

Technische Universität Berlin
Fakultät Elektrotechnik und Informatik
Institut für Softwaretechnik und Theoretische Informatik
Fachgebiet Künstliche Intelligenz

Proposal zur Bachelorarbeit

Kamisado - Brettspiel in Java

**Strategisches 2-Personen Nullsummenspiel mit perfekter
Information**

Murat Akgül
Matrikelnr.: 221131
E-Mail: osbornx22@gmail.com

5. Januar 2012

Betreut von Dr.- Ing. Stefan Fricke
E-Mail: stefan.fricke@dai-labor.de

Zusammenfassung

Kamisado ist ein “Schach-artiges” Brettspiel. Bei dieser Arbeit geht es um die Umsetzung als Computerspiel, welches in Java programmiert wird. Es werden entweder zwei Personen an einem Computer gegeneinander oder ein Spieler gegen den Computer spielen können. Dabei geht es hauptsächlich um die künstliche Intelligenz. Da es sich um ein strategisches 2-Personen Nullsummenspiel handelt, wird mittels Alpha-Beta-Suche der beste Zug ermittelt und ausgeführt. Außerdem wird durch iterative Tiefensuche gewährleistet, dass die Berechnungen in angemessener Zeit abgeschlossen werden.

Inhaltsverzeichnis

1	Einleitung	4
1.1	Zielstellung	4
1.2	Wissenschaftliche Fragestellung	4
2	Kamisado	5
2.1	Aufbau	5
2.2	Grundregeln der Einzelpartie	6
2.3	Ziel und Sieg	6
3	Grundbegriffe der Spieltheorie	8
3.1	Nullsummenspiel	8
3.2	Perfekte Information	8
4	Algorithmen der künstliche Intelligenz	9
4.1	Minimax-Algorithmus	9
4.2	Alpha-Beta-Suche	10
5	Aufgabenstellung und Lösungsansatz	11
5.1	Anforderung aus der Aufgabenstellung	11
5.2	Vorgehensmodell zur Entwicklung	12
5.3	Klassendiagramm zum Gesamtprojekt	12
5.4	Aktivitätsdiagramm zu Ausführung eines Zuges	13
5.5	Überlegungen zur Bewertungsfunktion	14
5.6	Aufbau der grafischen Benutzeroberfläche	15
5.7	schriftliche Ausarbeitung	15
5.8	Auswahl der Werkzeuge	16
6	Organisatorisches	17
6.1	Zeitplan	17
6.2	Eidesstattliche Versicherung	17
	Literaturverzeichnis	18

1 Einleitung

In diesem Einführungskapitel werden die Zielstellung und die dazugehörige wissenschaftliche Fragestellung erläutert.

1.1 Zielstellung

Das Ziel dieser Arbeit ist die Umsetzung des Brettspiels Kamisado in ein selbstständig ausführbares Java-Programm. Ein wichtiger Teil dieser Arbeit ist die Implementierung der künstlichen Intelligenz, welche aus vorhandenen Entwicklungsmustern übernommen und in diesem konkreten Fall angepasst werden muss.

Bei der Implementierung der Alpha-Beta-Suche stellt die Bewertungsfunktion für Spielsituationen eine große Rolle und besondere Herausforderung dar, da diese für jedes Spiel speziell entwickelt werden muss und dazu ein großes Spektrum an Möglichkeiten zur Auswahl steht.

1.2 Wissenschaftliche Fragestellung

Aus wissenschaftlicher Sicht ist klarzustellen, ob die gewählten Algorithmen korrekt und optimal für Kamisado sind, also ob das Spiel die Voraussetzungen erfüllt, die zur Anwendung der ausgewählten Entwicklungsmuster nötig sind. Desweiteren muss klargestellt werden, wie zu überprüfen ist, ob diese Algorithmen korrekt umgesetzt und angepasst wurden oder ob es noch Verbesserungspotenziale gibt.

Im besonderen ist herauszustellen, ob die Komplexität der Bewertungsfunktion in akzeptablen Grenzen liegt und wieviel Verbesserungspotenzial theoretisch noch auszuschöpfen wäre. Dazu ist zu klären, ob die gewählten Datenstrukturen optimal gewählt und implementiert sind oder ob es noch andere, eventuell bessere Möglichkeiten gibt, da diese einen besonderen Einfluss auf die Komplexität der Bewertungsfunktion haben.

