DriftingDroids
yet another Ricochet Robots solver program

Version 1.3.2 (2014-03-23)

Written by Michael Henke <smack42@googlemail.com>

Homepage https://github.com/smack42/DriftingDroids

Contents

Welcome

Ricochet Robots is a board game designed by Alex Randolph. If you don't know the game yet then you can start to read about it here:

Wikipedia (en)	https://en.wikipedia.org/wiki/Ricochet_Robot
Wikipedia (de)	https://de.wikipedia.org/wiki/Rasende_Roboter
BoardGameGeek (en)	http://boardgamegeek.com/boardgame/51/ricochet-robots

DriftingDroids is a computer version of the Ricochet Robots board game. It includes a **solver** algorithm that finds the **optimal solutions** to every game problem. You can use it as your **trainer** for solo playing or as a **referee** during real board gaming sessions.

How to install and run

Unpack the Zip archive you've downloaded to a directory of your choice.

Double-click the file **start.jar** to run DriftingDroids.

Java SE (JRE version **6 or newer**) is required to run DriftingDroids.

Windows users can download the latest version of Java here:
http://www.oracle.com/technetwork/java/javase/downloads/index.html

On other systems, such as **Mac OS X** or **Linux**, just use a Java Runtime Environment that is already installed or that can be installed easily, such as Java for Mac OS X or IcedTea / OpenJDK.

Advanced usage

The program can be started from the command line like this:

cd <path> (you had unpacked the archive here)

cd DriftingDroids_1.3.2

java -jar start.jar

The integrated program launcher (AppStart) evaluates the file **appstart.properties**, which you can modify using a standard text editor, for instance if you want to increase the amount of **memory** available to the program (option **-Xmx**) or if you want to set a certain **language** (option **user.language**).

The program launcher can be bypassed and the program be started directly, which allows you to use advanced Java VM options:
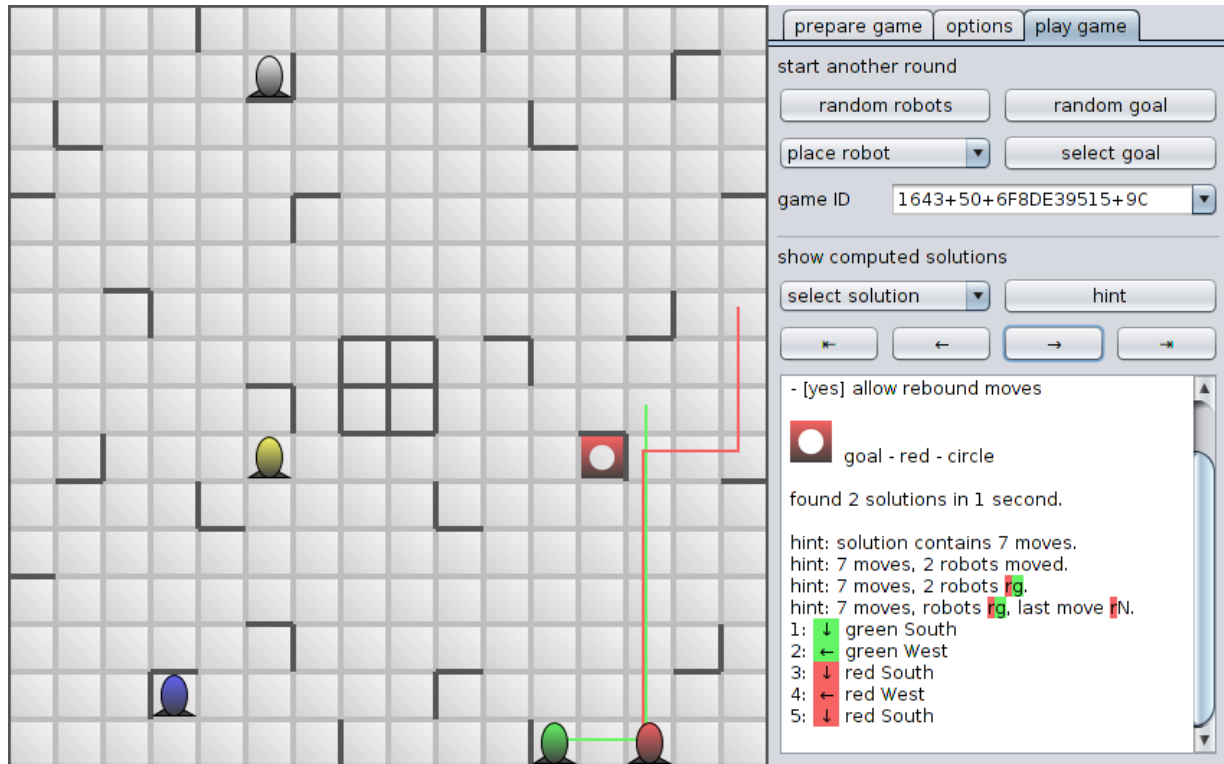
cd DriftingDroids_1.3.2

cd lib

**java [Java VM Options] -jar driftingdroids.jar**

Play game

The program starts in **play game** mode. You can **start another round** by setting robots and / or goal to new positions.


