

# MindRover: The Europa Project

**Linux** version by Linux Game Publishing LTD

Linux Game Publishing LTD  
17 Rossington Road  
Nottingham NG2 4HX  
England  
[www.linuxgamepublishing.com](http://www.linuxgamepublishing.com)

**Windows** version by CogniToy, LLC

CogniToy, LLC  
236 Central Street  
Acton, MA 01720  
USA  
[www.cognitoy.com](http://www.cognitoy.com)

©1999-2002 CogniToy, LLC. Linux version ©2000-2001 Loki Software, Inc., ©2002 Linux Game Publishing LTD. All rights reserved. CogniToy and MindRover are trademarks of CogniToy, LLC. Linux Game Publishing is a trademark of Linux Game Publishing LTD. Linux is a registered trademark of Linus Torvalds. All other trademarks and trade names are properties of their respective owners. Printed in the UK.

# Table of Contents

## Table of Contents

### Chapter 1

|                          |   |
|--------------------------|---|
| Introduction .....       | 4 |
| Quick Start.....         | 5 |
| Using This Manual .....  | 5 |
| System Requirements..... | 6 |
| Installation.....        | 6 |

### Chapter 2

|  |    |
|--|----|
| Concepts.....                            | 7  |
| Vehicle Types .....                      | 10 |
| Illegal Chassis and Components .....     | 11 |
| Saving, Deleting, Copying Vehicles ..... | 11 |
| Competing with your Friends .....        | 12 |
| Components .....                         | 14 |
| Points and Weights.....                  | 16 |
| Properties .....                         | 17 |
| Setting Initial Values .....             | 17 |
| Events .....                             | 18 |
| Wiring .....                             | 18 |
| Wire Properties .....                    | 19 |
| Multiple Wires.....                      | 21 |
| ICE Code .....                           | 21 |
| Running the Competition.....             | 22 |
| The GO Button.....                       | 23 |
| Camera Controls .....                    | 24 |
| Finding Bugs .....                       | 24 |

# Table of Contents

## Chapter 3

|                                  |    |
|----------------------------------|----|
| Your User ID .....               | 25 |
| Scenario Selection Screen .....  | 25 |
| Vehicle Selection Screen .....   | 27 |
| Component Selection Screen ..... | 28 |
| Wiring Screen .....              | 30 |
| The Competition .....            | 33 |

## Chapter 4

|                          |    |
|--------------------------|----|
| Component Reference..... | 35 |
| Weapon Statistics.....   | 49 |

## Chapter 5

|                                 |    |
|---------------------------------|----|
| Technical Support .....         | 50 |
| Credits .....                   | 51 |
| End User License Agreement..... | 54 |
| Notes .....                     | 59 |

# Chapter1

## Chapter1

### *Introduction*

Welcome to Europa, land of ice and more ice. With Jupiter constantly hovering on the horizon, we've found that homesickness among new arrivals is common, so let's just get started.

Your time here will present you with a new type of challenge - one that matches the excitement of an action game, the planning of a strategy game and the intense thinking required in a puzzle game.

Your goal is to create robotic vehicles using a wide array of different components, program their behavior, then set them free to compete with each other. Your progress through the levels will depend on cleverness, innovation, and even deception as you tackle some of the more challenging scenarios. We invite you to share your successes, get advice, download new challenges and compete with others by visiting the MindRover home office at <http://www.mindrover.com>, or for this Linux version at <http://www.linuxgamepublishing.com>.

MindRover probably isn't quite like anything you've seen before, so please give yourself a chance to learn it. The best way is to walk through at least the first tutorial. Alternatively, login and hit **F2** for tutorial prompts in the game.

Ready? Free your mind, grab your mouse, and enter into the world of MindRover!

# Chapter1

## ***Quick Start***

For the fastest introduction to MindRover, follow these steps:

Launch MindRover and press the **Start** button. Your vehicles will be stored in your home directory.

Go through the first 2 or 3 tutorials in the game. Follow the tutorial prompts.

Click on the Sports category and try Sumo Hover. This is a partially completed tutorial vehicle to help you get started. Of course, you can build a new vehicle if you prefer.

After that you should have a pretty good idea of how to go off and build your own rovers.

## ***Using this Manual***

### **Chapter 2**

The concepts section describes essential MindRover concepts in some detail. You will learn about scenarios, vehicles, components, wiring, and competitions.

You can read this chapter before you play to get a good feel for all aspects of the game. But if you like to jump right in and get started, just go to the first tutorial and come back to this chapter later.

### **Chapter 3**

The Console section goes into detail on each of the user interface screens. You can read through this chapter before you start, or just use it as a reference after you have started playing the game.

# Chapter 1

## Chapter 4

The components section contains specific information on each component in the game, listed alphabetically. Within the game, click on a component and press **F1** to get more details and examples.

### ***System Requirements***

The minimum system requirements for running Mindrover are:

- 200 MHz PC running Linux (300 MHz recommended)
- 32 MB RAM
- 40 MB Disc Space
- XFree86 3.3.x or 4.x with MESA/OpenGL drivers
- Supported 3D graphics acceleration card
- 4x CDRom

You will need to install and configure 3D hardware support for your computer. For further information on this, please see [www.linuxgamepublishing.com](http://www.linuxgamepublishing.com)

### ***Installation***

Insert the CD, mount it, and run the setup.sh script on the mounted CD. This is accomplished with the following commands.

```
mount /mnt/cdrom
sh /mnt/cdrom/setup.sh
```

You will then be prompted for install options.

If you have not yet installed the LGP Update tool, the installer for this will run before the initial game installation. This will automate the process of finding and applying patches. You may wish to use it periodically to check for patches and add-ons for MindRover.

## Chapter 2

### **Concepts**

When you play MindRover, you are given a series of scenarios or challenges. Your job is to program a robotic vehicle to solve them. There is no single solution to each problem. MindRover lets you devise your own personal way of getting through a level. This chapter will provide the basic concepts for playing MindRover. The next chapter goes into detail on how to use the interface.

Some scenarios may require you to build a vehicle to complete a series of simple tasks. Others will require you to program a set of vehicles that work together to defeat another team.

You can equip your vehicles with everything from rocket launchers to radars and speakers. You can program them to do anything from following a track to finding a path through a maze to seeking and destroying other vehicles. The behaviors you can create are limitless - and the game will grow with your abilities.

There are five basic steps in playing mindrover:

1. Choose a scenario
2. Choose a vehicle
3. Add components
4. Wire it all together
5. Go!

### **Scenarios**

A scenario is a challenge or competition with a goal, such as "push your opponent off the wrestling mat". Each scenario has particular rules that you must follow in order to win.

If this is your first time using MindRover, you should go to the Tutorial category first and choose Tutorial:

## Chapter 2

### **Mindrover Basics**

After the first few tutorials you should try a level one scenario in another category.

After selecting a new category and scenario, read its description in the property box. The **F1** key brings up in-game help. If you have just selected a scenario, it will give you detailed information about the challenge you've selected.

### **Choosing a Scenario**

Scenarios are organized into five categories: Battles, Sports, Races, Tutorials, and Miscellaneous. Within each category there are several levels of difficulty, and each level contains one or more scenarios.

To see a list of scenarios in a category, click on the category name on the top line of the screen. Select a scenario within a category by clicking on its name. The preview screen in the lower right corner will show a flyby of the scenario. The property box on the right gives a description.

### **Completed Scenarios**

After you complete a scenario it is "checked off" on the screen. This is a visual indicator of the scenarios you have completed so far.

You can play the scenarios in any order, but you'll probably do best if you complete the scenarios within a category in the order they're presented.

### **Vehicles**

All vehicles in MindRover are based on one of three types: Hovercraft, Wheeled, or Treaded. There are three sizes of each. The smaller sizes are lighter and take less power to accelerate but are more easily pushed around. Larger chassis, on the other hand, are heavier and contain many more attach points for mounting



## Chapter 2

components. However they also take longer to accelerate and require a higher engine throttle setting to move them.

Once you have chosen a scenario and entered the vehicle selection screen, both empty chassis and pre-made vehicles will generally be available for the competition. The scenario may restrict the use of certain chassis types or components. In this case any vehicles using forbidden components will not be available for use in the scenario.

For many scenarios there may be 2 or 3 opponent vehicles to choose from. Some of the lower-level scenarios have tutorial vehicles to help you get started. Chapter 3 has more details.

Your vehicle is automatically saved whenever you run a competition or log out of the game. It is saved under the row label of the scenario type in which you created it or most recently modified it.

For example, if you created a vehicle while in a Battle scenario, then it will be listed in the row labeled "Battle". You can still use it in a Race or Sports scenario as long as it's chassis and components are legal.

Your vehicles will always be team 1 (blue), and your opponents will be team 2 (red). Some scenarios compete one vehicle against one opponent. Others compete two on two or one against the clock.

