

# ROBO RALLY



**von Gruppe 11:**  
**Clemens Schiebel**  
**Julius Giesler**  
**Kilian Klasen**  
**Mirjam Michalek**  
**Stefan Eller**  
**Tutor: Jan Hättig**

# Inhaltsverzeichnis

## Spielkonzept

Zusammenfassung des Spiels.....	3
Alleinstellungsmerkmal.....	3

## Benutzeroberfläche

Spieler-Interface	
Programmierphase.....	4
Zugphase.....	6
Menü-Struktur.....	7
Menü-Diagramm.....	8

## Technische Merkmale

Verwendete Technologien.....	9
Mindestvoraussetzungen.....	9

## Spiellogik

Objekte & Aktionen	
Spieleraktionen.....	10
Roboteraktionen.....	11
Roboterspezialaktionen.....	12
Spezialfelder.....	12
Bewegungsbefehle.....	13
Spielkonzepte.....	14
Spielablauf.....	14
Statistik.....	16
Achievements.....	17

## Screenplay

Storyline.....	19
----------------	----

# Spielkonzept

## *Zusammenfassung des Spiels*

*“Monatelang hatte er auf diesen Moment hin trainiert. Alle seine Systeme aktualisiert und die Gelenke geölt. Doch als der Roboter XF435-C7 an diesem alles entscheidenden Tag die Testhalle für Neuroboter betrat, überkam ihn das Gefühl, dem Schicksal ausgeliefert zu sein...”*

In “ROBO RALLY” schlüpfen Sie in die Rolle eines futuristischen Roboters. Finden Sie den idealen Algorithmus, um die anfallende Arbeit in der Fabrik vor Ihren Konkurrenten zu erledigen und bewahren Sie sich somit vor der Verschrottung. Doch Vorsicht! Überall lauern fiese Laserstrahlen, Förderbänder und Löcher, die Sie umgehen müssen. Ihre neidischen Kollegen werden alles daran setzen, Sie am Erfolg zu hindern. Geben Sie also Acht, dass Sie nicht von einem ihrer Laser erwischt oder gar in ein Loch geschoben werden! Klappern Sie nacheinander die verschiedenen Checkpoints ab, wobei Sie immer fünf Schritte auf einmal planen müssen. Nutzen Sie die Reparaturfelder oder den “Power-Down“-Modus ihres Roboters um entstandenen Schaden zu heilen. Erleben Sie an eigener Haut, wie es ist, wenn die Programmierung die Kontrolle übernimmt...

In “ROBO RALLY” können Sie sich mit einer oder mehreren KIs messen. Noch spannender wird dieses packende SciFi-Spiel jedoch durch den eingebauten Netzwerk-Modus, der es Ihnen ermöglicht, sich mit Ihren Freunden anzulegen. Sie haben außerdem die besondere Möglichkeit, sich unter mehreren Robotern Ihren Eigenen auszuwählen. Dieser verfügt dann über eine besondere Spezialfähigkeit, wie beispielsweise den starken Doppellaser oder den praktischen Laserspiegel. Durch diese Personalisierung kann es Ihnen gelingen, schon vor Spielbeginn die Weichen auf Sieg zu stellen. Viel Spaß!

## *Alleinstellungsmerkmal*

Das Besondere an „ROBO RALLY“ ist die einzigartige Spiellogik. Die Spieler müssen immer fünf Roboterschritte im Voraus planen, und sich für die Schrittfolge entscheiden, die sie für die Beste halten. Doch unvorhergesehene Ereignisse oder die Aktionen der anderen Roboter werfen diesen Plan immer wieder über den Haufen, was dazu führen, dass der vermeintlich gelungene Algorithmus den Roboter auf Abwege bringt.

Das macht „ROBO RALLY“ zu einem leicht verständlichen Strategiespiel, das auch nach sehr häufigem Spielen nicht langweilig wird!

# Benutzeroberfläche

„ROBO RALLY“ ist ein 2D Strategie Spiel. Während des gesamten Spiels schaut der Spieler aus der Vogelperspektive auf die Fabrikhalle. Am rechten Bildrand sind, je nach Spielphase, zwei verschiedene Interfaces zu sehen. Erläuterungen zu den Spielphasen finden sich im Abschnitt „Spielstruktur“.

