1 Einleitung

1.1 Minimalanforderungen

CeeBot erfordert einen Computer neueren Datums. Ein optimales Erscheinungsbild kann nur mit einer guten 3D-Karte erzielt werden.

- Prozessor 300 MHz, 64 MB RAM
- 3D-Grafikkarte mit 16 MB RAM
- 100 MB freier Speicherplatz auf der Festplatte
- Windows XP, 2000, ME, 98 oder 95
- DirectX 8a (Wenn nötig wird DirectX 8a von CeeBot-A installiert)

Empfohlene Grafikkarten:

- nVidia GeForce 2, 3 oder 4
- Matrox G400, G450, G550 oder Parhelia

3dfx Voodoo Grafikkarten werden offiziell nicht unterstützt, können aber durchaus funktionieren (ohne Garantie).

Mit einigen Grafikchips in Laptops werden keine zufriedenstellenden Ergebnisse erzielt.

1.2 Installation von CeeBot-A

Legen Sie die CD-ROM CeeBot-A in das Laufwerk ein.

Normalerweise sollte das Installationsprogramm nach einigen Sekunden automatisch starten. Klicken Sie auf **Installieren**.

Wenn das Installationsprogramm nicht automatisch startet, öffnen Sie das CD-ROM-Laufwerk und Sie zweimal auf **install**.

Wenn Sie CeeBot-A über eine ältere Version installieren, werden alle Ihre gespeicherten Programme und die Informationen über die schon vollendeten Übungen gelöscht. Installieren Sie CeeBot-A ggf. in einem anderen Ordner oder sichern Sie Ihre Dateien zuvor.

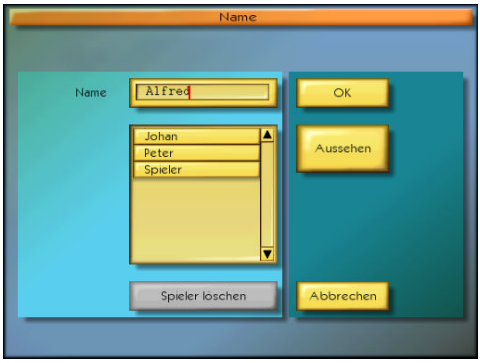
Bei der Installation wird geprüft, ob auf Ihrem Computer DirectX 8a installiert ist. Wenn nicht bestätigen Sie bitte, dass Sie DirectX 8a installieren wollen.

1.3 Deinstallieren von CeeBot-A

- Klicken Sie auf **Start** - **Systemsteuerung** - **Software**.
- Wählen Sie links oben **Programme ändern oder entfernen**.
- Wählen Sie in der Liste das Programm **CeeBot-A** aus.
- Klicken Sie auf **Ändern/Entfernen**.

2 Erste Schritte

In diesem Abschnitt wird Schritt für Schritt erklärt, wie Sie die erste Übung absolvieren können.



Zuerst bittet CeeBot Sie, Ihren Namen einzugeben. Jeder Spieler sollte einen anderen Namen benutzen, da CeeBot Ihre Programme automatisch speichert und die absolvierten Übungen kennzeichnet. Geben Sie also Ihren Namen ein, er wird später in der Liste erscheinen. Klicken Sie auf OK, dann auf

Auf dem Bildschirm erscheinen nun die verschiedenen Übungen. In der linken Kolonne erscheinen die verschiedenen Kapitel, und rechts die verschiedenen Übungen des ausgewählten Kapitels. Ein Haken bedeutet, das die Übung erfolgreich abgeschlossen wurde.

Die Übungen werden schrittweise schwieriger. Wir empfehlen, dass Sie sie der Reihe nach absolvieren, auch wenn dies nicht zwingend ist.

Klicken Sie auf **1: Grundlagen** und **1: Move**, dann klicken Sie auf **Spielen**.

