

# Spielanleitung APP

## Projektgruppe Blocus\_longus

Clemens Beeken, Hanna Holderied, Clara Holzhüter,  
Jonas Klug, Malin Lachmann, Tobias Lücking

27. Juli 2015

*Bemerkung: In dieser Spielanleitung wird zu Gunsten der Lesbarkeit ein generisches Femininum verwendet, wenn allgemeine Begriffe zur Bezeichnung von Personen gleich welchen Geschlechts verwendet werden.*

## 1 Einführung - Das Spiel

Bei dem vorliegenden Spiel handelt es sich um ein Projekt im Rahmen des Allgemeinen Programmierpraktikums an der Fakultät für Mathematik und Informatik der Universität Göttingen im Sommersemester 2015. Aufgabe der Gruppe war es, das Spiel Blocus zu implementieren.

Bei diesem gibt es ein aus  $20 \times 20$  quadratischen Vertiefungen bestehendes quadratisches Spielfeld und in vier Farben 21 Spielsteine (Polyominos), die sich aus bis zu fünf kleinen Quadraten zusammensetzen (siehe Abbildung 1) und für jede Farbe genau einmal vorkommen. Im Zweispielerinnenspiel bekommt jede Spielerin zwei Farben, der eine blau und rot, der andere gelb und grün. Abwechselnd werden nun Spielsteine aufs Feld gelegt, wobei die Reihenfolge Blau, Gelb, Rot, Grün ist. Der erste blaue Stein muss die linke obere Ecke abdecken und die ersten Steine in anderen Farben die entsprechende im Uhrzeigersinn nächste Ecke. Jeder weitere Stein einer Farbe muss die eigene Farbe an einer Ecke berühren, keinesfalls aber an einer Kante. An fremde Steine kann man beliebig anlegen. Wer keinen Zug mehr machen kann oder möchte, pausiert.

Es wird so lange gezogen, bis alle Spielerinnen pausieren oder eine Spielerin alle ihre Steine auf das Feld gelegt hat. Wer am Schluss die meisten Punkte hat, gewinnt. Dabei zählt jedes kleine Quadrat eines eigenen Spielsteins auf dem Spielfeld einen Punkt. Zusätzlich gibt es fünfzehn Punkte für die Spielerin, die alle ihr Steine auf dem Spielfeld unterbringen konnte. Beim Zweispielerinnenspiel werden für jede Spielerin die Punkte ihrer beiden Farben addiert.

## 2 Parameter

### 2.1 Spielerinnentypen

Es wurden vier Spielerinnentypen implementiert: Eine interaktive Spielerin, eine zufällig spielende, eine „leichte“ KI-Spielerin und eine („schwere“) KI-Spielerin. Wählen der interaktiven Spielerin ermöglicht eine eigene Eingabe von Zügen, die zufällig spielende Spielerin wählt aus ihren möglichen gültigen Zügen einen zufällig aus.

Die „leichte“ KI-Spielerin genügt den in Punkt elf der Projektanforderung gestellten Forderungen. Wenn sie einen Zug setzen kann, der ihr direkt zum Sieg verhilft, setzt sie diesen. Einen Zug, nach dem ihre nächste Gegnerin im nächsten Zug gewinnen kann,

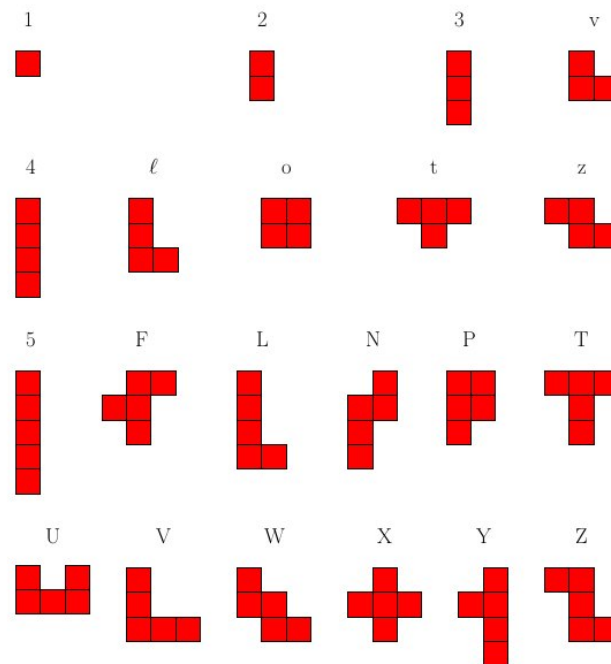


Abbildung 1: Die einer Farbe zur Verfügung stehenden Spielsteine.

setzt sie nur, wenn sie keine Alternativen hat. Unter allen möglichen Zügen wählt sie nun zufällig einen aus der Menge der Züge, bei denen die Summe der Punkte, die dieser Zug bringt und der Quadrate, die auf einer der Spielfeld-Diagonalen liegen, maximal ist.

Die KI-Spielerin hat drei verschiedene Schwierigkeitsgrade, beim leichten versucht sie, möglichst große Steine zu legen, beim zweiten spielt sie so, dass sie die Anzahl ihrer nach dem Zug möglichen Züge maximiert und versucht, die Gegner(in) zu blockieren. Auf der dritten Stufe maximiert sie wieder die Anzahl ihrer möglichen Züge und berücksichtigt auch die Größe der gespielten Steine.

## 2.2 Ein- und Ausgabe

Die Ein- und Ausgabe kann textbasiert, also über die Konsole, oder grafisch erfolgen.

