

**Manual  
for ITASbasic 0.96**

Tarek Hassan  
([thassan@connectmail.carleton.ca](mailto:thassan@connectmail.carleton.ca))

Carleton University  
The Department of Systems and Computer Engineering  
1125 Colonel By Drive  
Carleton University  
Ottawa, Ontario  
K1S 5B6  
Canada  
+1 (613) 520 5740  
+1 (613) 520 5727

© 2002 Carleton University

## Table of Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	ITAS	4
1.2	RoboCup	4
1.3	RoboCup Soccer Simulator Server	4
<b>2</b>	<b>Getting Started</b>	<b>5</b>
2.1	What you need	5
2.2	Joining a game	5
<b>3</b>	<b>The Display</b>	<b>6</b>
3.1	The Field	7
3.2	The View Cone	7
3.3	The Players & the Ball	8
<b>4</b>	<b>The Controls</b>	<b>8</b>
<b>5</b>	<b>Developer Guide</b>	<b>9</b>
<b>6</b>	<b>Contacts &amp; More Information</b>	<b>10</b>
6.1	ITAS Development Team	10
6.2	Useful Links	10

# **1 Introduction**

## **1.1 ITAS**

ITAS = In The Agent's Shoes

This tool was developed to put a human user in the place of a RoboCup agent for the purpose of logging their actions and their reactions to different game situations. This log could then be used for machine learning for development of software agents. ITAS could also be used to train, test or challenge an AI team by allowing the human players to create desired game situations through their actions (e.g. try different attack approaches to challenge a goalie agent's adaptability).

## **1.2 RoboCup**

RoboCup is an international research initiative driven by competition. Its ultimate goal statement is, "By 2050, develop a team of fully autonomous humanoid robots that can win against the human world champion team in soccer." [1] By providing a standard and well-known challenge, RoboCup promotes AI and robotics research and innovation worldwide. An annual world cup competition is held with different leagues that focus on different aspects of the RoboCup challenge. There are four different robotic leagues and one RoboCup Simulation league. More information about RoboCup is available at [www.robocup.org](http://www.robocup.org).

## **1.3 RoboCup Soccer Simulator Server**

The RoboCup Simulation league focuses on the artificial intelligence aspect of the RoboCup project by abstracting away from the problems of robotics. The physical field environment is replaced with a simulated field, provided by The RoboCup Soccer Simulator Server (RCSSS hereon) [2]. RCSSS is a server that simulates the soccer field environment and the objects on the field. Players (the software agents) are clients that log onto this server and interact with it by receiving world information and sending commands. Essentially, each agent is the brain of a player, receiving sensory stimuli and issuing commands based on these stimuli. The basic commands available to

the agent for controlling the players actions are 'dash', which makes the player run forward with a given power, 'turn', which makes the player turn with a given moment and 'kick' which kicks the ball with a given power. Additional information about RCSSS is available at [sserver.sourceforge.net](http://sserver.sourceforge.net).

## 2 Getting Started

### 2.1 What you need

First, to run ITAS, you will need a Java Runtime Environment (J2RE 1.4 or higher is recommended) and the ITAS software. If a game server is already running, all one needs is the ITAS software, and the server's address and port number. If you wish to start your own server, you will also need the RCSSServer software and a monitor program to setup and control the server environment.

What you need	Where to get it
ITASbasic (latest version)	<a href="http://itas.sourceforge.net">itas.sourceforge.net</a>
J2RE 1.4 or higher	<a href="http://sun.java.com/j2se">sun.java.com/j2se</a>
RCSSServer 8.0 or higher (if running a server)	<a href="http://sserver.sourceforge.net">sserver.sourceforge.net</a>
RCSS monitor (if running a server)	<a href="http://sserver.sourceforge.net">sserver.sourceforge.net</a>

### 2.1 Joining a game

- 1- Make sure that your JRE is properly installed and that your CLASSPATH is set to the right location.
- 2- Unzip ITASbasicx.xx-bin.zip (or compile ITASbasicx.xx-src) into a directory of your choice.
- 3- From a terminal or prompt, go to your ITAS directory and run ITAS using the following command line:

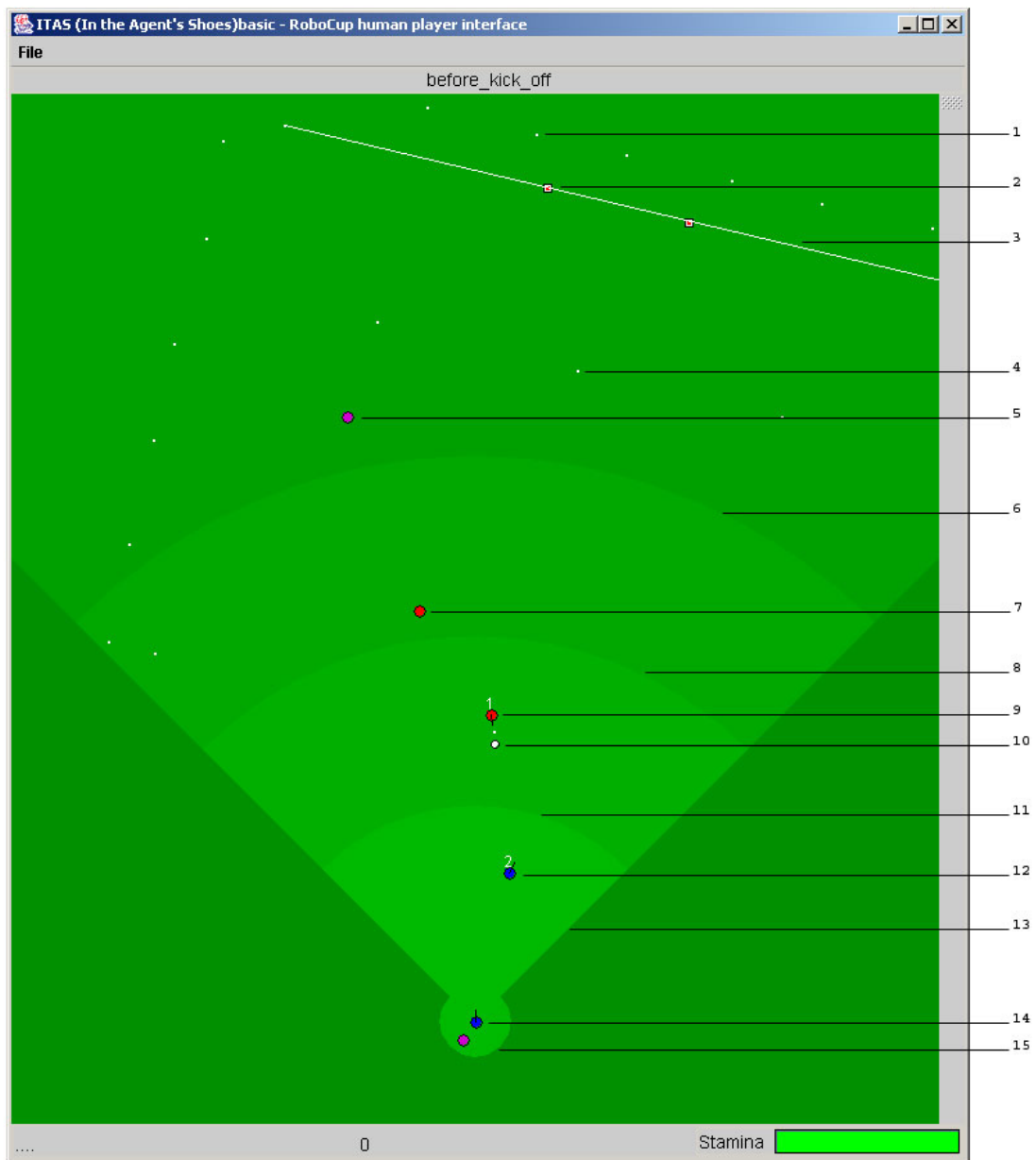
```
java ITAS -host <hostname> -port <portnumber> -team  
<teamname>
```

(default: hostname="", port="6000", team="GreyMatterUtd")

### 3 The Display

Your player's view is displayed in a bird's-eye perspective (top-down). Everything in the display is relative to your on-screen player(14). i.e. the player and their view cone are static and everything else moves relative to them. In this version of ITAS the user is restricted to seeing only what the agent can see.