2 Kamisado

Kamisado ist ein abstraktes Brettspiel. Es wurde von im Jahre 2009 von Peter Burley entwickelt und vom Verlag HUCH & friends veröffentlicht. Kamisado ist ein strategisches 2-Personen-Spiel und eine normale Partie dauert 20 bis 30 Minuten. (Ledinger, 2010)



Abbildung 2.1: Der Aufbau zu Beginn des Spiels (Luibl, 2011).

2.1 Aufbau

Das Spielfeld besteht aus 8 mal 8 Feldern. Jedes Feld hat eine bestimmte Farbe, wobei es insgesamt nur 8 verschiedene Farben im Spiel gibt: Braun, Grün, Rot, Gelb, Rosa, Lila, Blau und Orange. Jeder Spieler hat 8 Figuren, die in der Grundfarbe weiß oder schwarz sind, allerdings zusätzlich mit einer der 8 Farben markiert sind. Die Figuren werden

wie in Abbildung 2.1 an der Grundlinie der Farbe entsprechend aufgestellt. (HUCH! & friends, 2010)

2.2 Grundregeln der Einzelpartie

Bei der ersten Partie fängt der schwarze Spieler an, in allen folgenden Partien der jeweilige verlierer der vorherigen Partie. Die Figuren werden nun abwechselnd und nacheinander bewegt. Es darf immer nur die Figur mit der Farbe bewegt werden, welche der Feldfarbe entspricht, die der Gegner mit seinem letzten Zug erreicht hat. Dabei sind nur diagonale oder gerade Bewegungen in die Richtung der gegnerischen Grundlinie gestattet. Seitliche- oder Rückwärts-Bewegungen sind nicht gestattet. Beim ersten Zug darf der Spieler der anfängt eine beliebige Farbe auswählen. (Burley Games, 2011)



Abbildung 2.2: Eine Beispielhafte Aufstellung im Verlauf eines Spiels (Burley Games, 2011).

2.3 Ziel und Sieg

Ziel des Spiels ist es mit einer eigenen Figur auf die gegnerische Grundlinie zu kommen. Dazu muss der Gegner vorher eine Bewegung auf ein Feld mit einer Farbe machen, sodass

danach eine Bewegung der entsprechenden eigenen Figur auf die gegnerische Grundlinie möglich ist. Es gibt noch weitere Varianten dieses Spiel, allerdings wird darauf in dieser Arbeit nicht eingegangen. (HUCH! & friends, 2010)

3 Grundbegriffe der Spieltheorie

Die Spieltheorie war ursprünglich ein Teilgebiet der Mathematik, allerdings haben sich die Anwendungsbereiche auf Informatik, Psychologie, Wirtschaft und Andere ausgeweitet. In der Spieltheorie geht “es um die angemessene Beschreibung und Modellierung von Entscheidungssituationen bzw. Spielen” (Wiese, 2001, S. 2).

3.1 Nullsummenspiel

Bei einem Nullsummenspiel im Sinne der Spieltheorie handelt es sich um ein Spiel, bei dem die Summe aller Einzahlungen der Summe aller Auszahlungen entspricht, also der Gewinn des einen Spielers ebenso der Verlust des anderen Spielers ist. (Berninghaus u. a., 2010, S. 406)

Diese Eigenschaft trifft für Kamisado zu, weil immer genau einer der beiden Spieler gewinnt (+1) und der andere Verliert (-1). Die Summe der Gewinne und Verluste ($+1 + -1 = 0$) beträgt also Null.