- **random robots** The program places all robots on random board positions.

- **place robot** First choose a robot color from the list and then click on the new board position where you want to put this robot. Repeat this for all robots you want to place.

- **random goal** The program selects one of the available goals at random.

- **select goal** First click this button and then select one of the available goals on the board by clicking on it.

- **game ID** The program generates a unique Game ID for each game configuration, which includes the layout of the 4 board tiles and the positions of robots and goal. You can select a Game ID from the list or you can enter (or copy&paste) a Game ID into this text box. You can copy the Game ID of an interesting game problem and save it in a text file or send it to a friend who can run it in her copy of DriftingDroids.

The solver runs in the background whenever you change the starting position (as described above) and presents its results in the lower part of the control area.

Use these controls to examine the **computed solutions**:

- **select solution** For some game configurations the solver finds more than one optimal solution. This list of solutions is sorted according to the option **prefer solution with minimum / maximum number of robots in motion** on the options page. The best solution is situated on top of the list and is selected by default.

- **hint** Gives a hint: how many moves are in the solution. Click again for a second hint: how many moves and how many moved robots are involved in the solution. Click again for the third hint: how many moves and which robots are moved. Click again for the fourth hint: how many moves, which robots are moved and what's the last move of the solution.

- ⇤ Undo all shown moves.

- ← Undo last shown move.

- → Show next move.

- ⇥ Show all moves.

The functions **place robot** and **select goal** are also available via **context menu**. To use it, you just click the appropriate mouse button (usually the right button) on the desired position of the board.
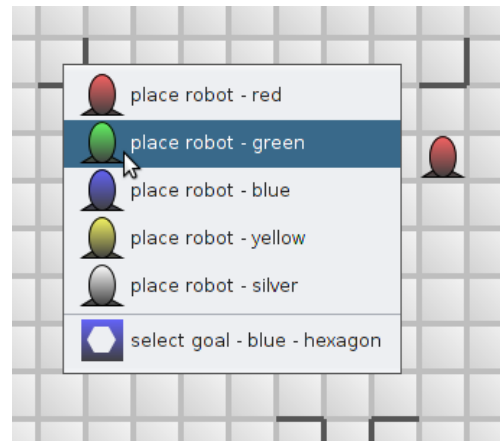
Most of the functions mentioned above are also available as **keyboard shortcuts** (hotkeys). The keys are displayed as part of the tooltip that appears when the mouse pointer hovers over a control. (for example, press **Ctrl-H** for **hint**)

Prepare game

Here you can arrange the 4 board tiles and change the number of robots. The board view on the left side has all goals made visible so that you can recognize the Ricochet Robots board tiles more easily.
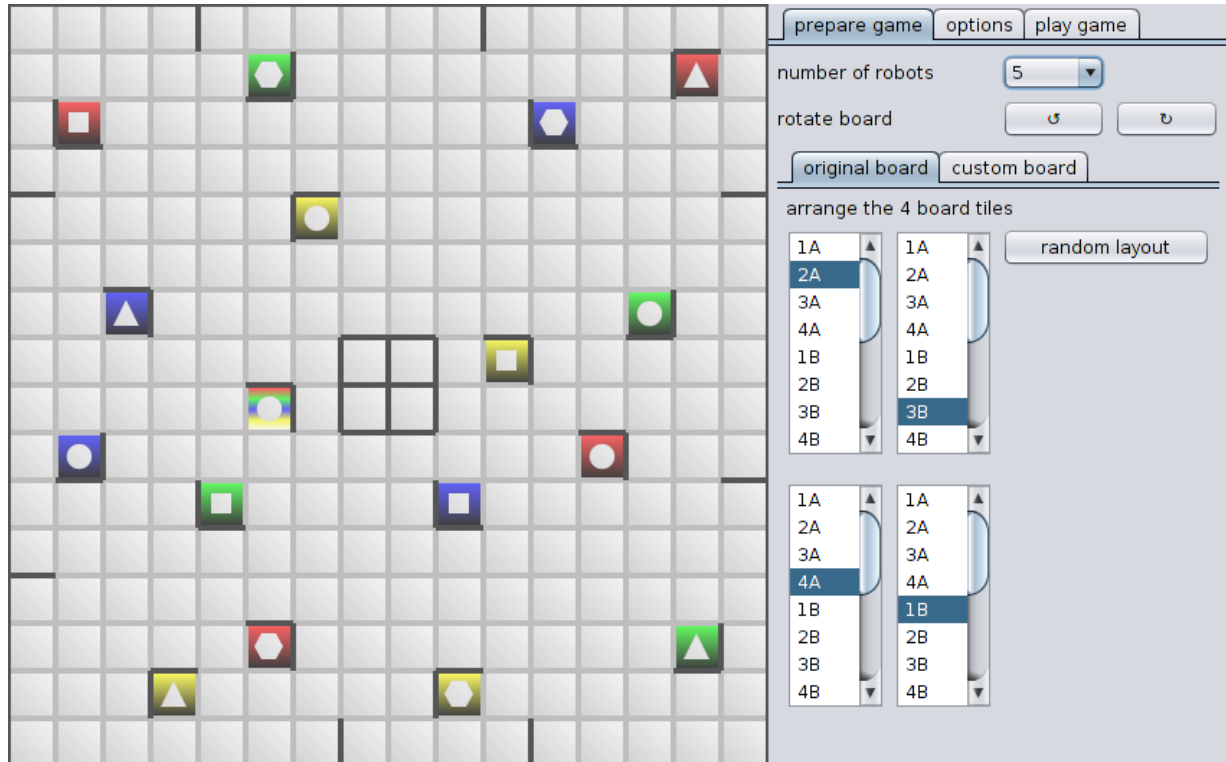
Select from the list the **number of robots** that will be present on the board. The first edition of the Ricochet Robot(s) board game came with 4 robots (red, green, blue, yellow), while newer editions of the game include a 5th (silver) robot.

