



abalone

Handbuch

Exposee

Abalone ist ein bereits existierendes Brettspiel, das hier als Multiplayer-Spiel in Client-Server Architektur in Java implementiert wurde. Das Spielfeld besteht aus 61 hexagonal angeordneten Mulden und jeder Spieler hat zu Beginn 14 Kugeln.

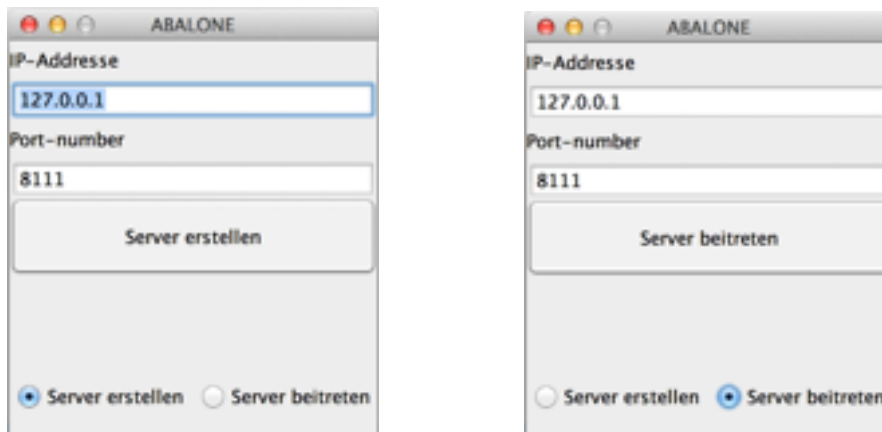
Tim Steindel
Lukas Drescher
Diego Renner

Inhalt

Lobby:	3
GameGUI:	4
Spielregeln	5
Ziel des Spieles	5
Beispielsession	6
Software-Qualitätssicherung	7
Struktur des Programms	7
Funktionalität des Spiels	8
Bibliotheken	8
Nachwort	8

Einem Server beitreten/Einen Server erstellen

Um das Spiel zu starten, genügt ein Doppelklick auf das File "Abalone.jar". Daraufhin öffnet sich ein Menü, das das Erstellen und Betreten eines Servers erlaubt. Beim Erstellen eines Servers sollte die eigene IP-Adresse sowie eine Port im Bereich ab 1025 gewählt werden.



Nach erfolgreicher Anmeldung an einem Server kann man in der Lobby chatten und ein Spiel erzeugen oder einem existierenden Spiel beitreten.

Lobby:

