

ReadMe for Robocode

Updated 04-Mar-2012 by Flemming N. Larsen

Send all feedback, comments, and issues to the tracker at the [Robocode project](#) at SourceForge.net.

Robocode Home Page: <http://robocode.sourceforge.net/>

TABLE OF CONTENTS

1. [What is Robocode?](#)
2. [History of Robocode](#)
3. [System Requirements](#)
4. [Getting Started](#)
5. [Robocode API](#)
6. [Robocode Repository](#)
7. [Community](#)
8. [Challenges](#)
9. [Competition](#)
10. [Command Line](#)
11. [Links](#)
12. [Reporting Defects](#)
13. [Feature Requests](#)
14. [Versions](#)
15. [News](#)
16. [How to contribute](#)

1. WHAT IS ROBOCODE?

Robocode is a programming game where the goal is to develop a robot battle tank to battle against other tanks in Java or .NET. The robot battles are running in real-time and on-screen.

The motto of Robocode is: Build the best, destroy the rest!

Besides being a programming game, Robocode is used for learning how to program, primarily in the Java language, but other languages like C# and Scala are becoming popular as well.

Schools and universities are using Robocode as part of teaching how to program, but also for studying artificial intelligence (AI). The concept of Robocode is easy to understand, and a fun way to learn how to program.

Robocode comes with its own installer, built-in robot editor and Java compiler, and only pre-requires a Java Virtual Machine (JVM) to exist on the system where it must be installed. Hence, everything a robot developer needs to get started is provided with the main Robocode distribution file (`robocode-xxx-setup.jar`). Robocode also supports developing robots using external IDEs like e.g. [Eclipse](#), [IntelliJ IDEA](#), [NetBeans](#), [Visual Studio](#) etc., which supports the developer much better than the robot editor in Robocode.

The fact that Robocode runs on the Java platform makes it possible to run it on any operating system with Java pre-installed, meaning that it will be able to run on Windows, Linux, Mac OS, but also UNIX and

variants of UNIX. Note that Java 5.0 or newer must be installed on the system before Robocode is able to run. See the [System Requirements](#) for more information.

Be aware that many users of Robocode (aka Robocoders) find Robocode to be very fun, but also very addictive. :-)

Robocode comes free of charge and is being developed as a spare-time project where no money is involved. The developers of Robocode are developing on Robocode because they think it is fun, and because they improve themselves as developers this way.

Robocode is an Open Source project, which means that all sources are open to everybody. In addition, Robocode is provided under the terms of [EPL](#) (Eclipse Public License).

2. HISTORY OF ROBOCODE

The Robocode game was originally started by Matthew A. Nelson, aka Mat Nelson, as a personal endeavor in late 2000 and became a professional one when he brought it to IBM, in the form of an AlphaWorks download, in July 2001.

In the beginning of 2005, Robocode was brought to SourceForge as Open Source with Robocode version 1.0.7. At this point, the development of Robocode had somewhat stopped. In the meanwhile, the community around Robocode began to develop their own versions of Robocode in order to get rid of bugs and also to put new features into Robocode, the Contributions for Open Source Robocode and later on the RobocodeNG project by Flemming N. Larsen.

As nothing seemed to happen with Robocode in more than a year, Flemming N. Larsen took over the Robocode project at SourceForge as administrator and developer in July 2006. The RobocodeNG project was dropped, but the Robocode 2006 variant, which contained a lot of contributions from the Robocode community, was merged into the official Robocode with version 1.1. Since then, lots of new versions of Robocode have been released with more and more features and contributions from the community.

Recently (from version 1.7.2.0), the .NET platform is supported through a .NET plug-in provided by Pavel Savara based on [jni4net](#), also developed by Pavel Savara.

3. SYSTEM REQUIREMENTS

In order to run Robocode, Java 5.0 Standard Edition (SE) or a newer version of Java must be installed on your system. Both the Java Runtime Environment (JRE) and the Java Developer Kit (JDK) can be used. Note that the JRE does not include the standard Java compiler (javac), but the JDK does. However, Robocode comes with a built-in compiler (ECJ). Hence, it is sufficient running Robocode on the JRE.

Also note that it is important that these environment variables have been set up prior to running Robocode:

- `JAVA_HOME` must be setup to point at the home directory for Java (JDK or JRE).
Windows example: `JAVA_HOME=C:\Program Files\Java\jdk1.5.0_22`
UNIX, Linux, Mac OS example: `JAVA_HOME=/usr/local/jdk1.5.0_22`
- `PATH` must include the path to the `bin` of the Java home directory (`JAVA_HOME`) that includes `java.exe` for starting the Java virtual Machine (JVM).
Windows example: `PATH=%PATH%;%JAVA_HOME%`

UNIX, Linux, Mac OS example: `PATH=${PATH}:${JAVA_HOME}/bin`

You can read more details from here:

- [System Requirements](#)

If you want to program robots in .NET or control Robocode from a .NET application, you need to install the Robocode .NET API plug-in on top of Robocode. The plug-in is installed by double-clicking the `robocode.dotnet-xxx-setup.jar` the same way as Robocode itself is installed.