## Spieler-Interface

### Programmierphase

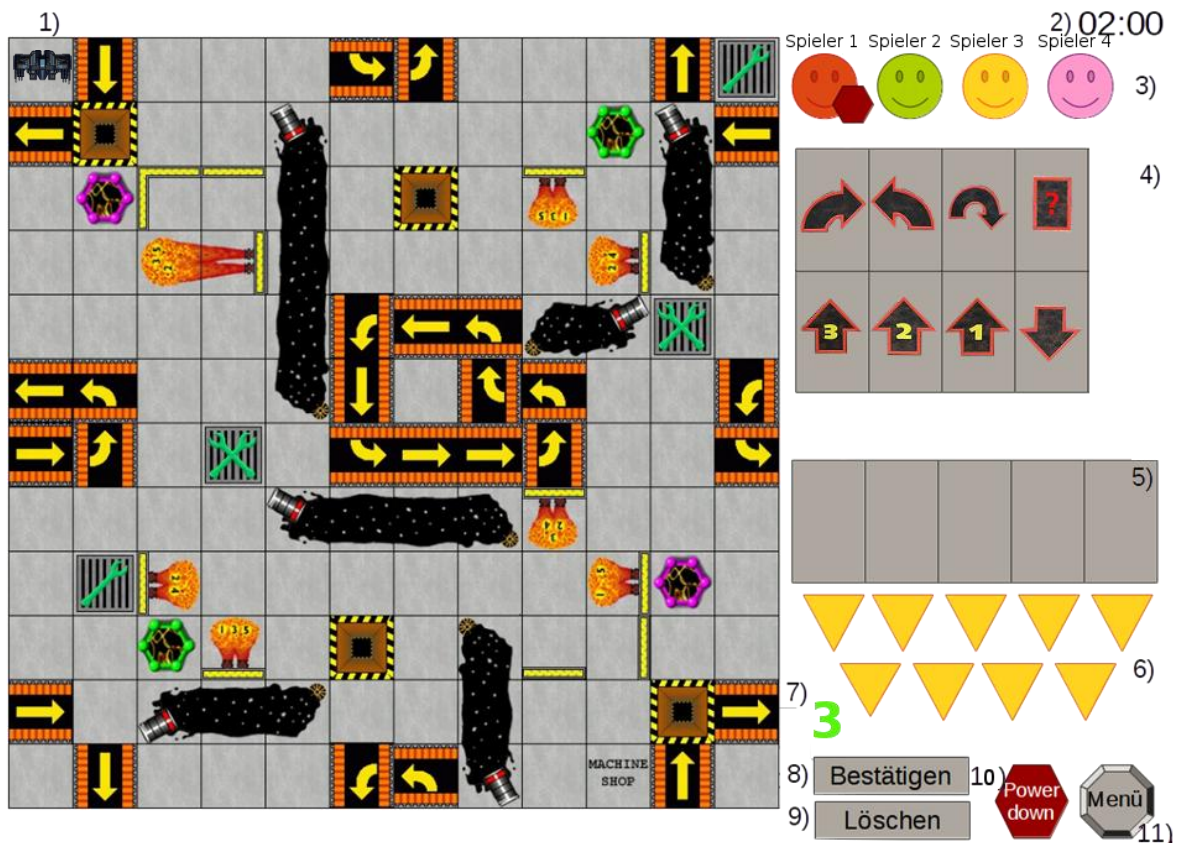


Abb. 1 - Programmierphase

### Erläuterungen zu Abbildung 1:

- 1) Hier befindet sich das Spielfeld, worauf das Spielgeschehen stattfindet (hier nur eine grobe Skizze). Es gibt verschiedene Feldertypen, welche im Abschnitt „Spielbojekte“ erläutert werden. Pro Mitspieler befindet sich ein Roboter auf dem Spielfeld(hier: oben links in der Ecke), welcher wie Abschnitt „Spielstruktur“ beschrieben gesteuert wird.
- 2) Zeigt an, wie viel Zeit noch verbleibt, um das Programm auszuwählen
- 3) Hier werden die Mitspieler mit Namen und einem Bild ihres Roboters angezeigt. Klickt man auf die Grafik, so öffnet sich ein Fenster mit Informationen über diesen Spieler. (Achievements, Robotertyp, Schadenspunkte, ...)
- 4) Hier befinden sich alle Bewegungen, die ein Roboter ausführen kann (siehe Abschnitt „Bewegungsbefehle“). Wie viele Bewegungsbefehle welcher Art man in der aktuellen Runde zur Verfügung hat, wird durch die blauen Zahlen angegeben. Steht von einer Bewegungsart keine zur Verfügung, so wird das Symbol transparent angezeigt. Das

Symbol mit dem Fragezeichen steht für eine aktivierbare Spezialfähigkeit, die bei Klick ausgeführt wird.

- 5) Hier werden die ausgewählten Bewegungsbefehle angezeigt. Ist ein Programmschritt „festgebrannt“ (siehe Abschnitt „Spielkonzepte“), so wird er farbig umrandet.
- 6) Dies sind die Schadenspunkte des eigenen Roboters. Nicht zugefügte Schadenspunkte transparent.
- 7) Hier werden die verbleibenden Lebenspunkte angezeigt.
- 8) Klicken beendet den Zug und verhindert weitere Änderungen des Programms. (Shortcut: Enter)
- 9) Dieser Button löscht alle bisher ausgewählten Programmschritte. (Shortcut: Entf)
- 10) Leitet einen Power-Down für die nächste Runde ein. (siehe Abschnitt „Spieleraktionen“)
- 11) Mit dem Menübutton gelangt man ins In-Game-Menü, wobei allerdings die Zeit weiterläuft. (Shortcut: Escape)