Help:
Hier findet man das Handbuch
und die Regeln

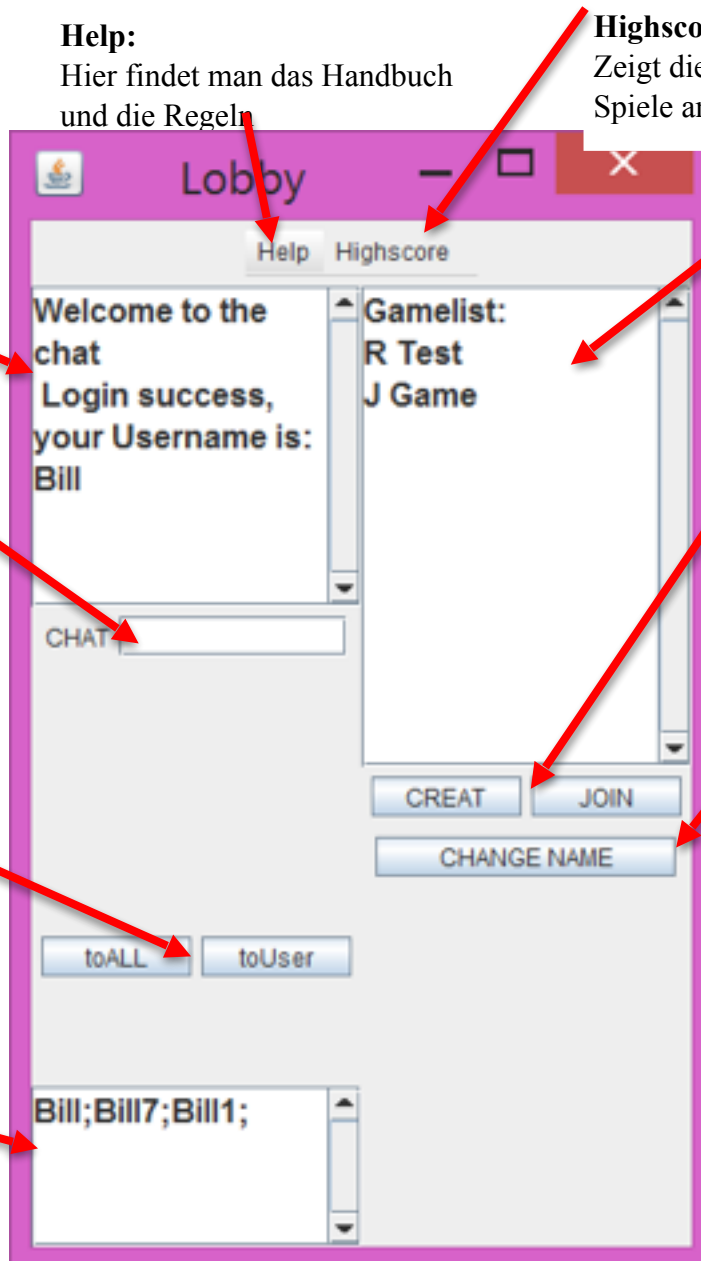
Highscore:
Zeigt die alten abgeschlossenen
Spiele an

Zeigt den Chat an

Hier können Sie eine
Nachricht verfassen
oder einen neuen
Benutzernamen
eintragen. Für eine
Privatnachricht
verwenden Sie bitte
folgende Methode:
Empfänger,Nachricht

toAll:
sendet an alle
toUser:
sendet die Nachricht an
den genannten
Empfänger

Liste der momentan
aktiven User



Liste der Spiele. Spiele können drei
Zustände haben
R –das Spiel ist voll und läuft
J: Sie können dem Spiel beitreten
F: das Spiel ist beendet

Create & Join:
Öffnet ein neues Fenster in dem Sie
den Namen des Spiels, welches Sie
erzeugen oder beitreten wollen,
eintragen (! Gross-/Kleinschreibung
beachten)

Change Name:
Wechselt Ihren Namen in den
Namen, den Sie durch das
Chatfenster gewählt haben

GameGUI:

Help:

Hier finden du das Handbuch und die Regeln

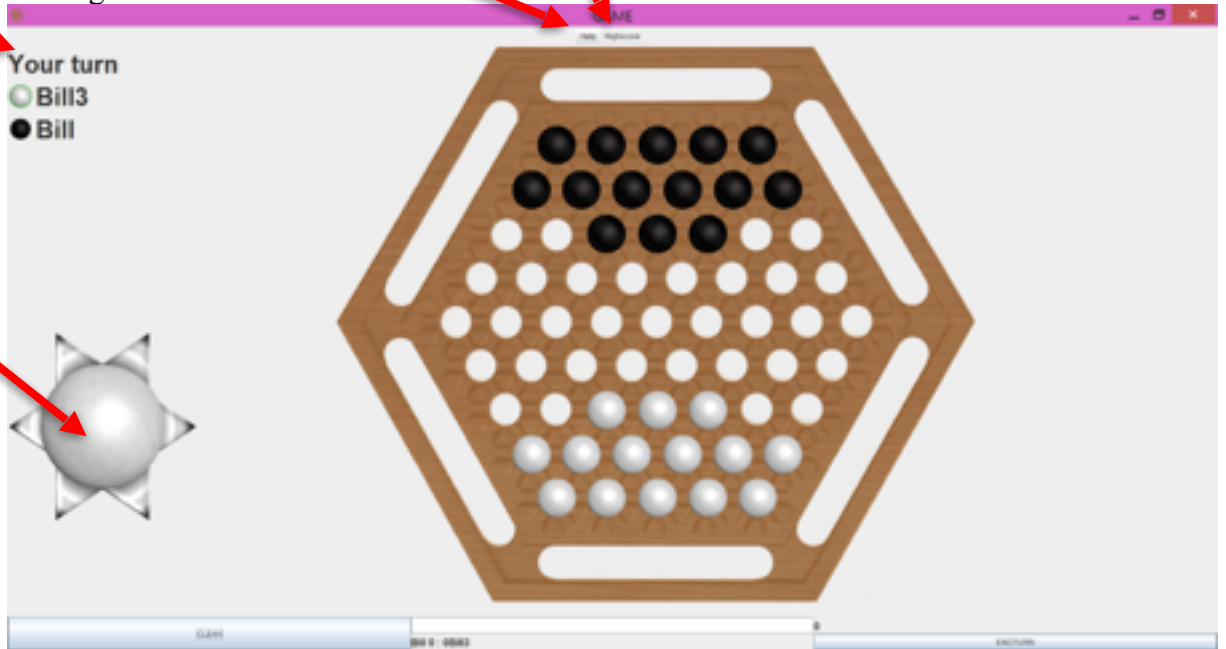
Highscore:

Zeigt die alten abgeschlossenen Spiele an

Zeigt wer gerade dran ist und auch, welche Farbe Sie und Ihr Gegner haben.

Ein **Kompass** für die Richtungsauswahl

Clean: Lösche deine momentane Auswahl



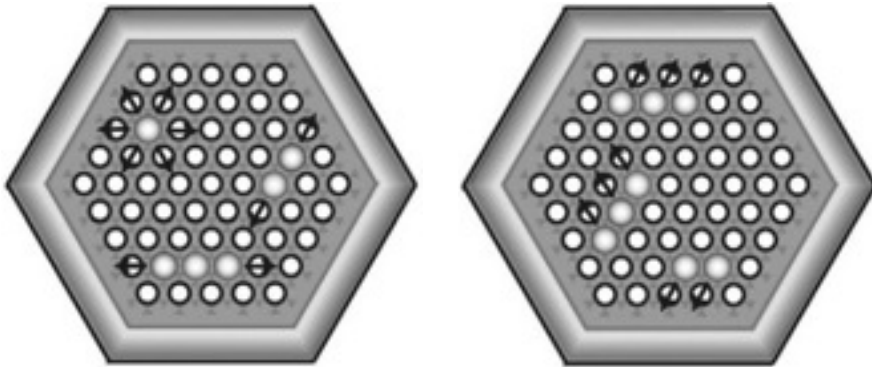
Zeigt den momentanen **Punktstand** an

Sollten Sie einen fehlerhaften Zug durchführen, wird es hier angezeigt

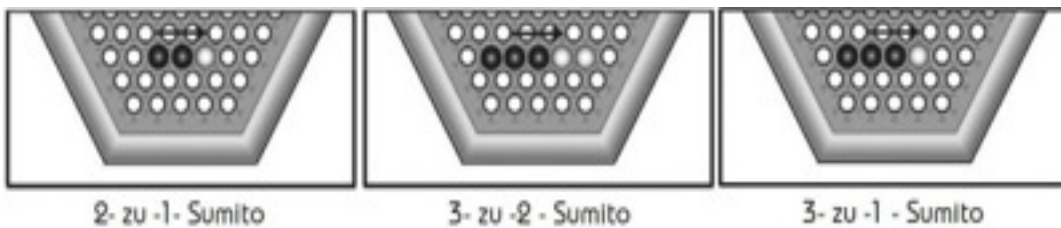
Wenn Sie zufrieden mit ihrer Auswahl sind, können Sie hiermit Ihren Zug beenden

Spielregeln

Es dürfen pro Zug bis zu drei eigene, nebeneinanderliegende Kugeln in eine der sechs möglichen Richtungen verschoben werden, sofern diese nicht von eigenen oder gegnerischen Kugeln blockiert werden. Auch eine parallele Bewegung ist erlaubt. Hier einige Beispiele:



Um gegnerische Kugeln zu stossen, muss die eigene Anzahl Kugeln die der Gegner überwiegen.