4. GETTING STARTED

Most documentation about Robocode is provided thru the [RoboWiki](#), which contains the official documentation about Robocode, but which also hosts the community around Robocode. It is highly recommended to read the articles on the RoboWiki for getting started with Robocode. These articles are provided from here:

- [Getting Started](#)

You should read about the anatomy of a robot, the game physics, scoring etc.

To learn more about developing robots for .NET, these articles are a good starting point:

- [Create a .NET robot with Visual Studio](#)
- [Debug a .NET robot with Visual Studio](#)

5. ROBOCODE API

The Robocode API is provided as HTML pages for both the Java and .NET platform.

- [Java API](#)
- [.NET API](#)
- [.NET Control API](#)

The Robocode API actually consists of 3 different APIs.

- **Robot API:** Within the Java package `robocode` and .NET namespace `Robocode`.
The Robot API is used for developing robots, and is the only part of the API that robots are allowed to access.
- **Robot Interfaces:** Within the Java package `robocode.robotinterfaces` and .NET namespace `Robocode.RobotInterfaces`.
The Robot Interfaces are used for developing new robot types with a different API than the standard Robot API.
Note: The game rules and robot behaviors cannot be changed.
- **Control API:** Within the Java package `robocode.control` and .NET namespace `Robocode.Control`.
The Control API is used for letting another application start up battles with selected robots in Robocode and retrieve the results. It is also possible to get snapshots of robots and bullets (like position, heading, energy level etc.) at a specific time in a battle.

6. ROBOCODE REPOSITORY

If you want to try out new robots than the sample robots that come with Robocode, you should visit the [Robocode Repository](#).

Robots are available under the [Bots](#) section of the repository.

The Robocode Repository is developed and maintained by David Lynn as a project independently of the Robocode project.

7. COMMUNITY

The community around Robocode is using the RoboWiki as communication channel. At the RoboWiki, people share new ideas, code snippets, algorithms, strategies, and lots of other stuff about Robocode. New official documentation from the developers of Robocode will be put at the RoboWiki as well.

On the RoboWiki, these strategies are provided:

- [Radar](#)
- [Movement](#)
- [Targeting](#)

The code snippets are also provided on the RoboWiki:

- [Code Snippets](#)

8. CHALLENGES

A good way to improve you self as a robot developer is to try out some real challenges. On the RoboWiki, two famous challenges exist for testing/studying a robots movement, targeting, and gun abilities:

- [Movement Challenge](#)
- [Targeting Challenge](#)
- [RoboRumble Gun Challenge](#)

9. COMPETITION

If you want to challenge your robot(s) and yourself as robot developer, the [RoboRumble@Home](#) is the best way to do it. RoboRumble is the ultimate collaborative effort to have a live, up-to-date ranking of Robocode bots. It uses the power of available Robocoders' computers to distribute the effort of running battles and building the rankings.

RoboRumble is actually 3 different rumbles:

- **RoboRumble** (aka 1v1): One robot versus another robot - both picked at random. These two robots a alone on the battle field.
- **MeleeRumble**: Ten robots picked at random all battle against each other..
- **TeamRumble**: One team versus another team - both picked at random. Each team consists of five or less robots.

In order to get started with RoboRumble, you should read this page:

- [Starting With RoboRumble](#)

Note that the RoboRumble@Home client is built into Robocode and can be started using the batch/shell/command files:

	Windows	UNIX / Linux	Mac OS
RoboRumble	<code>roborumble.bat</code>	<code>roborumble.sh</code>	<code>roborumble.command</code>
MeleeRumble	<code>meleerumble.bat</code>	<code>meleerumble.sh</code>	<code>meleerumble.command</code>
TeamRumble	<code>teamrumble.bat</code>	<code>teamrumble.sh</code>	<code>teamrumble.command</code>

Two other competitions exists which are:

- [Twin Duel](#): Two teams battle on an 800x800 field. Each team consists of two robots (twins).
- [Hat League](#): Two teams not knowing each other are paired together at random (like drawing names from a hat). Each team consists of two robots. These two teams must work together and defeat two other teams that have also been picked at random.

10. COMMAND LINE

It is possible to specify options and predefined properties from the command-line when running Robocode. The usage of these can be listed by writing this from a command prompt or shell:

```
robocode -help
```

For example, it is possible to:

- disable the graphical user interface (GUI).
- disable security that is specific to Robocode (but does not override the security that comes with the JVM).
- enable/disable the debugging mode, useful when debugging robots.
- play a battle based on an existing Robocode .battle file.
- replay a recorded battle visually.
- save the results of battles in a comma-separated file.