To get credit for beating a scenario you must beat one of the opponents supplied by MindRover. You can, however, choose to compete against a different opponent. Please see the section below on how to compete with other players.

### ***Vehicle Types***

#### **Hovercraft**

A Hovercraft is a light vehicle that floats on a cushion of air. It has very little friction and can slide freely in any direction. Its friction with the ground depends on the weight it carries.

## Chapter 2

A Hovercraft has no built-in thrust system - you will usually want to use at least one Thruster to move it. For better direction control, use multiple Thrusters.

### **Wheeled**

A Wheeled vehicle is like a car with no engine. It has four wheels. It rolls freely fore-and-aft when unpowered, but strongly resists being pushed from side-to-side.

It is possible to power wheeled vehicles with Thrusters, but most of the time you'll want to use one of the Engine components. An engine applies its power directly to the wheels of the vehicle; the throttle setting on the engine will control the speed. The throttle can be set from -100% to 100% of full power. You will need a large amount of throttle if you put a small engine on a medium or large size vehicle.

You also need a Steering component to turn the wheels of the vehicle. The Steering component only allows 30 degrees of turning to the left or right. To reverse direction, you should use a negative engine throttle.

### **Treaded**

A Tank has two independent treads, and never rolls freely. It strongly resists being shoved in any direction. In order to move a tank, you need an Engine to provide the power, and a TreadControl to distribute the power to the treads.

Both the Engine throttle and the TreadControl must be set for the tank to move. It is easiest to set the engine throttle on full (100%) and then vary the distribution of that power to each tread using a TreadControl component. If the left and right treads are both set to 50%, the tank will move forward using half of the engine's throttle.

## Chapter 2

### ***Illegal Chassis and Components***

In order to make each scenario interesting and challenging, you will often find that certain chassis types and components have been disallowed. For instance, in the Drag Race you can only choose from hovercraft chassis to build your vehicle.

Once you have chosen a scenario, any vehicle containing forbidden components will be filtered out of the available choices. So if you can't find a vehicle you made, perhaps it is not legal for the scenario you have chosen.

All user-created vehicles will show up in the Tutorial scenario called Testing Ground, which has no illegal components and allows all chassis types. If you can't find a vehicle that you created in the past, choose the Testing Ground scenario, then go to the Vehicle selection screen.

### ***Saving, Deleting, Copying Vehicles***

Vehicles are automatically saved in MindRover when you leave the Wiring or Component screen. If you drag a pre-made vehicle into a vehicle slot and give it a new name, it will be saved with that new name. The old vehicle is no longer available.

Once you have been playing MindRover for a while you will have many vehicles in your vehicles folder. To clean up your folder and get rid of vehicles you don't want, highlight a vehicle in the holobox and click on **Delete** in the upper right of the screen.

Deleting vehicles from the holobox removes them from your directory and you cannot get them back. MindRover will not allow you to delete the chassis from the "chassis" row.

If you like a particular vehicle and wish to use it as the basis for a new vehicle, you can clear out the opponent slot, then drag the desired vehicle into slot 1. Drag the same vehicle from the holobox into slot 2. When you complete the second drag, this vehicle automatically gets a new name which is the old name plus

## Chapter 2

a number. You cannot compete two vehicles with the same name so MindRover automatically renames it.

You now have a copy of the original vehicle as well as the original. Both will be saved.

### ***Competing with Your Friends***

All scenarios will allow you to remove the MindRover opponent and replace it with any vehicle you like. If you want to beat all the levels of the game, you have to beat the opponents we supply. However, at any time you can take on some real competition from other players. Come to **[www.mindrover.com](http://www.mindrover.com)** and find a worthy opponent!

Click on the MindRover opponent and click on Clear in the upper right of the screen to remove the vehicle. Then drag any vehicle into its place as Team 2, and drag your vehicle into Team 1. Then see how you do.

If you would like to play against the MindRover opponent again, you must choose a new scenario, then choose the scenario you were playing. The default MindRover opponent will fill in the Team 2 slot again.

### **Email or Floppy Exchange**

Vehicle files are quite small (usually only 20-30 Kb) and can be copied to a floppy disk, emailed as an attachment, or uploaded to a website where people are collecting them, exchanging them, or competing them against each other.

The Vehicle Transport feature will let you save your vehicles to a floppy disk or to another area on your hard drive. The Vehicle Transport will also let you import other vehicles into your MindRover folder.

You can manipulate your vehicle files outside of MindRover. Your vehicle files are stored in your MindRover configuration directory. The default is:

## Chapter 2

```
~/lgp/mindrover/Vehicles/<userid>
```

There are three files associated with each vehicle you have created. The one with the .vmf extension (for instance: kim\_dragster.vmf) is the one that you need to copy or upload for others to compete against you.

If you give someone all three files (kim\_dragster.ice, kim\_dragster.wst, and kim\_dragster.vmf), then they will get owner information about your vehicle and they will be able to modify it, rename it, and claim it as their own.

If you just give someone the vmf file, then they cannot see into your vehicle's brain. They can compete against your vehicle, but they cannot see your wiring and they cannot make any changes to your vehicle, even its name!

If you rename a .vmf vehicle file outside the game, MindRover will no longer recognize it. Also, please be careful to create unique names for your vehicles. You cannot compete two vehicles with the same name.

MindRover gives you default names that start with your login name. If you keep that part and just change the rest of the name, you are less likely to have problems with duplicate vehicle names.

The owner of the vehicle (the one who has the .wst and .ice files) is the only one that can change the name of the vehicle.

### **Components**

After choosing a scenario and choosing a vehicle chassis, the next step is to add physical components: movement, sensors, weapons, navigation/communication, and extras.

In the Component selection screen you get a top-down view of the chassis with grid markings. Your components must fit in the open grid spaces on the top of the vehicle.

## Chapter 2

Components cannot be placed on top of other components and they cannot span across different grids. The different grids light up as your mouse passes over them.

You can rotate the component by 90 degrees as you are placing it by clicking on the right mouse button. Some components can be rotated during play. If you plan on rotating a component while playing (like a ProximityRadar or SpinThruster), then we recommend that you don't rotate it while placing it. It becomes confusing to try to rotate it in more than one place.

Chapter 4 will give you details on each component. While in the game, you can also click on a component, then press **F1** for detailed help.

### **Movement**

The movement components are used to propel a vehicle or affect its direction.

A Thruster is used primarily for hovercraft movement. The location of a thruster affects how the vehicle will behave. If you put the thruster on the far left of the vehicle facing forward, when it turns on it will tend to turn your hovercraft towards the right. There is also a SpinThruster which can rotate.

The three different Engines (small, medium, and large) are used with the wheeled and treaded vehicles. These components take up 1, 2 or 4 grid spaces. The placement of the engine does not affect the vehicle's performance.

A Steering component is used only with a wheeled vehicle to control its front wheels. A TreadControl is used only with a treaded vehicle to control each of its treads. It doesn't matter where on the vehicle you place a TreadControl or Steering component.

## Chapter 2

### Sensors

Sensors are the 'eyes' and 'ears' of the vehicles giving you information on what is in the arena. You use this information to decide how to move or steer your vehicle or when to fire your weapon.

Sensor types include the TrackSensor, BumpSensor, Sonar, Filter, Speedometer, SpinOmeter, LootSensor and three types of Radar. The Radar, TrackSensor, and Sonar are location-dependent, while the others are not. For instance, the BumpSensor detects any collision of your vehicle whether it was in the front, back or on either side. You can place the BumpSensor in any open grid space on your vehicle.

### Weapons

Weapons are used to damage, slow down or move another vehicle. They do varying amounts of damage. In some scenarios, after a certain amount of damage the vehicle will be destroyed. When this happens, in some cases the competition will end; in other cases, the vehicle will "respawn" and can continue to play.

Most weapons are fixed in place, which means that you must rotate the entire vehicle to aim them. You can set their initial orientation by using the right mouse button to set the mounting position. However, the Machine Gun does have a limited ability to adjust its aim.

### Nav/Comm

Navigation and communication components help you to find things and communicate with your team. BearingSensor, WaypointSensor, XYSensor and XYFinder are used to find specific points in the arena; RadioTransmitter and Receiver allow you to send and receive information to a teammate.

Please see Chapter 4 for more details on using these components.

## Chapter 2

### **Extras**

The Extra components are mostly just for fun but can be used to help debug your vehicle if something isn't working correctly.

For instance, the Speaker can be set up to say the numbers from 1-5. You can have the speaker sound the number 1 when your vehicle has reached the first waypoint.

The RunningLight was designed to be connected to the ModeSwitch to tell you what mode color the vehicle is in. See ModeSwitch for more information on this component.