Patt:

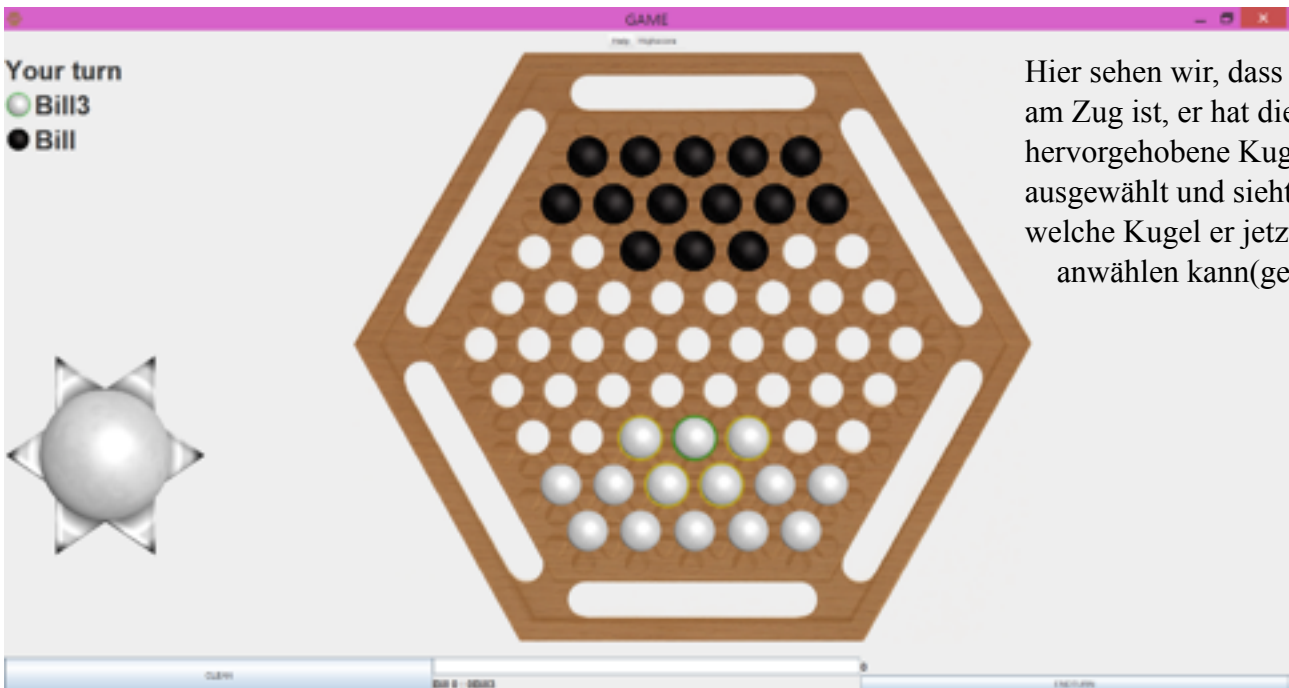
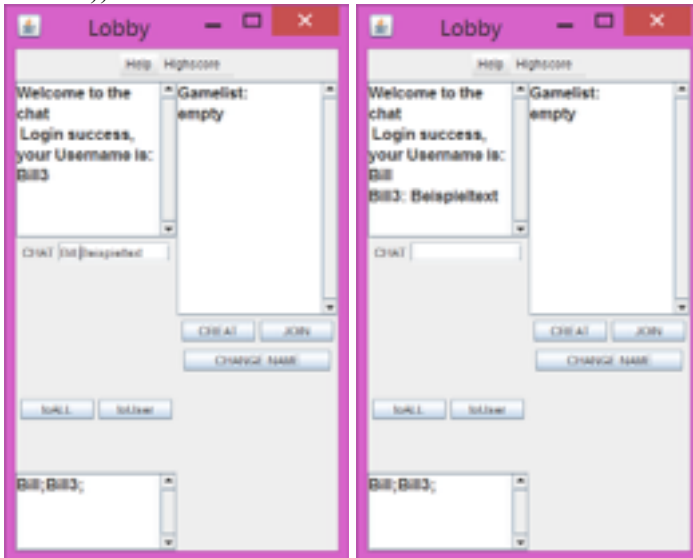
Sollte die Anzahl der Kugel beider Spieler gleich sein, ist kein verschieben der Kugelreihe möglich. Wobei es ab der Konfiguration 3 vs. 3 nicht auf weitere Kugeln ankommt.

