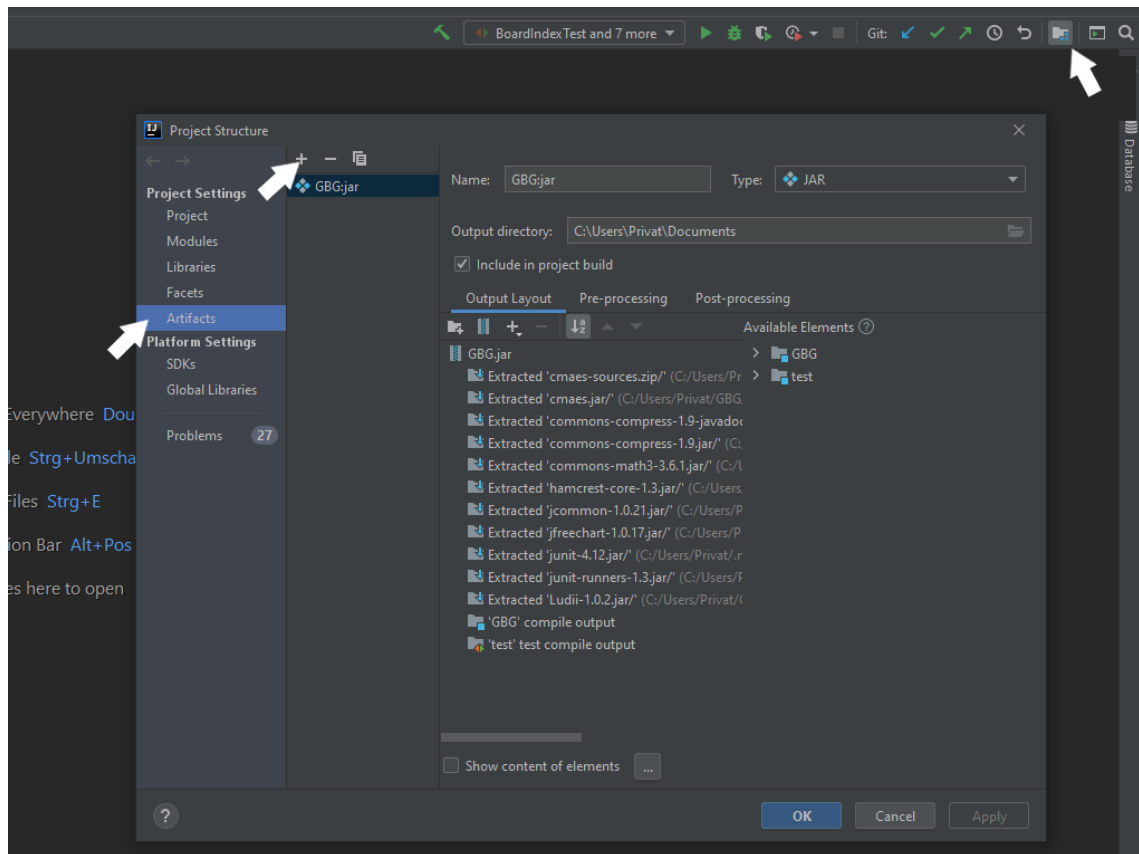


Abbildung 13: Screenshot der IntelliJ IDEA

Import in der grafischen Ludii-Oberfläche

Sofern ein der Pseudo-Agent als Java Archive vorhanden ist, kann dieser in der grafischen Ludii-Oberfläche importiert werden. Dazu sollte die kompatible Ludii-Version verwendet werden, die im GitHub-Repository im `lib`-Verzeichnis zu finden ist. In der Ludii-Oberfläche sollte dann "From JAR" für den entsprechenden Spieler ausgewählt werden. Daraufhin öffnet sich ein Dialog, in dem das Java Archive des Pseudo-Agenten ausgewählt werden kann. Im nächsten Schritt erscheint eine Auswahl, in welcher eine Klasse auszuwählen ist, welche von Ludii's AI-Klasse erbt. Hier ist `ludiiInterface.gateway.GbgAsLudiiAgent` zu selektieren. Im Anschluss öffnet sich erneut ein Datei-Dialog, welcher nach einem exportierten GBG-Agenten fragt (vgl. Abbildung 14). Hier kann ein beliebiger Agent des GBG-Frameworks importiert werden. Nach Bestätigung der Auswahl beginnt die Partie automatisch.

5.3.2 Programmgesteuerte Verwendung

Sofern programmgesteuert Partien zwischen Ludii- und GBG-Agenten stattfinden sollen, lässt sich dazu das im GitHub-Repository zu findende Java-Skript `src/ludiiInterface/evaluation/RunCustomMatch.java` verwenden. Nachdem dieses gestartet wurde, öffnet sich das Dialog-Fenster aus Abbildung 14, welches zur Auswahl eines exportierten GBG-Agenten auffordert. Nach getätigter Auswahl erscheint ein ähnlicher Dialog, in dem eine Datei angegeben werden soll, in welche die Ergebnisse der simulierten Partien geschrieben werden sollen.

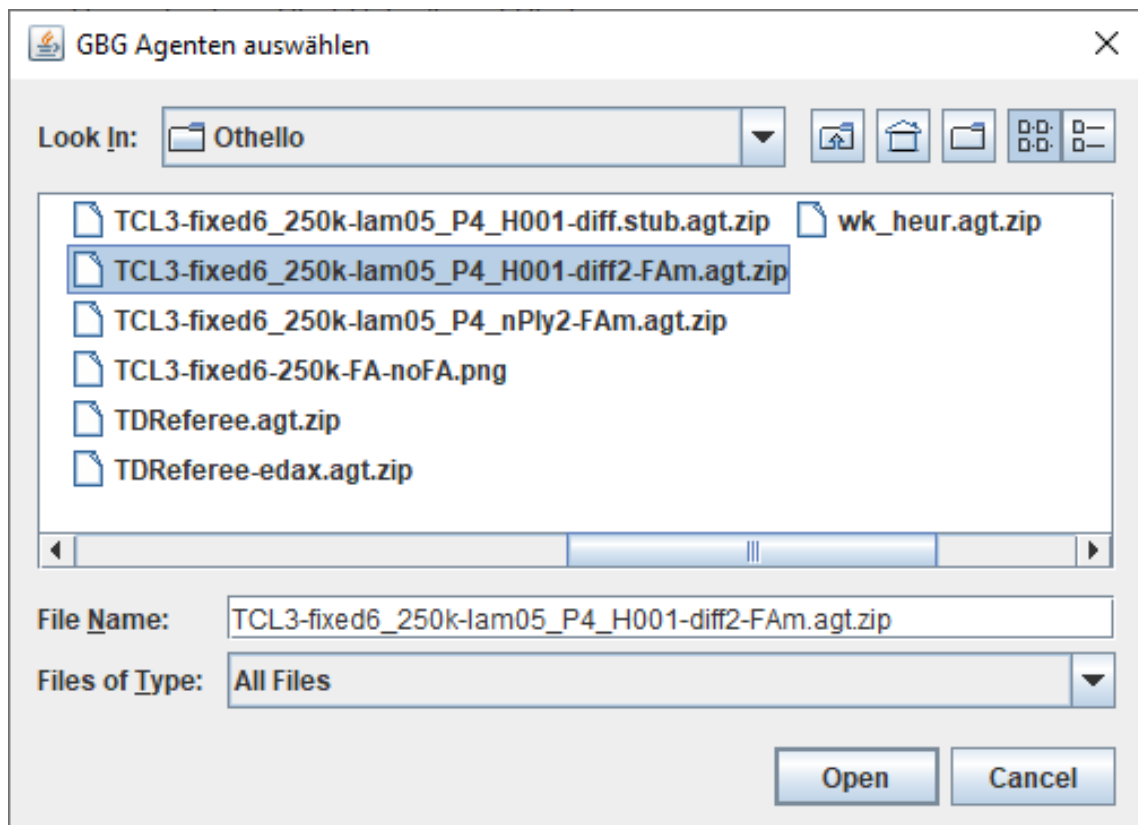


Abbildung 14: Auswahl eines beliebigen exportierten GBG-Agenten

5.4 Fazit

Das Informatikprojekt hat neben einer Einführung in das Themengebiet General-Board-Game playing auch erfolgreich tiefere Einblicke in die GBG- und Ludii-Frameworks gegeben. Teilweise brachte das Projekt aufgrund der beschriebenen Hürden und Fallstricke teilweise ziemliche Frustrationen mit sich, dennoch ist es rückblickend eine tolle Erfahrung gewesen, welche auch das Interesse für dieses Themengebiet gestärkt hat. Das Projekt wurde vor allem durch die Veröffentlichung der Ludii-Version 1.0.2 interessant, wodurch sich das Projektergebnis signifikant verändert hat. In Summe bietet das Projekt eine optimale Grundlage für eine aufbauende Bachelorarbeit.

Literatur

- [1] Maciej Świechowski u. a. “Recent Advances in General Game Playing”. In: *The Scientific World Journal* 2015 (Aug. 2015), S. 986262. ISSN: 2356-6140. DOI: 10.1155/2015/986262. URL: <https://doi.org/10.1155/2015/986262>.
- [2] Éric Piette u. a. *Ludii – The Ludemic General Game System*. 2019. arXiv: 1905.05013 [cs.AI].
- [3] Cameron Browne u. a. *Foundations of Digital Archæoludology*. 2019. arXiv: 1905.13516 [cs.AI].

Anhang

Erhobene Spieldaten

Nachfolgend werden die erhobenen Spieldaten aufgelistet. Eine Spielpartie wird dabei als chronologische Abfolge der Spielfelder auf denen ein Zug stattgefunden hat dargestellt. Begonnen mit dem ersten Wert beschreibt dabei jeder zweite Index einen Spielzug des schwarzen Spielers. Die verbleibenden Werte repräsentieren Spielzüge des weißen Spielers. Bei den Indizes handelt es sich um die Ludii-basierten Indizes. Diese lassen sich bei Bedarf in die entsprechende GBG-Repräsentation umrechnen. Der Zusammenhang zwischen Ludii- und GBG-Spielfeldindizes wird in Kapitel 2.2.1 näher erläutert.