Es erscheint ein grüner Text oben am Bildschirm der Sie auffordert, auf **F1** zu drücken, damit die Anweisungen angezeigt werden. So erhalten Sie alle nötigen Informationen. Sie können auch auf die folgende Schaltfläche klicken:

Lesen Sie die Anweisungen sorgfältig durch. Sobald Sie fertig sind, klicken Sie auf folgende Schaltfläche links unten am Bildschirm:

Sie können ggf. jederzeit wieder die Anweisungen einsehen, indem Sie auf **F1** drücken.

Der Roboter, den Sie programmieren müssen, ist schon angewählt: Sein Symbol oben links am Bildschirm ist orange. Um einen Roboter anzuwählen klicken Sie auf das entsprechende Symbol, oder direkt auf den Roboter.

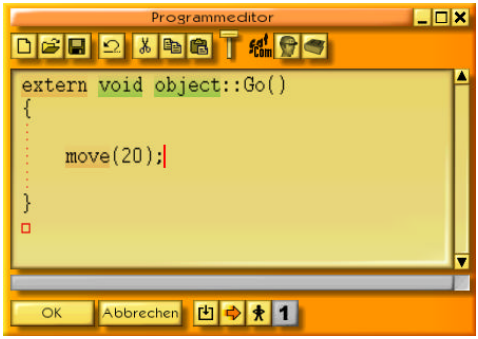
Wählen Sie das erste Programm in der Liste aus. Jeder Roboter hat 10 Slots für Programme. So können Sie verschiedene Lösungen ausprobieren. Die Programme werden automatisch für jeden Spieler getrennt gespeichert.

Programm 4 enthält die Lösung der Übung, so dass Sie nicht stecken bleiben. Versuchen Sie jedoch, nicht zu früh aufzugeben.

Um in den Programmeditor zu gelangen, klicken Sie auf das Symbol **{..}**.

Der Editor erscheint mit einem leeren Rahmen für ein Programm, in den Sie Ihr Programm einfügen können.

Die Einfügemarke befindet sich zwischen den geschweiften Klammern, wo Sie die Anweisungen für das Programm eingeben müssen. Verändern Sie nicht die erste Zeile des Programms oder die geschweiften Klammern.

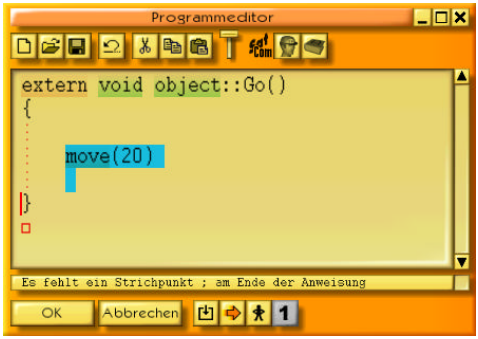


Das Ziel der Übung ist es, den Roboter gerade nach vorne zu fahren, so dass er auf dem blauen Kreuz 20 Meter weiter vorne zum stehen kommt.

Geben Sie dafür die folgende Anweisung ein:

```
move(20);
```

Vergessen Sie nicht den Strichpunkt ";" am Ende der Anweisung.



Klicken Sie auf **OK**

Wenn Sie eine Fehler gemacht haben, wird die entsprechende Stelle in blau angewählt, und eine entsprechende Meldung erscheint unten am Bildschirm. Im Beispiel rechts wurde z.B. der Strichpunkt ";" am Ende der Zeile vergessen.

Wenn alles richtig ist, wird der Editor durch einen Klick auf **OK** geschlossen.

Klicken Sie auf das Pfeilsymbol um Ihr Programm auszuführen. Der Roboter sollte dann bis zum blauen Kreuz fahren.

Bravo, Sie haben gerade Ihr erstes Programm geschrieben.

Gehen Sie dann zur nächsten Übung über. Sie ähnelt der ersten Übung, Sie müssen jedoch einen roten Strich ziehen.

Als erstes brauchen Sie also eine Anweisung, die den roten Bleistift auf das Papier setzt. Achten Sie auf das **R** von **Red**, das groß geschrieben ist:

```
pendown(Red);
```

Geben Sie auf der nächsten Zeile eine Anweisung ein, die den Roboter nach vorne bewegt:

```
move(20);
```

Vergessen Sie den **;** nicht.

