



A GAME ØF STØNES

Read Me

ALLGMEINES PROGRAMIERPRAKTIKUM

FREDERIK KETTEL, PHILIPP THUROW UND MAXIMILIAN PESCH

Inhalt

Das Projekt	2
Die Projektstruktur	2
Das Interface.....	2
Die Spieler.....	4
Ant	5
Targets	5
Die jar-Datei.....	6
Die Kommandozeilenargumente	6

Das Projekt

„Die Breakthrough-Variante des Allgemeinen Programmierpraktikums (BreakthroughPP) ist ein Spiel für zwei Spieler, wobei ein Spieler (Rot) mit den roten und der andere Spieler (Blau) mit den blauen Steinen spielt.“

Quelle: Projektbeschreibung

Das Projekt kann einfach mit dem Archivmanager von Linux entpackt werden oder mit dem Explorer in Windows.

Die Projektstruktur

Im Hauptverzeichnis des Projektes befinden sich eine Datei und ein Verzeichnis. Im Source-Verzeichnis befindet sich der Quellcode des A Game Of Stones Projekts. Der in der Projektbeschreibung gestellte Quellcode befindet sich im Paket „preset“. Im preset Paket befinden sich auch noch ein weiteres Paket, was weitere Exception Typen enthält.

Die Datei, die im Hauptverzeichnis liegt ist die Ant-Build file.

Das Interface

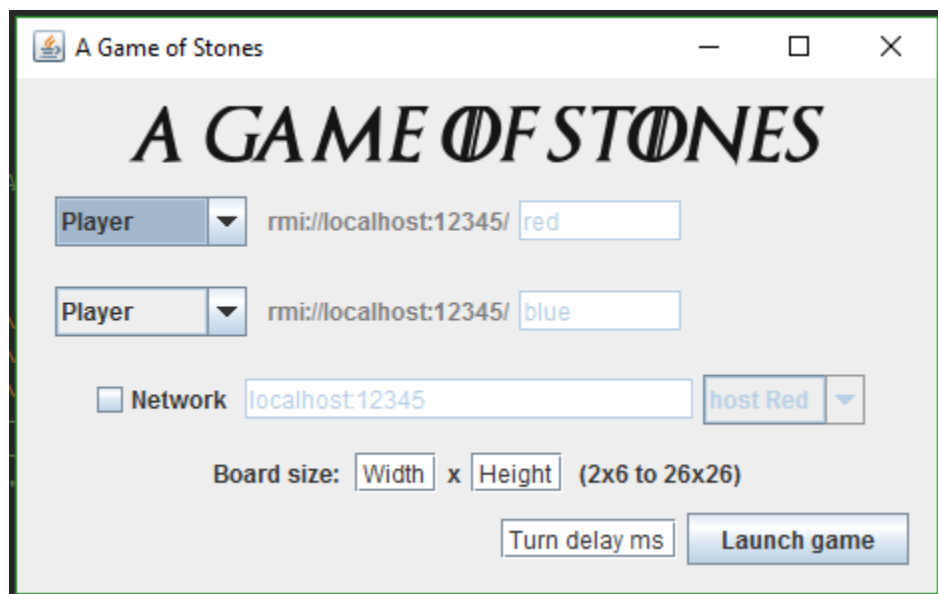


Figure 1 Das Spielsetup

Hier kann man in den ersten beiden Auswahlboxen die Art der Spieler auswählen. Wenn man selber spielen möchte, wählt man „Player“ aus ansonsten hat man die Auswahl auf einen NetPlayer für Spiele übers Netzwerk oder verschiedene KIs. Sollte man sich für den NetPlayer entschieden kann man dann noch die RMI Adresse angeben, falls diese nicht gleich dem Standard ist.