Ludii (Version 0.9.3) spielt Schwarz und GBG spielt Weiß

Gültige Siege des Ludii-Agenten

1. 27, 36, 28, 35, 34, 44, 37, 26, 51, 42, 19, 45, 53, 20, 43, 50, 13, 18, 17, 25, 33, 41, 52, 32, 40, 48, 29, 21, 22, 30, 39, 38, 31, 46, 58, 47, 55, 24, 54, 12, 16, 8, 11, 10, 2, 6, 5, 4, 9, 15, 3, 23, 7, 14, 0, 1, 56, 63, 62, 61, 59, 60, 49, 57
2. 27, 36, 28, 35, 29, 19, 10, 45, 42, 43, 50, 37, 26, 18, 46, 38, 47, 34, 53, 21, 13, 20, 11, 44, 25, 51, 52, 33, 32
3. 27, 36, 28, 35, 20, 37, 42, 34, 26, 18, 44, 43, 10, 19, 50, 45, 21, 29, 53, 13, 5, 17, 16, 11, 2, 12, 4, 25, 24, 51, 59
4. 27, 36, 28, 35, 29, 19, 42, 43, 44, 45, 46, 20, 11, 34, 37, 18, 25, 21, 22, 10, 2, 12, 3, 33, 32, 41, 40, 17, 16, 9, 0, 1, 26, 50, 58, 30, 31, 4, 5
5. 27, 36, 28, 35, 43, 26, 19, 44, 45, 37, 46, 18, 17, 29, 30, 20, 12
6. 27, 36, 28, 35, 20, 37, 46, 18, 26, 19, 42, 34, 10, 21, 44, 45, 53, 29, 13, 43, 52, 51, 59, 50, 57, 12, 4
7. 27, 36, 28, 35, 43, 26, 20, 29, 19, 44, 22, 37, 25, 42, 46, 18, 34, 45, 41, 11, 10, 33, 2, 21, 40, 12, 13, 38, 47, 52, 60, 53, 61, 51, 59, 17, 8
8. 27, 36, 28, 35, 20, 37, 43, 34, 42, 19, 41, 21, 11, 18, 38, 45, 10, 29, 22, 44, 17, 26, 53, 30, 31, 52, 60
9. 27, 36, 28, 35, 29, 19, 18, 37, 20, 26, 44, 45, 42, 34, 33, 43, 53, 41, 40, 25, 16, 52, 51, 50, 58, 46, 47, 38, 39, 11, 3
10. 27, 36, 28, 35, 29, 19, 42, 43, 44, 45, 10, 51, 58, 41, 40, 50, 59, 49, 56, 57, 37, 34, 53, 48, 18, 32, 33, 26, 24, 25, 17, 21, 13, 20, 12, 11, 2, 5, 3, 6, 9, 30, 39, 0, 16, 38, 31, 46, 47, 60, 61, 52, 22, 54, 63, 4, 7, 14, 62, 55, 23, 8, 1, 15

Ungültige Siege des Ludii-Agenten

1. 27, 36, 28, 35, 34, 44, 21, 20, 13, 42, 50, 18, 37, 43, 19, 29, 52, 45, 30, 12, 4, 38, 47, 53, 61, 51, 60, 46, 39, 33, 32, 26, 10, 22, 23, 31, 54, 5, 14, 7, 15, 3, 11, 41, 40, 49, 56, 57, 58, 63, 55, 17, 2, 6, 16, 25, 9, 59, 24, 0, 1, 8

2. 27, 36, 28, 35, 29, 19, 34, 43, 18, 21, 13, 26, 20, 42, 50, 17, 25, 37, 46, 51, 16, 12, 60, 22, 52, 45, 53, 30, 23, 11, 5, 10, 2, 38, 47, 61, 44, 59, 58, 57, 54, 63, 62, 49, 33, 40, 56, 55, 24, 9, 41, 39, 31, 15, 14, 7, 6, 4, 1, 0, 3, 8
3. 27, 36, 28, 35, 34, 44, 37, 26, 52, 42, 19, 45, 53, 20, 43, 50, 33, 10, 58, 18, 12, 29, 51, 25, 16, 21, 13, 30, 38, 22, 17, 3, 2, 9, 0, 1, 23, 8, 4, 24, 32, 11
4. 27, 36, 28, 35, 43, 26, 21, 29, 37, 45, 19, 20, 53, 44, 51, 34, 12, 13, 5, 18, 42, 30, 23, 22, 31, 38, 17, 11, 3, 25, 39, 46, 16, 33, 47, 14, 7, 15, 32, 41, 40, 52, 61, 58, 59, 50, 57, 60, 49, 56
5. 27, 36, 28, 35, 20, 37, 43, 34, 45, 21, 29, 52, 59, 61, 22, 19, 18, 13, 5, 44, 42, 30, 23, 10, 2, 51, 33, 14, 7, 6, 58, 25, 60, 57, 50, 17, 16, 26, 38, 11, 12, 32, 40, 48, 49, 3, 4, 9, 0, 39, 31, 56, 41, 53, 47, 24, 46, 8, 62, 63
6. 27, 36, 28, 35, 43, 26, 19, 44, 25, 42, 37, 18, 34, 45, 53, 33, 10, 52, 50, 41, 61, 38, 40, 29, 39, 51, 58, 17, 20, 22, 24, 21, 23, 32, 30, 46, 47, 54, 63, 62, 60, 16, 8, 11, 3, 12, 9, 0, 13, 48, 1, 5, 4, 31, 6, 2
7. 27, 36, 28, 35, 34, 44, 53, 18, 19, 26, 33, 25, 16, 20, 12, 45, 17, 29, 46, 13, 30, 22, 21, 4, 5, 6, 37, 43, 52, 31, 23, 15, 14, 50, 42, 60, 58, 61, 51, 41, 40, 59, 62, 7
8. 27, 36, 28, 35, 43, 26, 18, 44, 34, 19, 20, 42, 45, 37, 50, 11, 52, 13, 29, 21, 5, 51, 2, 12, 59, 22, 4, 17, 25, 3, 23, 30, 10, 24, 39, 31, 16, 8, 38, 53, 33, 47, 55, 46, 61, 9, 0, 32, 40, 41, 49, 60, 54, 58, 57, 14, 7, 6
9. 27, 36, 28, 35, 43, 26, 18, 44, 45, 19, 20, 21, 34, 42, 50, 37, 22, 11, 2, 25, 16, 10, 29, 17, 9, 30, 31, 51, 58, 53, 61, 38, 52, 13, 5, 46, 39, 12, 4, 59, 60, 47, 54, 23, 14, 7, 15, 3, 55, 63
10. 27, 36, 28, 35, 29, 19, 42, 43, 44, 45, 46, 20, 26, 37, 13, 21, 30, 22, 11, 53, 61, 52, 59, 60, 51, 2, 3, 4, 5, 6, 23, 38, 12, 34, 25, 18, 39, 54, 17, 31, 63, 47, 10, 15, 14, 16, 9, 7
11. 27, 36, 28, 35, 43, 26, 19, 44, 45, 37, 25, 52, 59, 42, 20, 29, 21, 18, 51, 34, 33, 50, 11, 10, 17, 13, 58, 22, 5, 30, 2, 12, 23, 46, 47, 53, 61, 41, 40, 38, 39, 54, 3, 31, 63, 24, 32, 48, 49, 56, 57, 60
12. 27, 36, 28, 35, 29, 19, 34, 43, 26, 37, 51, 21, 30, 20, 11, 42, 33, 12, 22, 18, 4, 13, 10, 50, 58, 25, 44, 17, 16, 45, 5, 9, 46, 38, 0, 3, 47, 2, 1, 40, 41, 23, 39, 31, 15, 52, 60, 53, 54, 63, 61, 55
13. 27, 36, 28, 35, 34, 44, 29, 20, 43, 42, 51, 26, 45, 37, 17, 18, 11, 21, 33, 19, 13, 12, 5, 10, 2, 25, 16, 41, 40, 53, 61, 52, 60, 50, 58, 49, 56, 48, 32, 38, 39, 30, 31, 46, 47, 22, 23, 4, 3, 14, 7, 6
14. 27, 36, 28, 35, 20, 37, 46, 18, 26, 19, 17, 25, 24, 21, 22, 12, 29, 34, 5, 30, 38, 45, 43, 44, 53, 42, 41, 11, 4, 2, 3, 6, 13, 10, 1, 0
15. 27, 36, 28, 35, 34, 44, 53, 18, 19, 26, 17, 25, 16, 11, 29, 20, 3, 42, 50, 41, 40, 52, 45, 37, 13, 58, 46, 21, 51, 38, 47, 30, 23, 12, 5, 4, 33, 43, 22, 60, 61, 62, 59, 54, 63, 49, 56, 10, 9, 14, 57, 6, 7, 0, 8, 24, 32, 48

16. 27, 36, 28, 35, 29, 19, 42, 43, 34, 37, 20, 41, 51, 26, 45, 44, 40, 21, 33, 52, 13, 18, 53, 12, 5, 11, 3, 10, 2, 50, 58, 25, 32, 49, 56, 48, 17, 30, 38, 22, 23, 46, 47, 16, 24, 14, 7, 4, 8, 0
17. 27, 36, 28, 35, 43, 26, 19, 44, 45, 37, 30, 18, 17, 52, 21, 29, 61, 38, 20, 34, 47, 25, 16, 42, 50, 22, 23, 51, 58, 33, 32, 41, 40, 59, 60, 49, 56, 39, 31, 57, 46, 53, 54, 12, 4, 10, 2, 24, 11, 9, 1, 13, 5, 14, 7, 15
18. 27, 36, 28, 35, 20, 37, 42, 34, 26, 18, 10, 29, 38, 43, 52, 44, 51, 22, 30, 19, 45, 21, 23, 46, 47, 11, 2, 12, 5, 13, 6, 50, 58, 4, 3, 53, 61, 54, 63, 55, 39, 25, 24, 41, 33, 32, 40, 17, 16, 9, 0, 1
19. 27, 36, 28, 35, 34, 44, 45, 26, 18, 37, 43, 42, 50, 19, 11, 10, 2, 25, 30, 17, 24, 33, 16, 41, 20, 12, 40, 3, 32, 51, 58, 53, 5, 4, 29, 46, 61, 21, 47, 38, 39, 59, 22, 13, 60, 49, 56, 48, 52, 23, 15, 7, 6, 9, 0, 14, 8, 55, 63, 1, 54, 31
20. 27, 36, 28, 35, 43, 26, 17, 45, 21, 29, 37, 44, 53, 42, 34, 52, 61, 38, 41, 46, 47, 30, 20, 19, 18, 51, 39, 33, 40, 50, 58, 22, 23, 25, 16, 54, 63, 62, 60, 32, 24, 49, 56, 57, 59, 11, 4, 2, 12, 10, 9, 13, 5, 0, 1, 31, 3, 6, 14, 7, 15, 48
21. 27, 36, 28, 35, 20, 37, 43, 34, 29, 21, 25, 19, 13, 26, 30, 51, 59, 42, 33, 44, 45, 12, 3, 52, 11, 18, 61, 10, 2, 17, 16, 60, 53, 41, 40, 38, 47, 22, 23, 46, 55, 5, 4, 1, 14, 6, 62, 63
22. 27, 36, 28, 35, 34, 44, 37, 26, 52, 42, 19, 45, 43, 18, 17, 51, 41, 50, 53, 25, 16, 33, 58, 20, 40, 49, 21, 13, 56, 57, 59, 29, 5, 30, 31, 22, 23, 12, 4, 11, 3, 61, 54, 10, 2, 14, 38, 24, 32, 62, 7, 63, 46, 55, 47, 39
23. 27, 36, 28, 35, 29, 19, 18, 37, 45, 26, 34, 42, 50, 44, 43, 51, 60, 41, 40, 33, 17, 25, 16, 58, 12, 59, 57, 52, 61, 53, 46, 38, 47, 21, 13, 11, 20, 10, 2, 32, 24, 3, 4, 22, 23, 30, 31, 14, 7, 6, 5, 9, 0, 1
24. 27, 36, 28, 35, 34, 44, 53, 18, 19, 26, 20, 45, 37, 29, 10, 12, 5, 11, 21, 43, 2, 13, 51, 52, 6, 42, 61, 50, 58, 4, 3, 25, 33, 17, 16, 9, 0, 8, 24, 38, 39, 41, 40, 30, 31, 32
25. 27, 36, 28, 35, 34, 44, 21, 20, 37, 26, 18, 19, 53, 25, 16, 45, 46, 17, 10, 29, 24, 38, 13, 12, 5, 22, 23, 42, 50, 52, 43, 30, 31, 33, 41, 61, 59, 11, 2, 9, 0, 8, 62, 51, 58, 63
26. 27, 36, 28, 35, 29, 19, 42, 43, 10, 45, 50, 37, 44, 53, 61, 38, 18, 46, 47, 51, 60, 20, 21, 34, 52, 22, 23, 26, 33, 12, 4, 41, 40, 25, 17, 39, 31, 54, 63, 55, 30, 11, 2, 13, 5, 24, 32, 48, 49, 56, 57, 16, 9, 14, 7, 62, 15, 0, 1, 58, 59, 8