## Zugphase

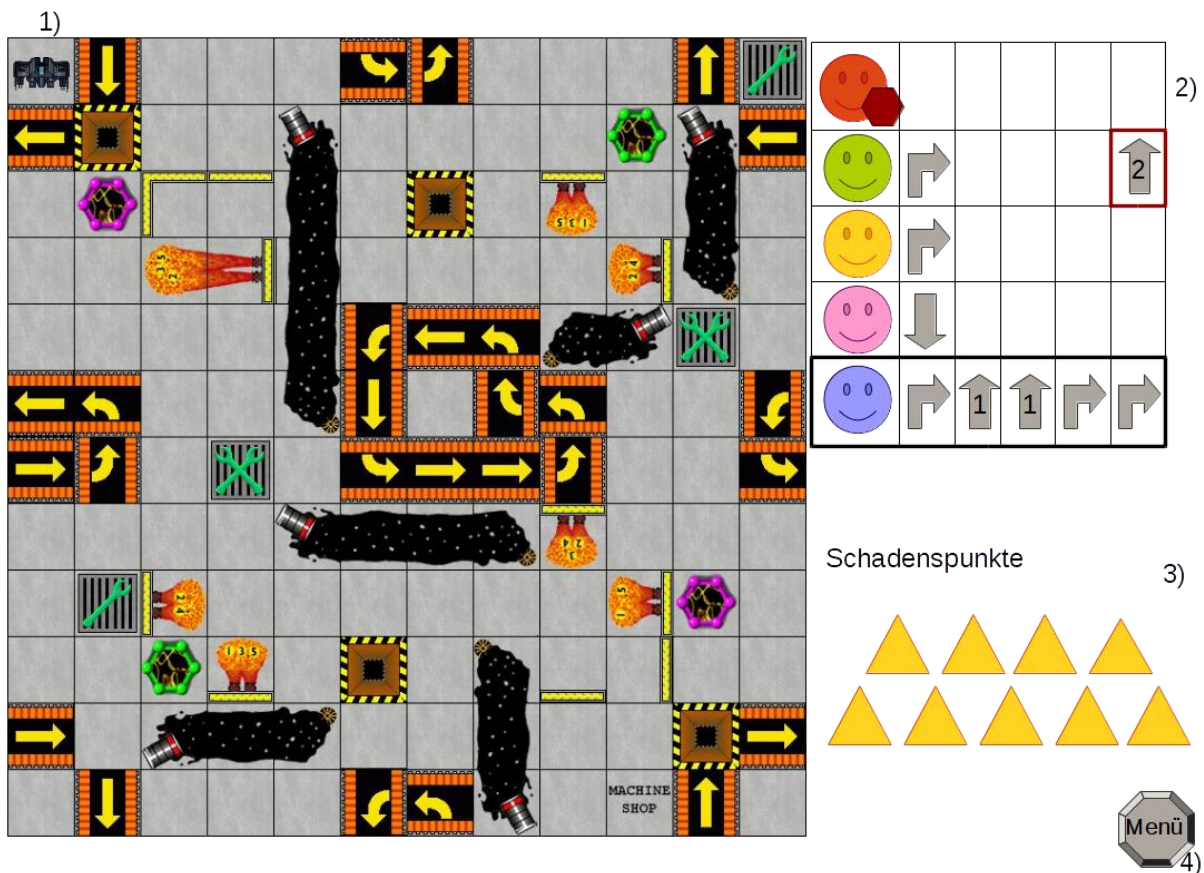


Abb. 2 Zugphase

### Erläuterungen zu Abbildung 1:

- 1) Spielfeld siehe oben.
- 2) In dieser Anzeige sieht man, welche Bewegungsbefehle die Mitspieler ausgewählt haben. Die eigenen Bewegungen werden in der untersten Zeile angezeigt. Pro Programmschritt werden erst alle Befehle sichtbar, danach werden sie ausgeführt. Die Eigenen und die „Festgebrannten“ der Gegner (siehe Abschnitt „Spielkonzepte“) sind von Anfang an zu sehen. Ist ein Roboter im Power-Down, so bleibt seine Zeile leer. Stirbt ein Roboter, so verfärbt sich seine Robotergrafik für den Rest der Runde und seine restlichen Züge werden nicht gezeigt.
- 3) Dies ist eine Anzeige der eigenen Schadenspunkte, die sich nach jedem Programmschritt aktualisiert.
- 4) In-Game-Menü siehe oben.

In beiden Phasen wird bei einem Mouseover auf einem Objekt (Spielfelder, Buttons, Leben, Schadenspunkte, ...) eine Kurzbeschreibung angezeigt.

## Menü-Struktur

Im Spiel wird mit der Maus navigiert. Für manche Buttons gibt es Shortcuts.

Im Hauptmenü (welches beim Start des Spiels aufgerufen wird) gibt es folgende Schaltflächen:

- Spiel starten: Hostet ein neues Spiel und betritt die Lobby
- Spiel beitreten: Der Spieler tritt dem Netzwerkspiel bei, falls es existiert.
- Einstellungen: Öffnet das Einstellungen-Menü.
- Statistiken & Achievements: Zeigt eigene Statistiken und Achievements an (vgl. Abschnitte „Statistik“ und „Achievements“).
- Beenden: Beendet das Spiel (Shortcut: Esc)

In der Lobby kann der Host

- Einen Spielplan aus einer Liste auswählen
- Die Anzahl der abzufahrenden Checkpoints einstellen
- Neue KIs hinzufügen und deren Roboter wählen
- KIs oder Mitspieler entfernen
- Die Länge des Timers für die Programmierphase einstellen oder ihn ganz deaktivieren
- Sofern alle anderen Spieler bereit sind:
  - Spiel starten: Startet ein neues Spiel mit den obigen Einstellungenoder
  - Spiel laden: Öffnet das „Spiel laden“-Menü

Alle Spieler können in der Lobby

- Ihren eigenen Roboter auswählen
- Die oben genannten Einstellungen des Hosts einsehen
- Alle Mitspieler und deren Roboter sehen
- Signalisieren, dass sie bereit sind

Im In-Game-Menü gibt es folgende Optionen:

- Zurück zum Spiel: Kehrt zum Spiel zurück. (Shortcut: Esc)
- Spiel speichern: Öffnet den „Spiel speichern“-Dialog. Im Netzwerkmodus kann nur der Host das Spiel speichern.
- Einstellungen: Öffnet das "Einstellungen"-Menü.
- Hauptmenü: Man wird gefragt, ob man das Spiel wirklich verlassen will, und gelangt bei Bestätigung ins Hauptmenü.
- RoboRallye beenden: Beendet nach einer Bestätigung das Spiel und kehrt zu Windows zurück.

Zudem wird, falls das Menü während der Programmierphase geöffnet wird, die verbleibende Rundenzeit angezeigt.