The LootCarrier is used to pick up objects in the scenario such as the enemy flag. Anything which might be carried by a vehicle will require a LootCarrier.

### ***Points and Weights***

Each component has a point cost and weight associated with it. The scenario you choose will determine the total amount of weight and total points you have to configure on your vehicle.

For scenarios with more than one vehicle/team, the total points and weight is the sum from each vehicle. So if you build your first vehicle with many heavy components, you may not have enough weight budget left to add basic components on your second vehicle.

If your team has exceeded the points or weight limit, these numbers will turn red and the competition will not run until you remove some components.

In general the engines and large weapons weigh the most. Components which are technically complex require the most amount of points. The purely logical components that aren't physical (found in the Wiring screen) have no weight and little or no point cost.



## Chapter 2

### ***Properties***

Each component has a set of properties. A property defines some characteristic of the component, such as the range of the radar, or the distance to a waypoint.

Some properties are input properties -- they control some value on the vehicle and affect the way it responds. For example, you use the Steering Angle property to control the steering.

Some properties are output properties -- they're reported by the component as information, such as the CurrentSpeed property of the Speedometer.

Finally, some properties are activate properties -- they cause the component to take some action, such as the Fire property on weapons.

### ***Setting Initial Values***

There are a number of ways you can find out about the properties and events of a component.

The initial value for input properties can be set on the wiring screen. When you select a component (by clicking on it) on the wiring screen, the property area on the right side of the screen shows you a list of the component's input properties. You can adjust the properties here; they are initialized to these values when the scenario starts.

If you hold your mouse still over a component in the Wiring screen, a help box will appear which shows you these properties and their current settings.

To get more detailed help on a component and read about its events as well as properties, select a component (by clicking on its icon) and press **F1**.

## Chapter 2

### **Events**

Events are triggers that originate from a component and can be used to cause the vehicle's behavior to change.

For example, the BumpSensor has an event called Bump. If your vehicle has a BumpSensor on it and it bumps into the wall, it will trigger a Bump event. You can use a wire to cause this Bump event to set the Angle property of a Steering component.

### **Wiring**

To give your vehicles intelligence, you wire components together such that an event from one component (the source component) sets the property of another component (the destination component).

To create a wire, drag the source component with the left mouse button. You'll see a rectangular outline of the component icon following the mouse. When you pass over a valid destination component, the rectangle will vanish and be replaced by a wire connecting the two components. Drop the component when you see the wire and a connection will be made.

If you do not see a wire then you may have the source and destination components backwards. The source component must be able to trigger events and the destination component must have settable input properties.

For example, you can drag a MediumRadar component over a Steering component and you will get a wire. The MediumRadar is the source and can trigger many different events, such as TurnOn, TurnOff and Change. The Steering wheel has one settable property, which is the steering Angle.

If you try to drag the Steering wheel over the MediumRadar component, you will not get a wire because the Steering

## Chapter 2

component triggers no events. It cannot be the source of a wiring connection.

Once you have created a wire between components you will see an arrow from the source to the destination, indicating the direction of information flow.

### ***Wire Properties***

Like components, wires also have properties. These properties describe how the wire behaves. Selecting a wire (by clicking on its direction arrow) will show you the properties for the wire in the property area along the right side of the screen.

#### **Source Event**

The source and its event are the first items in the wire property box. You will see the name of the source component, followed by a pull-down box containing all of its events. Most components have from one to three events that they can trigger.

#### **Destination Property**

The next item in the wire property box is the destination component and its property. You will see the name of the destination component and a pull-down box containing the list of all its settable properties. Components may have as many as five different settable properties.

#### **Value/Source Property**

In the third section of the wiring property box you decide what you want to set the destination property to. There are two choices.

You can set the destination to a specific value, such as a number or mode color. Or you can set the destination property to an output property of the source component. For instance, you can set the Steering Angle (destination) to the Output property of the Randomizer (source).

## Chapter 2

### Activate Property

There are a few components whose settable property is an 'activate' command such as Fire or Set. You don't give this destination property a value, you simply activate it. For instance the RocketLauncher's only property is Fire.

### An Example

As an example of different wiring properties, let's say you want the Speaker to play a random sound every 2 seconds from its selection of 5 different sound choices.

We need a Speaker component, a Randomizer, and a LoopTimer. To set the LoopTimer to trigger a Tick event every 2 seconds, simply click on it and set its TickTime property to 2.

Similarly, set the Minimum and Maximum property values for the Randomizer by clicking on the Randomizer icon. Set the minimum to 1 and the maximum to 5.

When you click on an icon and change its property values you are setting its initial values. When the vehicle starts running in the competition, these values can change depending on the wiring conditions you set.

If you then create a wire from the LoopTimer to the Randomizer, you will see that the LoopTimer is the source component, and its trigger event is called Tick.

The destination component is the Randomizer and its property, Trigger is an activate property. Trigger means "roll the dice". It is an activate property because you don't need to tell it how much or how many. Just "roll the dice" and get a new random number.

Next create a wire from the Randomizer to the Speaker. Here, the Randomizer is the source and its event is called Set. The Speaker is the destination and the property we want to set is PlaySound.

## Chapter 2

Finally, set the PlaySound property to the Output property of the Randomizer. Now, when you "roll the dice", the Speaker plays the sound specified by the number you rolled.

### **Multiple Wires**

Each wire you create can set one property on the destination component. You can add as many wires as you need between components to set multiple properties.

For instance, to set both the left and right TreadControl to 100% when the radar turns on, you need to create two wires from the Radar component to the TreadControl component. One wire will set the left tread to 100%, and the other will set the right.

A component like the TreadControl requires that two wires be used for each direction you want to move. So to move forward, left and right would require 6 wires to the same TreadControl component.

To help organize your wiring, use Broadcast components. With three Broadcast components, one labeled 'TurnLeft', one labeled 'TurnRight', and one labeled 'Forward', you can spread out the wires going to your TreadControl. Click on the Broadcast component and press **F1** to get more details on its use.

### **ICE Code**

Behind the graphical interface which allows you to wire components together is a programming language called ICE. Every time you add a component, set its properties, or add wires between components, new lines of ICE code are generated.

The ICE code for each vehicle is saved with the extension '.ice'. Outside of the game, you can look at this code with a text editor. You will find the '.ice' file for your vehicle in

```
~/.lgp/mindrover/Vehicles/<userid>
```

## Chapter 2

In this 1.07b release of MindRover, it does you no good to modify the ICE code. It is regenerated and recompiled each time you hit **GO**. This is in place for a possible future update.

Your vehicle is automatically saved whenever you move to a new screen or hit the "GO" button. When it is saved, three files are generated. The .ice file is the ICE code. The .wst file contains the wiring state information which will allow MindRover to recreate the component and wiring screen for this vehicle. The last file is the .vmf, virtual machine file. This is the file that tells your vehicle what to do in a machine code that it understands.

Come back to the CogniToy website often to get new components and scenarios as well as information on creating your own objects in ICE.

### ***Running the Competition***

Once you have added components and wired them up, it's time to see what your vehicle can do on its own! Hit the **GO** button and watch what happens.

Each scenario has objectives and when they are met by your team or the opposing team, you will get a win/lose screen. At this point you have several choices:

The Instant Replay will play the exact same competition with the same starting conditions. This is useful when you want to analyze what your vehicle did or what the opponent did.

The Play Again option will allow the scenario to chose different starting positions (if possible) or other random events so that you will not get the exact same result. Some scenarios, such as Drag Race, do not vary their starting conditions, in which case Play Again and Instant Replay will have the same result.

Return to the Console allows you to go back and make changes to your vehicle, or to choose a new scenario or new vehicle.

## Chapter 2

### **The GO Button**

The **GO** button is only enabled when the conditions are right to launch a competition. There are a number of reasons why it might not be enabled:

You don't have all the vehicle slots filled. Go into the Vehicle selection screen and make sure there is a chassis in every empty slot.

Your vehicle or your team has exceeded the Point or Weight limits. If you have gone over the budget in either area, the numbers which display the points and weights will turn red. Remember to check both vehicles on your team if it is a 2-on-2 competition. If you have gone over the limit, you need to remove some components before continuing.

### **Camera controls**

During a competition you may want to change your camera angle to get a better view. The following table shows you which keys to press:

| <b>Key</b> | <b>Point of View</b>              |
|------------|-----------------------------------|
| 1          | look at vehicle 1 (your vehicle)  |
| 2          | look at vehicle 2                 |
| 3          | look at vehicle 3                 |
| Shift+1    | follow camera for vehicle 1       |
| Shift+2    | follow camera for vehicle 2       |
| Shift+3    | follow camera for vehicle 3       |
| 9          | Scoreboard view (if there is one) |
| 0          | Autocam or "best" view            |
| Shift+0    | drive the camera with your mouse  |

### **Finding Bugs**

You can (and should) go back and forth between testing your vehicle in the competition and fine tuning it in the console. If it

## Chapter 3

isn't going well in the competition you don't have to wait for one team to win; just hit the ESCAPE key, go back to the console and make changes to your vehicle.