Gültige Siege des GBG-Agenten

1. 27, 36, 28, 35, 43, 26, 20, 29, 19, 44, 30, 42, 51, 34, 52, 37, 45, 21, 25, 33, 40, 12, 5, 10, 41, 3, 4, 6, 18, 11, 13, 17, 16, 46, 47, 38, 31, 50, 2, 1, 58, 49, 56, 48, 9, 22, 23, 0, 8, 24, 32, 57, 59, 39, 14, 7, 15, 61, 60, 53, 54, 63, 62, 55
2. 27, 36, 28, 35, 34, 44, 29, 20, 45, 42, 11, 37, 50, 21, 43, 19, 13, 18, 25, 38, 31, 22, 23, 10, 2, 52, 53, 47, 51, 61, 30, 26, 39, 15, 46, 55, 17, 33, 41, 32, 24, 16, 12, 5, 54, 63, 62, 58, 59, 60, 14, 7, 6, 4, 3, 1, 9, 40, 49, 56, 48, 57, 0, 8

3. 27, 36, 28, 35, 34, 44, 29, 20, 11, 42, 43, 26, 18, 19, 41, 37, 25, 33, 52, 51, 50, 40, 32, 24, 12, 16, 46, 17, 45, 53, 61, 38, 47, 54, 63, 62, 60, 48, 30, 58, 49, 10, 1, 21, 13, 2, 3, 5, 4, 0, 8, 9, 56, 31, 39, 6, 23, 55, 22, 14, 7, 57, 59, 15
4. 27, 36, 28, 35, 29, 19, 18, 37, 20, 26, 33, 21, 45, 44, 22, 12, 5, 34, 53, 43, 42, 13, 4, 11, 30, 38, 46, 23, 31, 39, 3, 10, 2, 25, 16, 17, 24, 41, 47, 55, 40, 32, 14, 15, 54, 63, 62, 51, 50, 61, 60, 52, 59, 7, 6, 1, 9, 58, 49, 56, 57, 0, 8, 48

Ungültige Siege des GBG-Agenten

1. 27, 36, 28, 35, 29, 19, 26, 37, 11, 21, 45, 44, 30, 20, 46, 38, 31, 53, 43, 52, 60, 61, 12, 59, 54, 50, 42, 51, 58, 13, 5, 34, 33, 22, 23, 18, 17, 57, 49, 4, 3, 14, 7, 6, 62, 63, 55, 25, 24, 15, 41, 56, 48, 40, 32, 16, 10, 47, 39, 1, 2, 9, 0
2. 27, 36, 28, 35, 34, 44, 45, 26, 18, 37, 38, 42, 51, 19, 33, 41, 40, 25, 21, 43, 50, 52, 20, 53, 61, 29, 32, 30, 23, 58, 59, 46, 47, 54, 63, 55, 17, 31, 39, 12, 22, 13, 11, 10, 2, 3, 5, 4, 57, 16, 24, 48, 49, 60, 9, 8, 0, 56, 1, 15, 7, 62
3. 27, 36, 28, 35, 20, 37, 42, 34, 46, 18, 44, 45, 26, 43, 10, 19, 12, 11, 53, 52, 51, 50, 2, 60, 59, 58, 49, 61, 29, 38, 39, 47, 54, 31, 30, 23, 22, 63, 55, 21, 13, 57, 14, 25, 24, 17, 16, 7, 15, 33, 32, 56, 48, 40, 41, 8, 9, 4, 3, 62, 5, 6
4. 27, 36, 28, 35, 34, 44, 37, 26, 19, 45, 33, 20, 43, 29, 21, 18, 46, 12, 13, 38, 10, 30, 11, 42, 47, 22, 50, 39, 31, 5, 4, 3, 2, 1, 9, 41, 32, 0, 25, 17, 8, 6, 14, 58, 49, 51, 59, 60, 52, 61, 53, 62, 54, 7, 15, 63, 55, 23
5. 27, 36, 28, 35, 34, 44, 37, 26, 20, 19, 25, 33, 32, 43, 12, 45, 51, 42, 50, 41, 29, 18, 40, 52, 61, 11, 4, 17, 16, 38, 53, 58, 31, 21, 59, 13, 5, 10, 3, 24, 49, 46, 47, 56, 9, 60, 57, 62, 54, 0, 8, 63, 55, 39, 30, 23, 22, 15, 14, 48
6. 27, 36, 28, 35, 34, 44, 37, 26, 52, 42, 33, 43, 17, 18, 51, 50, 21, 45, 38, 20, 12, 29, 25, 41, 19, 10, 3, 2, 22, 4, 11, 59, 40, 5, 58, 57, 1, 0, 13, 6, 14, 7, 15, 23, 30, 53, 46, 9, 60, 61, 54, 8, 16, 24, 32, 48, 49, 31, 39, 56
7. 27, 36, 28, 35, 34, 44, 37, 26, 52, 42, 20, 45, 41, 43, 38, 33, 32, 19, 12, 18, 25, 17, 11, 51, 16, 29, 59, 50, 58, 49, 30, 21, 22, 46, 56, 13, 47, 48, 5, 40, 24, 23, 14, 31, 39, 7, 15, 55, 54, 63, 53, 57, 62, 2, 4, 3, 10, 6
8. 27, 36, 28, 35, 43, 26, 20, 29, 19, 44, 22, 37, 46, 21, 25, 45, 52, 53, 42, 34, 38, 12, 61, 30, 31, 51, 60, 50, 3, 13, 11, 4, 58, 2, 5, 6, 10, 54, 63, 55, 14, 7, 39, 47, 15, 33, 17, 18, 32, 23, 9, 41, 40, 8, 0, 1, 16, 62

Gültige untentschiedene Parteien

1. 27, 36, 28, 35, 34, 44, 29, 20, 43, 42, 51, 26, 33, 45, 11, 13, 37, 52, 19, 18, 10, 17, 61, 40, 32, 24, 12, 25, 53, 21, 5, 58, 50, 57, 16, 8, 49, 56, 48, 41, 0, 9, 2, 38, 39, 22, 23, 30, 46, 1, 31, 3, 4, 14, 15, 59, 60, 62, 54, 63, 55, 7, 6, 47

Ungültige untentschiedene Parteien

1. 27, 36, 28, 35, 20, 37, 42, 34, 44, 19, 30, 21, 10, 18, 26, 43, 22, 29, 45, 38, 46, 41, 40, 13, 4, 12, 5, 11, 2, 3, 14, 33, 25, 7, 23, 24, 17, 16, 9, 53, 61, 47, 31, 50, 58, 6, 51, 0, 1, 49, 56, 48, 32, 15, 8, 39

GBG spielt Schwarz und Ludii (Version 0.9.3) spielt Weiß**Gültige Siege des Ludii-Agenten**

1. 27, 36, 28, 35, 34, 42, 43, 44, 51, 58, 50, 26, 45, 46
2. 27, 36, 28, 35, 34, 26, 19, 44, 18, 10, 43, 42, 41, 40, 37, 25, 20, 38, 11, 12, 29, 30
3. 27, 36, 28, 35, 34, 44, 29, 18, 26, 42, 17, 16, 10, 2, 9, 0, 19, 20, 11, 38, 25, 43, 50, 58, 52, 61
4. 27, 36, 28, 35, 34, 42, 43, 44, 51, 60, 45, 52, 37, 46, 50, 41, 59, 58, 29, 22, 53, 61, 38, 47, 21, 20, 19, 11, 30, 23, 12, 5, 13, 4, 26, 18, 10, 2, 25, 24, 17, 16
5. 27, 36, 28, 35, 34, 42, 43, 26, 33, 40, 41, 44, 18, 10
6. 27, 36, 28, 35, 34, 42, 43, 44, 51, 58, 50, 26, 45, 46
7. 27, 36, 28, 35, 34, 42, 43, 44, 51, 58, 50, 26, 45, 46
8. 27, 36, 28, 35, 34, 42, 43, 26, 33, 45, 18, 32, 44, 10, 41, 40, 21, 14
9. 27, 36, 28, 35, 34, 42, 43, 44, 51, 58, 50, 26, 45, 46
10. 27, 36, 28, 35, 34, 42, 43, 44, 51, 58, 50, 26, 45, 46
11. 27, 36, 28, 35, 34, 42, 43, 44, 51, 58, 50, 26, 45, 46
12. 27, 36, 28, 35, 34, 26, 19, 10, 11, 44, 18, 42, 37, 12, 45, 29, 21, 30, 43, 46, 20, 13, 38, 39, 52, 60
13. 27, 36, 28, 35, 34, 42, 43, 44, 51, 58, 50, 26, 45, 46
14. 27, 36, 28, 35, 34, 26, 19, 10, 11, 42, 20, 12, 18, 13, 21, 30, 22, 29, 37, 23, 43, 51, 2, 38, 44, 25, 46, 52, 5, 4, 3, 45, 47, 31, 17, 16, 33, 24, 59, 60, 61, 53, 41, 39, 15, 49, 58, 50, 57, 9, 0, 1, 62, 54, 63, 55, 6, 7, 8, 14, 32, 48, 40, 56
15. 27, 36, 28, 35, 34, 44, 29, 22, 30, 18, 26, 33, 42, 50, 25, 38, 40, 21, 41, 32, 24, 37, 10, 20, 51, 16, 8, 43, 19, 11, 17, 2, 4, 60, 58, 59, 61, 12, 13, 5, 6, 49, 52, 45, 47, 31, 53, 46, 56, 48, 57, 0, 14, 9, 23, 39, 54, 63, 62, 55, 1, 15, 3, 7
16. 27, 36, 28, 35, 34, 42, 43, 44, 51, 60, 45, 52, 37, 25, 50, 38, 26, 41, 59, 58, 40, 33, 29, 32, 24, 16, 8, 20, 53, 61, 19, 11, 18, 17, 46, 47, 12, 4, 3, 10, 5, 13, 30, 21, 39, 22, 55, 54, 63, 62, 57, 49, 48, 56, 9, 0, 2, 1, 31, 6, 14, 7, 15, 23
17. 27, 36, 28, 35, 34, 42, 43, 26, 33, 40, 41, 51, 44, 45, 50, 58
18. 27, 36, 28, 35, 34, 42, 43, 44, 51, 18, 45, 33, 41, 52, 37, 40, 26, 38, 50, 58, 53, 62
19. 27, 36, 28, 35, 34, 44, 29, 18, 26, 42, 17, 16, 10, 2, 9, 0, 19, 12, 45, 30, 43, 46, 21, 20, 37, 22, 11, 3, 53, 61
20. 27, 36, 28, 35, 34, 42, 43, 44, 51, 58, 50, 18, 45, 26, 29, 46, 21, 20, 37, 13, 19, 11, 12, 4

