

Projekt zur GdI 1 – Sommersemester 2009

Technische Universität Darmstadt



Gruppe 3
„Little Buddha Shisen“

Chandra Kirchrath, Thomas Lack, Eugen Grenz, Benjamin Mayer

Inhaltverzeichnis

1. Einleitung	3
2. Ausgangssituation	3
3. Klassendiagramm	4
4. Aufgetretene Probleme	5
5. Fazit der Gruppenmitglieder	5

1 Einleitung

Die Aufgabenstellung war, in Kleingruppenarbeit das Spiel Shisen in Java zu implementieren. Für eine Einführung in das eigentliche Spiel verweisen wir auf <http://en.wikipedia.org/wiki/Shisen-Sho>.

Organisiert haben wir das Projekt im Stil einer typischen Open Source Entwicklung, was einer aufgabenbezogenen Arbeitsweise in Eigenverantwortung mit durchgängigem Teamkontakt über gängige Kommunikationsplattformen im Internet entspricht (IRC, ICQ). Zusätzlich fanden im einwöchigen Rhythmus ca. ein- bis zweistündige Treffen statt, um Probleme direkter zu besprechen, Aufgaben umzuverteilen und erreichte Milestones zu besprechen.

Das gemeinsame Arbeiten am Quellcode wurde mittels SVN abgewickelt, wobei kurz vor Abgabeschluss der im Rechnerpool der TU Darmstadt eingerichtete Server seinen Dienst versagte. So waren wir gezwungen das Projekt in der letzten Phase noch umzuziehen. Zu diesem Zweck wurde „google code“ als neue Plattform ausgesucht: <http://code.google.com/p/gdi1shisen/>.

2 Ausgangssituation

Ausgehend von folgendem Use Case Diagramm (Abb. 1) wurde die Entwicklung des Projektes begonnen. Dabei ist zu beachten, dass die „include“ und „extend“ Beziehungen nur der übersichtlicheren Darstellung geschuldet sind und nicht direkt in dieser Form eine Umsetzung fanden.

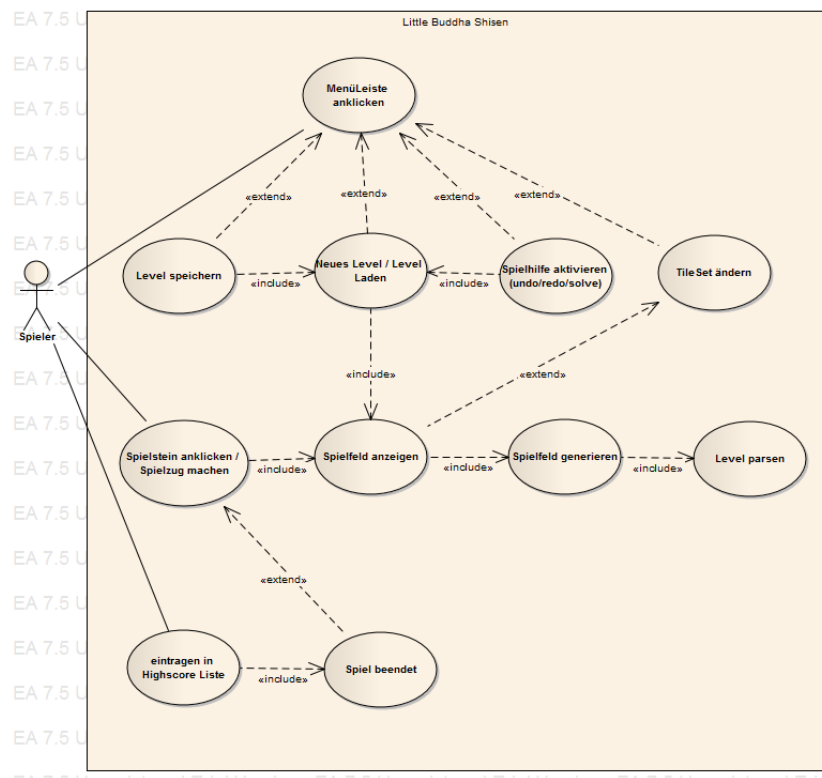


Abbildung 1: Use Case Diagramm

3 Klassendiagramm

In dem finalen Klassendiagramm (Abb. 2) haben wir versucht, einerseits unsere Aufteilung in Model, View und Controller Klassen zu visualisieren und andererseits auf die Beziehung der Klassen untereinander einzugehen. Ein Pfeil mit einer offenen Pfeilspitze und einem Variablennamen stellt eine instanziierte Membervariable dar, ein Pfeil mit geschlossener weißer Pfeilspitze kennzeichnet eine Vererbungshierarchie.