If you test your vehicle after each new addition or sets of wires, you will be able to find and fix problems much more quickly. You can also use the 'Extra' components such as the Fireworks, Speaker and RunningLight to help you diagnose what your vehicle is thinking.

For instance, you can wire the Fireworks launcher up to the TrackSensor and have it fire the Fireworks whenever the TrackSensor turns on, or wire the RunningLight to the ModeSwitch and it will change color depending on the vehicle mode.

## Chapter 3

### ***Your User ID***

MindRover uses your Linux username as your User ID in the game. If you have not run MindRover before, a new directory called `~/ .1gp/mindrover` will be created in your home directory.

If you would like to play MindRover with a different User ID, you will need to create a new account on your Linux system and run MindRover logged in to that account.

The **Quit** button exits MindRover. To access this screen at any time, click the **Logout** button on any of the other console screens.

If you would like to exit from the game immediately without logging out first, you can hit **F12** at any time.

### ***Scenario Selection Screen***

After logging into MindRover you will be brought to the Scenario Selection Screen. In the upper left portion of the console are the

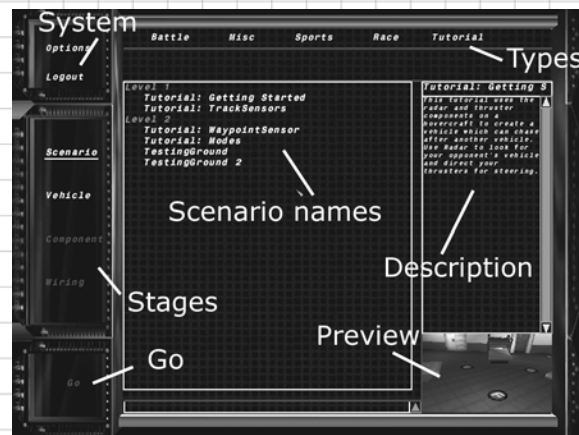


## Chapter 3

system buttons. **Logout** will bring you back to the main screen and allow you to quit.

The **Options** button will bring you to a console where you can change the music and sound effects volume. Hit **OK** to get back to the Scenario Selection Screen.

Along the left column are four buttons that will bring you to the four stages of MindRover play: Scenario, Vehicle, Component, and Wiring. Each of these is described in detail below. The lower left **GO** button will launch the competition with your vehicles. The other areas of the console vary depending on the stage you have chosen.



The first step to playing MindRover is choosing which scenario you would like to play. Click the button on the left side of the screen labeled Scenario to get to the Scenario Selection Screen. If you have just logged in, you are automatically placed in the Scenario Selection Screen. Along the top of the screen you will see the various categories of scenarios: Tutorial, Battles,

## Chapter 3

Races, Sports, and Misc. Clicking on one of these will bring up a list of scenarios in that category.

On the right side of the console is the Property Box. When you click on a scenario a description of it will appear in the Property Box. Read these carefully. In many cases there is specific information that you will need in order to successfully play a scenario.

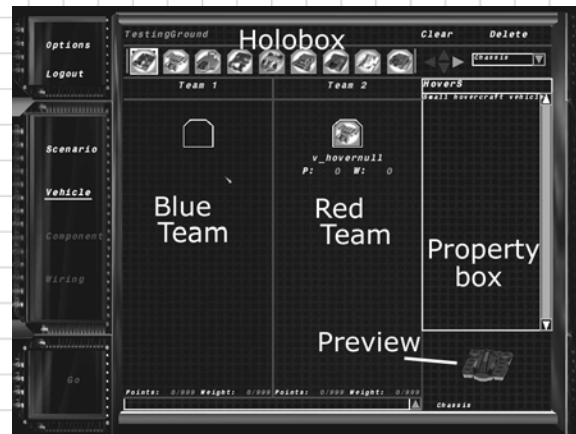
Also, you can hit the **F1** key to get information on the scenario you have chosen, including a map of waypoints. Once the help screen has popped up, you can scroll through other help topics as well.

Below the Property Box on the right is the Preview screen. In the preview you will see a flyby that shows the area in which the scenario will take place.

### ***Vehicle Selection Screen***

Once you have chosen a scenario you may begin building a vehicle for that scenario. Clicking the **Vehicle** button on the left side will bring you to the Vehicle Selection Screen.

## Chapter 3



Along the top of this screen you will see the holobox. The holobox contains a set of icons representing the various chassis for you to choose from as the base for your vehicle.

There are several ways to navigate in the holobox: First, you may click the four arrow buttons on the right side of it. You can change rows with the pulldown to the right. For faster navigation you may also 'right-drag' the contents of the box around. Do this by clicking and holding the right mouse button in the holobox and dragging it.

The top row of vehicles in the holobox are the empty chassis used when you want to build a vehicle from scratch. Other rows will contain your previously created vehicles, sorted by the scenario type in which you created them. You can scroll through these rows with the arrows or the pull-down box to the right of the holobox. If you have just logged in there will be no other rows.

Not all chassis will be displayed in the holobox, only those allowed by the currently selected scenario.

## Chapter 3

Once you select a chassis you can read its name and description in the Property box on the right.

In the center of the screen you will see Vehicle Slots arranged by teams. When playing against the computer you are always the blue team (Team 1), on the left, and the computer represents the red team (team 2), on the right.

To select a vehicle for play, drag the vehicle icon into one of the open slots on your team. Once it has been dropped into the slot you can see what it looks like in the lower right Preview Screen, and you can edit its name and description in the Property Box.

### ***Component Selection Screen***

Clicking on the ***Component*** button on the left side of the screen will bring you to the *Component Selection Screen*. This is where you add physical components to your vehicle.

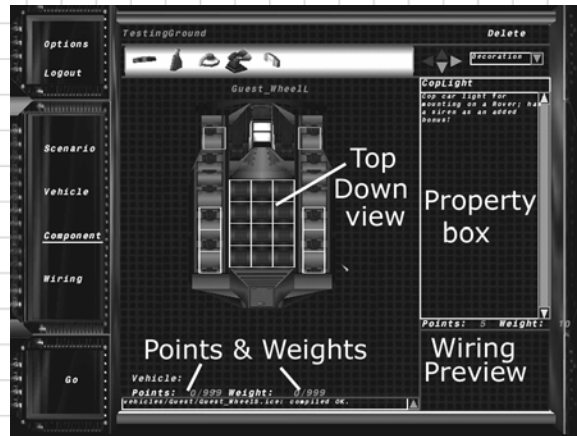
The components available for this scenario are displayed in the holobox at the top of the screen. The center of the screen displays a top-down view of your vehicle showing its grid points.

Each component requires a certain number of grid squares depending on its size. To see how much space a component will take up, drag it from the holobox onto the vehicle without letting go of the mouse. A set of colored squares outlines the spaces the component will take up on your vehicle.

You will also see the grid squares light up as a group when you fly your mouse over them. Highlighted grid squares are all part of

## Chapter 3

one group and are at the same level on the vehicle. You cannot drop a component across different levels of grid.



To place a component, drag it from the holobox and release the mouse button over the grid spaces on which you wish to place it. To change the orientation of the component, you can click the right mouse button while you are dragging the component. To remove the component, pick it up again with the left mouse button and drag it off the edge of the vehicle.

You can view your vehicle in 3D with the components attached to it by clicking and dragging the right mouse button over the vehicle. This will cause the vehicle to rotate. Letting go of the mouse button will return to the top-down view.

The Preview Box (lower right corner) will show you a preview of the Wiring Screen.

Each component has a specific point cost and weight associated with it in addition to taking up grid space on your vehicle. You can see these values in the Property Box when you select a component.

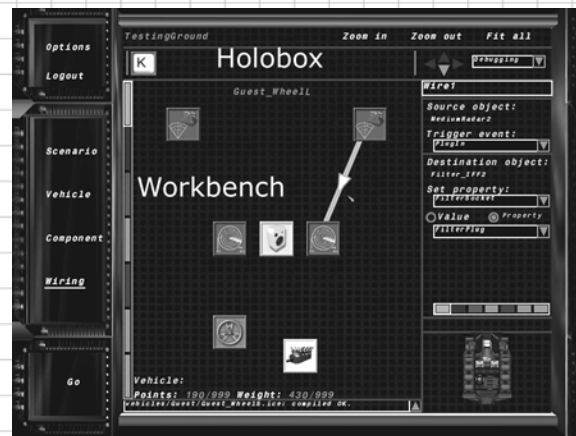
## Chapter 3

The total points and weight available to you are for all vehicles on your team. If you only have one vehicle, then all the points and weight can be used by that one vehicle. You can see the total and the maximum allowable points and weight below the vehicle you are working on.