21. 27, 36, 28, 35, 34, 42, 43, 26, 33, 40, 41, 51, 44, 45, 50, 58
22. 27, 36, 28, 35, 34, 42, 43, 26, 33, 40, 41, 44, 18, 10
23. 27, 36, 28, 35, 34, 44, 29, 22, 30, 18, 26, 42, 17, 21, 13, 16, 10, 31, 43, 2, 9, 0, 19, 12, 11, 3, 1, 4, 37, 52, 41, 40, 33, 25, 20, 45, 50, 58, 51, 59, 38, 47, 46, 55, 5, 6
24. 27, 36, 28, 35, 34, 42, 43, 44, 51, 58, 50, 26, 45, 46
25. 27, 36, 28, 35, 34, 42, 43, 26, 33, 40, 41, 45, 18, 19, 20, 44, 51, 12, 50, 58, 25, 17, 37, 29, 52, 61, 53, 62, 11, 3

Ungültige Siege des Ludii-Agenten

1. 27, 36, 28, 35, 34, 26, 19, 42, 20, 12, 11, 44, 21, 18, 17, 13, 29, 37, 43, 2, 4, 10, 45, 16, 38, 53, 30, 51, 46, 47, 52, 23, 59, 61, 50, 60, 62, 57, 22, 5, 6, 15, 39, 31, 54, 63, 55, 58, 7, 14, 3
2. 27, 36, 28, 35, 34, 42, 43, 44, 51, 18, 45, 50, 33, 59, 52, 24, 41, 26, 32, 38, 16, 40, 17, 25, 37, 46, 21, 20, 48, 22, 61, 60, 58, 49, 19, 11, 29, 9, 57, 30, 53, 62, 63
3. 27, 36, 28, 35, 34, 44, 29, 20, 21, 18, 26, 22, 45, 46, 10, 17, 42, 2, 53, 61, 12, 3, 4, 11, 1, 52, 37, 41, 51, 38, 60, 59, 30, 33, 23, 43, 47, 14, 13, 19, 25, 24, 16, 8, 32, 40, 7, 5, 6, 9, 50, 15, 31, 39, 58, 57, 49, 56, 48, 55, 63
4. 27, 36, 28, 35, 34, 26, 19, 10, 11, 42, 20, 29, 18, 3, 41, 40, 43, 50, 17, 16, 45, 25, 21, 37, 44, 46, 33, 13, 51, 58, 52, 61, 38, 53, 22, 39, 24, 23, 12, 32, 30, 5, 4, 31, 2, 1, 59, 9, 60
5. 27, 36, 28, 35, 34, 26, 19, 12, 11, 44, 18, 3, 20, 10, 43, 42, 45, 21, 51, 52, 50, 58, 53, 46, 37, 29, 30, 61, 13, 23, 22, 5, 59, 31, 60
6. 27, 36, 28, 35, 34, 42, 43, 44, 51, 18, 45, 50, 33, 26, 41, 40, 32, 46, 17, 16, 52, 53, 19, 20, 25, 24, 37, 30, 58, 59, 21, 12, 13, 5, 22, 23, 14, 29, 60, 7, 61, 49, 38, 39, 57, 54, 11, 2, 62, 10, 63, 55, 47
7. 27, 36, 28, 35, 34, 44, 29, 33, 41, 21, 13, 26, 18, 17, 20, 10, 42, 51, 37, 45, 19, 5, 43, 12, 11, 2, 24, 40, 52, 59, 50, 58, 53, 61, 49, 56, 16, 25, 4, 9, 0, 3, 32, 8, 48
8. 27, 36, 28, 35, 34, 44, 29, 18, 26, 33, 42, 51, 43, 37, 25, 21, 45, 53, 50, 24, 46, 58, 41, 39, 32, 40, 52, 61, 22, 23, 30, 31, 47, 38, 15, 55, 63
9. 27, 36, 28, 35, 34, 26, 19, 10, 11, 42, 20, 37, 18, 13, 21, 29, 43, 12, 45, 17, 33, 40, 41, 44, 51, 46, 50, 58, 53, 61, 49, 56, 47, 39, 31, 52, 30, 23, 15, 38, 4, 3, 55, 54, 63, 5, 22, 14, 2, 1, 59, 60, 62
10. 27, 36, 28, 35, 34, 44, 29, 20, 21, 22, 45, 26, 43, 41, 33, 50, 42, 46, 18, 10, 51, 58, 25, 40, 37, 32, 52, 61, 53, 30, 38, 31, 47, 54, 19, 11, 17, 16, 24, 49, 12, 5, 63, 13, 56, 55, 39, 57, 59, 60, 62

11. 27, 36, 28, 35, 34, 44, 29, 20, 21, 22, 45, 43, 42, 18, 37, 51, 30, 33, 13, 23, 19, 46, 52, 5, 38, 61, 50, 58, 26, 53, 47, 39, 10, 17, 12, 4, 41, 40, 25, 16, 59, 60, 14, 7, 6, 3, 11, 2, 15, 31, 54, 63, 49, 55, 62, 56, 57
12. 27, 36, 28, 35, 34, 42, 43, 44, 51, 58, 50, 18, 45, 41, 37, 59, 26, 52, 53, 46, 19, 61, 21, 20, 29, 38, 33, 22, 60, 17, 30, 23, 25, 16, 40, 24, 32, 48, 47, 31, 13, 5, 14, 7, 15
13. 27, 36, 28, 35, 34, 42, 43, 26, 33, 24, 18, 25, 19, 12, 17, 44, 20, 10, 41, 21, 11, 4, 2, 40, 3, 1, 5, 6, 13, 22, 29, 38, 37, 45, 30, 23, 46, 47, 31, 39, 53, 61, 52, 59, 54, 51, 60, 63, 62, 9, 32
14. 27, 36, 28, 35, 34, 42, 43, 26, 33, 24, 18, 37, 25, 10, 41, 50, 19, 40, 17, 16, 32, 20, 11, 2, 12, 5, 29, 21, 44, 45, 13, 6, 52, 61, 3, 4, 51, 59, 53, 54, 46, 47, 38, 39, 58, 49, 48, 56, 57
15. 27, 36, 28, 35, 34, 44, 29, 18, 26, 42, 17, 16, 10, 2, 9, 0, 19, 12, 45, 43, 50, 21, 20, 46, 13, 51, 25, 58, 4, 22, 37, 33, 11, 3, 41, 40, 49, 5, 30, 56, 38, 23, 31, 47, 39
16. 27, 36, 28, 35, 34, 26, 19, 42, 20, 13, 21, 29, 18, 12, 43, 17, 11, 2, 37, 52, 50, 58, 61, 44, 45, 53, 51, 10, 46, 47, 38, 30, 22, 23, 5, 4, 62, 60, 16, 63, 59, 25, 24, 33, 32, 40, 48, 41, 9, 8, 0, 6, 3, 1, 7, 54, 14, 15, 31, 39, 55

Gültige Siege des GBG-Agenten

1. 27, 36, 28, 35, 34, 26, 19, 44, 18, 33, 42, 10, 43, 17, 37, 41, 20, 13, 53, 46, 51, 60, 21, 52, 5, 22, 11, 2, 45, 38, 29, 4, 12, 30, 3, 6, 25, 16, 61, 62, 58, 50, 24, 14, 32, 40, 7, 15, 1, 9, 0, 8, 48, 49, 56, 57, 59, 54, 63, 55, 47, 39, 23, 31
2. 27, 36, 28, 35, 34, 44, 29, 18, 26, 33, 42, 17, 45, 51, 43, 30, 41, 53, 52, 50, 58, 40, 61, 46, 37, 59, 60, 38, 47, 54, 25, 32, 19, 20, 22, 21, 63, 23, 62, 55, 39, 31, 15, 14, 7, 13, 11, 10, 12, 5, 6, 2, 4, 3, 1, 9, 0, 8, 57, 49, 56, 48, 24, 16
3. 27, 36, 28, 35, 34, 26, 19, 42, 20, 11, 18, 37, 43, 10, 33, 40, 21, 29, 41, 44, 52, 22, 50, 58, 45, 61, 59, 53, 12, 60, 2, 4, 38, 13, 51, 39, 17, 16, 25, 24, 32, 3, 5, 30, 46, 14, 1, 47, 6, 9, 31, 23, 0, 8, 7, 15, 55, 54, 62, 63, 49, 48, 56, 57