3 Hauptmenü

3.1 Übungen

In diesem Teil von CeeBot-A lernen Sie, wie Sie die Roboter programmieren können, auch wenn Sie keinerlei Programmierkenntnisse haben. Die Übungen sind in Kapitel gruppiert, und weihen Sie schrittweise in die Grundlagen einer strukturierten und objektorientierten Programmiersprache ein. Sie können die Übungen in einer beliebigen Reihenfolge machen, es ist jedoch besser, mit den ersten Übungen anzufangen und sich nach und nach zu den schwierigeren Übungen vorzuarbeiten.

3.2 Challenges

Jeder Challenge ist eine Herausforderung an Ihre Programmierkenntnisse, und erfordert schon einiges an Erfahrung, die Sie sich mit den Übungen aneignen können. Die Challenges erlauben Ihnen somit, Ihre Fortschritte zu testen.

3.3 Einstellungen

Mit einem Klick auf **Einstellungen** gelangen Sie in fünf Registerkarten, in denen Sie verschiedene Einstellungen tätigen können.

3.3.1 Bildschirm

Wenn Sie CeeBot das erste Mal ausführen, hat der Bildschirm eine Auflösung von 640 x 480 x 16. Die meisten Computer vertragen auch eine bessere Auflösung:

Driver:

Wählen Sie wenn vorhanden einen Driver **HAL** (Hardware Abstraction Layer). Benutzen Sie Driver mit der Bezeichnung **Emulation** oder **T&L** nur in zweiter Wahl.

Auflösung:

Die ersten zwei Zahlen geben die Breite und Höhe des Bildschirms in Pixel an. Die dritte Zahl gibt die Anzahl anzeigbarer Farben an (16 für 65'000 Farben, 32 für 4 Millionen

Farben). Je größer die Auflösung ist, desto ästhetischer wird das Spiel, desto langsamer (ruckartiger) kann es aber auch werden. Beginnen Sie mit 640 x 480 x 16. Mit den meisten modernen Grafikkarten können Sie ohne weiteres bis 1024 x 768 x 16 oder noch höher gehen. Der beste Kompromiss hängt jedoch von jedem Computer ab und muss im Einzelfalle durch Probieren herausgefunden werden.

Vollbildschirm

Normalerweise nimmt COLOBOT den ganzen Bildschirm ein, egal bei welcher Auflösung. Wenn diese Option nicht angekreuzt ist, läuft COLOBOT in einem fixen Fenster mit 640 x 480 Pixel. So können Sie leicht zwischen verschiedenen Windows-Programmen hin- und herwechseln.

Änderungen ausführen

Klicken Sie auf diese Schaltfläche, damit die Änderungen in Kraft treten.

3.3.2 Grafik

Wenn die Darstellung am Bildschirm ruckelt, können Sie einige Optionen hier deaktivieren, wodurch allerdings die Ästhetik ein bisschen leiden kann.

Anzahl Partikel (0% bis 200%)

Mit den Partikeln werden Staub, Rauch, Splitter, Luftblasen usw. dargestellt.

Sichtweite (50% bis 200%)

Die Sichtweite ist die maximale Sichtdistanz, bevor alles im Nebel verschwindet. Mit einem großen Wert (z.B. 200%) können Sie weit sehen, benötigen aber eine gute Grafikkarte.

Details (0% bis 200%)

Wenn ein Objekt weit entfernt ist, wird es mit weniger Details dargestellt. Ein großer Wert hier schiebt diese Distanz weiter hinaus.

Anzahl Ziergegenstände (0% bis 100%)

Hier wird die Anzahl Ziergegenstände eingestellt, wie Pflanzen, Bäume, Kristalle, usw.

4 Der Bildschirm

Während einer Übung sieht der Bildschirm folgendermaßen aus:

1 Übersicht über die Bauten oder Roboter
2 Armaturenbrett für das ausgewählte Objekt
3 Die Minikarte
4 Zugang zum Menü
5 Kurzmeldungen

4.1 Die Übersicht

Oben am Bildschirm erscheint eine Übersicht über die verfügbaren Roboter und Bauten. Das ausgewählte Objekt erscheint orange. Wenn ein Roboter gerade ein Programm ausführt, blinkt der Rahmen.