3.2 Perfekte Information

Ein Spiel mit perfekter Information im Sinne der Spieltheorie ist ein Spiel in dem es keinerlei zufällige Elemente gibt und jedem Spieler alle ausgeführten Züge bekannt sind. Es darf auch keine verdeckten Karten oder ähnliches geben. Jedes Spiel mit perfekten Informationen ist auch Spiel mit vollständigen Informationen, da diese lediglich verlangen, dass es keine unquantifizierbaren Zufallselemente gibt. (Rasmusen, 2007, S. 65)

Bei Kamisado gibt es keine Würfel oder andere Zufallselemente. Das Spielfeld ist fest bestimmt und jedem gänzlich bekannt. Jede Figur und Zugmöglichkeit inkl. Regeln sind ebenso beiden Spielern bekannt. Kamisado ist ein Spiel mit perfekter Information.

4 Algorithmen der künstliche Intelligenz

Zum Aufbau einer künstlichen Intelligenz wurden bereits bestimmte mathematisch korrekte Entwurfsmuster entwickelt, die je nach Anforderung und Situation auszuwählen sind. Da Kamisado ein Nullsummenspiel mit perfekter Information ist, werden die dafür in Frage kommen Entwurfsmuster vorgestellt.

4.1 Minimax-Algorithmus

Mit dem Minimax-Algorithmus lassen sich optimale Züge bzw. die optimale Strategie für bestimmte Spiele finden. Insbesondere für strategische 2-Personen Nullsummenspiele mit perfekten Informationen ist der Algorithmus optimal und vollständig, allerdings ist die Vollständigkeit in der Praxis nicht immer erreichbar. (Russel und Norvig, 2003, S. 161)

Bei diesem Algorithmus, welche einer Tiefensuche entspricht, wird stets der minimal zu erwartende Gewinn maximiert bzw. der maximal zu erwartende Verlust minimiert. Dazu wird ein Suchbaum aufgestellt, wobei die Ebenen abwechselnd einen Zug der ersten und dann des zweiten Spieler darstellen. Zu jedem Zug des Spielers (Elternknoten) werden die darauf möglichen gegnerischen Folgezüge generiert (Kinderknoten). Mithilfe einer Bewertungsfunktion kann beurteilt werden, wie gut oder schlecht ein jeweiliger Pfad ist und somit kann der beste Zug für jede Spielsituation ausgewählt werden. (Fricke, 2010)

Es sind also mindestens zwei Hilfsfunktionen nötig, einmal ein Zuggenerator, der regelgerechte Folgezüge generiert, um daraus mögliche Folgesituationen zu ermitteln. Solch ein Zuggenerator sollte relativ einfach zu implementieren sein. Allerdings ist dies bei der Bewertungsfunktion etwas komplizierter, da es verschiedene Möglichkeiten gibt, um einer Situation im Spiel einen numerischen Wert zuzuteilen, aus dem im Vergleich zu anderen Situationen ein Vorteil oder Nachteil zu ermitteln ist. Dazu wird in der Regel eine Linearkombination aus verschiedenen Eigenschaften der jeweiligen Spielsituation gebildet. Außerdem ist noch eine Funktion nötig, die Sieg, Remi oder Verlustsituationen erkennt. (Russel und Norvig, 2003, S. 162f)

Wie schon kurz erwähnt, ist die Vollständigkeit nicht immer realisierbar. Dies kann aus der Größe des Baumes resultieren, also aus einem hohen Verzweigungsgrad und/oder einer hohen Baumtiefe. Sei b der Verzweigungsfaktor und m die Baumtiefe, dann beträgt die Zeitkomplexität vom Minimax-Algorithmus $O(b^m)$. Der Zeitaufwand steigt also für komplexere Spiele sehr stark an, deswegen ist ein leicht erweiterter Algorithmus für solche Fälle besser geeignet, nämlich die Alpha-Beta-Suche. (Fricke, 2010; Shah, 2007)

4.2 Alpha-Beta-Suche

Der grundlegende Unterschied der Alpha-Beta-Suche zum Minimax-Algorithmus besteht darin, dass nun nicht der ganze Suchbaum aufgespannt wird, sondern nur diejenigen Pfade berechnet werden, die das Endergebnis überhaupt noch beeinflussen können. Bei einem MIN-Knoten wird dies auch Alpha-Cutoff genannt und bei einem MAX-Knoten Beta-Cutoff. Bei optimaler Reihenfolge der Züge, die zuerst berechnet werden, beträgt der Zeitaufwand nur noch $O(b^{m/2})$. (Fricke, 2010)