Ungültige Siege des GBG-Agenten

1. 27, 36, 28, 35, 34, 26, 19, 12, 11, 10, 29, 42, 33, 40, 18, 43, 20, 25, 44, 13, 21, 22, 37, 45, 2, 17, 4, 5, 41, 3, 6, 49, 56, 50, 16, 9, 48, 32, 0, 1, 8, 7, 24, 57, 58, 59, 51, 60, 61
2. 27, 36, 28, 35, 34, 42, 43, 44, 51, 60, 45, 52, 37, 46, 50, 33, 26, 18, 19, 20, 10, 17, 25, 58, 11, 16, 41, 2, 40, 32, 59, 12, 53, 61, 38, 30, 21, 22, 29, 54, 63, 47, 62, 49, 57, 48, 13, 14, 3, 4, 5, 6, 7, 15, 1, 9, 0, 8, 23, 31, 39, 55, 56
3. 27, 36, 28, 35, 34, 26, 19, 12, 11, 42, 20, 21, 29, 10, 13, 37, 18, 30, 22, 5, 33, 40, 41, 31, 23, 17, 39, 14, 45, 38, 43, 44, 47, 25, 52, 60, 51, 59, 24, 16, 8, 54, 53, 62, 7, 9, 15, 46, 0, 1, 32, 6, 4, 2, 3

4. 27, 36, 28, 35, 34, 26, 19, 12, 11, 10, 29, 42, 33, 37, 21, 40, 18, 43, 20, 38, 44, 41, 45, 30, 51, 46, 50, 58, 25, 32, 49, 56, 13, 5, 22, 23, 48, 52, 47, 53, 16, 17, 3, 39, 4, 2, 61, 60, 31, 62, 15, 14, 7, 6, 1, 9, 0, 8, 24
5. 27, 36, 28, 35, 34, 26, 19, 12, 11, 44, 18, 25, 20, 10, 17, 42, 37, 13, 21, 29, 43, 16, 45, 52, 53, 61, 38, 31, 46, 47, 39, 30, 23, 15, 22, 14, 33, 32, 41, 40, 51, 59, 50, 49, 3, 2, 1, 9, 0, 8, 56, 58, 7, 6, 55, 54, 63, 62, 5, 4, 60
6. 27, 36, 28, 35, 34, 44, 29, 33, 41, 19, 20, 22, 18, 26, 43, 10, 11, 2, 40, 17, 25, 51, 16, 42, 21, 13, 37, 30, 12, 5, 53, 45, 23, 31, 39, 32, 24, 61, 50, 58, 52, 38, 47, 46, 55, 60, 49, 54, 63, 62, 59, 48, 14, 7, 56, 15, 3, 4, 57, 8, 0, 1, 6

Ludii (Version 1.0.2) spielt Schwarz und GBG spielt Weiß**Siege des Ludii-Agenten**

1. 27, 36, 28, 35, 20, 37, 46, 18, 26, 19, 10, 11, 45, 29, 30, 21, 12, 38, 47, 22, 23, 13, 5, 14, 34, 44, 7, 15, 51, 17, 16, 52, 60, 43, 50, 31, 39
2. 27, 36, 28, 35, 29, 19, 18, 37, 45, 26, 25, 21, 13, 44, 53, 30, 31, 20, 12
3. 27, 36, 28, 35, 29, 19, 10, 45, 44, 37, 42, 43, 46, 21, 13, 52, 18, 38, 61, 20, 53, 30, 22, 26, 33, 51, 12, 5, 58, 23, 50, 25, 17, 24, 32, 40, 41, 34, 14, 48, 49, 31, 39, 47, 54, 7, 15, 63, 62, 56, 55, 57, 59, 60, 2, 9, 11, 16, 4, 3, 6, 1, 0, 8
4. 27, 36, 28, 35, 34, 44, 37, 26, 18, 19, 52, 25, 20, 43, 16, 29, 30, 21, 13, 51, 59, 42, 49
5. 27, 36, 28, 35, 20, 37, 44, 19, 26, 45, 12, 34, 43, 42, 51, 52, 21, 13, 60, 29, 38, 10, 5, 18, 25, 33, 2, 41, 40, 50, 58, 17, 16, 11, 3
6. 27, 36, 28, 35, 29, 19, 10, 45, 44, 37, 46, 38, 47, 52, 60, 51, 34, 43, 59, 20, 12, 21, 13
7. 27, 36, 28, 35, 20, 37, 44, 19, 38, 21, 26, 45, 29, 18, 12, 42, 53, 34, 17, 43, 30, 11, 50, 25, 10, 33, 16, 41, 32, 13, 6
8. 27, 36, 28, 35, 29, 19, 26, 37, 45, 44, 43, 52, 61, 18, 11, 34, 25, 42, 51, 50, 58, 10, 20, 41, 40, 12, 2, 33, 5
9. 27, 36, 28, 35, 43, 26, 19, 44, 33, 42, 51, 18, 37, 34, 10, 41, 40, 25, 24, 20, 13
10. 27, 36, 28, 35, 43, 26, 19, 44, 45, 37, 29, 38, 30, 18, 42, 34, 47, 21, 13, 20, 11, 10, 2, 12, 5, 46, 55, 22, 23, 50, 57
11. 27, 36, 28, 35, 29, 19, 42, 43, 50, 21, 20, 30, 22, 12, 13, 37, 31, 38, 44, 45, 4, 11, 53, 46, 2, 3, 47, 52, 59, 51, 58, 54, 18, 10, 63, 62, 26, 25, 24, 17, 16, 5, 6, 9, 0, 61, 60, 49, 56, 34, 33, 41, 1, 39, 32, 57, 23, 48, 40, 55, 14, 8, 15, 7
12. 27, 36, 28, 35, 29, 19, 26, 37, 45, 44, 53, 18, 21, 20, 10, 30, 31, 22, 13, 34, 33
13. 27, 36, 28, 35, 43, 26, 21, 29, 20, 44, 45, 37, 38, 19, 34, 12, 11, 42, 25, 18, 3, 10, 33, 30, 2, 13, 5, 52, 22, 50, 41, 46, 47, 53, 61, 54, 63, 55, 58, 23, 39, 4, 24, 16, 8, 17, 32, 40, 48, 9, 60, 59, 0, 1, 31, 49, 62, 15, 56, 51, 57, 14, 7, 6
14. 27, 36, 28, 35, 20, 37, 44, 19, 30, 21, 12, 45, 53, 29, 26, 22, 38, 34, 23, 18, 17, 43, 41, 52, 50, 42, 59, 46, 47, 51, 58, 11, 3
15. 27, 36, 28, 35, 29, 19, 10, 45, 42, 43, 50, 37, 53, 21, 13, 22, 23, 46, 39, 51, 18, 44, 60, 26, 34, 33, 30, 41, 40, 20, 25, 38, 31, 11, 2, 12, 5, 52, 61, 17, 16, 14, 7, 15, 32, 24, 6, 4, 49, 56, 57, 58, 47, 55, 3, 54, 59, 48, 63, 62, 9, 0, 8, 1
16. 27, 36, 28, 35, 20, 37, 43, 34, 25, 21, 22, 19, 29, 44, 52, 45, 18, 53, 61, 42, 41, 30, 23, 12, 4, 13, 5, 11, 3, 38, 47
17. 27, 36, 28, 35, 34, 44, 37, 26, 51, 42, 33, 45, 53, 43, 19, 20, 12

18. 27, 36, 28, 35, 20, 37, 42, 34, 44, 19, 38, 21, 18, 26, 17, 25, 22, 43, 51, 13, 16, 29, 5, 11, 2, 10, 3, 30, 23, 4, 12, 41, 40, 24, 32, 9, 33, 1, 0, 45, 46, 53, 61
19. 27, 36, 28, 35, 34, 44, 45, 26, 52, 42, 33, 43, 50, 37, 19, 18, 38, 20, 10, 29, 21, 51, 59

Siege des GBG-Agenten

1. 27, 36, 28, 35, 20, 37, 42, 34, 41, 21, 22, 18, 38, 43, 50, 13, 12, 29, 5, 30, 23, 33, 32, 11, 2, 4, 3, 45, 10, 46, 19, 26, 25, 39, 53, 44, 47, 55, 52, 61, 17, 16, 51, 60, 9, 0, 1, 54, 8, 24, 63, 62, 59, 58, 57, 49, 56, 48, 31, 15, 14, 7, 6, 40
2. 27, 36, 28, 35, 29, 19, 10, 45, 44, 37, 53, 52, 59, 38, 18, 46, 31, 20, 26, 43, 13, 42, 34, 21, 41, 33, 22, 39, 32, 23, 47, 55, 30, 51, 60, 12, 14, 25, 11, 17, 16, 24, 9, 50, 58, 40, 48, 0, 49, 8, 1, 56, 57, 61, 54, 2, 3, 63, 62, 7, 15, 6, 5, 4
3. 27, 36, 28, 35, 34, 44, 45, 26, 18, 37, 29, 21, 13, 19, 20, 12, 42, 22, 52, 43, 51, 38, 4, 53, 61, 50, 31, 41, 58, 33, 40, 17, 16, 30, 23, 49, 25, 14, 57, 56, 48, 59, 7, 60, 32, 24, 5, 6, 11, 3, 62, 9, 8, 54, 0, 1, 2, 10, 15, 63, 55, 47, 39, 46
4. 27, 36, 28, 35, 34, 44, 29, 20, 21, 26, 12, 42, 43, 37, 41, 19, 18, 33, 53, 25, 40, 11, 45, 51, 50, 17, 58, 32, 24, 5, 16, 10, 2, 49, 3, 60, 59, 57, 56, 46, 47, 22, 23, 52, 61, 38, 39, 30, 31, 13, 4, 1, 9, 48, 14, 6, 7, 15, 55, 54, 63, 62, 0, 8
5. 27, 36, 28, 35, 34, 44, 21, 20, 19, 18, 17, 43, 52, 29, 42, 45, 46, 12, 22, 26, 4, 51, 58, 25, 24, 37, 11, 50, 33, 41, 30, 32, 40, 13, 59, 49, 56, 48, 5, 57, 23, 31, 39, 38, 10, 3, 15, 9, 61, 53, 47, 54, 2, 1, 63, 62, 0, 60, 6, 14, 55, 8, 7, 16
6. 27, 36, 28, 35, 34, 44, 21, 20, 19, 18, 17, 43, 51, 42, 50, 26, 53, 45, 46, 25, 33, 13, 5, 52, 61, 41, 40, 58, 59, 11, 12, 10, 2, 29, 38, 30, 39, 54, 63, 37, 60, 55, 47, 32, 24, 23, 31, 22, 15, 4, 3, 9, 0, 8, 16, 1, 57, 62, 56, 14, 7, 6, 49, 48
7. 27, 36, 28, 35, 43, 26, 20, 29, 22, 42, 19, 44, 52, 45, 50, 37, 33, 51, 59, 34, 38, 21, 41, 17, 18, 25, 16, 13, 5, 12, 4, 10, 2, 24, 32, 9, 0, 8, 11, 30, 23, 46, 47, 53, 61, 54, 63, 55, 39, 60, 31, 58, 49, 3, 1, 56, 57, 62, 14, 7, 6, 15, 48, 40
8. 27, 36, 28, 35, 29, 19, 34, 43, 18, 21, 20, 25, 22, 37, 50, 10, 2, 26, 11, 42, 45, 51, 59, 44, 32, 16, 33, 17, 52, 53, 24, 40, 61, 41, 49, 38, 30, 48, 31, 56, 57, 12, 3, 47, 39, 13, 4, 23, 14, 15, 7, 6, 58, 46, 8, 9, 60, 54, 63, 62, 5, 55, 0, 1
9. 27, 36, 28, 35, 34, 44, 45, 26, 43, 37, 38, 42, 41, 51, 60, 29, 20, 33, 24, 50, 19, 18, 25, 17, 10, 11, 4, 59, 12, 61, 58, 53, 52, 57, 62, 63, 22, 21, 40, 23, 32, 49, 16, 48, 30, 31, 8, 9, 56, 46, 0, 1, 54, 55, 5, 13, 47, 39, 15, 14, 6, 7, 2, 3
10. 27, 36, 28, 35, 43, 26, 17, 45, 37, 44, 21, 29, 53, 42, 41, 38, 18, 52, 47, 34, 33, 46, 39, 19, 61, 30, 20, 51, 50, 25, 16, 13, 5, 22, 31, 12, 3, 54, 63, 55, 11, 10, 2, 24, 32, 9, 0, 8, 1, 59, 58, 57, 49, 56, 48, 60, 62, 40, 4, 15, 23, 14, 7, 6
11. 27, 36, 28, 35, 20, 37, 43, 34, 29, 21, 38, 26, 19, 42, 22, 30, 25, 45, 41, 44, 31, 18, 10, 33, 24, 17, 16, 52, 60, 53, 61, 50, 59, 11, 2, 12, 5, 58, 57, 51, 13, 3, 4, 9, 49, 8, 0, 1, 56, 48, 62, 54, 63, 55, 47, 46, 39, 40, 32, 23, 14, 7, 15, 6

