

The Killbots Handbook

Parker Coates



The Killbots Handbook

Contents

1	Introduction	5
2	How to Play	6
2.1	Basics of Gameplay	6
2.2	Rounds, Scoring and Energy	8
2.2.1	Rounds	8
2.2.2	Scoring	8
2.2.3	Energy	8
2.3	Special Actions	9
2.3.1	Teleport	9
2.3.2	Teleport Safely	9
2.3.3	Use the Vaporizer	9
2.3.4	Wait Out Round	9
3	Interface Overview	11
3.1	Movement Controls	11
3.1.1	Keyboard Movement Controls	11
3.1.2	Mouse Movement Controls	11
3.2	Menu Items	11
4	Developer's Guide to Killbots	13
4.1	Creating Custom Killbots Themes	13
4.1.1	The Desktop File	13
4.1.2	The Preview File	13
4.1.3	The SVG File	14
4.1.4	Installing Custom Themes	15
4.2	Creating Custom Game Types	15
4.2.1	The Ruleset Desktop File	15
4.2.2	Installing Custom Game Types	17
5	Credits and License	18

Abstract

Killbots is a KDE port of the classic BSD console game **robots**.

Chapter 1

Introduction

GAME TYPE:
Strategy

NUMBER OF PLAYERS:
One

Killbots is a simple game of evading killer robots. Who created the robots and why they have been programmed to destroy, no one knows. All that is known is that the robots are numerous and their sole objective is to destroy you. Fortunately for you, their creator has focused on quantity rather than quality and as a result the robots are severely lacking in intelligence. Your superior wit and a fancy teleportation device are your only weapons against the never-ending stream of mindless automatons.

NOTE:
This documentation frequently refers to 'robots', 'junkheaps', etc., but Killbots is fully themeable. That means the game sprites you see while playing might not match the futuristic themed ones described here. The game play and strategies, however, will remain the same, even if your hero is a carrot being chased through a garden patch by ravenous rabbits.

NOTE:
Killbots supports multiple game types which control the game parameters and determine which game features are included. This documentation attempts to be generic and not specific to any one game type. Check the description and details of your current game type in the Configure Killbots dialog for more detailed information on game mechanics with respect to grid size, rounds, scoring, energy, etc.

Chapter 2

How to Play

2.1 Basics of Gameplay

OBJECTIVE:

Survive successive rounds of enemies while accumulating a high score.



Killbots is a turn-based game, played on a rectangular *grid*. The number of *cells* in each row and column in the grid is determined by the game type being played. Each cell can contain one of the following:



The Killbots Handbook

The hero is the unfortunate soul stuck in a room with a bunch of killer robots. As the player, it is your job to ensure that he remains alive for as long possible.

At the beginning of each *turn*, the hero can perform one of three types of actions: he can stay where he is, he can move to an unoccupied adjacent cell, or he can perform a special action. See and .



Robot

Robots are mechanical thugs desiring only to crush the hero with their metallic girth. They have never heard of the “Three Laws of Robotics” and would probably crush anyone who tried to explain them.

Each robot will take a single step toward the hero on every turn. If the robot is above the hero, it will step down. If the robot is to the left of the hero, it will step right. If the robot is below and right of the hero, it will step diagonally, up and left. Should a robot manage to reach the hero’s cell, the hero will be killed and the game will end.



Fastbot

Fastbots are a much speedier version of the basic robot. For each move the hero makes, they take two steps. Fortunately, their increased mobility was not paired with increased intelligence; they often only succeed in destroying themselves twice as quickly as their slower brethren.

Not all game types include fastbots.

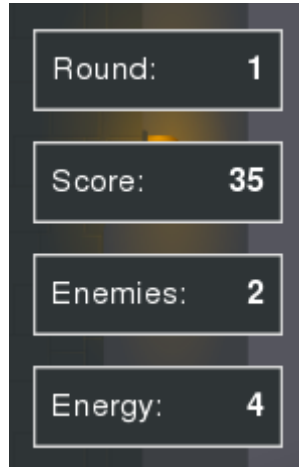


Junkheap

When two or more robots collide, they are destroyed and the resulting debris produces a junkheap. Enemies blindly ignore any junkheaps in their path and are destroyed upon crashing into them. You will find junkheaps to be extremely valuable, as they can serve as barricades between the hero and his attackers.