Mithilfe der Bewertungsfunktion können die zu berechnenden Folgesituationen vorsortiert werden. So können eventuell besonders vorteilhafte Züge vor den wahrscheinlich schlechteren Zügen berechnet werden, wodurch die schlechteren durch einen Cutoff weggelassen können, ohne viel Rechenzeit investiert zu haben. (Russel und Norvig, 2003, S. 169)

5 Aufgabenstellung und Lösungsansatz

Als Teile dieser Bachelorarbeit werden ein ausführbares Programm und eine schriftliche Ausarbeitung verlangt. Zum Abschluss wird diese Arbeit noch verteidigt werden müssen.

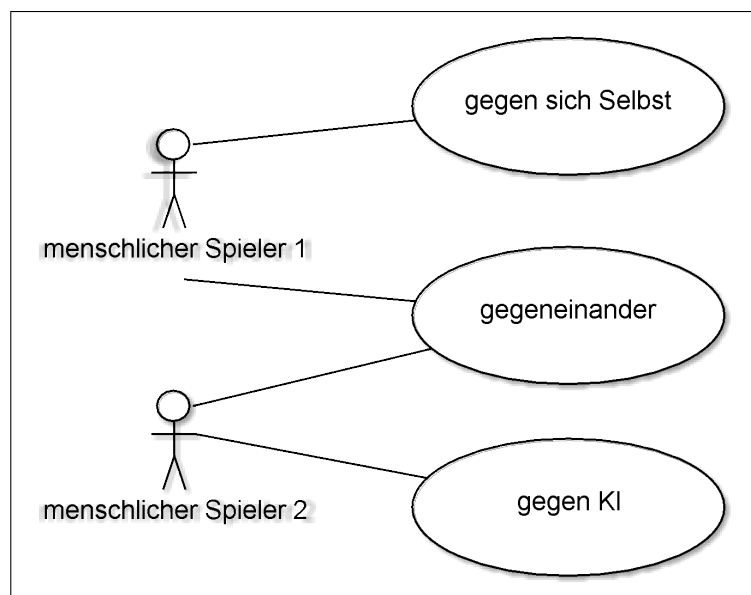


Abbildung 5.1: Ein Anwendungsfalldiagramm mit den Fällen, die umgesetzt werden müssen.

5.1 Anforderung aus der Aufgabenstellung

Bei dem ausführbaren Programm, handelt es sich um die Implementation von Kamisado. Dazu gehört zunächst eine grafische Benutzeroberfläche zur Darstellung des Spiels und der Interaktion mit dem Nutzer oder den Nutzern. Dabei soll es möglich sein, gegen sich Selbst, gegen einen anderen Spieler am gleichen Computer oder den Computer selbst zu spielen (siehe Abbildung 5.1). Der Computer bedient sich dabei einer künstlichen Intelligenz, dessen Strategie auf der Alpha-Beta-Suche mit iterativer Tiefensuche basiert. Es dürfen nur regelgerechte Züge ausgeführt werden und es wird automatisch der Sieger bzw. Verlierer erkannt.

5.2 Vorgehensmodell zur Entwicklung

Als ersten Schritt im Prozess zur Lösung dieser Aufgabe muss ein Vorgehensmodell aus der Softwareentwicklung ausgewählt werden. Hierfür ist aufgrund der charakteristischen Eigenschaften während der Entwicklung einer Bachelorarbeit das Spiralmodell gut geeignet. Dies ergibt sich daraus, dass eine Version gefertigt wird und entsprechend der Kritik des Betreuers eine Weiterentwicklung stattfindet. Wie in Abbildung 5.2 dargestellt, kann dieses Proposal als “Planung der Anforderungen” angesehen werden, darauf folgt ein “Entwicklungsplan”, der auch schon in diesem Proposal zumindest Teilweise stattfindet und danach folgen verschiedene “Entwurfs”-phasen mit “Testplanungen” und “Tests”, welche auch schon parallel zur Ausarbeitung dieses Proposals begonnen wurden. (Floyd, 2003)