Um ein Objekt auszuwählen, klicken Sie mit der Maus darauf, oder drücken Sie auf **Tab** um das nächste Objekt auszuwählen.

Mit der ersten Schaltfläche ganz links können Sie zwischen der Übersicht der Bauten und der Übersicht Roboter hin- und herwechseln.

4.2 Das Armaturenbrett

Am unteren Teil des Bildschirms erscheinen alle Informationen und Handlungen bezüglich des ausgewählten Objekts.

4.2.1 Die zehn Programme

Unten links erscheinen die jedem Roboter eigenen 10 Programm-Slots. Der vierte Slot enthält einen Lösungsvorschlag. Mit der Bildlaufleiste gelangen Sie zu den Slots 5-10.

Ausführen oder Anhalten des ausgewählten Programms.

Mit dieser Schaltfläche gelangen Sie in den Programmeditor (siehe Abschnitt 6.1). Beim Bearbeiten eines Programms wird das Spiel angehalten, so dass Sie in Ruhe arbeiten können.

4.2.2 Der *Sat*Com

Der ***Sat*Com** ist ein armbanduhrähnliches Gerät des Astronauten. Es stellt die Verbindung zur Einsatzzentrale her und zeigt alle nötigen Informationen über die Übung an (siehe Abschnitt 5).

Klicken Sie auf das Symbol **H** oder drücken Sie **F1** um die Anweisungen für die Übung anzuzeigen.

Klicken Sie auf **[ ]** um auf dem ***Sat*Com** Informationen über das angewählte Objekt zu erhalten.

4.2.3 Die Kamera

Mit dieser Schaltfläche können Sie den Standpunkt der Kamera ändern. Meistens wechselt sie zwischen einer nachfolgenden Kamera und der Bordkamera hin und her. Die Leertaste hat die gleiche Wirkung.

4.2.4 Zurücksetzen

Mit dieser Schaltfläche können Sie alle Objekte wieder an ihren Ausgangspunkt zurücksetzen. Schon geschriebene Programme werden nicht verändert. Benutzen Sie diese Schaltfläche bevor Sie Ihr Programm neu ausführen.

4.2.5 Zustandsanzeigen

Diese Anzeigen erlauben es Ihnen, auf einen Blick den Zustand des Roboters zu kennen. Links ist der Ladezustand der Batterie, in der Mitte der Grad der Beschädigung, rechts (nur bei fliegenden Robotern) die Triebwerkstemperatur.

4.3 Die Minikarte

Rechts unten am Bildschirm erscheint eine kleine Karte. Je dunkler die Zone, desto tiefer das Gelände. Roboter werden gelb, Bauten blau und Feinde grün angezeigt. Feinde sind auf der Minikarte nur sichtbar, wenn ein Radar in Betrieb ist.

Wenn der Mauszeiger auf einem Element ruht, wird das entsprechende Symbol und der Name angezeigt. Mit einem einfachen Klick können Sie es dann auswählen. Mit dem Schieberegler links können Sie den Maßstab der Karte einstellen.

4.4 Das Menü

Durch einen Klick auf das kleine Kreuz rechts oben wird folgendes Menü angezeigt:

Weitermachen	Schließt das Menü, und die Übung geht weiter.
Einstellungen	Einstellungen für das Spiel. Bestimmte Einstellungen sind hier nicht verfügbar. Um zu allen Einstellungen Zugang zu haben, verlassen Sie die aktuelle Mission, und klicken Sie auf **Einstellungen** im Hauptmenü (siehe Abschnitt 3.3).
Neu anfangen	Bricht die aktuelle Mission ab, und fängt sie von vorne wieder an.
Abbrechen	Bricht die Mission endgültig ab, und zeigt das Hauptmenü an.

5 Der *Sat*Com

Der ***Sat*Com** ist ein armbanduhrähnliches Instrument, das alle Informationen über die Übung enthält. Es hilft Ihnen auch mit der Programmiersprache weiter.