This is a screenshot of the ITAS interface.



1. outer flag
2. goal post
3. field line
4. inner flag
- 5,7,9,12. other players
- 6,8,11. vision quality ranges
10. ball
13. view cone
14. user's player
15. peripheral vision range

### **3.1 The Field**

The field is defined by flags and lines. Field lines(3) define the outer limits of the field and are only visible when directly in front of the player. This is why only one line is visible in the screenshot(the red team's goal line), not two. Flags are markers located on and outside of the field. 36 flags(1) are located around the outside of the field, 6 flags mark the actual perimeter of the field, 4 flags indicate the goalposts(2) for each side (the colour of the goalpost depends on the side) and 7 flags(4) mark key points on the field (center field and penalty boxes).

### **3.2 The View Cone**

The view cone(13) indicates the limits of the player's normal vision. This cone is further broken down into different ranges of vision quality. These can be differentiated by their different shades of green.

Beyond the farthest range(6), nothing is known about the player except for that they are a player. In the next farthest range(8), the player's team can be identified, but not consistently. In the second closest range, the player's team can be identified consistently and the player's number and direction can be identified inconsistently. Within the closest range(11), a player's number, direction and team are always identifiable.

The circle around the user's player indicates their peripheral vision(15). If an object is outside of the view

cone and in this circle, it is visible, but no details about the object are available.

### **3.3 The Players & the Ball**

A player is drawn as a circle with a black outline. Depending on the position and distance of the player, their player number(white numbers), the direction they are facing (marked by a small black dash) and their team colour (left=blue, right=red & unknown=purple) are visible. Different player representations are seen in the screenshot above (12, 9, 7 & 5).

The ball(10) is a small white circle with a black outline.

## **4 The Controls**

### **Dashing:**

Clicking the left mouse button will make your player dash forwards with a power relative to the distance of the mouse click from your player on screen.

### **Turning:**

Clicking the right mouse button will make the player try to turn to face the point where the mouse was clicked. The player will turn with a moment equal to the angle of the mouse click from the imaginary line straight ahead (upwards) of your player.

### **Kicking:**

Pressing the space-bar will make your player kick the ball towards the point where the mouse was clicked with a power relative to the distance of the mouse pointer from your player (like dash) and an angle equal to the angle of the mouse pointer (like turn).

To kick the ball, you must make contact with the ball. The efficiency (% of applied power) of the kick is affected by two factors, the distance and the positioning of the ball. If the ball is just barely in contact with the player, up to 25% power can be lost. The closer the ball is to the middle of the player, the more efficient the kick. Power



is also lost if the ball is not directly in front of the player (up to 25% if the ball is directly behind the player). So the least efficient kick would be when the ball is just barely in contact with and directly behind the player, losing 50% of the kicking power.

## **5 For Developers**

A developer guide for adding advanced functionality beyond that provided by ITASbasic will be available with ITASbasic1.0 (coming soon).

## **6 Contacts & More Information**

### **6.3 ITAS Development Team**

ITAS was developed in the Department of Systems and Computer Engineering at Carleton University in Ottawa, Ontario Canada by **Tarek Hassan** and **Paul Marlow** under the supervision of **Dr. Babak Esfandiari**.

For support or for any inquiries regarding ITAS, please contact:  
*thassan@connectmail.carleton.ca*

### **6.4 Useful Links**

**For more ITAS information, the latest news and releases of ITAS:**

*<http://Itas.sourceforge.net>*  
(official ITAS website)

*<http://www.sourceforge.net/projects/itas/>*  
(ITAS SourceForge page)

**For more information on RoboCup in general:**

*<http://www.robocup.org>*  
(official RoboCup site)

*<http://www.sce.carleton.ca/netmanage/robocup/>*  
(Carleton's RoboCup site - a little old, but has useful information and downloads)

**For more RCSSS information, news and releases of RoboCup simulation:**

*<http://sserver.sourceforge.net>*  
(official RCSSS website)

## References

- [1] <http://www.robocup.org>, RoboCup website
- [2] <http://sserver.sourceforge.net>, RCSS website