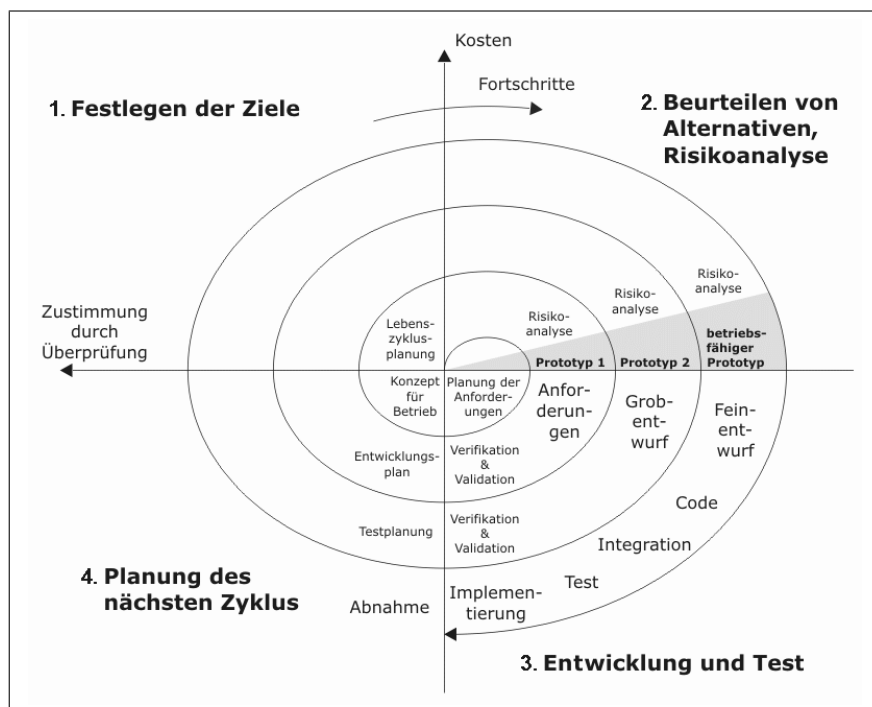


Abbildung 5.2: Die Phasen der Softwareentwicklung nach dem Spiralmodell. (Conny, 2004)

5.3 Klassendiagramm zum Gesamtprojekt

Da Java eine objektorientierte Programmiersprache ist, wurde zunächst eine intuitiv entsprechende Klassenmodellierung gewählt.

Die Objekte des Spiels sind schnell aufgezählt:

- Das Spielfeld besteht aus 64 Feldern.
- Es gibt 16 Figuren, die
 - einem Spieler gehören,
 - eine Grundfarbe und
 - eine Zusatzfarbe haben.

So lässt sich ganz einfach ein Klassendiagramm (siehe Abbildung 5.3) mit den entsprechenden Kardinalitäten, Assoziationen und Aggregationen aufbauen.