Es funktioniert ähnlich wie ein Internetbrowser. Wenn ein Wort unterstrichen ist, können Sie darauf klicken, um zur entsprechende Seite zu gelangen.

Vorhergehende Seite

Nächste Seite

Zurück zur Seite die beim Aktivieren des ***Sat*Coms** angezeigt wurde.

Kopiert die ausgewählten Zeichen in die Zwischenablage. Sie können so Programme in den Editor kopieren (siehe Abschnitt 6.1).

Zeigt die Anweisungen für die Übung. Mit **F1** gelangen Sie direkt auf diese Seite.

Zeigt die allgemeine Hilfe über die Programmiersprache CBOT. Hier finden Sie eine genaue Beschreibung jeder Anweisung. Mit **F2** gelangen Sie direkt hierher.

Schaltet den ***Sat*Com** aus.

Wenn eines der obigen Symbole grau erscheint, kann die entsprechende Schaltfläche nicht benutzt werden.

6 Programmieren

Alle Roboter sind programmierbar. In einigen Übungen ist jedoch einer der Roboter schon programmiert und kann nicht angewählt werden.

6.1 Programmeditor

Um in den Programmeditor zu gelangen gehen Sie wie folgt vor:

- Wählen Sie einen Roboter an.
- Wählen Sie unten links eines der verfügbaren 10 Programm-Slots aus.
- Klicken Sie auf die Schaltfläche **{..} Gewähltes Programm bearbeiten**.

Während der Bearbeitung eines Programms wird das Spiel automatisch angehalten. Im Hintergrund haben Sie jedoch weiterhin eine Übersicht über das Gelände. Die entsprechende Kamera können Sie jederzeit schwenken und nach vorne/hinten bewegen, indem Sie mit der Maus den Rand des Fensters berühren. Sie können das Fenster des Programmeditors wie jedes normale Windowsfenster anpassen:

 **Reduzieren**, reduziert das Fenster zu einer Titelleiste.

 **Großes Fenster**, lässt das Fenster den gesamten Bildschirm einnehmen.

 **Schließen**, entspricht der Schaltfläche **OK**.

Wenn Sie den Programmeditor aufrufen, während ein Programm ausgeführt wird, wird das Spiel nicht angehalten. Sie können so die Ausführung des Programms und den Inhalt der Variablen beobachten (siehe Abschnitt 6.1.11).

Das Fenster des Editors kann durch klicken mit der Maus auf die Titelleiste verschoben werden. Sie können auch die Größe des Fensters verändern, indem Sie mit der Maus die Ecken oder die Ränder verschieben. Diese Einstellungen werden gespeichert.

Folgende Schlüsselwörter der CBOT-Sprache erscheinen mit farbigem Hintergrund, um die Kontrolle der Syntax (Rechtschreibung) zu erleichtern:

Farbe	***Typ***	***Beispiel***
Orange	Anweisungen	`aim`, `fire`, `turn`, `goto`, `while`, usw.
Grün	Variabletypen	`object`, `float`, `int`, usw.
Rot	Konstanten, Kategorien	`TitaniumOre`, `Converter`, `BlackBox`, usw.

Wenn sich der Cursor in einem Schlüsselwort der CBOT-Sprache befindet, erscheint in der Statusleiste unter dem Editorfenster eine kurze Angabe der Syntax. Mit einem Klick auf diese Statusleiste oder mit der Taste **F3** können Sie den ***Sat*Com** mit mehr Informationen über das entsprechende Schlüsselwort aufrufen.

Um ein ganzes Wort auszuwählen, benutzen Sie einen Doppelklick. Wie in jedem Windows-Editor können Sie mehrere Zeichen auswählen, indem Sie die Maustaste gedrückt halten. All diese Handlungen können auch mit den Pfeiltasten durchgeführt werden. Benutzen Sie **Shift+Pfeiltasten** um Zeichen auszuwählen, **Ctrl+Pfeiltasten** um wortweise vorzurücken, und **Ctrl+Shift+Pfeiltasten** um ganze Wörter auszuwählen.