12. 27, 36, 28, 35, 20, 37, 45, 19, 18, 44, 29, 21, 22, 26, 42, 34, 52, 11, 2, 10, 41, 30, 23, 43, 51, 33, 40, 38, 31, 50, 58, 46, 53, 25, 47, 32, 24, 3, 4, 49, 17, 61, 56, 57, 60, 12, 5, 48, 59, 9, 0, 54, 13, 39, 63, 1, 8, 16, 55, 62, 14, 7, 15, 6
13. 27, 36, 28, 35, 43, 26, 17, 45, 37, 44, 51, 52, 61, 29, 53, 34, 38, 18, 25, 42, 19, 50, 58, 46, 47, 54, 63, 55, 21, 30, 39, 10, 12, 41, 40, 33, 32, 60, 59, 22, 20, 24, 2, 48, 16, 8, 14, 62, 9, 13, 49, 0, 4, 1, 11, 3, 5, 7, 6, 15, 23, 31, 56, 57
14. 27, 36, 28, 35, 20, 37, 46, 18, 34, 45, 26, 29, 10, 19, 38, 43, 52, 25, 11, 44, 16, 42, 41, 12, 17, 30, 31, 33, 24, 21, 22, 13, 5, 51, 58, 50, 49, 53, 60, 56, 57, 59, 61, 48, 40, 32, 47, 39, 23, 54, 8, 9, 4, 2, 3, 1, 0, 14, 15, 7, 63, 6, 55, 62
15. 27, 36, 28, 35, 29, 19, 34, 43, 20, 21, 13, 26, 25, 37, 45, 44, 18, 17, 52, 42, 50, 51, 16, 41, 40, 33, 58, 53, 61, 38, 24, 46, 47, 49, 56, 57, 30, 48, 11, 3, 22, 23, 12, 4, 32, 39, 31, 10, 2, 6, 5, 1, 9, 0, 8, 60, 59, 54, 63, 62, 55, 14, 7, 15
16. 27, 36, 28, 35, 43, 26, 21, 29, 37, 45, 53, 34, 42, 44, 18, 20, 19, 51, 60, 22, 52, 50, 58, 13, 5, 12, 23, 30, 11, 38, 47, 46, 39, 31, 54, 10, 2, 59, 61, 63, 55, 25, 32, 17, 16, 24, 33, 9, 0, 1, 3, 62, 57, 49, 56, 48, 40, 41, 4, 14, 7, 15, 6, 8
17. 27, 36, 28, 35, 20, 37, 42, 34, 41, 21, 22, 18, 38, 43, 44, 13, 5, 33, 26, 19, 24, 51, 50, 45, 17, 29, 46, 30, 31, 53, 61, 12, 60, 10, 2, 23, 11, 39, 47, 52, 15, 58, 14, 25, 16, 7, 59, 6, 57, 55, 4, 3, 54, 1, 9, 32, 40, 0, 8, 48, 49, 56, 62, 63
18. 27, 36, 28, 35, 29, 19, 26, 37, 44, 18, 21, 20, 12, 13, 22, 30, 34, 45, 23, 38, 4, 11, 53, 46, 47, 43, 2, 52, 10, 41, 42, 51, 40, 33, 25, 17, 16, 3, 61, 50, 60, 31, 39, 59, 58, 49, 56, 57, 5, 6, 48, 14, 7, 15, 55, 54, 1, 9, 32, 24, 62, 63, 0, 8
19. 27, 36, 28, 35, 20, 37, 44, 19, 38, 21, 18, 26, 22, 11, 34, 29, 12, 25, 17, 10, 2, 13, 16, 4, 30, 3, 5, 43, 42, 23, 31, 39, 14, 47, 46, 52, 60, 51, 59, 15, 45, 50, 58, 33, 32, 24, 55, 54, 63, 53, 7, 6, 1, 9, 8, 0, 40, 41, 62, 61, 57, 49, 48, 56
20. 27, 36, 28, 35, 34, 44, 29, 20, 43, 42, 11, 26, 25, 18, 41, 19, 17, 33, 40, 10, 45, 13, 21, 4, 2, 53, 5, 3, 61, 12, 37, 16, 51, 24, 9, 30, 31, 22, 23, 59, 60, 0, 1, 52, 58, 50, 14, 7, 6, 38, 47, 39, 46, 8, 57, 15, 55, 54, 32, 48, 49, 56, 63, 62
21. 27, 36, 28, 35, 43, 26, 21, 29, 20, 44, 37, 45, 34, 18, 53, 42, 19, 50, 33, 52, 17, 46, 58, 38, 47, 51, 59, 22, 23, 25, 16, 39, 31, 54, 63, 62, 30, 13, 61, 55, 5, 41, 40, 10, 2, 60, 12, 24, 32, 11, 3, 9, 49, 56, 0, 8, 48, 1, 57, 4, 14, 6, 7, 15
22. 27, 36, 28, 35, 29, 19, 10, 45, 43, 18, 17, 34, 44, 37, 53, 50, 42, 51, 60, 59, 58, 52, 26, 33, 40, 41, 32, 46, 47, 20, 61, 49, 21, 38, 56, 57, 30, 48, 25, 2, 11, 22, 23, 39, 31, 3, 12, 4, 5, 6, 1, 0, 13, 14, 7, 15, 55, 54, 63, 62, 9, 8, 16, 24
23. 27, 36, 28, 35, 34, 44, 45, 26, 43, 37, 18, 19, 11, 42, 41, 52, 59, 53, 38, 29, 46, 50, 58, 20, 22, 21, 60, 51, 12, 33, 13, 17, 16, 40, 24, 32, 48, 10, 25, 3, 4, 5, 2, 1, 9, 6, 14, 7, 15, 23, 31, 30, 39, 47, 54, 55, 8, 0, 57, 56, 49, 63, 62, 61

24. 27, 36, 28, 35, 29, 19, 42, 43, 10, 45, 44, 37, 46, 21, 34, 30, 22, 53, 38, 52, 61, 26, 18, 51, 13, 39, 58, 20, 50, 31, 23, 15, 12, 11, 4, 2, 3, 25, 17, 33, 16, 41, 47, 55, 24, 5, 40, 49, 14, 1, 56, 57, 59, 32, 54, 6, 9, 60, 62, 7, 63, 0, 8, 48
25. 27, 36, 28, 35, 29, 19, 42, 43, 10, 45, 44, 37, 46, 21, 20, 38, 22, 30, 23, 31, 18, 26, 17, 34, 47, 39, 25, 41, 40, 11, 2, 33, 32, 12, 5, 13, 6, 16, 24, 48, 14, 50, 58, 51, 59, 52, 9, 8, 49, 60, 61, 56, 57, 62, 53, 3, 4, 0, 1, 7, 15, 55, 63, 54
26. 27, 36, 28, 35, 20, 37, 42, 34, 41, 21, 22, 18, 44, 29, 26, 43, 38, 45, 10, 46, 47, 30, 23, 33, 32, 11, 2, 12, 13, 4, 3, 1, 19, 51, 58, 5, 50, 25, 24, 52, 61, 17, 16, 9, 0, 8, 6, 54, 53, 31, 39, 55, 63, 59, 60, 62, 48, 40, 49, 56, 57, 14, 7, 15
27. 27, 36, 28, 35, 29, 19, 42, 43, 50, 21, 11, 10, 13, 26, 17, 3, 18, 25, 20, 44, 2, 1, 45, 4, 34, 51, 58, 33, 32, 37, 30, 12, 5, 6, 52, 16, 24, 40, 41, 8, 9, 48, 49, 0, 14, 56, 57, 59, 60, 61, 53, 62, 54, 38, 39, 63, 55, 47, 46, 7, 15, 23, 31, 22
28. 27, 36, 28, 35, 43, 26, 19, 44, 45, 37, 33, 18, 17, 52, 29, 34, 59, 61, 42, 20, 13, 50, 58, 12, 21, 41, 40, 30, 22, 51, 23, 25, 32, 10, 2, 11, 5, 53, 60, 57, 3, 4, 54, 62, 49, 24, 16, 56, 48, 8, 9, 63, 38, 46, 47, 55, 39, 31, 15, 14, 7, 6, 0, 1
29. 27, 36, 28, 35, 34, 44, 37, 26, 52, 42, 18, 19, 50, 25, 16, 17, 10, 20, 21, 11, 2, 33, 43, 29, 40, 41, 48, 24, 32, 9, 0, 8, 22, 56, 49, 57, 58, 51, 1, 13, 5, 45, 12, 59, 53, 30, 38, 46, 39, 47, 60, 61, 54, 31, 23, 15, 62, 63, 55, 6, 7, 14, 4, 3
30. 27, 36, 28, 35, 20, 37, 42, 34, 41, 21, 38, 46, 29, 19, 45, 44, 13, 12, 51, 22, 47, 18, 11, 30, 39, 5, 23, 52, 17, 58, 4, 3, 53, 61, 43, 26, 59, 60, 10, 31, 33, 2, 6, 7, 15, 14, 54, 16, 24, 63, 55, 62, 8, 25, 50, 57, 49, 56, 48, 1, 9, 0, 32, 40
31. 27, 36, 28, 35, 20, 37, 46, 18, 26, 19, 10, 11, 2, 25, 44, 45, 53, 17, 16, 21, 22, 52, 60, 29, 38, 30, 39, 43, 32, 12, 50, 23, 13, 31, 42, 47, 14, 15, 33, 34, 41, 7, 6, 55, 54, 63, 62, 61, 59, 51, 58, 57, 49, 56, 48, 5, 4, 3, 1, 9, 0, 8, 24, 40