Die Felder für „board size“ und „turn delay“ sind optional und werden mit Standardwerten belegt falls nichts eingegeben ist. (Board size: 2x6 und Turn delay: 0 ms)

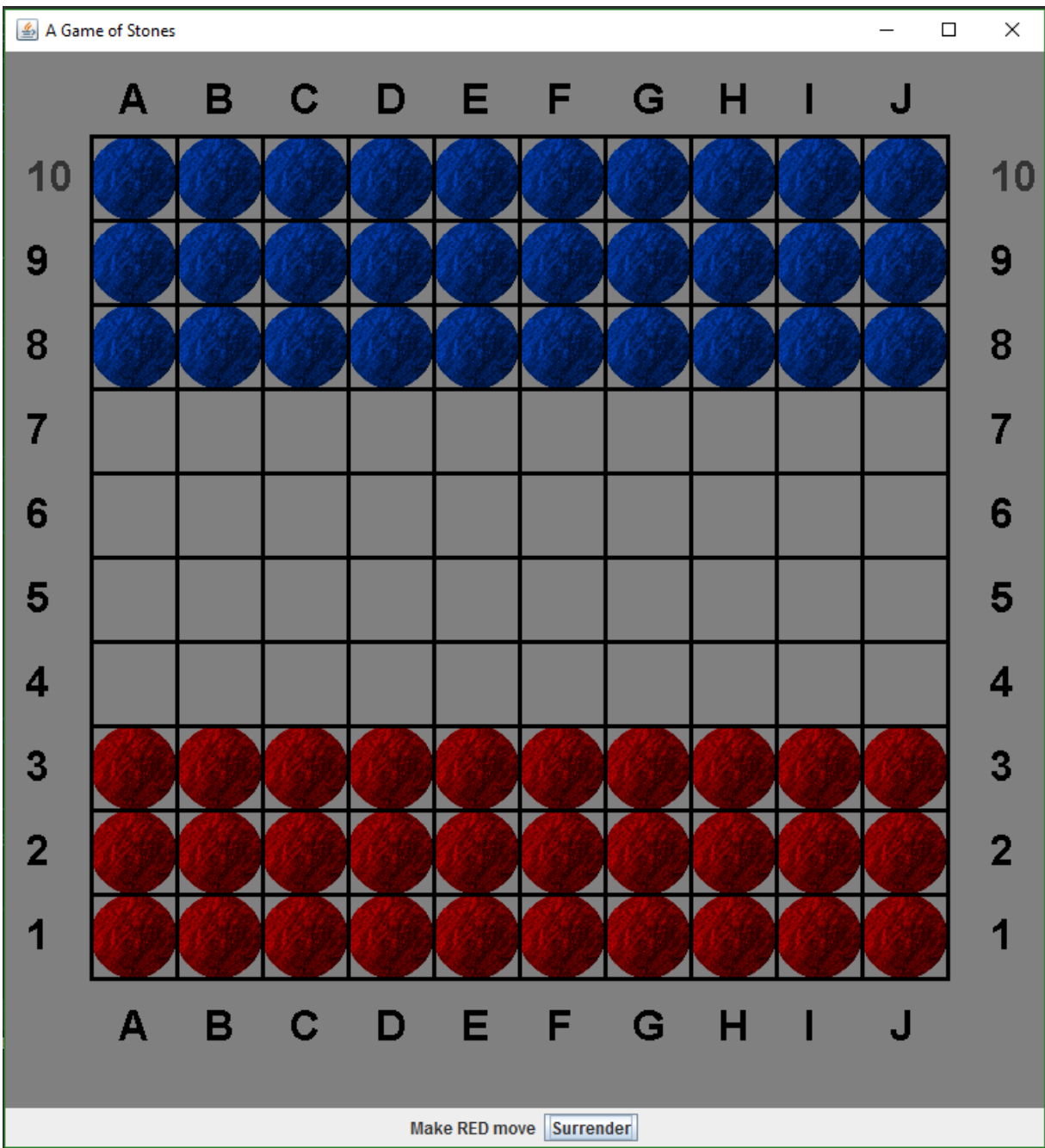


Figure 2 Ein neues 10x10 Brett

Sobald man auf „Launch Game“ geklickt hat wird ein neues Spielfeld der angegebenen Größe erzeugt. Die Oberfläche ist nun ziemlich simpel gehalten. Unten wird angezeigt welcher Spieler den nächsten Zug machen muss. Des Weiteren wird hier dem aktuellen Spieler die Möglichkeit gegeben aufzugeben, in dem er auf den „Surrender“-Button klickt. Ein Spielzug wird gemacht in dem man auf den zu ziehenden Stein klickt, der dann als ausgewählt markiert ist und dann in einem zweiten Klick das Zielfeld auswählt. Die möglichen Züge der jeweiligen Spieler werden auch farblich angezeigt (von Rot sind mögliche Züge Gelb und Blau hat Cyan farbige Züge)

Sollte das Spiel dann für eine Seite gewonnen sein, wird dies unten angezeigt.