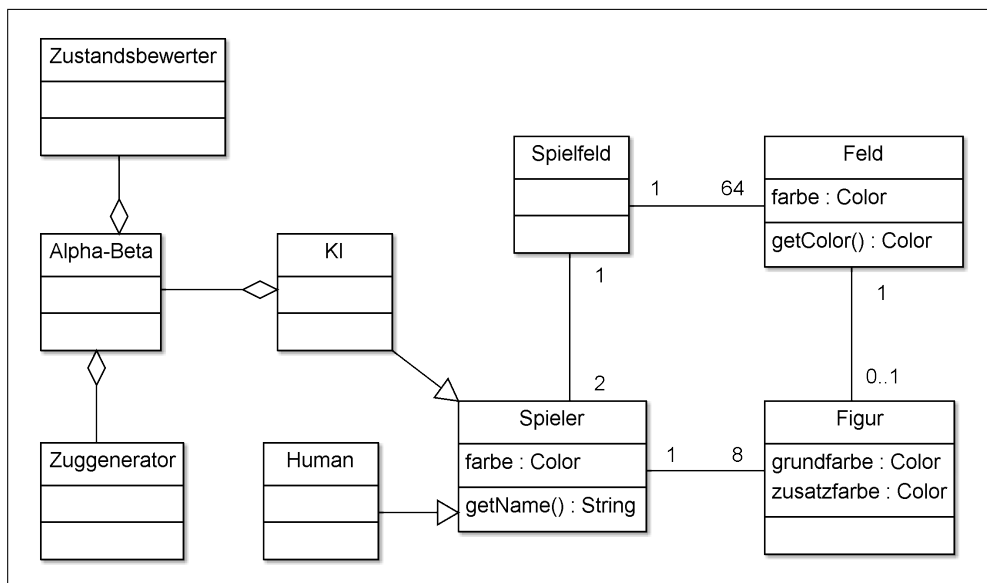


Abbildung 5.3: Ein Klassendiagramm zu Kamisado mit Objekten, Methoden und Attributen.

5.4 Aktivitätsdiagramm zu Ausführung eines Zuges

Um die Funktionsweise des Programms etwas besser verstehen zu können, ist in Abbildung 5.4 ein Aktivitätsdiagramm zur Ausführung eines Zuges im Laufe des Spiels dargestellt.

Abgesehen vom ersten Zug, ist in allen nachfolgenden Zügen eindeutig bestimmt, welche Figur der Spieler bewegen muss, da dies von der Zielfeldfarbe des vorherigen Zuges vom Gegner abhängt. Deswegen soll der Spieler einfach nur das “Zielfeld auswählen”. Falls das Zielfeld mit der jeweiligen Figur “regelkonform” erreichbar ist, dann wird die “Figur auf

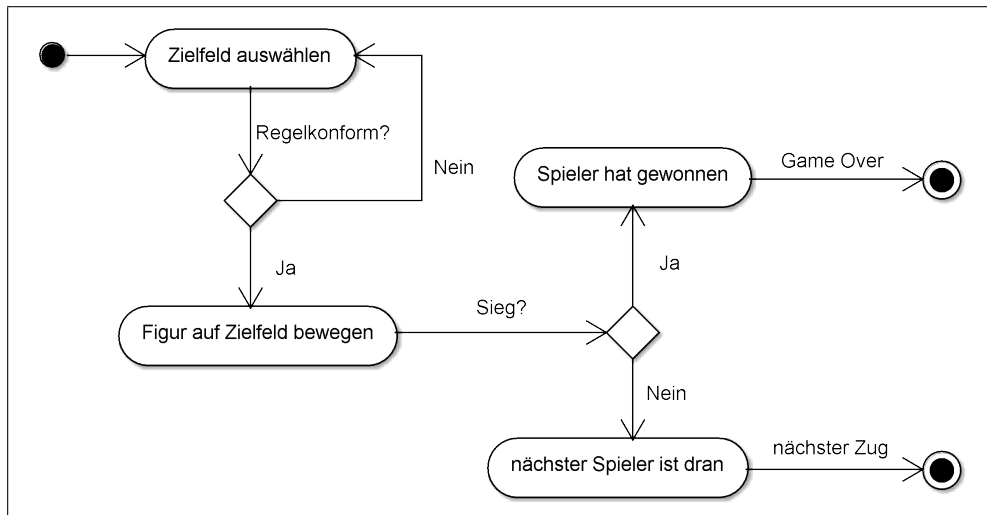


Abbildung 5.4: Aktivitätsdiagramm zur Ausführung eines Zuges von einem Spieler im Laufe eines Spiels.

das Zielfeld bewegt“, andernfalls muss der Spieler ein neues “Zielfeld auswählen“. Falls durch diesen Zug der “Spieler gewonnen hat“, dann ist das Spiel vorbei (Game Over), andernfalls ist der “nächste Spieler dran“. Diese Aktivität wird solange wiederholt bis ein Sieger feststeht (Game Over).

5.5 Überlegungen zur Bewertungsfunktion

Wie bereits erwähnt, ist die Bewertungsfunktion ein entscheidender Teil der Alpha-Beta-Suche und somit auch wichtig für die Stärke der künstlichen Intelligenz. Zur Ermittlung einer guten Bewertungsfunktion werden einige Versuche und Tests nötig sein. Dazu könnten auch verschiedene Bewertungsfunktion gegeneinander spielen, um deren jeweilige Stärke zu ermitteln.