6.1.1 Neu

Löscht das Programm im Editor und ersetzt es durch das Skelett eines neuen Programms:

```
extern void object::Go( )
{

}
```

`Go` ist der Name des Programms, Sie können ihn durch einen Namen Ihrer Wahl ersetzen. Der Name eines Programms darf jedoch keine Leerstellen enthalten. Benutzen Sie am besten Namen, bei denen die ersten Buchstaben der Wörter groß, der Rest aber klein geschrieben ist, wie z.B. `Suche`, `SuchTitan`, `ZurückZurBasis`, usw.

Wenn Sie unabsichtlich auf diese Schaltfläche geklickt haben, können Sie diese Handlung mit der Schaltfläche **Widerrufen** rückgängig machen.

6.1.2 Öffnen und Speichern

Alle von Ihnen erstellte Programme werden automatisch für jede Übung oder Challenge gespeichert. Wenn Sie hingegen ein Programm in einer anderen Mission benutzen wollen, speichern Sie es auf der Festplatte, und lesen Sie es in einer anderen Übung ein.

Privat Das Programm wird in einem privaten Ordner des jeweiligen Spielers gespeichert.

Öffentlich Das Programm wird in einem allen Spielern zugänglichen Ordner gespeichert.

Der Name des Ordners erscheint unter der Titelleiste des Dialogs. Der Pfad ist jedoch relativ zum Ordner, in dem CeeBot-A installiert wurde. **Savegame\Spieler\Program** heißt z.B., dass der Pfad **C:\Program Files\CeeBot-A\Savegame\Spieler\Program** lautet (je nach dem Ordner, in dem das CeeBot-A installiert wurde).

Tastaturkürzel: **Ctrl+O** bzw. **Ctrl+S**.

6.1.3 Widerrufen

Widerruft die letzte Änderung. Sie können die 20 letzten Änderungen widerrufen.

Tastaturkürzel: **Ctrl+Z**.

6.1.4 Ausschneiden, Kopieren und Einfügen

Mit diesen Befehlen können Sie die ausgewählten Zeichen ausschneiden, in die Zwischenablage kopieren und anschließend z.B. in ein anderes Programm einfügen.

Tastaturkürzel: **Ctrl+X**, **Ctrl+C** bzw. **Ctrl+V**.

6.1.5 Zeichengröße

Mit diesem Schieberegler können Sie die Zeichengröße im Programmeditor und im **SatCom** einstellen. Die Einstellung wird gespeichert.

6.1.6 SatCom: Anweisungen

Hier erhalten Sie die Anweisungen für die Mission oder Übung. In diesen Text gelangen Sie auch mit der Taste **F1**.

6.1.7 SatCom: Hilfe über Programmieren

Hier finden Sie mit verschiedenen Hypertextlinks Hilfe über die Programmiersprache **CBOT**. In diesen Text gelangen Sie auch mit der Taste **F2**.

6.1.8 OK

Kompiliert das Programm und schließt den Editor. Wenn beim Kompilieren ein Fehler auftritt, wird die entsprechende Stelle ausgewählt, und eine Fehlermeldung erscheint in der Statusleiste.

6.1.9 Abbrechen

Schließt den Editor, ohne zu kompilieren. Die Änderungen werden jedoch gespeichert.

6.1.10 Kompilieren

Kompiliert das Programm, ohne den Editor zu schließen. So können Sie zwischendurch nachprüfen, ob das Programm Fehler enthält.

Ein Programm kann nur dann laufen wenn es korrekt kompiliert worden ist. Ein korrekt kompiliertes Programm ist jedoch keine Garantie dass das Programm auch wirklich korrekt funktioniert.

6.1.11 Start/Stop

Startet bzw. stoppt die Ausführung des Programms, ohne den Editor zu schließen. So können Sie das Programm *debuggen*, d.h. auf Fehler untersuchen. Ein dunkles Rechteck zeigt an, welche Anweisung gerade ausgeführt wird, und in der unteren Hälfte des Fensters können Sie in Echtzeit sehen, wie sich der Inhalt der Variablen verändert.

