

# **RoboClient**

**Fuzzy LoKick**

**Benutzerhandbuch**

## Einführung:

RoboClient simuliert eine Fußballmannschaft aus elf autonom agierenden Spielern. Das plattformunabhängig in Java geschriebene Programm stützt sich auf Protokoll 7 der offiziellen RoboCup Soccer Simulation League. Über einen RCSS-Server kann das Team gegen andere Mannschaften antreten. Dabei wird als Default-Teamname "Fuzzy LoKick" verwendet. Es ist möglich, die Spieler auf mehrere Rechner zu verteilen. Dies empfiehlt sich bei schwächeren Systemen. Vor allem zu Entwicklungszwecken weist das Programm noch etliche Features auf, deren Aktivierung und Funktion in diesem Handbuch vorgestellt werden soll.

*(Nähere Programmspezifikationen finden sich im Pflichtenheft).*

## Inhaltsverzeichnis:

- Seite 3:           **Kommandozeilenparameter**
- Seite 4:           **Interner Viewer**
- Seite 5:           **Zusätzliche Funktionen**

## Programmstart

### Aufruf:

```
java RoboClient [ switches ] [ Teamname [ Servername [ Port ] ] ] [ switches ]
```

### Kurzbeschreibung der einzelnen Parameter:

(ausführlichere Erklärungen und Funktionen weiter unten)

Alle Parameter sind optional, jedoch müssen folgende drei in einer festen Reihenfolge angegeben werden:

Parameter	Beschreibung	Default Wert	Wertebereich*
Teamname	Teamname unter dem sich die Spieler am SoccerServer anmelden	Fuzzy_LoKick	max. 20 Zeichen (keine Leerzeichen)
Servername	Netzwerkname des SoccerServers	localhost	-
Port	Port des SoccerServers	6000	1024 bis 65534

Die übrigen (switches) können frei platziert werden - sogar zwischen den obigen. Sie beginnen alle mit einem Slash, auf den ein Kleinbuchstabe und evtl noch weitere Zeichen folgen:

Switch	kurze Beschreibung	Default Wert	Wertebereich*
/nxx	Spieleranzahl	11	1 bis 11
/v	internen Weltmodellviewer starten	Viewer deaktiviert	-
/g	ohne Torwart starten wenn das Team auf mehrere Rechner verteilt werden soll (der Viewer ist dann deaktiviert)	Spieler Nummer eins ist der Torwart	-
/tx	Servvertimeout einstellen Timeout: $2 \cdot (x + 1)$ Sekunden	4 (entspricht 10s)	0 bis 9 (entspricht 2s bis 20s)
/l	Logging aller Servernachrichten aktivieren	Logging deaktiviert	-
/dxxx	Debuginfo xxx auf der Konsole ausgeben	keine Debugausgabe (entspricht 000)	000 bis 999 (aber nur wenige belegt)
/s	nur nach see-msgs denken und handeln	in jeder Runde handeln	-
/p	Benutzerkonsole aktivieren	Konsole deaktiviert	-
/ax	Taktik x auswählen	0	0 bis 9