By clicking ↺ or ↻ you rotate the board by 90 degrees in the respective direction.

original board

The board tiles in DriftingDroids are modeled after the tiles of the original Ricochet Robots board. They are simply named 1, 2, 3, 4 and the 2 x two sides front / back are named A / B / C / D. (see photos in directory doc)

Click on button **random layout** to let the program select a layout: all four different tiles will be used, they are placed at random positions with a random side (A, B, C or D) facing up.

You can arrange the board tiles manually by selecting any entry 1A … 4D in each of the four lists. The program imposes no restrictions on the possible combinations of tiles, so you can build a board from four copies of the same tile if you want to.

custom board

Here you can create board layouts that don't exist in Ricochet Robots. Every wall and every goal can be set or removed individually. To do this, just click on the desired positions of the board.

By clicking the **primary mouse button** (usually the left one) you **set the walls** that are highlighted in white (1 or 2) or, if the pointer is over the center of a cell then you **set the currently selected goal**. (color and shape are specified in the two lists)

By clicking the **secondary mouse button** (usually the right one) or **shift key + primary** mouse button you **remove** the highlighted **walls** or the **goal** under the mouse pointer.

You can clear the board by using `delete walls` and `delete goals`.

The `width (columns)` and `height (rows)` of the board can be adjusted freely, but the maximum supported size of the board is **4096** fields, which is equivalent to 64 x 64 or 100 x 40 fields.

Note: The outer walls at the edges of the board cannot be deleted.
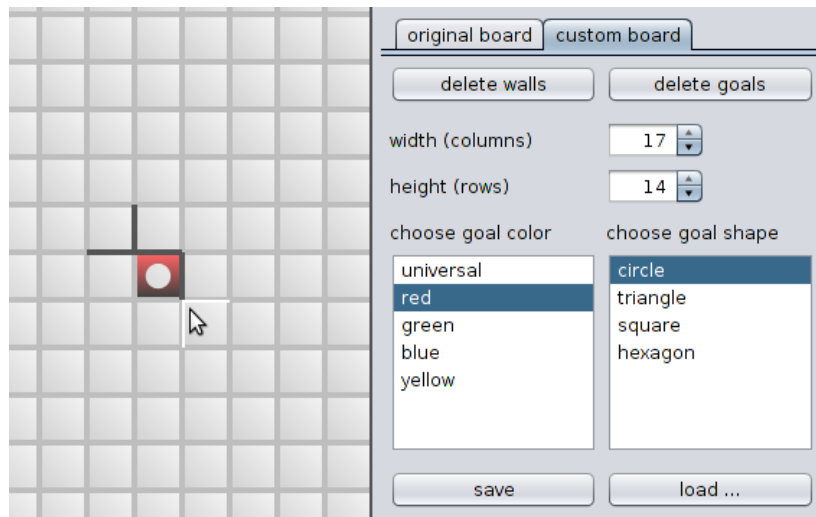
When you click `save` the current board will be **copied** to the **clipboard** as **text**, which you can paste into a text document or e-mail and then save it.

After you click on `load...` an input box opens where you can **paste** from the clipboard a board that had been saved as **text**.