6.1.12 Pause/Weitermachen

Wechselt zwischen dem Schritt-für-Schritt-Modus und dem normalen Modus hin und her.

6.1.13 Ein Schritt

Führt im Schritt-für-Schritt-Modus die nächste Anweisung aus. Im unteren Teil des Fensters wird dabei jedes Mal der Inhalt der Variablen angezeigt.

6.2 Die Programmiersprache CBOT

Die CBOT-Sprache ähnelt in Syntax und Struktur den Standardsprachen C++ und Java? . Sie wurde für die Benutzung in CeeBot und für ein leichtes Erlernen des Programmierens angepasst. Im vorliegenden Handbuch wird nur eine kurze Einführung in die einfachsten Anweisungen geboten. Für genauere Angaben benutzen Sie bitte den **SatCom** mit der Taste **F2** (Hilfe über Programmieren).

Achten Sie immer auf Groß- und Kleinschreibung. Z.B. `Radar` ist nicht gleich `radar`.

Jede Anweisung wird mit einem Strichpunkt **;** abgeschlossen.

Kommentare beginnen mit // und enden am Ende der Zeile.

6.2.1 Die Anweisung `radar`

Mit der Anweisung `radar();` können Sie mit dem bordeigenen Radar nach bestimmten Objekten wie feindlichen Wesen, Robotern, Gebäuden oder Rohstoffen suchen. Schreiben Sie in Klammern den Namen des Objekts, das Sie suchen. Geben Sie das Ergebnis in eine Variable vom Typ `object`. Hier ist ein Beispiel, das nach der nächsten Ameise sucht:

```
                              // Am Anfang des Programms:
object  item;                 // Deklaration der Variable
item = radar(AlienAnt); // sucht die nächste Ameise
```

6.2.2 Die Anweisung `goto`

Mit der Anweisung `goto();` wird dem Roboter der Befehl erteilt, eine bestimmte Position zu erreichen. Die häufigste Anwendung besteht darin, den Roboter anzuweisen, eine mit `radar();` gefundene Position zu erreichen. Wenn die von der Anweisung `radar();` zurückgegebenen Informationen in einer Variable gespeichert wurden, schreiben Sie den Namen der Variable gefolgt von `.position`, um die Position des Objekts zu erhalten. Hier ist ein Beispiel eines Programm, das einen Titanwürfel sucht, die entsprechende Position erreicht, und den Würfel ergreift:

```
object item;
item = radar(Titanium);
goto(item.position);
grab();
```

6.2.3 Die Anweisungen `grab` und `drop`

Der Befehl `grab();` weist den Roboter an, mit seinem Greifer einen Gegenstand aufzunehmen, sei es vom Boden, von der Basisplattform von verschiedenen Bauten oder von der Batterieladefläche eines Roboters. Die Anweisung `drop();` legt den transportierten Gegenstand wieder ab. Hier ist ein kurzes Programm, das einen Gegenstand vor dem Roboter ergreift und 5 Meter weiter wieder ablegt:

```
grab();   // Greift den Gegenstand
move(5);  // Fährt 5 m weiter
drop();   // Legt den Gegenstand ab
```

6.2.4 Die Anweisung `fire`

Die Anweisung `fire();` feuert die Kanone des Roboters ab. Meistens wird diese Anweisung benutzt, um Salven von einer Sekunde abzufeuern:

```
fire(1);
```

6.2.5 Die Anweisung `while`

Die Anweisung `while () {}` wird benutzt, um eine Gruppe von Anweisungen mehrmals zu wiederholen. Die häufigste Anweisung von `while` besteht darin, dass eine Gruppe von Anweisungen unendlich lang wiederholt wird. Für eine unendliche Schleife schreiben Sie `while (true) {}` und setzen Sie die zu wiederholenden Anweisungen zwischen die geschweiften Klammern `{}`. Hier ist als Beispiel ein Programm, das unendlich oft eine Spinne sucht, sich zu ihr dreht und schießt:

```
while (true)
{
        item = radar(AlienSpider);
        turn(direction(item.position));
        fire(1);
}
```

Sie müssen dieses Programm nur gerade einmal ausführen, und keine Spinne rundherum in Reichweite wird überleben.