Ziel des Spieles

Ziel des Spieles ist es, durch kluge Verschiebungen der Kugeln, sechs der gegnerischen Kugeln vom Spielfeld zu stossen.

Beispielsession

Ein Beispiel für das Schreiben an eine Person
(links der Sender , rechts der Empfänger, nachdem der Sender die Nachricht gesendet hat(**durch toUser Button**))



Hier sehen wir, dass Bill3 am Zug ist, er hat die grün hervorgehobene Kugel ausgewählt und sieht, welche Kugel er jetzt noch anwählen kann(gelb)

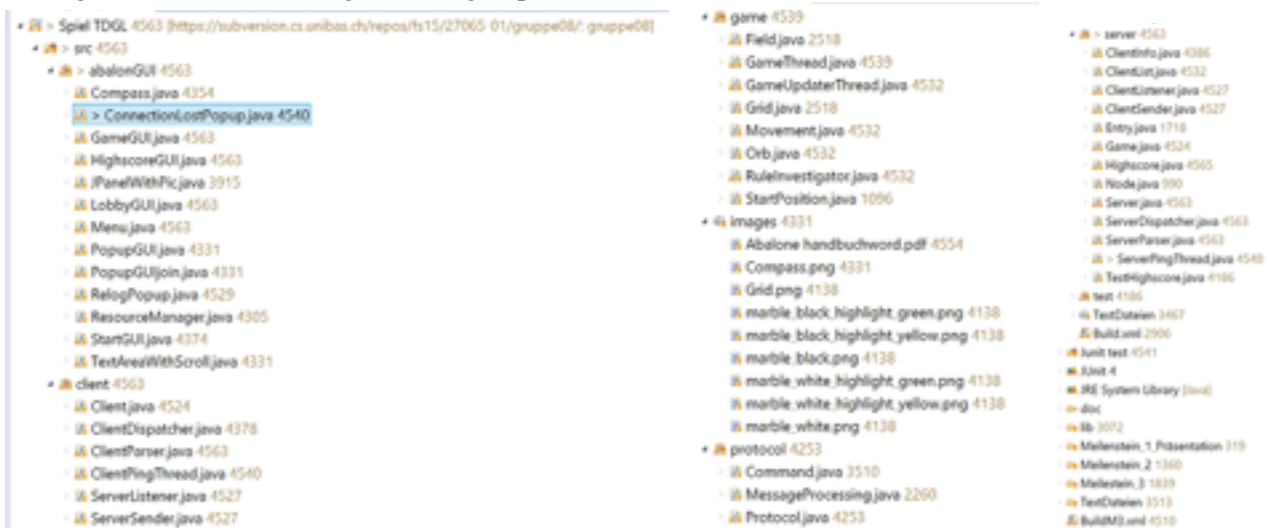
Software-Qualitätssicherung

Der erste wichtige Stützpfiler für unsere Qualitätssicherung ist die Einhaltung unser Code-Convention, die auf unserem Blog zu finden sind. Durch das Benutzen unseres Abalone-Protokolles, also einem textbasierten Protokoll ist die Kommunikation einheitlich und intuitiv, sowie leicht erweiterbar und/oder veränderbar. So muss nur in der Klasse „*Protocol*“ der Befehl geändert werden und nicht mühselig innerhalb der anderen Klasse. Eine ähnliche Methodik haben wir auch in der Klasse „*ResourceManager*“ verwendet, in diesem werden unsere Bilder erstellt und gespeichert.