If you have exceeded the maximum allowed in either points or weight for your team, these values will turn red. Remove some components or exchange them for lighter or cheaper ones until the points and weight turn green again. The **GO** button will be disabled and you will not be able to run the competition if your team vehicles are over either limit.

### Wiring Screen

Clicking the Wiring button or the wiring preview in the component selection screen will bring you to the Wiring Screen. This is where you teach your vehicle how to use all the components you have equipped it with in order to beat the scenario.



## Chapter 3

### **Navigating the Wiring Screen**

The center of the wiring screen is your workbench. This is the space in which you actually 'program' your vehicle.

To move around your workbench right-drag on the workbench, similar to how you move around in the holobox. Clicking on Fit All in the upper right or hitting the Tab key on your keyboard will zoom in or out to fit all of the components in your view. You can use the **Zoom in** and **Zoom out** buttons in the upper right to move around on your workbench. Alternatively, holding down Z and left-clicking will zoom in; holding down Z and right-clicking will zoom out.

You may select the various components by clicking on them with the left mouse button. You may also select a group of components by dragging a box around multiple icons. All selected icons have a white outline around them. You may freely move these components around the workbench by dragging this group with the left mouse button.

If you select an icon that represents a physical component (see next section for differences between physical and logical) the viewport in the lower right will zoom in on the component you have selected. This will remind you where you put the component on the vehicle. Feel free to move the icon by dragging it anywhere you would like on the workbench.

The holobox along the top of the wiring screen now contains logical components that you can use to help in programming your vehicle (see below). The property box to the right of the workbench displays component and wire properties.

### **Logical Components**

Logical components are components that have no physical representation on your vehicle. They cannot sense the outside world or affect your vehicle directly in any way. Instead, these components are used to help you control the physical components which you added to your vehicle in the Component Selection Screen.

## Chapter 3

To add a logical component, simply drag the icon out of the holobox onto the workbench. Clicking on each of the components in the holobox will give you a short description in the property box. Also, you can hit **F1** to get more information for each component.

To delete a logical component, simply select the component or components you wish to delete and press delete on your keyboard.

To delete a physical component, you must go back to the Component Selection Screen (click on the **Component** button on the left), highlight the component and click on **Delete** in the upper right of the screen. Or you can move the component off the vehicle. You cannot delete a physical component from within the wiring screen.

### Setting Properties

The most basic form of programming in MindRover is setting the properties of the components you have previously placed. Selecting a component will bring up a list of its properties and controls with which to set them. For example, selecting a SpinThruster will bring up an Angle control for setting the angle at which the thruster pushes and a Thrust control for setting how much force should be applied. When the vehicle is started it will take on these specified properties until a wire causes them to change.

### Creating Wires

Wires are messengers of events from one component to another. Most components can cause events. A MediumRadar, for instance, causes an event called TurnOn whenever something passes in front of its line of sight. Wires allow you to harness these events to change the properties of other components. In effect, that is all a wire does: It listens for an event on the source component and changes a property on the destination



## Chapter 3

component. With this seemingly simple model you can teach your vehicle to do almost anything.

To create a wire, click and drag the component you want to be the source of the signal onto the component you want to receive the message. When the mouse cursor moves over another component a line will appear. Releasing the mouse button while this line is present will create a wire.

There is no one way to solve a given problem and there are thousands of ways to wire your vehicle. Explore!

### ***The Competition***

Once you have wired your vehicle, it's time to send it to the competition. Click the **GO** button. After loading the scenario and the vehicles, you will see a count down screen, and then the competition starts.



Hit the **ESCAPE** key to go back to the console and make adjustments.

## Chapter 3

Within the competition you can change your camera view as follows:

| <b>Key</b> | <b>Point of View</b>              |
|------------|-----------------------------------|
| 1          | look at vehicle 1                 |
| 2          | look at vehicle 2                 |
| 3          | look at vehicle 3                 |
| 4          | look at vehicle 4                 |
| Shift+1    | follow camera for vehicle 1       |
| Shift+2    | follow camera for vehicle 2       |
| Shift+3    | follow camera for vehicle 3       |
| Shift+4    | follow camera for vehicle 4       |
| 9          | Scoreboard view (if there is one) |
| 0          | Autocam or "best" view            |
| Shift+0    | drive the camera with your mouse  |

For most scenarios that are 1 on 1, you are vehicle 1 (team 1) and your opponent is vehicle 2. In a 2 on 2 match, you have vehicles 1 and 2 (both team 1) and your opponent has vehicles 3 and 4.

Once you have successfully completed a scenario, it will be checked off in the Scenario Selection Screen.

## Chapter 4

### **Component Reference**

In this reference you will find the component description, category, properties, events, points and weight. You can get more detailed information and usage notes from within MindRover by selecting a component and pressing the **F1** key.

Check the MindRover website at [www.mindrover.com](http://www.mindrover.com) for downloadable add-on components.

#### **Add**

A component that adds its two inputs and generates an output equal to the sum.

Category: Logical  
Properties: Input 1, Input 2, Output  
Weight/Points: 0/0  
Events: Change, Set



#### **BearingSensor**

A sensor that will tell you if you are left of, right of, or directly heading for a reference bearing.

Category: Physical  
Properties: DeltaBearing, FuzzyAngle, RefBearing, TrueBearing  
Weight/Points: 10/30  
Events: Change, LeftOfRef, OnRef, RightOfRef



#### **Broadcast**

A component used to reproduce activate events.

Category: Logical  
Properties: Trigger  
Weight/Points: 0/0  
Events: Set



## Chapter 4

### BumpSensor

A sensor that detects a collision between your vehicle and another object.

Category: Physical  
Properties: FilterPlug  
Weight/Points: 30/10  
Events: PlugIn



### Compare

A component that will compare two numeric values. This component can be set to trigger on GreaterThan, LessThan, or EqualTo.

Category: Logical  
Properties: Input1, Input2  
Weight/Points: 0/0  
Events: EqualTo, GreaterThan, LessThan



### CopLight

A cop car light bar with a siren.

Category: Physical  
Properties: Play  
Weight/Points: 0/0  
Events: None



### Deadweight

A large piece of lead used to add extra weight to your vehicle.

Category: Physical  
Properties: None  
Weight/Points: 100/0  
Events: None



## Chapter 4

### DebugMessage

Allows you to display a message string while the scenario is running.

Category: Logical/Debugging  
Properties: Color, Message  
Weight/Points: 0/0  
Events: None



### Divide

A component that divides Input1 by Input2 and will give you the quotient and remainder.

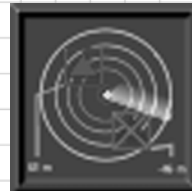
Category: Logical  
Properties: Input1, Input2, Quotient, Remainder  
Weight/Points: 0/0  
Events: Change, Set



### Filter\_IFF

A filter (Identify Friend or Foe) that is used to discriminate between friend, foe, and projectile.

Category: Physical  
Properties: EnemyVehicle, FilterSocket, Other, Projectile, State, Teammate  
Weight/Points: 10/40  
Events: Change, TurnOff, TurnOn



### Fireworks

A component that shoots off a brief firework display when activated.

Category: Physical  
Properties: FireColor  
Weight/Points: 0/0  
Events: None



## Chapter 4

### KeySensor

A component that will allow you to use the Keyboard to trigger events in the world. It is not a legal component in most scenarios, but a good debugging tool.

Category: Logical  
Properties: KeyLast, KeyList  
Weight/Points: 0/0  
Events: Key1, Key2, Key3, Key4, Key5



### LargeEngine

An engine that can be used with a wheeled or treaded chassis.

Category: Engine  
Properties: Throttle  
Weight/Points: 300/30  
Events: None



### Laser

A laser gun with a pretty good range and a moderate repeat rate.

Category: Weapon  
Properties: Fire  
Weight/Points: 75/25  
Events: None



### LogicalAND

A component that will tell you when two logical inputs are both true.

Category: Logical  
Properties: InputA, InputB, State  
Weight/Points: 0/0  
Events: Change, TurnOff, TurnOn

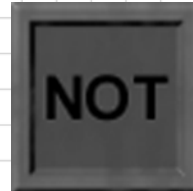


## Chapter 4

### LogicalNOT

A component that will give you the opposite of the input state.

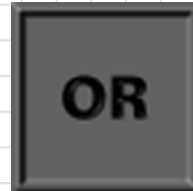
Category: Logical  
Properties: InputA, State  
Weight/Points: 0/0  
Events: Change, TurnOff, TurnOn



### LogicalOR

A component that will tell you when either of its two logical inputs are true.

