

# ATEO-SYSTEM

## KOMPONENTE: SAMS

### VERHALTENSSPEZIFIKATION

<b>Version</b>	<b>2.0 / 2.2<sup>1</sup></b>
<b>Datum</b>	<b>20.10.2009</b>
<b>Autoren<sup>2</sup></b>	Prof. Dr. Klaus Bothe Michael Hildebrandt Nicolas Niestroj

---

<sup>1</sup> Dieses Dokument ist hauptsächlich auf die SAMs Version 2.0 zugeschnitten. Ein Großteil der Aussagen kann jedoch auch für die Version 2.2 übernommen werden.

<sup>2</sup> Dieses Dokument basiert auf den studentischen Lösungen aus dem Projektseminar „Mensch-Technik-Interaktion in Echtzeit“ aus dem Wintersemester 2007/2008.

**1 INHALT**

2	Zweck und Aufbau des Dokuments.....	4
3	Zielbestimmung.....	5
3.1	Musskriterien.....	5
3.2	Wunschkriterien .....	5
3.3	Abgrenzungskriterien .....	5
4	Produkteinsatz .....	6
4.1	Anwendungsbereiche .....	6
4.1.1	Forschung .....	6
4.1.2	Lehre .....	7
4.1.3	Art des Einsatzes .....	7
4.2	Zielgruppen.....	9
5	Beschreibung der Anwendungsfälle.....	10
5.1	Anwendungsfälle im Überblick.....	10
5.2	Tracking durchführen .....	10
5.2.1	Überblick .....	10
5.2.2	Abfrage und Verarbeitung der Steuereingaben .....	11
5.2.3	Erkannte Fehler und Probleme im realen System.....	13
5.3	Tracking überwachen und führen.....	14
5.3.1	Überblick .....	14
5.3.2	Zugriff auf die Simulationsdaten .....	14
5.3.3	Eingriffe .....	14
5.3.4	Protokollierung von Fehlern des Operators .....	16
5.4	Versuch konfigurieren .....	16
5.4.1	Überblick .....	16
5.4.2	Funktionale Beschreibung.....	16
5.4.3	Beschreibung der Daten.....	17
5.5	Versuch steuern.....	18
5.6	Versuchsbeschreibung entwerfen und einlesen .....	19

5.6.1	Überblick .....	19
5.6.2	Versuchs- und Streckenkonfiguration .....	20
5.6.3	Zuweisung der Strecken zu den Versuchsschritten .....	22
5.6.4	Fehler und Probleme im realen System .....	23
5.7	Versuchsdaten protokollieren .....	23
5.7.1	Überblick .....	23
5.7.2	Funktionale Beschreibung .....	23
5.7.3	Beschreibung der Daten .....	24
5.7.4	Fehler und Probleme im realen System .....	28
6	Nichtfunktionale Anforderungen .....	28
6.1	Zuverlässigkeit / Effizienz .....	28
6.2	Bedienbarkeit .....	28
6.3	Wartbarkeit .....	29
6.4	Variabilität .....	29
7	Technische Systemumgebung .....	30
7.1	Referenzsystem .....	30
7.1.1	Hardware .....	30
7.1.2	Software .....	30
7.2	Squeak .....	30
8	Literatur .....	31
9	Glossar .....	32

## 2 ZWECK UND AUFBAU DES DOKUMENTS

Dieses Dokument beschreibt in verbaler Form die Anforderungen an die Software SAMs. Die Struktur dieses Dokuments orientiert sich an der in [1] vorgestellten Struktur für ein Pflichtenheft.

Hierbei wird in *Abschnitt 3* zunächst eine Zielbestimmung durchgeführt, welche die Muss-, Wunsch- und Abgrenzungskriterien von SAMs erläutert. Daraufhin erfolgt in *Abschnitt 4* die Darstellung des Produkteinsatzes, welche die Anwendungsbereiche und die jeweiligen Zielgruppen vorstellt. In *Abschnitt 5* werden die Anwendungsfälle in der Übersicht und auch im Detail beschrieben. Danach erfolgt eine Darstellung der nichtfunktionalen Anforderungen, welche an SAMs gestellt werden (*Abschnitt 6*). Die technische Versuchsumgebung wird dann in *Abschnitt 7* vorgestellt. In *Abschnitt 8* wird die in diesem Dokument verwendete Literatur angegeben. Der Glossar, welcher die Bedeutung der hier verwendeten Begriffe erläutert, ist im letzten Abschnitt angegeben.

### 3 ZIELBESTIMMUNG

#### 3.1 MUSSKRITERIEN

Die Software SAMs soll die Anwender in die Lage versetzen, rechnergestützte Versuche durchzuführen, um die Funktionsteilung zwischen Mensch und Maschine untersuchen zu können. Insbesondere sollen auch Untersuchungen zur Arbeitsteilung zwischen Entwicklern und Operateuren möglich sein (siehe hierzu [2],[3] und [4]).

Es lassen sich die folgenden Anforderungen formulieren:

- Durchführung einer Tracking-Aufgabe innerhalb einer Straßenlandschaft durch ein bis zwei Versuchspersonen.
- Die Protokollierung der relevanten Versuchsdaten während Versuchsdurchführung.
- Die Konfiguration einzelner Versuchsparameter vor der Versuchsdurchführung durch den Versuchsleiter.
- Die Steuerung des Ablaufs der Versuchsdurchführung durch den Versuchsleiter.
- Die Ermöglichung der Überwachung und Führung des Trackings durch eine weitere Versuchsperson oder durch ein System.
- Ein Versuch soll mit seinen einzelnen Schritten und Strecken durch externe Text- und Grafikdateien beschrieben werden können. Diese Dateien sollen eingelesen und die entsprechende Strecken sowie der entsprechende Versuchsablauf durch SAMs dargestellt bzw. ausgeführt werden.