Für die weitere Qualitätssicherung ist der Client, Server, GUI sowie das Spiel an sich voneinander getrennt. Dadurch kann zum Beispiel mit wenigen Veränderungen ein anderes Spiel über unsere Server-Client-Verbindung gespielt werden.

Struktur des Programms

Package: abalonGUI, client, game, images, protocol, server, test



Die Packagenamen zeigen ihre Funktion, da der ServerParser, sowie der ClientParser mit Hilfe des Protokolls kommunizieren. „protocol“ ist ein eigenes Package. In diesem findet sich auch der Escaper und auch ein Nachrichtenzerteiler.

Die allgemeine Struktur im Server und Client ist gleich: der Hauptträger sind die Dispatcher/Parser. Die Dispatcher bekommen die Nachrichten und diese wird dann durch den Parser analysiert und je nachdem weitergegeben. Die Listener und Sender funktionieren komplett gleich, nur dass auf Serverseite für jeden Client ein Paar von ihnen existiert. Diese haben die Aufgabe, entweder auf eine Nachricht zu warten und diese an ihren Dispatcher weiterzugeben oder aber eine Nachricht auszugeben.

Die PingThread auf beiden Seiten sind vermutlich selbsterklärend, wobei die Besonderheit daran liegt, dass eine neue Verbindung aufgebaut wird, sollte die Verbindung verloren gehen. Aufgrund der TCP Verbindung wird die ältere Verbindung nach etwa 12 Sekunden gekappt.

Der Ablauf des Spieles läuft durch den GameUpdaterThread und durch den RuleInvestigator. Der GameUpdaterThread verwaltet die Spiele und der RuleInvestigator überprüft, ob der Spielzug regelkonform war.

Funktionalität des Spiels

Alle wichtigen Chateigenschaften werden durch unser Spiel unterstützt, so können Privatnachrichten ausgetauscht werden, aber auch öffentliche. Sollte der User wider Erwarten mit seinem Namen nicht einverstanden sein, kann dieser auch seinen Namen ändern. Der Spieler erkennt ein öffentliches Spiel und kann diesem beitreten.

Für kompetitive Spieler gibt es den Highscore, in dem sie versuchen, das beste Ergebnis zu haben. Damit aber keiner schummelt, wird nicht nur durch die GUI schon lokal mögliche falsche Eingaben herausgefiltert, sondern auch nochmal auf Seiten des Servers kontrolliert.

Eine persönliche Gestaltung der Kugel oder eine Auswahl aus verschiedenen Kugelfarben wäre denkbar und da es rein kosmetischer Natur wäre, auch einfacher zu implementieren.

Das Regelwerk des Spieles ist durch seine Einfachheit nicht sehr gut erweiterbar, doch kann relativ einfach eine neue Startpositionen erstellt werden, so müsste nur in der Klasse „Startposition“ die Kugel anders platziert werden durch einen anderen Konstruktor. Auch eine Vergrößerung des Spielfeldes wäre denkbar, doch hier wäre eine Veränderung deutlich schwieriger, da es durch die spezifische Form des Spielfeldes zu Komplikationen in der Graphik, sowie zu einigen Veränderungen in der Positionsberechnung führen würde.

Aufgrund vom javadoc und selbsterklärenden Variablennamen ist ein Einlesen in den Code und das Erweitern durch Dritte möglich sein.

Bibliotheken

Wir benutzen keine weiteren Bibliotheken außerhalb der von Java.

Nachwort

Für die Zeichnungen der Kugel bedanken wir uns an Milan van Zanten. Außerdem bedanken wir uns bei Laura Feuck, für ihre Hilfe am Handbuch und bei der Berechnung der Kugelpositionen. Auch möchten wir uns bedanken bei Loris Sauter und Filip Brinkmann für Hilfestellung und Code-Review.

Alle Rechte vorbehalten. Patent N° DM/012362

® Abalone ist ein registriertes Warenzeichen von Abalone s.a.

© Abalone s.a. - Domaine De Longue Plaine, 37250 Sorigny France

Autoren: Laurent Lévi & Michel Lalet.