Zunächst ist es wichtig zu wissen, welche Kriterien zur Aufstellung einer Bewertungsfunktion überhaupt zur Verfügung stehen, wobei zu unterscheiden ist, welcher Spieler gerade dran ist:

- Anzahl der eigenen, möglichen Zielfelder (möglichst hoch halten)
- Anzahl der gegnerischen, möglichen Zielfelder (möglichst gering halten)
- Summe der Entfernungen (Anzahl der Felder) von jeder eigenen Figur zur gegnerischen Grundlinie (möglichst niedrig halten)
- Summe der Entfernungen von jeder gegnerischen Figur zur eigenen Grundlinie (möglichst hoch halten)

Es bleibt zu überprüfen, ob eine Linearkombination aus diesen Eigenschaften zum schnellen Erfolg führen wird und welche der Kriterien wie stark zu werten sind, also wie die Koeffizienten zu wählen sind. Eventuell stellen sich noch im Verlaufe der Entwicklung weitere Bewertungskriterien heraus.

5.6 Aufbau der grafischen Benutzeroberfläche

Beim Aufbau des Graphical User Interface (GUI) wird versucht, möglichst getreu dem Konzept Model-View-Control (MVC, Modell-Präsentation-Steuerung) nachzugehen, da diese eine halbwegs überschaubare Implementierung ermöglichen soll.

Das Modell enthält dabei alle wichtigen Daten bzw. Datenstrukturen, also Spieler, Farben, Figuren, Positionen usw. Hier werden auch die meisten Spielregeln realisiert, also logische Zusammenhänge implementiert.

Die Präsentation ist die eigentliche Darstellung der Daten aus dem Modell auf dem Monitor mit der Präsentation des Spielfelds, die Visualisierung der Figuren. Befehle, die durch die Interaktion mit dem Benutzer zustande kommen, werden an die Steuerung weitergeleitet.

Die Steuerung nimmt Befehle des Benutzers entgegen und leitet daraus entsprechenden Modifikationen im Modell ein.

(Krasner und Pope, 1988; Morse und Anderson, 2004)

5.7 schriftliche Ausarbeitung

Die schriftlich Ausarbeitung wird eine genauere Beschreibung des Programms enthalten, also wie tatsächlich die Algorithmen umgesetzt wurden und auf welchen wissenschaftlichen Erkenntnissen dies basiert.

Die schriftlichen Ausarbeitung wird unter anderem Folgendes beinhalten:

- Einleitung
- Erklärung zur Implementierung der Alpha-Beta-Suche inkl. der iterativen Tiefensuche (richtig, vollständig, optimal)
- Erläuterungen zur Bewertungsfunktion (Vergleich)
- Zeit- und Speicherkomplexität
- Weitere Entwicklungsmöglichkeiten, weiterführendes

5.8 Auswahl der Werkzeuge

Windows 7 dient als Betriebssystem. Für Implementierung von Kamisado wird Eclipse Helios Service Release 2 als Entwicklungsumgebung und Java 1.6 als Programmiersprache genutzt. Zur Erstellung von UML Modellen wird ArgoUML 0.32.2 genutzt. Alle anderen Bilder werden mit GIMP 2.6.11 bearbeitet. Für die schriftliche Ausarbeit und dieses Proposal wird MiKTeX 2.9 mit TeXworks 0.4.0, also Latex verwendet.

6 Organisatorisches

In diesem Abschnitt werden einige organisatorische Daten angegeben.

6.1 Zeitplan

Im folgenden wird eine ganz grobe Schätzung über den zeitlichen Ablauf der gesamten Bachelorarbeit angegeben, dabei ist zu beachten, dass einige Aufgabenteile parallel abgearbeitet werden:

- ein Monat für dieses Proposal
- einen Monat für die Implementierung in Java
- einen Monat für die schriftliche Ausarbeitung
- einen Monat für Korrekturen (Test, Bugfix), Verbesserungen (Patch), Erweiterungen (Upgrade), etc. der Implementierung oder der schriftlichen Ausarbeitung

Zusammengerechnet ergibt sich daraus, dass diese Bachelorarbeit spätestens im August 2011 abgabebereit sein sollte. Falls nichts unerwartetes dazwischen kommt, also die Arbeit einigermaßen Planmäßig vorangeht, dann ist eine Anmeldung beim zuständigen Prüfungsamt im Juni 2011 in Betracht zu ziehen.