Das Einstellungen-Menü bietet folgende Einstellungsmöglichkeiten:

- Programmierhilfe („Ghost“) ein/aus (siehe Abschnitt „Spielstruktur“)
- Ton ein/aus
- Gesamtlautstärke

- Musik ein/aus
- Musikk Lautstärke
- Effektlautstärke
- Bildschirmauflösung ändern (Vollbild an/aus)

Der "Spiel speichern"-Dialog zeigt eine Liste bereits vorhandener Spielstände, die nach einer zusätzlichen Bestätigung überschrieben werden können, und hat ein Textfeld zur Eingabe eines neuen Dateinamens sowie zwei Buttons, um das Spiel unter dem eingegebenen Namen zu speichern oder zum Pause-Menü zurückzukehren.

Der "Spiel laden"-Dialog zeigt eine Liste von Spielständen, die mit den aktuell in der Lobby vorhandenen Spielern geladen werden können, und bietet einen Button, der den ausgewählten Spielstand lädt und einen, der zur Lobby zurück führt.

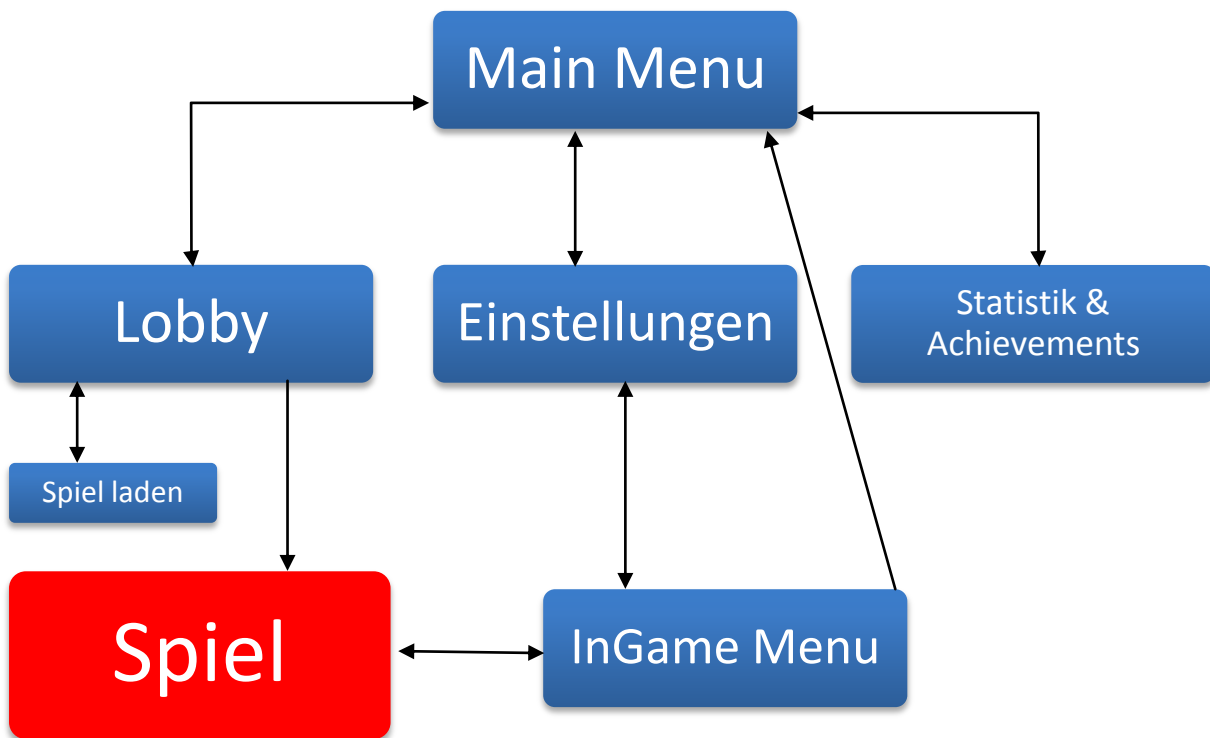


Abb.3 Menüdiagramm



# Technische Merkmale

## *Verwendete Technologien*

- Microsoft Visual Studio 2010/2013
- Microsoft .NET Framework 4.5
- Microsoft XNA 4.0
- Microsoft C#
- Resharper
- Trac
- SVN
- Google Docs

## *Mindestvoraussetzungen*

Prozessor: Intel Celeron 550 / 2 GHz

Betriebssystem: Windows Vista ab Service Pack 2 oder neuer

Arbeitsspeicher: 2 GB

freier Festplattenplatz: 1 GB

Grafikkarte mit Unterstützung für DirectX 9.0c und Shader Model 1.1

# Spiellogik

In „ROBO RALLY“ verfügt jeder Spieler über einen Roboter, den er über das oben beschriebene Interface steuern kann. Der Roboter ist das zentrale Spielobjekt, er verfügt über eine Spielfeldposition sowie über eine Richtung und zu Spielbeginn über drei Lebenspunkte. Außerdem hat jeder Roboter eine Spezialfähigkeit, die der Spieler zu Spielbeginn auswählt. Hier unterscheiden sich aktive, das heißt vom Spieler aktivierbare und passive Spezialfähigkeiten. Das Spielfeld, auf dem die Roboter sich bewegen, besteht aus verschiedenen Feldern, die ebenfalls auf den Roboter einwirken. Im Folgenden sind Spieleraktionen, Roboteraktionen und die verschiedenen Feldertypen erklärt.

## Objekte & Aktionen

### Spieleraktionen

Hier sind die Aktionen aufgelistet, die der Spieler während dem Spiel ausüben kann. Die Aktion IDS05 kann nur ausgeführt werden, wenn der Spieler über die entsprechende Spezialfähigkeit verfügt.