Some game types permit the hero to *push* junkheaps around the grid. The hero can then use junkheaps to construct ‘bunkers’ that offer protection on multiple sides. Junkheaps can even be used offensively in some game types. Pushing a junkheap on top of an enemy squashes it and can even result in a bonus.

2.2 Rounds, Scoring and Energy



2.2.1 Rounds

Gameplay in Killbots is divided into *rounds*. Each round begins with the hero in the middle of the grid and with enemies placed randomly around him. The round ends when all enemies have been destroyed. As each round is completed, the number of enemies increases. The number and type of the enemies added is game type dependent.

The grid will continue to fill with enemies until one of two things occurs: enemies occupy more than 50 percent of the cells or a safe teleport is attempted, but there are no safe cells in the grid. Once either of these occurs, the current round is abandoned and the enemy counts are reset to those used in the first round.

2.2.2 Scoring

There really is no such thing as winning in Killbots; the hero attempts to survive as long as possible, but the depressing reality is that, in the end, his luck will run out. Therefore, the goal in Killbots is to accumulate the highest *score* possible before the inevitable occurs. The ten highest scores (along with the round reached) are recorded in the high score dialog.

Points are awarded for destroying enemies. Fastbots are usually worth more than regular robots. Bonus points may also be awarded for enemies destroyed while [pushing a junkheap](#) or [waiting out the round](#).

2.2.3 Energy

In addition to points, the hero can also collect *energy* as the game progresses. Energy allows the hero to perform [safe teleports](#) and/or use the [vaporizer](#). Depending on the game type, energy can be awarded for a variety of reasons including completing a round, [pushing a junkheap](#) or destroying enemies while [waiting out the round](#).

The amount of energy that the hero can store is capped at a game type determined maximum. In some game types the maximum increases as rounds are completed. The energy status display is only visible if the current game type includes energy and safe teleports.

2.3 Special Actions

In addition to simple movement from cell to cell, the hero has up to four special actions at his disposal.

2.3.1 Teleport

The hero somehow managed to get his hands on a personal teleporter. Unfortunately, when operated without an energy source, it is not very reliable. Activating the teleporter causes the hero to be transported to a randomly selected cell anywhere on the grid. There is no limit to the number of teleports the hero can perform.

The destination cell is guaranteed to be unoccupied, but there is a very real chance that it will be adjacent to an angry robot. Due to the risk inherent in teleporting, it should generally be reserved as a last ditch maneuver, when there are no longer any safe moves available.

2.3.2 Teleport Safely

The teleporter has a secondary operating mode that will transport the hero to a randomly selected, but safe cell. Safe teleporting requires [energy](#) though, and energy is generally in short supply. Obviously, the ability to teleport safely at key moments can greatly extend the hero's lifespan.

Not all game types allow safe teleports and the amount of energy required to teleport safely varies between game types.

2.3.3 Use the Vaporizer

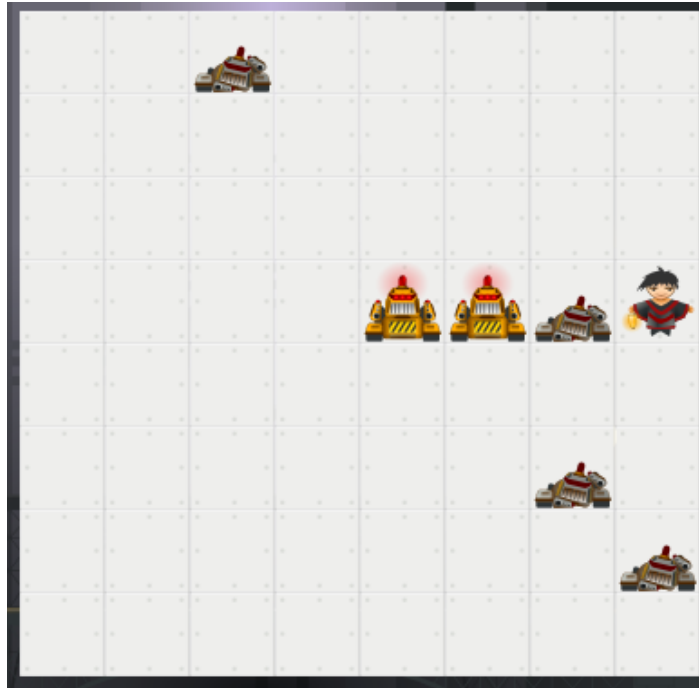
The third use for the teleporter device doesn't actually teleport the hero at all. This somewhat uncommon operating mode teleports all robots in cells adjacent to the hero. No one knows exactly where the robots are teleported to, but it seems to be sufficiently far away that they bother the hero again. Because the robots disappear into thin air without leaving a junkheap, this mode is nicknamed the 'vaporizer' and enemies dispatched through vaporisation are considered as killed for scoring purposes.