Category: Logical  
Properties: InputA, InputB, State  
Weight/Points: 0/0  
Events: Change, TurnOff, TurnOn



### LongRangeRadar

A radar that can detect objects as far away as 15 meters with as much as 30 degrees scan width.

Category: Physical  
Properties: Angle, FilterPlug, MaxRange, ScanWidth, State  
Weight/Points: 0/0  
Events: Change, PlugIn, TurnOff, TurnOn



### LoopTimer

A settable timer that automatically restarts after counting down.

Category: Logical  
Properties: TickTime  
Weight/Points: 0/0  
Events: Tick



## Chapter 4

### LootCarrier

A component that allows your vehicle to carry a flag, jewels or other loot.

Category: Physical  
Properties: Drop, LootColor, State  
Weight/Points: 30/10  
Events: Change, TurnOff, TurnOn



### LootSensor

A sensor that will give you the bearing and distance to the loot - jewels, enemy flag, or other good stuff.

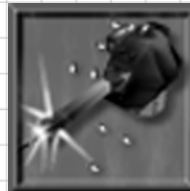
Category: Physical  
Properties: Bearing, Distance, LootColor  
Weight/Points: 30/30  
Events: Change



### MachineGun

A rapid fire gun mounted on a pivoting turret.

Category: Weapon  
Properties: Angle, Fire  
Weight/Points: 100/15  
Events: None



### MediumEngine

An engine that can be used with a wheeled or treaded chassis.

Category: Engine  
Properties: Throttle  
Weight/Points: 200/30  
Events: None





## Chapter 4

### MediumRadar

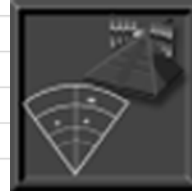
A radar that can detect objects within 5 meters with as much as 90 degrees scan width.

Category: Physical

Properties: Angle, FilterPlug, MaxRange, ScanWidth, State

Weight/Points: 30/30

Events: Change, PlugIn, TurnOff, TurnOn



### MineLayer

A component that drops explosive proximity mines.

Category: Weapon

Properties: Fire

Weight/Points: 75/20

Events: None



### ModeSwitch

A component that will change your vehicle's mode during a competition. You can define up to seven modes, specified by the colored wires in the wiring screen.

Category: Logical

Properties: Mode

Weights/Points: 0/0

Event: Change, Leave, Set



### Multiply

A component that multiplies its two inputs and generates an output equal to the product.

Category: Logical

Properties: Input1, Input2, Output

Weight/Points: 0/0

Events: Change, Set



## Chapter 4

### ProximityRadar

A radar for detecting objects within 3 meters with as much as 360 degrees scan width.

Category: Physical

Properties: Angle, FilterPlug, MaxRange, ScanWidth, State

Weight/Points: 30/30

Events: Change, PlugIn, TurnOff, TurnOn



### RadioReceiver

A component used to receive signals from a RadioTransmitter.

Category: Physical

Properties: IncomingNumber, Station

Weight/Points: 10/30

Events: NumberReceived



### RadioTransmitter

A component used to send signals to a RadioReceiver.

Category: Physical

Properties: SendNumber, Station

Weight/Points: 10/30

Events: None



### Randomizer

A component that generates a random number within a specified range.

Category: Logical

Properties: Maximum, Minimum, Output, Trigger

Weight/Points: 0/0

Events: Set



## Chapter 4

### RangeTest

A component used to detect whether a value is above, below, or within a certain range of values.

Category: Logical

Properties: Input, Maximum, Minimum, Output

Weight/Points: 0/0

Events: AboveRange, BelowRange, Change, InRange, Set



### RocketLauncher

A component that fires a slow but deadly rocket - watch out for the splash damage.

Category: Weapon

Properties: Fire

Weight/Points: 150/20

Events: None



### RunningLight

A component that helps you find your vehicle in a crowd, or helps you debug your vehicles behavior.

Category: Physical

Properties: Mode

Weight/Points: 0/0

Events: None



### SmallEngine

An engine that can be used with a wheeled or treaded chassis.

Category: Engine

Properties: Throttle

Weight/Points: 100/30

Events: None



## Chapter 4

### Sonar

A component that will give you the distance to the nearest object in front of it.

Category: Physical

Properties: Angle, Distance, FilterPlug, Fire, MaxRange

Weight/Points: 30/30

Events: NoPing, Ping, PlugIn



### Speaker

A component that plays a number of different sounds. Use for taunting an opponent or for debugging your vehicle.

Category: Physical

Properties: PlaySound, SoundGroup

Weight/Points: 0/0

Events: None



### Speedometer

A sensor used to give you the speed of your vehicle.

Category: Physical

Properties: CurrentSpeed, Fuzziness

Weight/Points: 30/30

Events: Change



### SpinOMeter

A component that tells you how fast you are spinning.

Category: Physical

Properties: CurrentSpeed, Fuzziness

Weight/Points: 30/30

Events: Change



## Chapter 4

### SpinThruster

A thruster that is rotatable, a bit heavier and more costly than the simple Thruster.

Category: Physical  
Properties: Angle, Thrust  
Weight/Points: 120/30  
Events: None



### Splice

Combines text and/or numbers into a single message.

Category: Logical/Debugging  
Properties: Calc, message, Part1, Part2, Part3, Part4  
Weight/Points: 0/0  
Events: Change, Set



### Startup

A component that gives a single tick at the beginning of the scenario.

Category: Logical  
Properties: None  
Weight/Points: 0/0  
Events: Tick



### Steering

A component that allows you to steer a wheeled vehicle.

Category: Physical  
Properties: Angle  
Weight/Points: 30/10  
Events: None



## Chapter 4

### Subtract

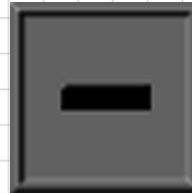
A component that subtracts Input2 from Input1 and generates an output equal to the difference.

Category: Logical

Properties: Input1, Input2, Output

Weight/Points: 0/0

Events: Change/Set



### Switch

A component that reproduces events if it is enabled.

Category: Logical

Properties: Enabled, Toggle, Trigger

Weight/Points: 0/0

Events: Change, Set, TurnOff, TurnOn



### TaxiLight

A Taxicab light bar for mounting on a vehicle. It plays a sound when triggered.

Category: Physical

Properties: Play

Weight/Points: 0/0

Events: None



### Thruster

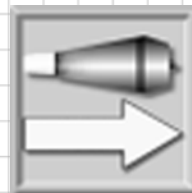
A component that applies a force in the desired direction.

Category: Physical

Properties: Thrust

Weight/Points: 75/10

Events: None



## Chapter 4

### Timer

A one-shot settable timer which can be paused and restarted.

Category: Logical

Properties: PauseTimer, RestartTimer, TickTime

Weight/Points: 0/0

Events: Tick



### TrackSensor

A component that detects whether the sensor sees the track.

Category: Physical

Properties: Angle, Range, State

Weight/Points: 20/30

Events: Change, TurnOff, TurnOn



### TreadControl

A component that controls power to the left and right treads of a treaded vehicle.

Category: Physical

Properties: LeftTread, RightTread

Weight/Points: 50/10

Events: None



### Variable

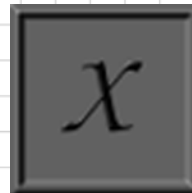
A component used to store a numeric variable. Can also be used as a counter.

Category: Logical

Properties: IncrementBy, Input, Output, Query

Weight/Points: 0/0

Events: Change, Querying, Set



## Chapter 4

### WaypointSensor

A sensor that gives you the distance and bearing to the next waypoint.

Category: Physical

Properties: Bearing, CurrentWaypoint, Distance, WaypointList

Weight/Points: 10/30

Events: Change, HitWaypoint



### WeldingTorch

A weapon used to inflict close-range damage.

Category: Weapon

Properties: Fire

Weight/Points: 100/10

Events: None



### XYFinder

A component that will report the bearing and distance to a specified XY position.

Category: Physical

Properties: Bearing, Distance, XPosition, YPosition

Weight/Points: 10/30

Events: Change



### XYSensor

A sensor used to determine your position in the world.

Category: Physical

Properties: XPosition, YPosition

Weight/Points: 10/30

Events: Change





## Chapter 4

### **Weapon Statistics**

#### **Laser**

- Refire Rate: 2.5 seconds
- Damage: 10 hit points max - varies with distance
- Range: 10 meters
- Speed: Instant

#### **Machine Gun**

- Refire Rate: 1 round/second
- Damage: 3 hit points/bullet, 10 bullets/round
- Range: 8 meters
- Speed: Instant
- Pivots: +45 to -45 degrees at 20 degrees per second

#### **Mine Layer**

- Refire Rate: 0.8 seconds
- Damage: 40 hit points for direct hit
- Range: 0 meters

#### **Rocket Launcher**

- Refire Rate: 3 seconds
- Damage: 15, 25 or 50 hit points for direct hit
- Range: Infinite
- Speed: 1, 2 or 3 meters/second

#### **Welding Torch**

- Refire Rate: 2 seconds
- Damage: 20 hit points/second
- Range: 1 meter
- Speed: Instant

## Chapter 5

### Chapter 5

#### ***Technical Support***

Please be sure to read the README document on the game CD, and review the Frequently Asked Questions on our online support page: **support.linuxgamepublishing.com**. Here you can also check for any updates to ensure you have the latest version of the software.