GBG spielt Schwarz und Ludii (Version 1.0.2) spielt Weiß

Siege des Ludii-Agenten

1. 27, 36, 28, 35, 34, 26, 19, 44, 18, 10, 43, 42, 41, 40, 37, 20, 21, 38, 25, 17, 11, 2, 33, 32, 13, 22, 29, 5, 12, 4, 30, 39, 23, 31, 47, 46, 15, 14, 3, 7, 6, 16, 24, 48, 9, 55, 45, 50, 58, 53, 61, 51, 52, 59, 60, 0, 8, 1, 56, 54, 62, 63, 57, 49
2. 27, 36, 28, 35, 29, 37, 44, 19, 45, 30, 21, 12, 20, 52, 13, 38, 18, 26, 39, 5, 22, 23, 3, 4, 6, 31, 15, 10, 42, 43, 60, 51, 50, 41, 53, 61, 62, 58, 46, 55, 25, 32, 11, 2, 1, 33, 17, 34, 40, 24, 16, 14, 47, 7, 57, 0, 59, 8, 9, 54, 63, 48, 49, 56
3. 27, 36, 28, 35, 34, 26, 19, 42, 20, 13, 21, 12, 18, 30, 43, 10, 33, 44, 17, 16, 50, 58, 25, 24, 51, 59
4. 27, 36, 28, 35, 20, 19, 26, 17, 25, 37, 18, 24, 34, 12, 21, 11, 29, 22, 42, 50, 45, 43, 44, 53, 52, 60

5. 27, 36, 28, 35, 20, 19, 26, 37, 18, 12, 21, 25, 34, 33, 29, 22, 11, 2, 13, 10, 17, 5, 32, 16, 45, 53, 43, 42, 41, 52, 44, 51, 24, 40, 30, 23, 38, 47, 50, 46, 14, 58, 49, 6, 7, 15, 31, 39, 4, 56, 55, 9, 61, 54, 60, 48, 0, 8, 1, 3, 57, 59, 63, 62
6. 27, 36, 28, 35, 34, 44, 29, 18, 26, 22, 10, 25, 19, 33, 42, 43, 41, 17, 21, 2, 30, 40, 45, 46, 53, 61, 47, 11, 20, 38, 37, 13, 23, 39, 31, 52, 12, 14, 3, 5, 1, 4, 6, 7, 15, 0, 51, 55, 50, 58, 49, 9, 24, 32, 48, 56, 57, 16, 8, 59, 60, 54, 63, 62
7. 27, 36, 28, 35, 43, 42, 34, 44, 51, 18, 45, 46, 52, 59, 50, 41
8. 27, 36, 28, 35, 43, 26, 20, 51, 50, 45, 29, 21, 13, 30, 37, 46, 18, 10, 34, 5, 38, 41, 59, 52, 23, 47, 53, 44, 22, 39, 19, 61, 60, 31, 55, 58, 42, 33, 25, 11, 17, 12, 40, 57, 32, 49, 2, 3, 15, 14, 4, 1, 6, 7, 62, 63, 48, 56, 9, 0, 8, 16, 24, 54
9. 27, 36, 28, 35, 20, 21, 29, 37, 30, 18, 45, 22, 12, 13, 5, 31, 19, 38, 26, 10, 11, 2, 42, 34, 41, 53, 44, 3, 43, 40, 46, 47, 33, 24, 17, 16, 32, 25, 52, 4, 1, 61, 51, 60, 58, 50, 39, 14, 54, 9, 0, 8, 55, 23, 57, 59, 49, 56, 48, 63, 7, 62, 6, 15
10. 27, 36, 28, 35, 20, 21, 29, 19, 12, 5, 13, 37, 18, 17
11. 27, 36, 28, 35, 34, 42, 43, 44, 51, 18, 45, 46, 52, 50, 53, 61, 41, 33, 26, 40, 58, 59, 19, 20, 29, 17, 13, 12, 21, 38, 30, 31, 47, 39, 23, 37, 25, 32, 55, 14, 11, 3, 10, 2, 16, 24, 48, 9, 60, 22, 62, 49, 8, 54, 56, 57, 7, 63, 4, 15, 0, 1, 5, 6
12. 27, 36, 28, 35, 29, 19, 34, 45, 37, 21, 46, 47, 53, 42, 50, 61, 54, 63, 20, 33, 44, 11, 43, 41, 52, 58, 51, 18, 26, 59, 10, 2, 12, 3, 25, 32, 17, 13, 24, 16, 60, 62, 55, 30, 9, 38, 40, 0, 39, 49, 31, 48, 8, 22, 23, 15, 56, 57, 1, 14, 7, 6, 5, 4
13. 27, 36, 28, 35, 34, 26, 19, 42, 20, 21, 29, 18, 12, 45, 13, 22, 37, 5, 30, 31
14. 27, 36, 28, 35, 20, 21, 29, 37, 30, 18, 45, 31, 19, 38, 22, 23, 42, 53, 26, 33, 44, 34, 25, 52, 43, 50, 17, 16, 41, 40, 46, 47, 51, 58, 61, 59, 49, 56, 39, 57, 48, 32, 24, 60, 62, 12, 54, 11, 10, 63, 55, 3, 5, 4, 2, 13, 6, 9, 0, 1, 8, 7, 14, 15
15. 27, 36, 28, 35, 29, 19, 34, 37, 21, 13, 26, 42, 43, 51, 44, 18, 20, 45, 52, 61, 10, 2, 50, 53, 30, 58, 12, 22, 46, 5, 41, 33, 11, 38, 17, 16, 25, 47, 54, 40, 23, 3, 32, 63, 55, 31, 62, 14, 4, 24, 39, 15, 60, 59, 7, 6, 49, 56, 48, 57, 9, 0, 1, 8
16. 27, 36, 28, 35, 34, 44, 29, 18, 26, 42, 17, 16, 10, 2, 9, 0, 19, 30, 37, 46, 38, 39
17. 27, 36, 28, 35, 29, 19, 34, 45, 37, 46, 21, 13, 18, 43, 44, 25, 20, 17, 26, 51, 42, 33, 41, 40, 50, 58, 12, 4, 30, 31
18. 27, 36, 28, 35, 20, 19, 26, 33, 25, 21, 34, 42, 43, 17, 41, 44, 18, 40, 12, 3, 53, 45, 13, 5, 24, 10, 11, 16, 32, 61, 4, 9, 29, 37, 0, 2, 52, 50, 51, 58, 8, 1, 48, 59, 60, 49, 46, 38, 39, 30, 31, 22, 56, 57, 23, 47, 55, 15, 7, 54, 63, 62, 6, 14
19. 27, 36, 28, 35, 34, 42, 43, 26, 33, 40, 41, 44, 18, 10
20. 27, 36, 28, 35, 34, 42, 43, 44, 51, 58, 50, 26, 45, 46

21. 27, 36, 28, 35, 43, 44, 37, 26, 45, 51, 42, 46, 34, 33, 41, 40

Siege des GBG-Agenten

1. 27, 36, 28, 35, 34, 26, 19, 12, 11, 44, 18, 3, 20, 10, 43, 25, 45, 33, 17, 42, 32, 46, 29, 37, 40, 16, 51, 58, 60, 13, 21, 59, 57, 38, 24, 30, 8, 41, 2, 9, 22, 31, 0, 1, 4, 14, 49, 56, 52, 48, 50, 61, 7, 6, 5, 53, 15, 47, 39, 23, 54, 63, 62, 55
2. 27, 36, 28, 35, 29, 19, 34, 45, 37, 41, 53, 30, 21, 44, 43, 52, 60, 12, 42, 61, 62, 51, 18, 17, 26, 33, 13, 5, 11, 20, 38, 10, 59, 46, 22, 23, 4, 3, 58, 50, 57, 56, 54, 25, 24, 49, 48, 40, 32, 16, 9, 8, 31, 47, 39, 55, 63, 0, 1, 2, 6, 14, 7, 15
3. 27, 36, 28, 35, 29, 19, 34, 37, 21, 41, 45, 46, 20, 44, 52, 51, 53, 43, 38, 22, 30, 47, 13, 59, 12, 23, 18, 5, 42, 10, 11, 33, 60, 2, 3, 4, 58, 61, 62, 50, 25, 17, 16, 26, 40, 54, 63, 24, 57, 8, 31, 39, 14, 55, 15, 49, 48, 6, 7, 1, 32, 56, 9
4. 27, 36, 28, 35, 43, 42, 34, 26, 33, 45, 18, 10, 25, 41, 17, 16, 50, 51, 44, 53, 19, 59, 40, 20, 37, 46, 29, 52, 38, 32, 61, 31, 47, 39, 23, 22, 30, 21, 2, 11, 15, 62, 63, 3, 4, 55, 54, 7, 1, 13, 60, 58, 57, 12, 9, 49, 56, 48, 24, 8, 0, 5, 6, 14
5. 27, 36, 28, 35, 34, 42, 43, 26, 33, 24, 18, 10, 41, 51, 50, 37, 25, 17, 19, 20, 44, 52, 11, 40, 32, 16, 13, 21, 12, 2, 4, 5, 61, 60, 59, 3, 29, 38, 22, 23, 46, 30, 39, 45, 31, 47, 53, 58, 57, 49, 56, 48, 9, 0, 8, 62, 63, 1, 55, 54, 15, 14, 6, 7
6. 27, 36, 28, 35, 29, 37, 44, 21, 43, 42, 34, 45, 51, 18, 50, 58, 38, 41, 26, 59, 22, 30, 19, 47, 10, 33, 20, 17, 49, 23, 32, 56, 25, 2, 11, 16, 24, 3, 31, 39, 12, 13, 46, 40, 52, 54, 63, 9, 0, 1, 4, 53, 55, 5, 15, 62, 61, 60, 8, 14, 7, 6, 48, 57
7. 27, 36, 28, 35, 29, 19, 34, 45, 37, 21, 46, 42, 44, 52, 43, 33, 41, 53, 26, 51, 50, 40, 20, 39, 30, 58, 22, 12, 13, 5, 31, 23, 14, 7, 15, 17, 38, 18, 25, 16, 61, 60, 6, 54, 63, 62, 59, 4, 3, 11, 10, 9, 0, 2, 1, 47, 55, 57, 49, 56, 48, 32, 24, 8
8. 27, 36, 28, 35, 43, 44, 37, 46, 38, 42, 29, 22, 45, 47, 21, 20, 19, 53, 12, 18, 52, 5, 50, 58, 31, 39, 55, 59, 34, 51, 26, 41, 33, 32, 10, 17, 25, 2, 61, 60, 57, 24, 4, 11, 3, 49, 56, 48, 40, 13, 16, 9, 62, 54, 23, 30, 14, 6, 15, 8, 63, 7, 0, 1
9. 27, 36, 28, 35, 34, 26, 19, 12, 11, 42, 20, 13, 21, 44, 29, 37, 43, 10, 18, 25, 45, 53, 52, 50, 51, 61, 60, 59, 5, 4, 3, 2, 1, 9, 0, 17, 6, 8, 16, 24, 32, 33, 41, 40, 48, 14, 7, 30, 22, 15, 23, 31, 39, 38, 57, 58, 62, 49, 56, 54, 63, 55, 47, 46
10. 27, 36, 28, 35, 29, 19, 34, 37, 21, 13, 26, 33, 45, 20, 18, 53, 12, 10, 22, 5, 25, 16, 17, 24, 3, 11, 46, 47, 44, 23, 42, 50, 61, 60, 59, 30, 41, 58, 57, 52, 51, 43, 32, 40, 38, 54, 4, 2, 63, 62, 49, 56, 55, 39, 31, 15, 14, 48, 7, 6, 1, 9, 0, 8
11. 27, 36, 28, 35, 34, 44, 29, 18, 26, 17, 42, 33, 45, 51, 25, 16, 43, 21, 20, 46, 52, 22, 58, 11, 10, 60, 4, 19, 50, 41, 59, 2, 61, 12, 53, 37, 40, 49, 38, 39, 30, 31, 13, 6, 56, 54, 24, 32, 8, 9, 63, 57, 48, 62, 0, 1, 3, 5, 7, 14, 15, 23, 47, 55
12. 27, 36, 28, 35, 29, 37, 44, 21, 43, 26, 18, 10, 38, 52, 45, 31, 46, 53, 34, 47, 61, 33, 20, 42, 51, 59, 30, 39, 22, 60, 58, 19, 11, 23, 41, 4, 17, 16, 40, 32, 24, 50, 25, 48, 3, 2, 12, 5, 13, 6, 62, 14, 57, 49, 8, 0, 56, 54, 63, 55, 15, 9, 1, 7