Like teleporting safely, using the vaporizer requires [energy](#). Not all game types include the vaporizer and the amount of energy required to use it varies between game types.

2.3.4 Wait Out Round

There may come a time when the grid is laid out such that all the robots can be destroyed without the hero having to move at all. In such a situation, you have the option to wait out the round.

Example 2.1 An Opportunity to Wait Out the Round



Waiting out the round causes the hero to freeze in place while the enemies continue their regular movement. This continues until one of two things happen: the enemies destroy themselves and the round finishes, or an enemy manages to reach the hero and the game ends. Once a wait out round command has been issued, it cannot be canceled, so one must be certain that the hero is, in fact, safe.

So why would one take an action that eliminates the opportunity to change one's mind? Well, it adds a bit of flourish and drama to the end of a round, but more importantly, there is a bonus awarded for each enemy destroyed while waiting out the round. The size and type (points or energy) of the bonus is determined by the game type.

Chapter 3

Interface Overview

3.1 Movement Controls

The player has the option of using either the keyboard, the mouse or both to control the hero.

3.1.1 Keyboard Movement Controls

There are nine different movement actions: left, right, up, down, the four diagonals, and stand still. Each of these actions can be assigned to two different keyboard keys. By default these nine actions are assigned to the nine number keys of the numeric keypad and to a 3 by 3 block of letters at the left end of a QWERTY keyboard. Of course, users using a different keyboard layout or one lacking a numeric keypad can modify these key assignments using the shortcuts configuration dialog.

Each of the special movement commands (such as teleport) can also be issued from the keyboard. See [Move](#).

3.1.2 Mouse Movement Controls

The mouse (or any other pointing device) can be used to navigate in a more natural, intuitive way. As the pointer moves over the main game area, the cursor will change to indicate the direction of the pointer relative to the hero. For example, moving the pointer to the space above the hero will change the cursor to an up arrow. If the pointer is placed immediately over the hero, the cursor changes to indicate no movement.

A single left-click will cause the hero to step in the indicated direction (provided such a step is permitted). For convenience, one can assign additional actions to the right and middle mouse buttons in the [Configure Killbots dialog](#). These actions include those found in the [Move](#) menu, as well as additional option: Repeated Step. Performing a repeated step causes the hero to move in the indicated direction for several consecutive turns until an obstruction is reached or continued movement would be unsafe. One can also halt a repeated step by clicking again while the move is in progress.

3.2 Menu Items

Game → New (Ctrl+N)

Starts a new game, discarding any game that may currently be in progress.

Game → Show High Scores (Ctrl+H)

Shows the high score dialog, containing past scores.

Game → Quit (Ctrl+Q)

Quits Killbots.

Move → Teleport (R)

Teleports the hero to a random, empty cell anywhere within the playing field. See .

Move → Teleport Safely (T)

Teleports the hero to a random, empty cell that is guaranteed to be safe from attack on that turn. See .

Move → Teleport, Safely If Possible (Space)

A convenience action. Teleports the hero safely if there is enough energy available, otherwise just teleports randomly.

Move → Vaporizer (F)

Destroys all robots in cells adjacent to the hero. See .

Move → Wait Out Round (V)

Causes the hero to freeze in place until the end of the round. See .

Settings → Configure Killbots...

Displays the Killbots configuration dialog, where one can adjust gameplay parameters, choose a different set of game rules, or change the game theme.