You may also launch `lgp_update` as the user who installed the game to check for product updates, while connected to the Internet.

If you are still having problems, please contact our technical support team. Full information about how to do this is found on the above website.

By email, please contact **support@linuxgamepublishing.com** and we will do all we can to resolve your MindRover problem. In your email, please include the following information:

- Complete product title and version number (found by typing `mindrover -v`)
- Exact error message (if any)
- Linux distribution
- Linux version (found by typing `uname -a`)
- Computer processor type and speed (e.g. Pentium 3 - 500)
- Video and sound card make and model

#### **Faulty Disks**

If you receive a faulty CD, please contact the company you purchased the game from.

## Chapter 5

### **Credits**

#### **Windows Version**

**Kent Quirk:** Game Architect, Lead Designer and Programmer, Management.

**Kim Quirk:** Management, Marketing, QA, Documentation

**Zach Morong:** Lead Artist, Arena and Vehicle Design

**Nat Goodspeed:** Implementation Architect, Programmer

**Brian Sharp:** Graphics Engine Programmer

**Charlie Cleveland:** Game Programmer, Playability Design

**Steve Maitland Audio Design:** Music and Sound Effects

**Interns:** Matt Cole, Jeff Dubrule, Al Reed, Lincoln Quirk

**Consulting Artists:** Richard Bornemann, Janet Bornemann, David Burke, Randall McLamb

**QA:** Matt Cole, Patricia Pizer, Lincoln Quirk, Al Reed, Glenn Sugden

**Contributing Beta Testers:** Sam Adelman, Pat Allred, Craig Backlin, Justin Bailey, Shannon Beagle, Bill Benedict III, Steve and Reed Benjamin, James Berge, Rick Blair, Ross Borgeson, Daniel Bryant, Jason Coan, Ron Coit, Chris Cole, Ryan Cornetta, Alex Crew, Jonathon Deonarine, Patrick Dodson, Eric Ellingson, Damian Frank, Jason Giannini, Daryl Gleason, Ben Goodman, Peter Goodspeed, Eric Gross, Steve Hodson, Dan Holmes, David Huang, Peter and Natalie Kertzner, Niklas Konstenius, Daniel LaLiberte, James MacIntosh, Dave McClosky, Les Nelkin, Bill Oakford, Chris Quirk, Morgan Quirk, Eyal Ron, Eric St. Onge, Neil Stern, Brian Stormont, Dan Tevin, Paul Vadine, Sylvester Wong, Michael Zaroinski, Tijs Zwinkels, "Chaos", "FEPSDevGroup", "Hugo", "Jediah", "Mike"

## Chapter 5

**Special Thanks to:** Chuck Olson, Wendell Smith, Frank Zenie, Burleigh Hutchins, Art Bardige, Sandy Goseland, Rick Goodman, Cindy Null, Ryan Cornetta, Ned Roos.

**Extra appreciation** to the friends and family members who put up with our neglect while we got this out.

**Canine Companionship:** Bella

### **Linux Version (Loki Software)**

**President:** Scott Draeker

**Programmers:** Sam Lantinga

**Installer:** Sam Lantinga, Stéphane Peter

**QA/Tech Support:** Andy Mecham, Mike Phillips

**Artwork:** Katie Phillips, Jason Kim

**Beta Testers:** Lee Anderson, Mark "Shamgar" Bainter, Brandon Beattie, Fionn Behrens, Scott "Blaine" Bowden, Gary "Chunky" Briggs, Carl "DuckWing" Constantine, Frank "Svartalf" Earl, Matt qfingers Gerassimoff, Christopher Hahn, John "OverCode" Hall, David "NeoTron" Hedbor, Hayden "hjames" James, Will Johansson, Jason "deadman" Lundy, Brian "IndyZ" Martin, Brian "bwebpages" Mastenbrook, Gregory "Centove" McLean, Micro "MacSlow" Mueller, Michael "Herkemer" Parker, Bill "EmacsWeenie" Perry, Tim "BZFlag" Riker, Michael M00k3y Ritner, Michael "mIc" Sauer, Syphonius, Alexander "Omega" Trauzzi, Terry "keerf" Warner

### **Linux Version (Linux Game Publishing)**

**PHB:** Michael Simms

**Programmers:** Peter Tirsek, Mike Phillips, David Thompson

**Production QA:** Gareth Bentley

## Chapter 5

**Beta Testers:** Tom Badran, Colin Bayer, Shane Bayer, Brandon "Morphideous" Beattie, Dominik Behning, Chris Debenham, Ian Hastie, David "NeoTron" Hedbor, Lee Henderson, Jason, Joshua Kleiner, Marko Klingner, Al Koskelin, Michael Monreal, Daniel Olson, Palle "Absent" Raabjerg, Filip Van Raemdonck, Bernd Ritter, Jeff Ryden, Marius Schfer, Zachery J. Slater, Marko Teiste, Andreas Vuorinen, Terry Warner, Bob Zimbinski, Stjepan Zlodi, D S Yates

### CD Notes

All MindRover music was created by Steve Maitland, of Steve Maitland Audio Design at [www.smaudio.com](http://www.smaudio.com)

| <b>Track</b> | <b>Name</b>               |
|--------------|---------------------------|
| 1 . . . . .  | MindRover Game Data track |
| 2 . . . . .  | <i>Ice Bell</i>           |
| 3 . . . . .  | <i>Life On Ice</i>        |
| 4 . . . . .  | <i>Absolute Zero</i>      |
| 5 . . . . .  | <i>Brain Flakes</i>       |
| 6 . . . . .  | <i>Cryogenic</i>          |
| 7 . . . . .  | <i>Run Below</i>          |
| 8 . . . . .  | <i>Frozen Assets</i>      |
| 9 . . . . .  | <i>Frosting</i>           |
| 10 . . . . . | <i>Icescape (part 1)</i>  |
| 11 . . . . . | <i>Icescape (part 2)</i>  |
| 12 . . . . . | <i>Incognito</i>          |
| 13 . . . . . | <i>Jovian Moon</i>        |
| 14 . . . . . | <i>Cold Solder</i>        |
| 15 . . . . . | <i>Convolve</i>           |

## Chapter 5

### ***End User License Agreement***

YOU SHOULD CAREFULLY READ THE FOLLOWING END USER LICENSE AGREEMENT BEFORE INSTALLING THIS SOFTWARE PROGRAM. BY INSTALLING OR OTHERWISE USING THE SOFTWARE PROGRAM, YOU AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, PROMPTLY RETURN THE UNUSED SOFTWARE PROGRAM TO THE PLACE OF PURCHASE WITHIN 30 DAYS OF THE ORIGINAL PURCHASE.

This software program (the "Program"), any printed materials, any on-line or electronic documentation (the "Manuals"), and any and all copies and derivative works of such software program are the copyrighted work of Linux Game Publishing LTD and its licensors. All rights reserved, except as expressly stated herein. All use of the Program is governed by the terms of this End User License Agreement ("License Agreement"). The Program is solely for use by end users according to the terms of the Licence Agreement. Any use, reproduction or redistribution of the Program not in accordance with the terms of the Licence Agreement is expressly prohibited.

1. Limited Use License. Linux Game Publishing ("LGP") hereby grants, and by installing the Program you thereby accept, a limited, non-exclusive license and right to install and use one (1) copy of the Program for your use on either a home, business, or portable computer. The Program is licensed not sold. Your license confers no title or ownership in the Program.

2. Ownership. All title, ownership rights and intellectual property rights in and to the Program and any and all copies thereof (including but not limited to any titles, computer code, themes, objects, characters, character names, stories, dialog, catch phrases, locations, concepts, artwork, animations, sounds, musical compositions, audio-visual effects, methods of operation, moral rights, and any related documentation, and "applets" incorporated into the Program) are owned by LGP and its licensors.

## Chapter 5

The Program is protected by the copyright laws of the United Kingdom, international copyright treaties and conventions and other laws. All rights are reserved. The Program contains certain licensed materials and LGP's licensors may protect their rights in the event of any violation of this Agreement.

