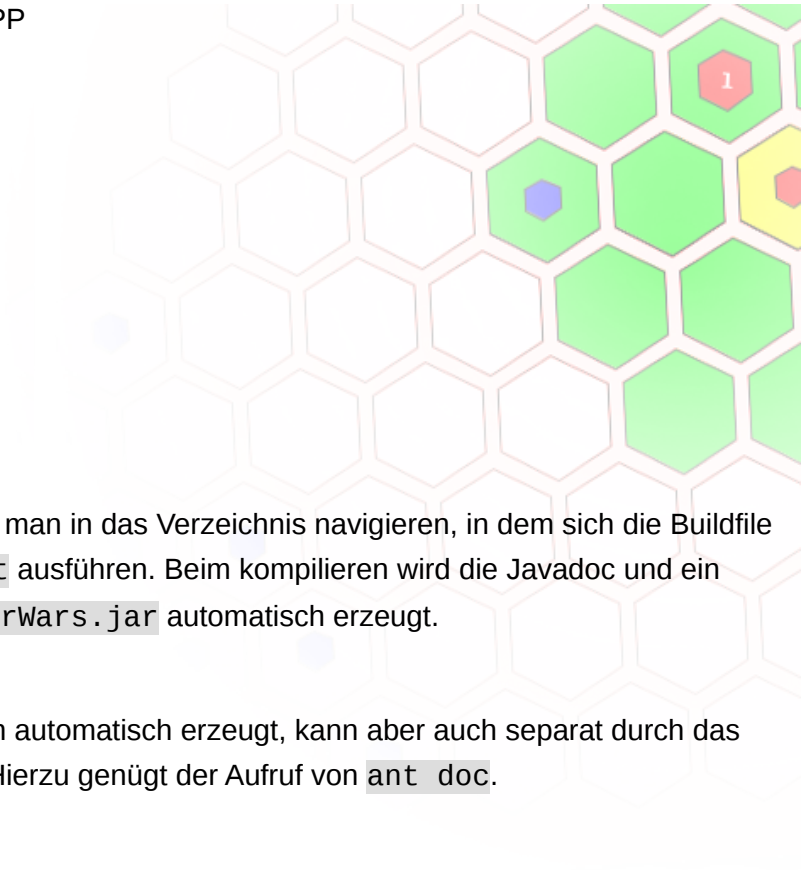
Projektgruppe *OSKAR* – TowerWarsPP

Anleitung



Inhalt

1 Kompilieren

Um das Spiel zu kompilieren kann man in das Verzeichnis navigieren, in dem sich die Buildfile `build.xml` befindet, und dort `ant` ausführen. Beim kompilieren wird die Javadoc und ein Java-Archiv mit dem Namen `TowerWars.jar` automatisch erzeugt.

2 Javadoc

Die Javadoc wird beim kompilieren automatisch erzeugt, kann aber auch separat durch das `ant`-Target `doc` erzeugt werden. Hierzu genügt der Aufruf von `ant doc`.

3 Ausführen

Das Spiel kann über `java -jar /<Pfad>/TowerWars.jar` ausgeführt werden und benötigt folgende Argumente:

`--standard`	um ein Spiel zu hosten (siehe 4) oder lokal zu spielen
`--debug` bzw. `--graphic`	für eine textuelle Ausgabe auf der Konsole bzw. eine grafische Ausgabe mit Java Swing mit visuellen Bedienelementen
`-size n`	erzeugt ein Spielbrett der Größe n mal n
`-red type` `-blue type`	Typ des roten bzw. blauen Spielers. Ersetze `type` mit einer der zur Verfügung stehenden Optionen:

- `remote` (Netzwerkspiel, siehe Punkt 4)
- `simple` (Simple KI)
- `adv1` (Fortgeschrittene KI)
- `human` (menschlicher Spieler)
- `random` (Zufällige Züge)

4 Netzwerkspiel

Um ein Netzwerkspiel bereitzustellen startet man eine rmi-Registry mit dem Befehl `rmiregistry -J-Djava.rmi.server.codebase=file:/<Pfad>/TowerWars.jar` und startet dann TowerWars.jar wie bereits in 3 erklärt, allerdings mit `--offer -type type` anstelle von `--standard`. `type` ist zu ersetzen mit einem der Typen, die auch für `-red` und `-blue` zur Verfügung stehen.

Um einen angebotenen Spieler anzunehmen startet man einen normalen Spieldurchlauf mit dem entsprechenden Spieler als `remote` und gibt dann host und Namen ein.

5 Ein- und Ausgabe

textuell: Die Züge werden im Format `Startposition->Endposition` übergeben. Gibt man anstelle eines Zugs `null` ein, so gilt das als Aufgabe. Ist ein Zug ungültig, erfolgt eine neue Anfrage des Zugs.

grafisch: Ein Klick auf einen eigenen Spielstein markiert eine Zelle gelb. Dies ist das Feld der Startposition. Alle grün hinterlegten Felder liegen in Reichweite dieses Spielsteins und sind potenzielle Endpositionen (nicht alle in Reichweite sind gültig). Ein Klick auf ein weiteres Feld übergibt den Spielzug an das Spielbrett.

6 Mitwirkende

Dieses Programm ist im Rahmen des Allgemeinen Programmierpraktikums der Universität Göttingen entstanden. Der betreuende Tutor war Sergio Perez.
Die Mitwirkenden in der Projektgruppe *OSKAR* waren: Marcel Hellwig, Andrej Liebert, Julian Lüken (Leitung) und Oskar Wittkamp.