ID/Name	Ereignisfluss	Anfangsbedingungen	Abschlussbedingungen
IDS01: Bewegungsbefehl einprogrammieren	Klickt der Spieler auf einen Bewegungsbefehl, erscheint dieser im ersten freien Slot. Durch Drag and Drop kann er einen Befehl an eine beliebige, nicht „festgebrannte“(siehe Abschnitt „Spielkonzepte“) Stelle ziehen. Ist dieser Slot bereits besetzt, so werden die beiden Befehle vertauscht.	Das Spiel befindet sich in der Programmierphase.	Der Befehl befindet sich an der entsprechenden Stelle im Programm.
IDS02: Einzelnen Befehl löschen	Durch Klick auf einen belegten Slot wird dieser Befehl aus dem Slot entfernt.	Das Spiel befindet sich in der Programmierphase.	Der entsprechende Slot ist leer.
IDS02: Alle Befehle löschen	Durch Klick auf das Feld „Löschen“ oder die Taste „Entf“ leert der Spieler alle Slots.	Das Spiel befindet sich in der Programmierphase.	Das Programmierfeld ist leer.
IDS03: Auswahl bestätigen	Durch Klick auf das Feld „Bestätigen“ wird das Programmierfeld gesperrt und das Programm „eingeloggt“.	Das Spiel befindet sich in der Programmierphase, alle Felder des Programmierfeldes sind belegt.	Die Programmierphase ist beendet. Wenn alle Spieler diese beendet haben, beginnt die Zugphase.
IDS04: Power-Down einleiten	Durch Klick auf das Feld „Power-Down“ wird der Power-Down für die nächste Runde eingeleitet.	Das Spiel befindet sich in der Programmierphase.	In der nächsten Runde geht der Roboter in den Power-Down. (siehe Abschnitt „Spielstruktur“)
IDS05: Bewegungsbefehle erneuern, SPEZIALFÄHIGKEIT	Durch Klick auf den Slot „Spezialfähigkeit“ werden die verfügbaren Bewegungsbefehle gelöscht und durch neue ersetzt.	Das Spiel befindet sich in der Programmierphase. Der Spieler hat den Roboter mit der entsprechenden Spezialfähigkeit ausgewählt.	Der Spieler erhält neue Bewegungsbefehle. Die Spezialfähigkeit ist für diese Runde blockiert.

## Roboteraktionen

Hier sind die Aktionen aufgelistet, die der Roboter während dem Spiel ausüben kann.

<b>ID/Name</b>	<b>Ereignisfluss</b>	<b>Anfangsbedingungen</b>	<b>Abschlussbedingungen</b>
IDR01: Schritte machen	Der Roboter geht ein, zwei oder drei Felder nach vorne, bzw. eins zurück. Stößt der Roboter auf eine Wand, bleibt er stehen.	Das Spiel befindet sich in der Zugphase. Im nächsten Slot liegt ein Schrittbefehl.	Der Roboter hat null bis drei Schritte oder einen zurück gemacht.
IDR02: Drehung	Der Roboter dreht sich um 90°, 180° oder 270°.	Das Spiel befindet sich in der Zugphase. Im nächsten Slot liegt ein Drehungsbefehl.	Der Roboter ist gedreht.
IDR03: Schießen	Der Roboter schießt einen Laserstrahl in Blickrichtung ab. Trifft der Strahl einen anderen Roboter, so bekommt dieser einen Schadenspunkt. Erhält der Spieler seinen neunten Schadenspunkt, stirbt er.	Das Spiel befindet sich in der Zugphase, nachdem alle Roboter ihre Bewegung gemacht haben.	Der Roboter hat geschossen. Die Laserstrahlen sind geflogen bis sie eine Wand, den Spielfeldrand oder einen Roboter getroffen haben.
IDR04: Sterben	Der Roboter explodiert.	Das Spiel befindet sich in der Zugphase.	Der Roboter verlässt das Spielfeld.
IDR05: Sicherungskopie anlegen	Steht der Roboter am Ende der Runde auf einem Reparaturfeld, oder erreicht der Roboter einen Checkpoint, erstellt er an dieser Stelle eine Sicherungskopie.	Das Spiel befindet sich in der Zugphase, der Roboter hat ein entsprechendes Feld erreicht.	Der Roboter hat eine neue Sicherungskopie angelegt.
IDR06: Respawn	Der Roboter wird an der Stelle seiner letzten Sicherungskopie gespawnt. Hat er keine erstellt, dann am Startfeld. Seine Schadenspunkte gehen ihm ebenso verloren wie einer seiner Lebenspunkte.	Das Spiel ist in der Initialisierungsphase. Der Roboter steht nicht auf dem Spielfeld und hat noch mindestens einen Lebenspunkt übrig.	Der Roboter verliert einen Lebenspunkt. Der Roboter steht an der Stelle seiner letzten Sicherungskopie/seinem Startfeld.
IDR07: Roboter Schieben	Betrifft ein Roboter ein Feld, auf dem bereits ein anderer Roboter steht, wird dieser in gleicher Richtung ein Feld weitergeschoben. Ist hier eine Wand, werden beide Bewegungen nicht ausgeführt.	Das Spiel befindet sich in der Zugphase, ein Roboter hat versucht ein Feld zu betreten, auf dem bereits ein Roboter stand.	Beide Roboter haben entweder keinen oder einen Schritt gemacht.
IDR08: In ein Loch fallen	Betrifft ein Roboter ein Feld des Typs „Loch“ (siehe „Feldertypen“), stirbt er.	Das Spiel befindet sich in der Zugphase, ein Roboter hat ein Loch betreten	Der Roboter stirbt. → IDR04
IDR09: Schadenspunkt erhalten	Der Roboter erhält einen Schadenspunkt. Ist es der neunte, stirbt er.	Das Spiel befindet sich in der Zugphase, der Roboter wurde beschossen oder ist in einen Feldlaser gelaufen. (siehe „Feldertypen“)	Der Roboter hat einen Schadenspunkt mehr. Ist es der neunte, stirbt er. → IDR04

### Roboterspezialaktionen

Hier sind die Aktionen aufgelistet, die der Roboter nur ausüben kann, wenn er über die entsprechende Spezialfähigkeit verfügt.