You can read more details here:

- [Console Usage](#)

11. LINKS

Links relevant to Robocode are provided on the home page of Robocode:

- [Home Page of Robocode](#)

Other links are provided from the RoboWiki - especially for challenges and competitions:

- [RoboWiki](#)

There is also a Wikipedia page available about Robocode, which provides good links to movement and targeting strategies, competitions, and other sites about Robocode:

- [Robocode on Wikipedia](#)

12. REPORTING DEFECTS

If you discover a defect (bug) in Robocode you are encouraged to report the issue as soon as you discover it - the sooner the better.

A bug report should be put on the [Bug Tracker](#) on the SourceForge site for Robocode. Each bug report will be prioritized among other bug reports depending on its impact on the game.

It will be a great help if you describe how to see or which steps to do in order to reproduce the defect. You are very welcome to provide a screen shot, source code or anything else that will show the bug. It is also a very good idea to write which system and version of Robocode and Java you are using.

If you are a registered user at SourceForge (register [here](#)) you will be able to add a "monitor" on your bug report. This way you will be able to receive notifications when someone add comments to your report, but will also be able to better track the current status of the bug, e.g. when the bug is fixed and with which version of Robocode the fix is available.

If you are a developer yourself, and have a good idea of how the bug could be fixed, you are more than welcome to do so. By fixing the bug, you will become a contributor of Robocode yourself. You can learn more about how to contribute [here](#). Note that we accept bug fixes under the terms of [EPL](#).

13. FEATURE REQUESTS

If you got an idea for a new feature or improvement for Robocode, you are very welcome to share your idea by submitting a feature request.

A feature request should be put on the [Feature Requests Tracker](#) on the SourceForge site for Robocode. Each feature request will be prioritized among other feature requests.

It will be a great help if you describe your idea in detail, and how you think it could be implemented into Robocode. For example, will it be possible to extend an existing feature with your idea?

If you are a registered user at SourceForge (register [here](#)) you will be able to add a "monitor" on your request. This way you will be able to receive notifications when someone add comments to your request entry, but will also be able to better track the current status of your entry, e.g. when the feature has been implemented and with which version of Robocode it will be available.

If you are a developer yourself, and have a good idea of how the feature could be implemented, you are more than welcome to do so if the feature is being accepted. By implementing the feature, you will become a contributor of Robocode yourself. You can learn more about how to contribute [here](#). Note that we accept implementations under the terms of [EPL](#).

14. VERSIONS

Robocode is continuously under development, and you should be aware that three different release types

exist for Robocode:

- **Alpha** version is an unofficial "snapshot" version of Robocode that is under development and not feature complete yet. It is normally provided for the person that has put a bug report or feature request on the tracker for Robocode, which needs to be verified. Alpha versions are not meant to be redistributed at all.
- **Beta** version is an official release that is considered feature complete, and which intended for everybody to try out and test regarding all new features and changes, but which is also considered "unstable", and might contain some unwanted side-effects due to the changes made to this version compared to the last final version of Robocode. Defects or undesired behaviors should be reported as soon as they are discovered so the issues can be fixed before the final release. Everybody is encouraged to take part in the testing Beta versions of Robocode.
- **Final** version is (as it says) the final version of a specific and official release of Robocode, and where new features and bug fixes have been tested. Note, that the release will not state "final" on the distribution files for Robocode like it is the case for Alpha and Beta versions.

15. NEWS

News about Robocode is put on the blog spot for Robocode. Here it is possible subscribe to a RSS feed to receive news about Robocode.

- [Robocode News](#)

You can also follow Robocode on Twitter and Facebook here:

- [Twitter for Robocode](#)
- [Robocode on Facebook](#)

The RoboWiki can be followed on Twitter as well:

- [Twitter for RoboRumble](#)

16. HOW TO CONTRIBUTE

If you want to contribute to Robocode with e.g. a new feature or bug fix, you should read the [Developers Guide for building Robocode](#).

Note that we accept code changes under the terms of [EPL](#).

There exist no or little documentation about the internals of Robocode, and the code base will need to be examined by a contributor in order to get an insight of how Robocode is implemented. Thus, it required a skilled Java developer to figure out how Robocode is put together.

Robocode is divided into several modules. You can read Pavel Savara's blog to get a good overview of Robocode here:

- [Robocode modules](#)

Help for Robocode internals can be provided through the [Robocode Developers Discussion Group](#) where

you can register yourself, and start up a new topic. This is the best way of getting information and asking about details for the internals in Robocode.

If a contribution is a somewhat small change to involves under 10 files, then the preferred way is to provide the contribution as a patch file that can be applied to the existing Robocode sources. Otherwise, you can be granted a Subversion branch for your work, where you can commit your changes. Later, this branch needs to be merged into the trunk be the administrators of Robocode and tested carefully. Additional work might be done by other Robocode developers to finalize the work.