Additionally Killbots has the common KDE **Settings** and **Help** menu items, for more information read the sections about the [Settings Menu](#) and [Help Menu](#) of the KDE Fundamentals.

Chapter 4

Developer's Guide to Killbots

Interested individuals can customize Killbots in two major ways: themes and rulesets.

4.1 Creating Custom Killbots Themes

Like most other KDE games, a Killbots theme consists of three files.

4.1.1 The Desktop File

The desktop file follows standard INI syntax. The key-value pairs store information about the theme and point to the other files included in the theme. The following example is taken from the default theme.

Example 4.1 An Example Theme Desktop File

```
[KGameTheme]
Name=Example Theme
Description=An example theme used to demonstrate theme creation. This one ←
    features pirates.
Author=Mr. Artist
AuthorEmail=mrartist@example.com
Type=SVG
FileName=example.svg
Preview=example.png
```

The first four parameters contain metadata about the theme and are straight forward enough. For Killbots themes, the `Type` parameter should always be set to `SVG`. The final two parameters contain the file names of the two other theme files. While it's not strictly necessary, it is highly recommended that, for consistency's sake, all three files share the same base file name.

4.1.2 The Preview File

The preview file is simply a PNG file containing a preview of the theme. The preview image should be 240 by 140 pixels and match layout of the default theme's preview as closely as possible, as this makes it easier to compare themes in the theme selector.

4.1.3 The SVG File

The SVG file is a scalable vector graphics image that contains the actual graphical elements used to make up the theme. The file may be either in uncompressed `.svg` or compressed `.svgz` format. Each game element is identified by a unique id in the SVG file. When creating a new theme, it is generally easiest to start with an existing theme and replace the elements one by one until the theme is complete. The SVG file *must* contain the following elements.

background

The image displayed behind the grid in the main game area.

status

The box used to display the current round, score, enemy count and energy.

textcolor

This element should be a rectangle of a single color. Its color is used to for the text in the status boxes.

cell

An individual grid cell.

hero

The character controlled by the player.

enemy

The basic enemy.

fastenemy

The faster enemy.

junkheap

The remains of a collision between enemies.

cursor0

A cursor indicating movement to the right.

cursor1

A cursor indicating diagonal movement up and to the right.

cursor2

A cursor indicating movement up.

cursor3

A cursor indicating diagonal movement up and to the left.

cursor4

A cursor indicating movement to the left.

cursor5

A cursor indicating diagonal movement down and to the left.

cursor6

A cursor indicating movement down.

cursor7

A cursor indicating movement down and to the right.

cursor8

A cursor indicating staying in place.

4.1.4 Installing Custom Themes

In order for Killbots to detect and use a new theme, the files must be moved to an appropriate directory. To install the theme for only the current user, copy the desktop, preview and SVG files to `killbots/themes` in the local folder in `qtpaths --paths GenericDataLocation`. To make the theme available system wide, move the files to `killbots/themes` in the system folder in `qtpaths --paths GenericDataLocation`. Note that this will likely require administrative privileges.

4.2 Creating Custom Game Types

4.2.1 The Ruleset Desktop File

Creating new game types is incredibly simple for Killbots, especially when starting from an existing ruleset file. Rulesets are stored as a single desktop file where the parameters and metadata are stored as simple key-value pairs.

Example 4.2 The Default Killbots Game Type

```
[KillbotsRuleset]
Name=Killbots
Name[gl]=Robots asasinos
Author=Parker Coates
AuthorContact=parker.coates@kdemail.net
Description=<qt><p>The default Killbots game type.</p><p>Includes a medium ←
  size grid, safe teleports, fast enemies and pushable junkheaps.</p></qt>
Description[gl]=<qt><p>O xogo tipo Killbots por omisión.</p><p>Inclúe unha ←
  grella mediana, teleportacións seguras, enemigos rápidos e chatarra que ←
  se pode empurrar.</p></qt>
Rows=16
Columns=16
EnemiesAtGameStart=8.0
EnemiesAddedEachRound=4.0
FastEnemiesAtGameStart=-2.0
FastEnemiesAddedEachRound=2.0
EnergyEnabled=true
EnergyAtGameStart=0
EnergyAddedEachRound=0
MaxEnergyAtGameStart=12.0
MaxEnergyAddedEachRound=0.0
SafeTeleportEnabled=true
CostOfSafeTeleport=1
VaporizerEnabled=false
CostOfVaporizer=0
PushableJunkheaps=Many
SquaskKillsEnabled=true
JunkheapsAtGameStart=0.0
JunkheapsAddedEachRound=0.0
PointsPerEnemyKilled=5
PointsPerFastEnemyKilled=10
WaitKillPointBonus=0
WaitKillEnergyBonus=1
SquashKillPointBonus=0
SquashKillEnergyBonus=1
```