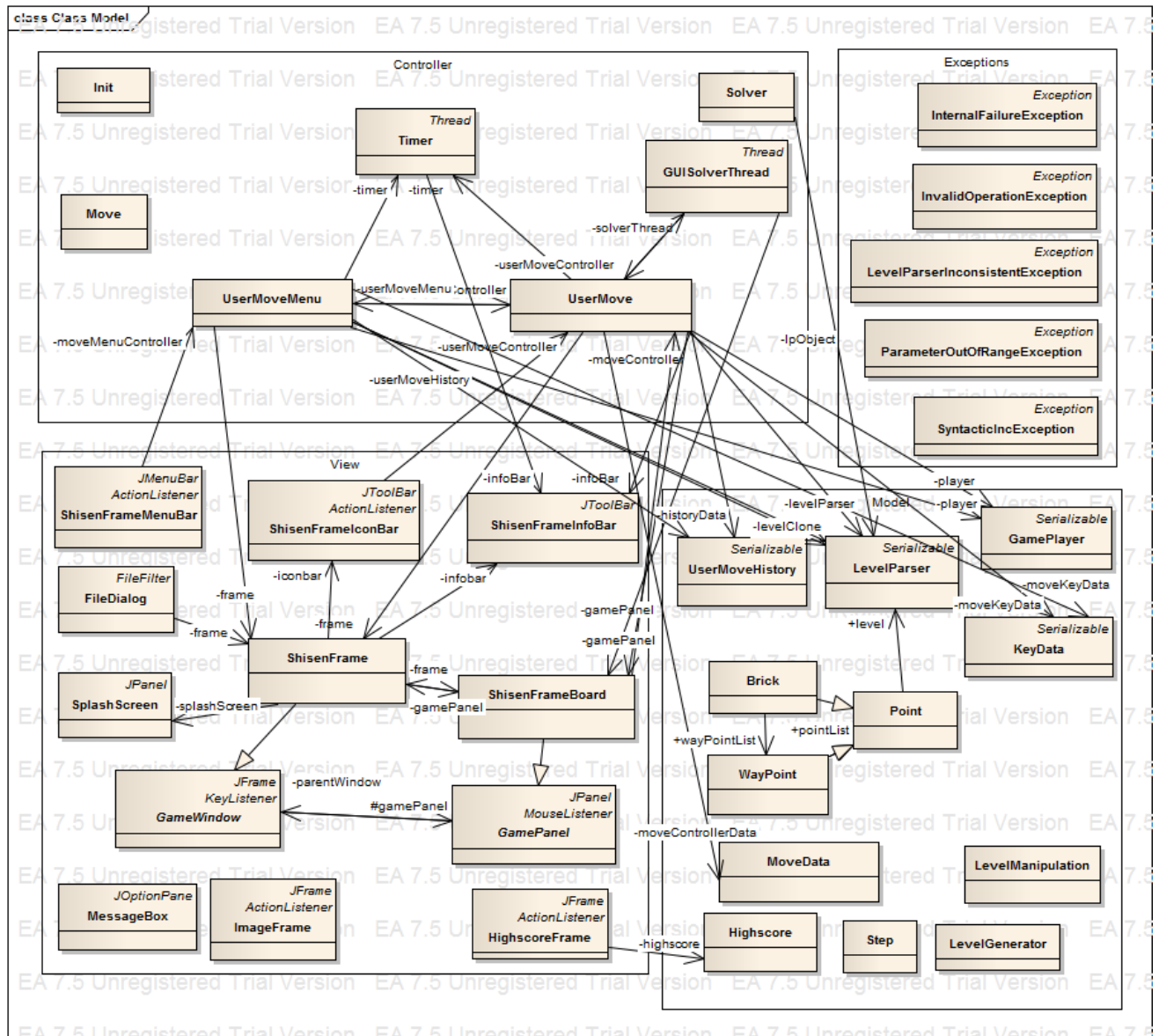


Abbildung 2: Klassendiagramm mit MVC Visualisierung

4 Aufgetretene Probleme

Neben den bereits angesprochenen Problemen mit dem Server für SVN traten vor allem Probleme mit dem mitgelieferten Framework für die graphische Ausgabe auf. Eine Lösung musste von uns selbstständig erarbeitet werden, was einige Zeit in Anspruch nahm. Für andere Projektgruppen wurde diese Lösung dann im Forum kommuniziert.

Von uns selbst verschuldete Probleme waren zum Einen eine nicht ganz ausreichende Planung der Klassenstruktur. Hier hätte man noch wesentlich schöner objektorientiert programmieren können. Wären die Grenzen des Projekts nicht so klar abgesteckt gewesen, hätte man gegen Ende der Entwicklung vielleicht über kleinere refactoring Maßnahmen nachdenken können.

Zum Anderen haben uns parallel zum Projekt stattfindende Klausuren zum ständigen umdisponieren bei der Aufgabenverteilung gezwungen, um einen konstanten Fluss an implementierten Funktionen zu gewährleisten.

Das letzte Problem war die Vorgabe der Länge der Dokumentation, die eigentlich nur als schlechter Scherz gedeutet werden kann. Alleine Deckblatt und Inhaltsangabe umfassen bereits zwei Seiten.