Options

The solver algorithm currently offers two options:

- **prefer solution with minimum / maximum number of robots in motion** Select **minimum** or **maximum** to let the solver finish the search for all optimal solutions and sort the list of solutions accordingly. (affects list **select solution** on tab **play game**)

- **allow rebound moves** Some people like to play Ricochet Robots by stricter rules that don't allow a robot to move back the way it came. Using DriftingDroids one can find that rebound moves make some solutions shorter. (example: game ID **0765+42+2E21BD0F+93** has 24-moves optimal solutions when rebound moves are allowed and 26-moves optimal solutions without rebounds)   There have been some discussions about these rebound moves in online forums like this: http://boardgamegeek.com/forum/22741/ricochet-robots/rules
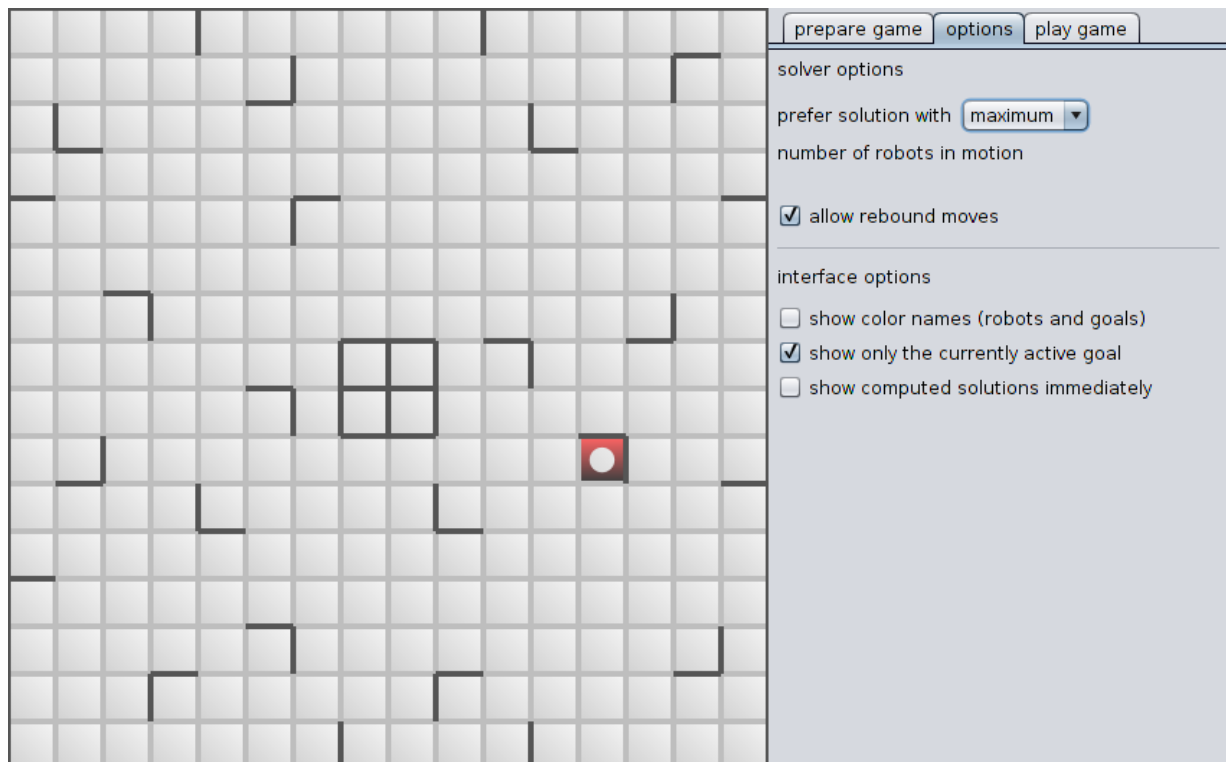
Currently there are three interface options you can set:

- **show color names** When active, the robots and goals are marked with the initial letter of their respective color names.

- **show only the currently active goal** When active, only one goal is visible on the board.

- **show computed solutions immediately** When active, the computed solutions will be shown automatically, as if you had clicked on button ⇥ (show all moves).

Acknowledgements

This program uses the following Java libraries:

- DesignGridLayout http://designgridlayout.java.net/
- AppStart http://code.google.com/p/appstart/

These tools are used to create this program:

- Eclipse http://www.eclipse.org/
- ProGuard http://proguard.sourceforge.net/

Thanks for ideas and inspiration go to:

- my sister who introduced the Ricochet Robots board game to me.
- all people who published review articles, scientific papers or computer programs about the Ricochet Robots game.
- David Hansel for sending me the sources of his fast solver.
- Michael Fogleman for his program with new ideas for a better solver algorithm.
- Jay (x11x@pisem.net) for contributing the Russian translation.

License

```
DriftingDroids - yet another Ricochet Robots solver program.
Copyright (C) 2011-2014 Michael Henke <smack42@googlemail.com>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see <http://www.gnu.org/licenses/>.
```