The Killbots Handbook

Note that all text parameters should be entered in American English. As seen in the example above, translations can be added to the desktop file by repeating the key and appending the language code between square brackets. Also note that rich text can be given to the `Description` field by wrapping the value in `<qt>` tags.

A ruleset file should contain all of the following parameters.

Name

the name of the ruleset as it will appear in the ruleset selector and the high score dialog.

Author

The creator of the ruleset

AuthorContact

The contact information of the creator. This is usually an email address.

Description

A description of the ruleset. This should give a decent overview of the game type, but it doesn't need to list the value of each parameter as this information can be seen in the Details dialog.

Rows

The number of rows in the game grid.

Columns

The number of columns in the game grid.

EnemiesAtGameStart

The number of regular enemies added to the grid at the start of the first round.

EnemiesAddedEachRound

The number of additional regular enemies added to the grid at the start of each round.

FastEnemiesAtGameStart

The number of fast enemies added to the grid at the start of the first round.

FastEnemiesAddedEachRound

The number of additional fast enemies added to the grid at the start of each round.

EnergyEnabled

If set to `'false'` the hero will be unable to earn or spend energy.

EnergyAtGameStart

The amount of energy awarded to the hero at the start of the first round.

EnergyAddedEachRound

The amount of energy awarded to the hero at the start of each round.

MaxEnergyAtGameStart

The maximum energy capacity of the hero at the start of the first round.

MaxEnergyAddedEachRound

The change in maximum energy capacity at the start of each round.

SafeTeleportEnabled

If set to `'false'` the safe teleport action will be disabled.

CostOfSafeTeleport

The amount of energy consumed by teleporting safely.

The Killbots Handbook

VaporizerEnabled

If set to 'false' the vaporizer action will be disabled.

CostOfVaporizer

The amount of energy consumed by using the vaporizer.

PushableJunkheaps

The number of junkheaps that can be moved with a single push. Possible values are 'None', 'One' and 'Many'

SquaskKillsEnabled

If true, the hero can squash enemies by pushing junkheaps onto them.

JunkheapsAtGameStart

The number of junkheaps placed on the grid at the start of the first round.

JunkheapsAddedEachRound

The number of additional junkheaps placed on the grid at the start of each round.

PointsPerEnemyKilled

The number of points awarded for destroying a regular enemy.

PointsPerFastEnemyKilled

The number of points awarded for destroying a fast enemy.

WaitKillPointBonus

The number of bonus points awarded for destroying an enemy while waiting out the round.

WaitKillEnergyBonus

The number of bonus energy units awarded for destroying an enemy while waiting out the round.

SquashKillPointBonus

The number of bonus points awarded for squashing an enemy with a junkheap.

SquashKillEnergyBonus

The number of bonus energy units awarded for squashing an enemy with a junkheap.

4.2.2 Installing Custom Game Types

In order for Killbots to detect and use a new game type, the ruleset file must be moved to an appropriate directory. To install the game type for only the current user, copy the file to `killbots/rulesets` in the local folder in `qtpaths --paths GenericDataLocation` . To make the game type available system wide, move the file to `killbots/rulesets` in the system folder in `qtpaths --paths GenericDataLocation` . Note that this will likely require administrative privileges.

Chapter 5

Credits and License

Killbots

Program copyright 2007–2009 Parker Coates parker.coates@kdemail.net

Documentation Copyright (c) 2008–2009 Parker Coates parker.coates@kdemail.net

This documentation is licensed under the terms of the [GNU Free Documentation License](#).

This program is licensed under the terms of the [GNU General Public License](#).