<b>ID/Name</b>	<b>Ereignisfluss</b>	<b>Anfangsbedingungen</b>	<b>Abschlussbedingungen</b>
IDR10: Doppellaser SPEZIALFÄHIGKEIT	Siehe IDR03. Einziger Unterschied: Es werden zwei Schadenspunkte hinzugefügt.	Siehe IDR03. Der Roboter hat die Spezialfähigkeit „Doppellaser“.	Siehe IDR03.
IDR11: Regenerierung SPEZIALFÄHIGKEIT	Der Roboter verliert einen Schadenspunkt. Dies passiert am Ende jeder Runde.	Der Roboter hat mindestens einen Schadenspunkt erhalten. Der Roboter hat die Spezialfähigkeit „Regenerierung“.	Der Roboter hat einen Schadenspunkt weniger.
IDR12: Spiegel SPEZIALFÄHIGKEIT	Wird der Roboter an der Rückseite getroffen, reflektiert er den Schuss. Der schießende Roboter erhält die Schadenspunkte, die er sonst verursacht hätte. Erhält er damit seinen neunten Schadenspunkt, stirbt er.	Der Roboter wurde an der Rückseite getroffen. Der Roboter hat die Spezialfähigkeit „Spiegel“.	Der schießende Roboter hat Schadenspunkte abbekommen.
IDR13: Pause SPEZIALFÄHIGKEIT	Der Roboter macht nichts.	Das Spiel befindet sich in der Zugphase. Im nächsten Slot liegt der Spezialbewegungsbefehl „Pause“. Der Roboter hat die Spezialfähigkeit „Pause“.	Der Roboter steht genauso da wie vorher.









### Spezialfelder

Manche der Felder verfügen über Spezialeigenschaften. Diese werden zu den in „Spielstruktur“ aufgeführten Zeitpunkten aktiviert. Der Checkpoint sowie das Loch werden sofort aktiviert, wenn ein Roboter sie betritt. Das Reparaturfeld wird am Ende der Runde aktiviert. Es gibt X nummerierte Checkpoints. Checkpoint 0 ist das Startfeld.

<b>ID/Name</b>	<b>Aktion bei Betritt</b>
IDF01: Loch	Der Roboter stirbt.
IDF02: Checkpoint	Der Roboter legt eine Sicherungskopie an, hakt Checkpoint Nr. x ab, wenn Nr. (x-1) schon abgehakt ist. Wer zuerst Checkpoint X abhakt, gewinnt das Spiel.
IDF03: (Doppel-) Reparaturfeld	Der Roboter verliert einen (bzw. zwei) Schadenspunkte.
IDF04: (Doppel-) Förderband	Verfügt über eine Richtung und einen (doppelten) Pfeil. Der Roboter wird ein (bzw. zwei) Felder in diese Richtung geschoben (kann Drehung beinhalten).
IDF05: Zahnrad	Verfügt über eine Drehrichtung. Der Roboter wird um 90° in diese Richtung gedreht.
IDF06: Wand	Liegt am Rand eines Feldes.
IDF07: Wand mit Laser	Liegt am Rand eines. Hat an einer Seite einen „Feldlaser“. Dieser schießt am Ende der Runde einen Laserstrahl, der allen Robotern, die er trifft, einen Schadenspunkt gibt.
IDF08: Ölfeld	Betritt der Roboter das Ölfeld, schlittert er geradeaus weiter bis zum Ende des Ölfelds, und ein Feld über das Ölfeld hinaus. Schlitternde Roboter schieben nicht, sondern bleiben stehen, wenn sie auf einen anderen Roboter treffen.

## Bewegungsbefehle

Hier sind die verschiedenen zur Verfügung stehenden Programmierbefehle aufgeführt. Jeder Bewegungsbefehl verfügt über eine Priorität, die bei der Ausführung verglichen wird.

Bild	Befehl	Bild	Befehl
	Einen Schritt nach vorne. → IDR01		Drehung um 90° nach rechts. → IDR02
	Zwei Schritte nach vorne. → IDR01		Drehung um 90° nach links. → IDR02
	Drei Schritte nach vorne. → IDR01		Drehung um 180°. → IDR02
	Einen Schritt zurück. → IDR01		Spezialeigenschaft. → IDS05 / IDR13

## *Spielkonzepte*

### Schieben:

Roboter schieben sich in intuitiver Weise, das heißt: falls ein Roboter in Richtung eines anderen steht und sich auf oder über dessen Feld bewegen würde, schiebt er den Roboter vor sich her. Dabei kann er den anderen Roboter nicht durch eine Wand schieben, seine eigene Bewegung verläuft also potentiell kürzer oder überhaupt nicht.