6.2 Eidesstattliche Versicherung

Die selbständige und eigenhändige Anfertigung versichert an Eides statt

Ort, Datum

Unterschrift

Literaturverzeichnis

- [Berninghaus u. a. 2010] BERNINGHAUS, S. ; EHRHART, K.M. ; GÜTH, W.: *Strategische Spiele: Eine Einführung in die Spieltheorie*. Springer, 2010. – ISBN 3642116507
- [Burley Games 2011] BURLEY GAMES: *Announcing Kamisado!* 2011. – URL <http://www.kamisado.com/index.php/kamisado>. – Zugriffsdatum: 5. Januar 2012
- [Conny 2004] CONNY: *Spiralmodell nach Boehm (1988)*. Wikipedia. November 2004. – URL http://de.wikipedia.org/w/index.php?title=Datei:Spiralmodell_nach_Boehm.png&filetimestamp=20080312012210. – Zugriffsdatum: 5. Januar 2012
- [Floyd 2003] FLOYD, Christiane: *Vorgehensmodelle für Software-Projekte*. Vorlesung 5 zu Softwaretechnik und Software-Ergonomie an der Universität Hamburg. 2003. – URL <http://swt-www.informatik.uni-hamburg.de/attachments/LVTermine/STE-VL4-WS-03-04.pdf>. – Zugriffsdatum: 5. Januar 2012
- [Fricke 2010] FRICKE, Stefan: *Grundlagen der Künstlichen Intelligenz - Strategische Spiele*. Vorlesungsfolien zum Modul Grundlagen der Künstlichen Intelligenz an der Technischen Universität Berlin. November 2010
- [HUCH! & friends 2010] HUCH! & FRIENDS: *Kamisado*. 2010. – URL <http://www.huchandfriends.de/page/de/Die-Spiele/Strategiespiele/kamisado.php>. – Zugriffsdatum: 5. Januar 2012
- [Krasner und Pope 1988] KRASNER, G.E. ; POPE, S.T.: A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System. In: *Journal of Object Oriented Programming* 1 (1988), Nr. 3, S. 26–49
- [Ledingner 2010] LEDINGER, Christoph: *Kamisado*. www.spieletest.at - Wir testen Spiele! Mai 2010. – URL <http://www.spieletest.at/spiel.php?ID=2766>. – Zugriffsdatum: 5. Januar 2012
- [Luibl 2011] LUIBL, Jörg: *Special: Kamisado*. 4Player.de - Das Spielemagazin. April 2011. – URL <http://www.4players.de/4players.php/dispbericht/Spielkultur/Special/26518/71553/0/Kamisado.html>. – Zugriffsdatum: 5. Januar 2012
- [Morse und Anderson 2004] MORSE, S.F. ; ANDERSON, C.L.: Introducing Application Design and Software Engineering Principles in introductory CS Courses: Model-View-Controller Java Application Framework. In: *Journal of Computing Sciences in Colleges* 20 (2004), Nr. 2, S. 190–201. – ISSN 1937-4771

- [Rasmusen 2007] RASMUSEN, E.: *Games and Information: An Introduction to Game Theory*. Blackwell, 2007. – ISBN 1405136669
- [Russel und Norvig 2003] RUSSEL, Stuart ; NORVIG, Peter: *Artificial Intelligence: A Modern Approach*. Second Edition. New Jersey : Pearson Education, 2003 (Prentice Hall Series in Artificial Intelligence). – ISBN 0137903952
- [Shah 2007] SHAH, Rajiv B.: *Minimax*. Dictionary of Algorithms and Data Structures [online], Paul E. Black, ed., U.S. National Institute of Standards and Technology. Januar 2007. – URL <http://www.nist.gov/dads/HTML/minimax.html>. – Zugriffsdatum: 5. Januar 2012
- [Wiese 2001] WIESE, H.: *Entscheidungs- und Spieltheorie*. Springer, 2001. – ISBN 3540427473