### 3. Responsibilities of End User.

A. Subject to the Grant of License hereinabove, you may not, in whole or in part, copy, photocopy, reproduce, translate, reverse engineer, derive source code, modify, disassemble, decompile, create derivative works based on the Program, or remove any proprietary notices or labels on the Program without the prior written consent of LGP.

B. The Program is licensed to you as a single product. Its component parts may not be separated for use on more than one computer.

C. You are entitled to use the Program for your own use, but you are not entitled to: (i) sell, grant a security interest in or transfer reproductions of the Program to other parties in any way, nor to rent, lease or license the Program to others without the prior written consent of LGP; or (ii) exploit the Program or any of its parts for any commercial purpose including, but not limited to, use at a cyber cafe, computer gaming center or any other location-based site. LGP may offer a separate Site License Agreement to permit you to make the Program available for commercial use. Contact LGP for details.

4. Program Transfer. You may permanently transfer all of your rights under this License Agreement, provided the recipient agrees to the terms of this License Agreement and you agree to remove the Program from any computer upon which it has been installed under the transferred License Agreement.

5. Termination. The License Agreement is effective until terminated. You may terminate the License Agreement at any time by destroying the Program. LGP may, at its discretion, terminate this License Agreement in the event that you fail to comply with the

## Chapter 5

terms and conditions contained herein. In such event, you must immediately destroy the Program.

6. Export Controls. The program may not be re-exported, downloaded, otherwise exported into (or to a national or resident of) any country to whom it would be illegal to export the Program under UK law.

7. Limited Warranty. Except as expressly and unambiguously set forth in this paragraph 7, LGP and its licensors disclaim any warranty for the Program, Editor and Manual(s). The Program, Editor and Manual(s) are provided "as is" without any warranty of any kind, either express or implied, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, or noninfringement. The entire risk arising out of use or performance of the Program, Editor, and Manual(s) remains with the User, however LGP warrants up to and including 90 days from the date of your purchase of the Program that the media containing the Program shall be free from defects in material and workmanship. In the event that the media proves to be defective during that time period, and upon presentation to LGP of proof of purchase of the defective Program, LGP will at its option: (i) correct any defect; (ii) provide a replacement copy of the same product; or (iii) refund your money.

8. Limitation of Liability. NEITHER LGP, ITS LICENSORS NOR ANY PARENT, SUBSIDIARY OR AFFILIATE OF THE FOREGOING SHALL BE LIABLE IN ANY WAY FOR LOSS OR DAMAGE OF ANY KIND RESULTING FROM THE USE OF THE PROGRAM, EDITOR, AND MANUAL(S) INCLUDING, WITHOUT LIMITATION, LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES.

9. Equitable Remedies. You hereby agree that LGP would be irreparably damaged if the terms of this License Agreement were not specifically enforced, and therefore you agree that LGP shall be entitled, without bond, other security, or proof of damages, to appropriate equitable remedies with respect to breaches of this



## Chapter 5

License Agreement, in addition to such other remedies as LGP may otherwise have available to it under applicable laws. In the event that any litigation is brought by either party in connection with this License Agreement, the prevailing party in such litigation shall be entitled to recover from the other party all the costs, attorney's fees and other expenses incurred by such prevailing party in the litigation.

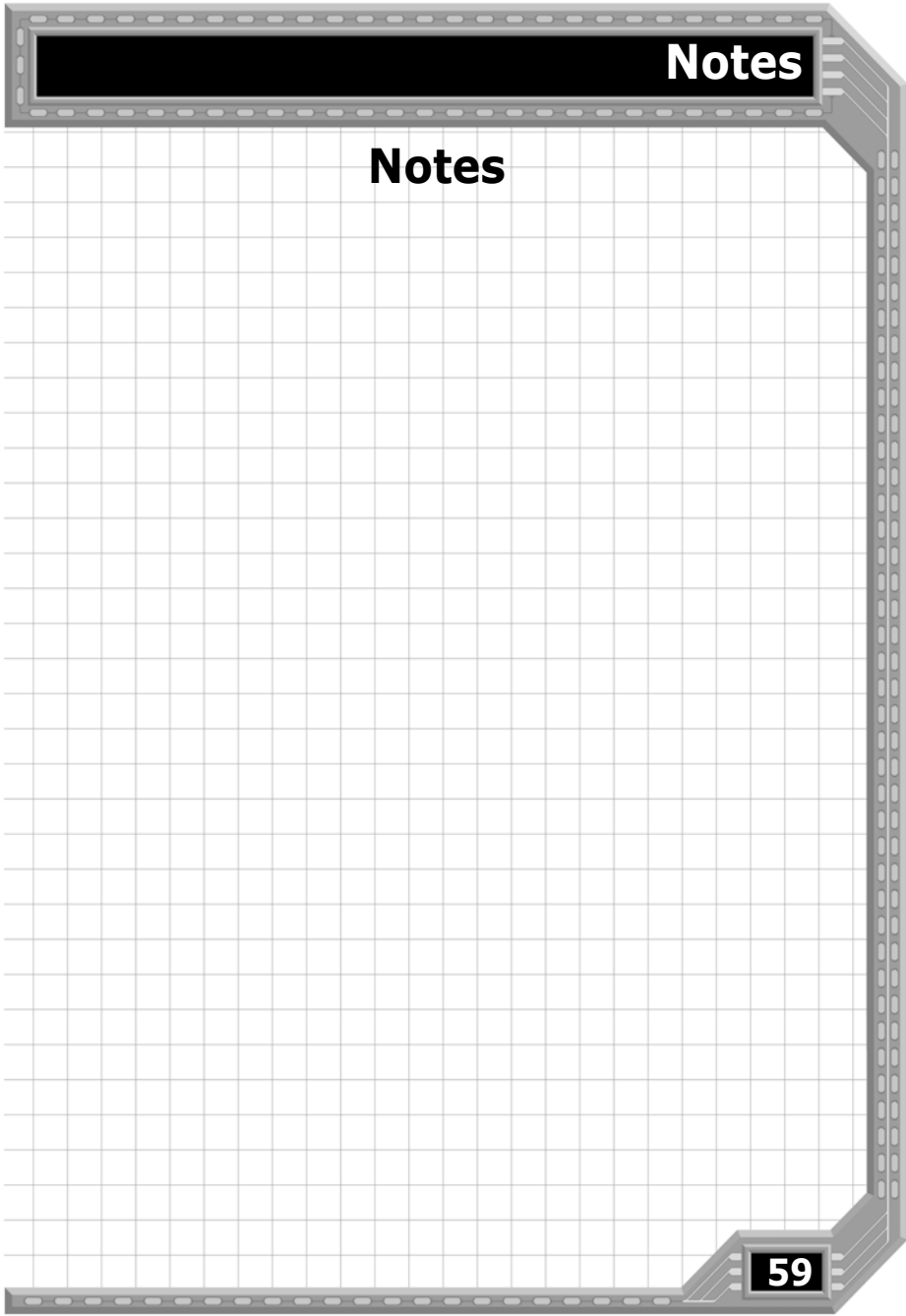
10. Limitations on License. Nothing in this License Agreement shall preclude you from making or authorizing the making of another copy of adaption of the Program provided, however, that (1) such new copy or adaption is created as an essential step in your utilization of the Program in accordance with the terms of this License Agreement and for NO OTHER PURPOSE; or (2) such new copy or adaption is for archival purposes ONLY and all archival copies are destroyed in the event of your Transfer of the Program, the Termination of this License Agreement or other circumstances under which your continued use of the Program ceases to be rightful.

11. Miscellaneous. This License Agreement shall be deemed to have been made and executed in the United Kingdom and any dispute arising hereunder shall be resolved in accordance with the laws of the UK. You agree that any claim asserted in any legal proceeding by one of the parties against the other shall be commenced and maintained in the appropriate court located in Nottingham, England, having subject matter jurisdiction with respect to the dispute between the parties. This License Agreement may be amended, altered or modified only by an instrument in writing, specifying such amendment, alteration or modification, executed by both parties. In the event that any provision of this License Agreement shall be held by a court or other tribunal of competent jurisdiction shall be unenforceable, such provision will be enforced to the maximum extent permissible and the remaining portions of this License Agreement shall remain in full force and effect. This License Agreement constituted and contains the entire agreement between the parties with respect to the subject matter hereof and supersedes any prior

## Chapter 5

oral or written agreements. You hereby acknowledge that you have read and understand the forgoing License Agreement and agree that the action of installing the Program is an express acknowledgment of your agreement to be bound by the terms and conditions of this License Agreement. You also acknowledge and agree that this License Agreement is the complete and exclusive statement of the agreement between you and LGP and that this License Agreement supersedes any prior or contemporaneous agreement, either oral or written, and any other communications between you and LGP.

Linux Game Publishing LTD  
17 Rossington Road  
Nottingham, NG2 4HX  
England



**Notes**

**Notes**

# Notes