5 Fazit der Gruppenmitglieder

Chandra Kirchrath:

Die Aufgabenstellung Projektes Gdl1 SS09 war relativ einfach zu verstehen. Ein bisschen problematisch war für mich, dass ich die interne Leveldarstellung auf einer 8x18 bzw. 10x20 Matrix ausgelegt hatte und ich anschließend eine Umgestaltung zu einer variablen Größe vornahm. Daher generiert der LevelGenerator auch nur Zufallslevel in der Größe 18x8.

Um sich Arbeit zu ersparen verwendeten wir das Framework. Dieses zeigte uns keine Bilder auf den Spielsteinen unter Linux an.

Der Fehler lag in dem Framework und den dort generierten URLs für die Pfade zu den Bildern. Dieses Problem sorgte für eine hervorragende aufgabenübergreifende Zusammenarbeit.

Die Aufgabe teilten wir in unterschiedliche Bereiche auf, vergaßen aber eine einheitliche Datenstruktur (bis auf das - für ein leeren Stein) zu definieren. Dies hat zur Folge, dass je nach Klasse und unterschiedlichem Ersteller unterschiedliche Datentypen verwendet wurden.

Wöchentliche Treffen im C-Pool und tägliche Treffen im irc/icq ermöglichten aber eine ansonsten nahezu perfekte Kommunikation.

Benjamin Mayer:

Das Projekt war relativ bereichernd an Wissen und vor allem "praktischer" Erfahrung. Also praktische Erfahrung in dem Sinne mal in einer Gruppe wirklich etwas zu programmieren mit Aufgabenteilung, Besprechungen und allem was sonst so zum programmieren im Team dazu gehört.

Da ich die Hausübung so gut wie immer alleine bearbeitet und abgegeben hatte war dies erst mal neu.

Desweiteren konnte beim Projekt das Wissen über GUI Programmierung in Java vertieft werden. Beim realisieren der Tastensteuerung musste zum Beispiel darauf geachtet werden den Eingabefokus

immer wieder an das Objekt in der die Tastensteuerung implementiert ist zurück zu geben und dass beim verschieben des Cursors dieser entsprechend neugezeichnet, an der alten Position entfernt, die neue Position gespeichert wird und so weiter, nur um mal ein paar Beispiele zu nennen.

Dies und vieles anderes bracht mir ein detaillierteres Wissen über Interaktion zwischen verschiedenen GUI Objekten. Ansonsten lernte man durch das MVC - Modell wie in ein größeres Projekt sinnvoll zu unterteilen und zu verwalten ist.

Kurz gesagt es konnten viele neue Erfahrungen gemacht werden und die Zusammenarbeit im Team hat Spaß gemacht auch wenn man teilweise doch sehr unabhängig voneinander gearbeitet hat. Außerdem ist natürlich extrem cool ein Spiel zu programmieren, zumindest bei mir war diese Idee einer der Auslöser dafür ein Informatik Studium überhaupt erst anzufangen.

Thomas Lack:

Im Nachhinein hätte man mehr Zeit für die Planung des Projektes aufwenden sollen. Zwar hat uns unsere Planung sicher bis ans Ende des Projektes getragen, aber man hätte noch einiges schöner und vor allem objektorientierter gestalten können mit etwas mehr Aufwand in der Planungsphase. Da habe ich mich von der „Aufbruchsstimmung“ meiner Kommilitonen mitreißen lassen, obwohl ich aufgrund meiner Vorkenntnisse aus dem Bereich Software Engineering etwas regulierender hätte eingreifen können. Aber das ist halt der Unterschied zwischen Theorie und Praxis.

Das Framework wurde mehrfach angesprochen, mit meinem jetzigen Wissen über Java und seine graphischen Ausgabemöglichkeiten würde ich es nicht mehr benutzen.

Schwierig fand ich zuweilen die Koordination der einzelnen Teammitglieder, was ihre Aufgabenstellungen betraf. Klausurbedingt mussten immer wieder Arbeiten unterbrochen oder von anderen Teammitgliedern übernommen werden, manchmal musste ich auch bremsen um wenigstens ein oder zwei interessante Aufgaben für Teilzeitabstinente zu retten. Und manchmal wurden natürlich auch Teile des Projekts nicht rechtzeitig fertig oder entgegen den Vorgaben umgesetzt, dann musste zuweilen auch der Rotstift herhalten oder auf Fehler aufmerksam gemacht werden.

Die Arbeit mit dem Team hat mir insgesamt sehr viel Spaß gemacht und ich kann mit Fug und Recht behaupten, dass ich meine Kenntnisse in der objektorientierten Programmierung und Java deutlich erweitern konnte.

Die Modalitäten zur Testierung finde ich mehr als misslungen. Bei Ausbaustufe 1 kann man vielleicht noch erwarten, dass jedes Teammitglied den gesamten Code kennt. Aber ab einer gewissen Projektgröße ist das einfach nicht mehr sinnvoll.

Eugen Grenz:

[...]