### Schadenspunkte, festgebrannte Befehle und Sterben:

Erhält ein Roboter einen Schadenspunkt (durch Respawn, Roboterlaser, oder Spezialfelder) kann das Folgen haben: Falls der Roboter fünf Schadenspunkte hat, brennt sich sein Befehlsschritt in Slot 5 fest. Dieser Befehl wird, so lange er festgebrannt ist, jede Runde ausgeführt und kann in der Programmierphase nicht verändert werden. Jeder weitere Schadenspunkt nach dem fünften brennt einen weiteren Befehlsschritt fest.

Falls der Roboter 9 oder mehr Schadenspunkte hat, stirbt er und wird sofort vom Spielfeld genommen.

Alternativ kann ein Roboter auch sterben, wenn er in ein Loch oder über den Kartenrand bewegt wird.

### Sicherungskopie:

Erreicht ein Roboter einen Checkpoint oder ein Reparaturfeld legt er dort eine Sicherungskopie an. Diese Sicherungskopie wird der nächste Respawnpunkt des Roboters. Ist beim Respawn das Feld auf dem die Sicherungskopie liegt bereits durch einen anderen Roboter blockiert, erfolgt der Respawn am Startfeld des Roboters.

### Respawn:

Gestorbene Roboter, die noch einen Lebenspunkt haben, werden neu gebaut und verlieren ein Leben, alle Schadenspunkte. Sie werden mit zwei Schadenspunkten dort auf das Spielfeld gesetzt, wo sich ihre Sicherheitskopie befindet.

## *Spielablauf*

Zu Spielbeginn befinden sich die Roboter aller am Spiel beteiligter Spieler (KIs und/oder Menschen) auf je einem Startfeld am Spielfeldrand. Auf dem Spielfeld sind mehrere aufsteigend nummerierte Checkpoints aufgestellt. Ziel des Spieles ist es, als Erster die Checkpoints mit seinem Roboter abzulaufen.

Eine Runde gliedert sich in drei Phasen:

1. Initialisierungs-Phase
2. Programmier-Phase
3. Zug-Phase

#### 1.) Initialisierungs-Phase:

Respawn gestorbener Roboter.

Alle Spieler bekommen bis zu 9 zufällige Bewegungsbefehle zugeteilt. Pro Schadenspunkt erhält der Spieler einen Bewegungsbefehl weniger.

## 2.) Programmierphase:

In der Programmierphase entscheidet der Spieler, was sein Roboter im nächsten Zug tun wird. Dies beinhaltet das Auswählen der Bewegungsbefehle, falls möglich das Aktivieren von Spezialfähigkeiten und, falls gewünscht, die Ankündigung eines Power-Downs.

### Auswählen der Bewegungsbefehle:

Der Spieler legt in jeden seiner freien Programmslots jeweils einen seiner erhaltenen Bewegungsbefehle. Jeder Befehl kann dabei nur in einen Slot gelegt werden.

Ist die Programmierhilfe aktiviert, zeigt ein farbiger Ghost-Roboter dem Spieler in Echtzeit die Position seines Roboters am Ende der Runde an, wenn das Programm wie geplant ablaufen würde.

### Aktivieren der Spezialfähigkeiten:

Hat der Spieler in der Lobby eine aktivierbare Spezifähigkeit (IDS05 oder IDR13) ausgewählt, kann er diese durch Klick auf den entsprechenden Button auswählen und je nach Fähigkeit entsprechend aktivieren.

### Ankündigung eines Power-Downs:

Wenn der Spieler die Schadenspunkte seines Roboters zurücksetzen möchte, kann er den Power-Down Button klicken. Dann wird sich der Roboter in der nächsten Runde abschalten. Die aktuelle Runde wird normal gespielt. In der Zugphase werden die anderen Spieler über die Ankündigung informiert. Am Ende der Zugphase schaltet sich der Roboter ab und heilt sofort alle Schadenspunkte. Die folgende Programmierphase wird für diesen Spieler übersprungen, der Roboter bewegt sich also eine Runde lang nicht von selbst. Er kann dennoch von anderen Robotern oder Spielfeldern Schaden nehmen oder geschoben werden.

Der Spieler kann die Programmierphase beenden, wenn in jedem Programmslot eine Programmkarte liegt und er durch Klick auf "Auswahl bestätigen" das Programm startet.

Die Programmierphase endet automatisch nach einer vom Host voreingestellten Zeit oder wenn alle menschlichen Spieler die Phase manuell beendet haben. (Falls KI im Spiel ist, erst nach [15] Sekunden).

Leere Befehlsslots am Ende der Programmierphase werden zufällig aufgefüllt.

## 3.) Zugphase:

Wenn alle Spieler gewählt haben, werden die Befehle ausgeführt. Die Roboter tun dies automatisch. Die Spieler haben keine Interaktionsmöglichkeiten mit dem Spielbrett und schauen zu. Sie können aber im Interface die aktuellen Programme und den Lebens/Schadenspunktstatus der anderen Roboter ansehen.

Die Zugphase gliedert sich in fünf Programmschritte. In Programmschritt 1 werden alle Befehle aller Roboter in Slot 1 der Reihe nach durchgeführt, usw.

Der Programmschritt ist wiederum in Phasen unterteilt:

1. Roboterbewegung
2. Ölfeld
3. Langsames Laufband
4. Schnelles Laufband
5. Zahnräder

6. Feldlaser
7. Roboterlaser

1. Roboterbewegung:  
Es beginnt der Roboter auf dem Spielfeld, der den Befehl mit der höchsten Prioritätszahl im Slot hat.  
Der Roboter bewegt sich entsprechend des Befehls. Dabei kann es zum Schieben anderer Roboter kommen.  
Anschließend werden die Befehle in Slot 1 der anderen Roboter auf dem Spielfeld nach absteigender Prioritätszahl ausgeführt. Wieder ist Schieben möglich.
- 2.- 6. Spezialfelder:  
Die entsprechenden Felder führen ihre Aktionen in dieser Reihenfolge (2. bis 6.) aus. Stirbt ein Roboter, stirbt er und wird sofort bis zur nächsten Initialisierungsphase aus dem Spiel genommen.
7. Roboterlaser:  
Die Roboter schießen mit ihren Lasern geradeaus. Die Laserstrahlen fliegen bis sie eine Wand, den Spielfeldrand oder einen Roboter treffen. Getroffene Roboter erhalten einen Schadenspunkt.