\* Bei Bereichsüberschreitungen wird der Defaultwert benutzt

### Beispiele:

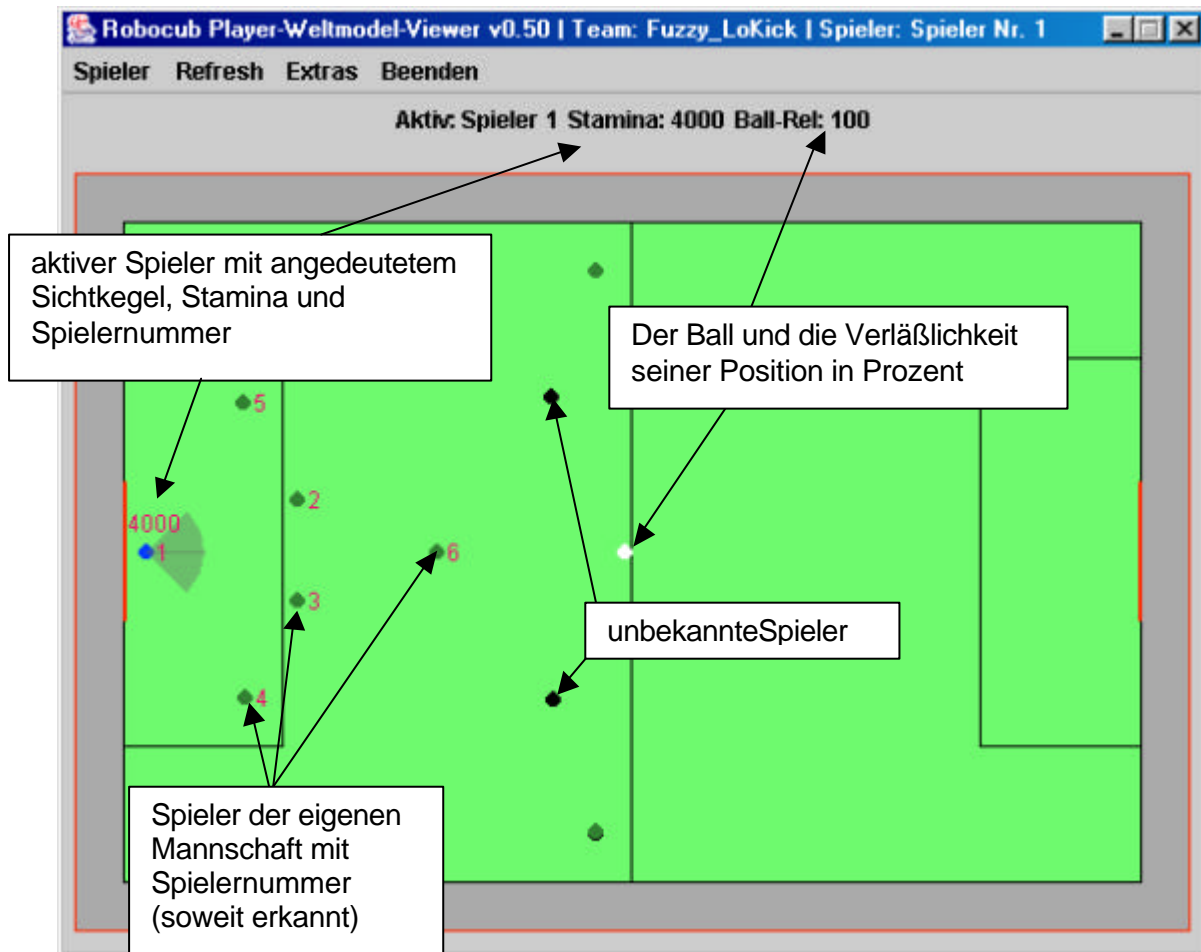
```
java RoboClient entspricht java RoboClient Fuzzy_LoKick localhost 6000
```

```
java /v Test /n2 host startet zwei Spieler und den Viewer. Das Team heißt 'Test' und der SoccerServer 'host'
```

## Interner Viewer

Unter einer grafischen Oberfläche ist es möglich, den Viewer per Kommandozeilenschalter '/v' zu starten.

Dieser visualisiert die Daten des Weltmodells und ist vor allem beim Debugging von Nutzen.



Viewer beim Programmstart

### Farbcodierung der beweglichen Feldobjekte:

blau	aktiver Spieler		weiß	Ball
grün	Mitspieler		gelb	Ball nächste Runde
rot	Gegner		grau	Mitspieler (Daten aus den jeweiligen Weltmodellen)
schwarz	nicht identifizierter Spieler			

Menüeintrag	Funktion / Optionen
Spieler	Auswahl des aktiven Spielers (dieser wird blau und mit zusätzlichen Informationen eingezeichnet)
Refresh	hier lässt sich einstellen, wie oft der Viewer neue Daten vom Weltmodell des aktiven Spielers holen soll. Per Default passiert dies nach jeder geparsten See-Nachricht, was aber weniger performante Systeme durchaus in die Knie zwingt.
Extras	'Geister ein-/ausschalten': Dadurch holt sich der Viewer die Position und Stamina der anderen Mitspieler aus deren Weltmodellen und zeichnet die Daten in einem transparenten grau ein. Außerdem wird die vom Weltmodell für die nächste Runde vorhergesagte Position des Balls gelb eingezeichnet.
Beenden	hier kann entweder nur der Viewer oder der ganze Client beendet werden

## Beschreibung der weiteren Features:

(diese dienen vor allem als Hilfsmittel zur weiteren Programmentwicklung)

### MessageLogging:

Wenn das Logging via '/' aktiviert ist, wird für jeden Spieler eine Datei der Form `_SvrMsgLog Player1.txt` angelegt. Hierin werden dann sämtliche Nachrichten vom und zum Server - mit einem Zeitstempel (Systemzeit in ms) versehen - abgespeichert.

### Aktivierbare Debuginformationen:

Um die Konsole im Normalbetrieb von den diversen Debuginfos zu befreien, haben wir ein System implementiert, bei dem diese Ausgaben per Parameter aktiviert werden können. So erspart man sich außerdem, jedesmal Zeilen auszukommentieren bzw. wieder aktivieren zu müssen.

*Die möglichen Debuginformationen finden sich unter `debuginfo.txt`.*

### Benutzerkonsole:

Wenn die Konsole aktiviert ist, kann der Client via 'quit' beendet werden.

### Auswahl verschiedener Taktiken:

Zur Zeit ist nur eine Taktik verfügbar, aber so können relativ einfach andere Aufstellungen oder die Auswahl anderer AI-Module realisiert werden.