6.2.6 Die Anweisung `distance`

Mit der Anweisung `distance( , )` können Sie die Distanz zwischen zwei Positionen berechnen. Wenn Sie `position` alleine schreiben erhalten Sie die Position des Roboters, der das Programm ausführt. Wenn Sie den Namen einer Variable gefolgt von `.position` schreiben, erhalten Sie die Position des Objekts, das von der Variable beschrieben ist. Hier ist ein Beispiel eines Programms, das den Roboter genau um die Distanz vorwärts bewegt, die den Roboter von der nächsten Ameise trennt:

```
item = radar(AlienAnt);
move(distance(position, item.position));
```

Dies wäre natürlich reiner Selbstmord. Es ist besser, 40 Meter vor der Ameise zu stoppen, um in Schussweite zu sein:

```
item = radar(AlienAnt);
move(distance(position, item.position) - 40);
```

6.2.7 Die Anweisung `if`

Mit den Anweisungen `if` und `else` können Sie erreichen, dass bestimmte Anweisungen nur dann ausgeführt werden wenn eine Bedingung wahr ist. Schreiben Sie die Bedingung in Klammern `()`, und die betroffenen Anweisungen in geschweiften Klammern `{}`. Hier ist ein einfaches Beispiel: Der Roboter wird nur schießen, wenn das Ziel näher als 40 Meter ist:

```
item = radar(AlienAnt);
if (distance(position, item.position) < 40)
{
        fire(1);
}
```

Sie können auch testen, ob es ein bestimmtes Objekt überhaupt in Reichweite ist. Wenn die Anweisung `radar();` den Gegenstand nicht findet, gibt sie den Wert `null` zurück. Mit `(item == null)` testen Sie, ob es den Gegenstand nicht gibt, mit `(item != null)` ob es ihn gibt. "==" bedeutet "ist gleich", "`!=`" bedeutet "ist nicht gleich". Hier ist ein Programm das die Batterie nur auflädt, wenn es ein Kraftwerk gibt:

```
item = radar(PowerStation);
if (item != null)
{
        goto(item.position);
        wait(5);
}
```

6.2.8 Die Anweisung `motor`

Mit `motor( , );` können Sie die Geschwindigkeit des rechten und des linken Motors des Roboters getrennt einstellen. Die so eingestellte Geschwindigkeit bleibt auch während den folgenden Anweisungen erhalten. So ist es möglich, den Roboter während dem Abfeuern der Kanone zu drehen, um eine ganze Zone abzudecken:

```
turn(45);          // Drehung 45 Grad links
motor(0.5, -0.5); // langsame Drehung nach rechts
fire(2);           // Feuer
motor(0, 0);       // Stoppt die Drehung
```

6.2.9 Die Anweisung `turn`

Der Befehl `turn();` weist den Roboter an, eine Drehung um die eigene Achse um einen bestimmten Winkel durchzuführen. Ein Drehung um 90 Grad ist eine Vierteldrehung, 180 Grad sind eine halbe Drehung. Ein positiver Winkel bedeutet eine Drehung im Gegenuhrzeigersinn, ein negativer Winkel eine Drehung im Uhrzeigersinn. Hier sind einige Beispiele mit `turn();`:

```
turn(90);   // Vierteldrehung links
turn(-90);  // Vierteldrehung rechts (negativer Winkel)
turn(180);  // Kehrtwendung
```

Um einen Roboter in die Richtung eines mit `radar();` gefundenen Objekts zu drehen, muss der Winkel mit der Anweisung `direction()` berechnet werden:

```
item = radar(AlienSpider);
turn(direction(item.position));
```

Feuern Sie die Kanone nach diesen Zeilen ab, und schon gibt es eine Spinne weniger.

7 Entwicklungsteam

Daniel Roux, Denis Dumoulin, Otto Kölbl, Michael Walz

EPSITEC SA, Mouette 5, CH-1092 Belmont
epsitec@epsitec.ch
www.epsitec.com