13. 27, 36, 28, 35, 43, 44, 37, 42, 29, 21, 20, 38, 22, 19, 18, 13, 46, 23, 5, 4, 3, 45, 12, 47, 30, 31, 52, 10, 39, 26, 50, 51, 34, 58, 11, 59, 53, 2, 1, 62, 6, 14, 17, 54, 63, 55, 33, 16, 41, 32, 7, 15, 25, 24, 60, 61, 57, 9, 40, 49, 56, 48, 8, 0
14. 27, 36, 28, 35, 20, 21, 29, 37, 30, 39, 45, 44, 38, 53, 31, 23, 46, 47, 42, 12, 43, 34, 13, 41, 3, 5, 52, 59, 22, 19, 11, 10, 2, 51, 61, 60, 58, 50, 62, 54, 33, 32, 26, 25, 16, 18, 4, 49, 6, 14, 57, 24, 40, 7, 15, 55, 63, 56, 48, 17, 8, 0, 9, 1
15. 27, 36, 28, 35, 34, 42, 43, 26, 33, 45, 18, 25, 19, 52, 41, 40, 21, 10, 44, 11, 51, 50, 17, 16, 2, 53, 58, 3, 4, 24, 20, 13, 29, 59, 60, 22, 30, 61, 62, 39, 37, 12, 1, 38, 46, 47, 5, 54, 57, 49, 56, 14, 63, 55, 31, 15, 48, 32, 8, 23, 6, 9, 7, 0
16. 27, 36, 28, 35, 29, 37, 44, 51, 52, 19, 45, 38, 43, 59, 53, 34, 20, 22, 21, 46, 60, 61, 30, 23, 47, 31, 26, 18, 10, 2, 17, 16, 25, 33, 13, 5, 11, 3, 14, 7, 6, 4, 41, 12, 32, 49, 42, 50, 24, 40, 58, 39, 62, 57, 56, 48, 8, 55, 15, 9, 63, 54, 1, 0
17. 27, 36, 28, 35, 34, 42, 43, 44, 51, 18, 45, 52, 37, 25, 50, 58, 21, 29, 20, 13, 41, 53, 19, 40, 33, 59, 16, 10, 12, 5, 17, 24, 30, 39, 22, 23, 11, 26, 4, 2, 6, 8, 32, 9, 0, 1, 3, 46, 47, 38, 31, 48, 49, 57, 56, 60, 61, 54, 55, 63, 62, 15, 14, 7
18. 27, 36, 28, 35, 29, 19, 34, 45, 37, 21, 46, 47, 53, 61, 54, 63, 44, 33, 26, 18, 38, 30, 17, 25, 62, 52, 20, 16, 13, 22, 12, 11, 41, 42, 43, 5, 10, 50, 40, 2, 51, 59, 24, 32, 8, 9, 0, 1, 3, 4, 6, 14, 7, 15, 23, 39, 31, 49, 48, 56, 57, 58, 60, 55
19. 27, 36, 28, 35, 20, 19, 26, 21, 34, 33, 25, 41, 42, 17, 18, 44, 43, 10, 37, 30, 45, 53, 29, 52, 38, 50, 23, 47, 22, 31, 39, 15, 32, 46, 51, 40, 16, 59, 24, 49, 48, 11, 12, 2, 3, 4, 5, 6, 13, 56, 57, 8, 9, 1, 0, 58, 60, 61, 62, 54, 63, 55, 7, 14
20. 27, 36, 28, 35, 29, 37, 44, 21, 43, 26, 18, 52, 34, 33, 42, 41, 51, 19, 20, 17, 50, 58, 25, 16, 45, 38, 53, 61, 46, 47, 40, 32, 54, 63, 11, 4, 10, 3, 13, 12, 2, 5, 6, 48, 22, 9, 39, 31, 55, 14, 7, 30, 0, 8, 1, 23, 15, 24, 56, 62, 60, 59, 49, 57
21. 27, 36, 28, 35, 34, 26, 19, 44, 18, 11, 20, 33, 42, 29, 43, 41, 21, 13, 12, 5, 10, 2, 37, 45, 52, 59, 50, 58, 61, 53, 46, 47, 51, 38, 30, 22, 39, 31, 25, 60, 57, 24, 17, 49, 62, 16, 56, 48, 4, 3, 40, 32, 8, 9, 1, 0, 15, 23, 55, 54, 63, 14, 7, 6
22. 27, 36, 28, 35, 29, 19, 34, 41, 33, 30, 21, 13, 26, 43, 42, 25, 37, 45, 44, 52, 38, 51, 46, 47, 53, 61, 50, 20, 22, 23, 40, 58, 18, 10, 16, 32, 24, 17, 12, 3, 11, 2, 54, 55, 48, 49, 56, 57, 59, 9, 4, 5, 63, 62, 60, 39, 31, 15, 14, 6, 7, 0, 1, 8
23. 27, 36, 28, 35, 34, 26, 19, 42, 20, 21, 29, 12, 18, 37, 22, 30, 41, 23, 33, 24, 32, 40, 10, 13, 5, 2, 17, 16, 25, 11, 6, 4, 14, 7, 43, 50, 44, 45, 51, 58, 52, 59, 61, 60, 57, 49, 56, 48, 8, 9, 53, 62, 63, 31, 38, 0, 1, 3, 39, 15, 54, 46, 55, 47
24. 27, 36, 28, 35, 29, 19, 34, 43, 42, 25, 51, 59, 50, 21, 20, 58, 13, 30, 37, 12, 26, 5, 44, 45, 33, 40, 18, 41, 24, 10, 17, 32, 48, 9, 38, 39, 52, 61, 22, 15, 46, 47, 53, 60, 0, 1, 2, 11, 23, 31, 4, 3, 6, 16, 8, 49, 56, 57, 54, 62, 63, 55, 7, 14

25. 27, 36, 28, 35, 43, 26, 20, 51, 50, 21, 22, 12, 34, 23, 19, 10, 13, 4, 29, 37, 30, 42, 44, 58, 33, 45, 38, 46, 18, 11, 2, 24, 40, 32, 16, 25, 39, 47, 55, 54, 63, 52, 53, 61, 41, 62, 60, 59, 57, 49, 31, 17, 8, 9, 0, 1, 3, 5, 15, 14, 6, 7, 48, 56
26. 27, 36, 28, 35, 34, 26, 19, 12, 11, 42, 20, 29, 37, 43, 21, 30, 22, 46, 45, 44, 5, 13, 39, 23, 52, 2, 18, 4, 38, 10, 50, 47, 31, 61, 51, 58, 60, 59, 53, 62, 54, 6, 3, 63, 55, 14, 7, 15, 33, 25, 41, 40, 1, 32, 49, 56, 17, 16, 24, 8, 0, 9, 48
27. 27, 36, 28, 35, 34, 42, 43, 26, 33, 41, 25, 24, 50, 58, 21, 52, 51, 29, 53, 13, 37, 61, 32, 59, 60, 40, 44, 38, 20, 19, 30, 39, 22, 23, 31, 45, 12, 3, 18, 17, 47, 55, 46, 54, 5, 4, 2, 14, 11, 10, 1, 9, 7, 6, 8, 0, 15, 16, 63, 62, 57, 49, 56, 48
28. 27, 36, 28, 35, 29, 21, 20, 19, 12, 3, 18, 17, 13, 22, 4, 5, 34, 45, 26, 41, 37, 30, 42, 25, 53, 44, 43, 46, 38, 47, 23, 61, 39, 31, 55, 33, 32, 40, 48, 54, 63, 52, 62, 51, 60, 59, 10, 2, 11, 16, 58, 50, 15, 14, 7, 6, 1, 49, 57, 9, 56, 0, 8, 24

Untentschiedene Partien

1. 27, 36, 28, 35, 43, 26, 20, 45, 44, 42, 53, 61, 46, 37, 38, 29, 52, 47, 51, 58, 50, 11, 34, 60, 30, 31, 39, 59, 17, 18, 41, 40, 33, 24, 32, 16, 22, 23, 21, 19, 25, 13, 10, 2, 12, 4, 3, 5, 54, 62, 63, 55, 57, 49, 14, 7, 56, 48, 9, 15, 6, 8, 0, 1

Vollständiger Quellcode

Der gesamte entwickelte Quellcode ist in dem GitHub-Repository
<https://github.com/WolfgangKonen/GBG> einsehbar.