Die Spieler

- Player

Dieser Spielertyp ist der grafische Spieler, der auf Usereingabe wartet.

- Random AI

Der Random Spieler wählt aus allen ihm möglichen Zügen zufällig einen aus. Er hat keine Gewinnstrategie.

- Easy AI

Die einfache KI schlägt Steine wo sie schlagen kann, versucht durchzubrechen und falls es Züge gibt, die den Gegner am Gewinnen hindern, werden diese ausgeführt. Sollten aber alle Züge keine dieser Bedingungen erfüllen, so zieht auch diese KI zufällig.

- Hard AI

Die harte KI verfährt nach dem in der Projekt Beschreibung beschriebenen Muster, um den Gegner möglichst effizient zu schlagen. Zusätzlich durchbricht die KI die gegnerischen Linien, wenn es möglich ist.

- NetPlayer

Um übers Netzwerk zu spielen, verwenden beide Seiten identische RMI-Adressen für Rot und Blau. Der Host des roten Spielers übernimmt die Spielleitung und muss zuerst auf den Launch Button drücken, der blaue Spieler hat danach 30 Sekunden Zeit, den Button ebenfalls zu drücken und sich zum Spiel zu verbinden. Anschließend läuft das Spiel wie üblich ab.

Netzwerkspiele sollten nicht unterbrochen oder geschlossen werden, um die belegten Adressen im RMI¹ ordnungsgemäß freizugeben.

¹ Ein RMI-Service kann in der Konsole mit dem Befehl "rmiregistry [port]" gestartet werden.

Ant

Die Ant- Buildfile ist im Hauptverzeichnis des Projektes zu finden und wird zur Kompilation, Dokumentationsgeneration und zum Ausführen verwendet.

Targets

- compile

Kompiliert das Projekt in einen neuen Ordner names „build“. Dabei wird das Compilerargument "-Xlint:unchecked" beachtet.

- clean

Löscht die Ordner, build und dist. Sodass man bei dem erneuten kompilieren keine Überreste von alten build Vorgängen hat.

- deepClean

DeepClean ist das gleiche Target wie clean jedoch wird dabei alles was von der Ant Build File erzeugt wird gelöscht. Das heißt nicht nur build und dist sondern auch schon erzeugte Dokumentationen.

- dist²

Das dist target kompiliert die aktuelle Version des A Game Of Stones Projekts und packt das gesamte build Verzeichnis zusammen mit den Assets in eine ausführbare jar-Datei dessen Hauptklasse SplinterTheOmniscientRat ist.

- javadoc

Dieses Target generiert aus allen im Sourcecode liegenden Dateien eine Dokumentation mithilfe des gleichnamigen Javadoc Tools und legt dieses in das Unterverzeichnis doc.

- run

Das run target erzeugt mit Hilfe des dist targets die aktuelle Version des Projekts als jar Datei und führt diese auch aus.

² dist ist kurz für distribution und enthält die jar-Datei die dann später autonom lauffähig ist.

Die jar-Datei

Nach dem ant die jar-Datei gebaut hat³, erhält man eine jar Datei. Diese kann man ohne jegliche Argumente auf der Kommandozeile oder per Doppelklick ausführen und das Spielerlebnis graphisch genießen.

Jedoch verfügt das A Game Of Stones Projekt auch über eine Kommandozeilen Version des Spiels.

Die Kommandozeilenargumente

- -d oder --debug

Startet das Spiel im Debug Modus. So bekommt man mehr Informationen über den Spielverlauf selber und bei Fehlern eine genauere Ausgabe was passiert ist mit der Java internen Stacktrace-Methode.

- -v oder --version

Gibt die aktuelle Version des Spiels aus.

- -h oder --help

Gibt eine Übersicht über die verfügbaren Argumente aus und die richtige Verwendung des Debug Modus.

³ siehe das [Target dist](#)