### 2.2.1 textbasierte Ein- und Ausgabe

Züge werden eingegeben, indem man die Koordinaten nennt, auf die der Stein gelegt werden soll. Die linke obere Ecke des Feldes hat die Koordinaten  $(A, 1)$ , die rechte obere  $(T, 1)$ , die linke untere  $(A, 20)$ . Die Koordinate  $(A, 1)$  wird als

`A,1`

eingegeben, mehrere Koordinaten werden durch Leerzeichen getrennt. Zum Pausieren kann man

`0`

eingegeben.

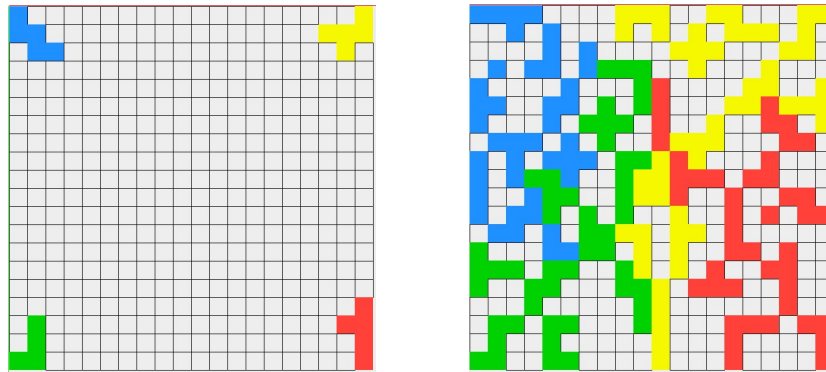


Abbildung 2: Das Spielbrett nach der ersten Runde des Spiel (links) und später im Spielverlauf.

### 2.2.2 grafische Ein- und Ausgabe

Züge werden gesetzt, indem man die entsprechende Felder in dem Grafikfenster anklickt. Durch Doppelklick nimmt man ausgewählte Felder wieder zurück. Sind alle Felder für den Zug ausgewählt, muss dieser über den Knopf „Zug überprüfen und beenden“ übergeben werden. Neben dem Spielbrett werden die noch nicht gespielten Spielsteine angezeigt. Zum Pausieren muss der „Pausieren“-Knopf gedrückt werden.

## 3 Kompilieren des Spiels

Dieser Anleitung beigelegt ist ein Archiv (blocus\_longus.tar.gz). Es werden nacheinander die untenstehenden Befehle in dem Ordner, in dem das Archiv liegt, ausgeführt.

```
tar -xzf blocus_longus.tar.gz
cd blocus_longus
```

Zum Übersetzen des Quelltext wird über

```
ant
```

die dort liegende Datei „build.xml“ ausgeführt. Nun kann das Spiel gestartet werden. Die API-Dokumentation kann über

```
ant javadoc
```

erzeugt werden.

## 4 Ausführen des Spiels

### 4.1 Default-Spiel

Im Ordner „trunk“ wird beim Kompilieren ein jar-Archiv erzeugt, welches im Ordner „jar“ liegt. Über die Eingabe von

```
java -jar jar/blocus_longus.jar d
```

wird das Spiel mit Default-Einstellungen gestartet. Dies ist ein Zweispielerinnenspiel, bei dem die starke KI-Spielerin mit den Farben Blau und Rot gegen eine interaktive Spielerin mit den Farben Gelb und Grün spielt. Die Ausgabe des Spiels findet über die Konsole statt.

## 4.2 Benutzerdefiniertes Spiel

Über die Eingabe von

```
java -jar jar/blocus_longus.jar c
```

wird das Spiel mit benutzerdefinierten Einstellungen gestartet. In der Konsole wird nachfolgend abgefragt, mit wie vielen Spielerinnen (zwei oder vier) gespielt werden soll. Als nächstes werden für jede Spielerin die Spielerinnentypen abgefragt (mehr dazu in Teil 4.3.2):

- (R)andomplayer: die zufällige Spielerin,
- (H)umanplayer: die interaktive Spielerin,
- (K)I-Player: die starke KI-Spielerin (die Stärke wird später abgefragt),
- (W)eak KI-Player: die leichte KI-Spielerin und
- (N)etworkplayer: die über ein Netzwerk angebotene Spielerin.

Abschließend wird erfragt, ob die Ein- und Ausgabe grafikbasiert oder über die Konsole erfolgen soll.

### 4.2.1 Debug-Schalter

Im Falle einer grafikbasierten Ein- und Ausgabe wird hier auch abgefragt, ob der Debug-Schalter gesetzt werden soll. Falls dies geschieht, gibt es zusätzlich zur Grafikausgabe auch eine Textausgabe nach jedem Zug, in der der gesetzte Zug, das Spielbrett und der Spielstand ausgegeben werden.

## 4.3 Spielen über Netzwerk

### 4.3.1 Anbieten einer Netzwerkspielerin

Über die Eingabe von

```
java -jar jar/blocus_longus.jar n
```

wird eine ausführbare Datei gestartet, die eine Netzwerkspielerin anbietet. Wieder wird die Anzahl der Spielerinnen und die gewünschte Darstellung abgefragt, zusätzlich noch, welche Farbe gespielt werden soll und ein Name, unter dem die Spielerin angeboten werden soll. Dann wird auf eine Verbindung gewartet.

### 4.3.2 Finden einer Netzwerkspielerin

Wird im benutzerdefinierten Spiel eine Netzwerkspielerin ausgewählt, so werden dann Name und Host abgefragt und falls möglich, wird sich mit dieser verbunden. Hier muss darauf geachtet werden, dass sowohl die Netzwerkspielerin als auch das Hauptspiel die gleichen Einstellungen wählen (Anzahl der Spielerinnen und Auswahl der Spielerinnentypen), geschieht dies nicht, wird mit den im benutzerdefinierten Spiel gewählten Einstellungen gespielt.