Die Runden wiederholen sich so lange, bis ein Roboter alle Checkpoints der Reihe nach erreicht hat, oder alle anderen Spieler ausgeschieden sind.

### *Statistiken*

Das Spiel sammelt folgende Statistiken:

- Felder gelaufen (vorwärts/rückwärts)
- Linksdrehungen
- Rechtsdrehungen
- Kehrtwenden
- Schaden genommen
- Tode gestorben
- Schaden verursacht
- Roboter zerstört
- Schadenspunkte repariert
- Power-Downs durchgeführt
- Gegner geschoben

Nach jedem Spiel werden die Statistiken für das eben abgeschlossene Spiel gezeigt. Vom Hauptmenü aus kann die Summe aus allen bisherigen Spielen sowie der beste in einem einzelnen Spiel erreichte Wert angesehen werden.



## Achievements

In allen Spielen (kumulativ):

Name	Bedingung
Erstes Öl	Ein Roboter zerstört
Ölrausch	Fünf Roboter zerstört
Amoklauf	Zehn Roboter zerstört
Kommunalpolitiker	Eine Kehrtwenden vollzogen
Mitglied des Bundestages	Fünf Kehrtwenden vollzogen
Bundeskanzlerin	Zehn Kehrtwenden vollzogen
Aua.	Fünf Schadenspunkte erhalten
Nur ein Kratzer	20 Schadenspunkte erhalten
Strukturelle Integrität gefährdet	50 Schadenspunkte erhalten
Für einen höheren Zweck	Einmal gestorben
Zombie	Fünfmal gestorben
Ich werde weiterleben!	Zehnmal gestorben
Ein Schritt nach dem anderen	Zehn Felder gelaufen
Achilles	50 Felder gelaufen
Schildkröte	100 Felder gelaufen
Taktischer Rückzug	Fünf Felder nach hinten gelaufen
Wilde Flucht	Zehn Felder nach hinten gelaufen
Vormarsch in die andere Richtung	20 Felder nach hinten gelaufen
Drehwurm	insgesamt 5 Drehungen vollzogen
Schwindel	insgesamt 20 Drehungen vollzogen
Totaler Orientierungsverlust	insgesamt 50 Drehungen vollzogen
Sylvester Stallone	Fünf Schaden an anderen Robotern verursacht
Hulk	20 Schaden verursacht
Godzilla	50 Schaden verursacht
Lackierung erneuert	Einen Schadenspunkt repariert
Öl gewechselt	Fünf Schadenspunkte repariert

Generalüberholt	20 Schadenspunkte repariert
Eine Pause tut gut.	Einen Power-Down durchgeführt
Ausgeschlafen	Fünf Power-Downs durchgeführt
Gut erholt	Zehn Power-Downs durchgeführt
Platz da, jetzt komme ich!	Zehn andere Roboter verschoben

in einem einzelnen Spiel:

<b>Name</b>	<b>Bedingung</b>
Unantastbar	Keinen Schaden genommen
Wer braucht schon Backups?	Nicht gestorben
Pazifist	Kein Schaden ausgeteilt

# Screenplay

## Storyline

“ROBO RALLY” spielt in einer nicht allzu weit entfernten Zukunft, in der autonome Roboter zu einem nicht wegzudenkenden Teil des alltäglichen Lebens geworden sind. Um die herausragenden Fähigkeiten dieser künstlichen Lebensformen sicherzustellen, hat die Roboterfabrik eine rigide Qualitätskontrolle eingeführt. Teil dieser ist ein sogenannter “Live-Konkurrenz-Test”, bei dem sich mehrere Roboter in einer Fabrikhalle miteinander messen. Die Aufgabe: An verschiedenen Checkpoints im Gebäude kleine Arbeiten zu verrichten. Die Roboter legen alles daran, schneller zu sein als ihre Konkurrenten, denn die Verlierer werden verschrottet. Und dieser Wettstreit läuft nicht friedlich ab. Nur der geschickteste Roboter kann der sicheren Vernichtung entgehen!

Die Fabrik bietet fünf verschiedene Robotermodelle zum Verkauf:

- 1.) **Tanky:** *“C’mon! Shoot me!”* Dieser Roboter ist ein robuster Typ. Regeneriert nach jeder Runde einen Schadenspunkt.
- 2.) **James Bot:** *“Version 0.07. Takes his oil shaken, not stirred.”* Ein hervorragender Schütze am Doppellaser. Verursacht pro Treffer zwei Schadenspunkte.
- 3.) **LazyBot:** *“He’s a chilling machine!”* Er ist einer von der gemütlichen Sorte. Kann pro Runde einmal pausieren.
- 4.) **Albot:** *“ $E=mc^2$ ”* Der intelligenteste Roboter in diesem Universum. Kann einmal pro Runde neue Bewegungsbefehle anfordern.
- 5.) **Peach:** *“Mirror, Mirror on the wall...”* Wer auf ihn schießt, lebt gefährlich. Reflektiert allen Schaden auf den Verursacher, wenn dieser ihn an der Rückseite trifft.