#### 3.2 WUNSCHKRITERIEN

Es wurden alle gewünschten Kriterien umgesetzt.

#### 3.3 ABGRENZUNGSKRITERIEN

Die folgenden Kriterien sollen von SAMs *nicht* umgesetzt werden:

- Die Auswertung des während der Versuchsdurchführung erstellten Logfiles.
- Die Funktionalitäten der Automaten und Assistenzsysteme.
- Die Funktionalität eines umfangreichen Operateursarbeitsplatzes.

## 4 PRODUKTEINSATZ

### 4.1 ANWENDUNGSBEREICHE

#### 4.1.1 FORSCHUNG

Das Produkt wird im Rahmen des Projektes **Arbeitsteilung Entwickler-Operateur** (ATEO) entwickelt und eingesetzt, welches sich mit der asynchronen Arbeitsteilung zwischen Entwicklern und Designern sowie Operateuren und Anwendern eines Mensch-Maschine-Systems beschäftigt (siehe [2]). Erstere planen, konzipieren und implementieren dieses System, während letztere das implementierte System schließlich benutzen.

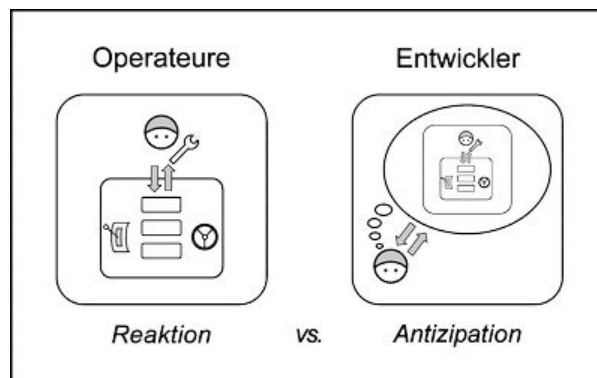


Abbildung 1 - Die Arbeitsteilung zwischen Entwicklern und Operateuren (siehe [2]).

Die Erkenntnisgewinnung im ATEO-Projekt erfolgt durch Experimente, welche rechnergestützt durchgeführt werden. Im Zentrum steht hierbei das Konzept der **Socially Augmented Microworld** (SAM) (siehe hierzu [3],[4]). Hierbei wird in SAM ein dynamischer Prozess in Form einer Tracking-Aufgabe modelliert, welcher zu einem gewissen Grad unvorhersehbar ist, aber dennoch in der Retrospektive erklärt werden kann. Dies wird dadurch erreicht, dass SAM einen sozialen Faktor enthält, nämlich die **Mikroweltbewohner** (MWB). Bei den MWB handelt es sich um ein bis zwei Versuchspersonen, welche die Aufgabe haben ein Tracking-Objekt auf einer vorgegebenen Strecke entlang zu steuern (vergleiche Abbildung 2).

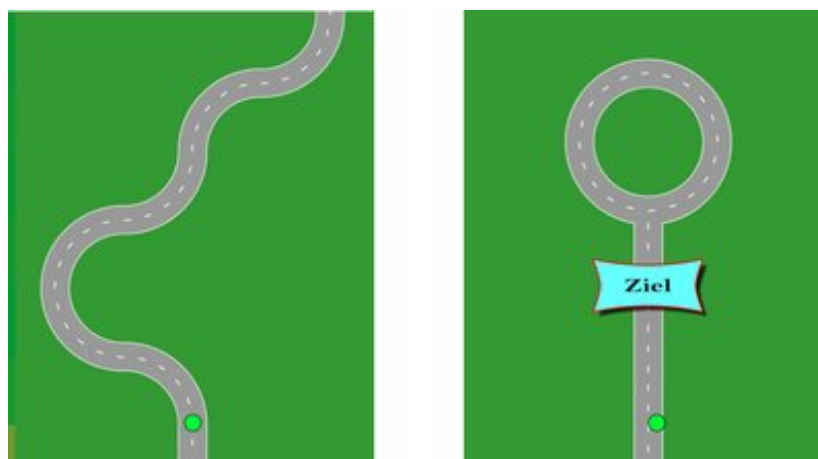


Abbildung 2 - Die Tracking-Aufgabe von SAM (siehe [2]).

Dies kann kooperativ oder nur durch einen der MWB durchgeführt werden. Ziel dieser Tracking-Aufgabe ist es, das Tracking-Objekt möglichst genau und dabei so schnell wie möglich durch den Parcours zu manövrieren. Eine weitere Versuchsperson wird als Operateur tätig, welcher das Tracking der MWB überwacht und Eingriffe in den Steuerungsprozess tätigt, falls dies nötig sein sollte. Neben dem menschlichen Operateur können auch

**Automatiken** (AM) für die Prozessüberwachung und -führung eingesetzt werden. Der Operateur bzw. die Automatik jedoch gehören selbst nicht mehr zu SAM. *Abbildung 3* verdeutlicht diesen Zusammenhang grafisch.

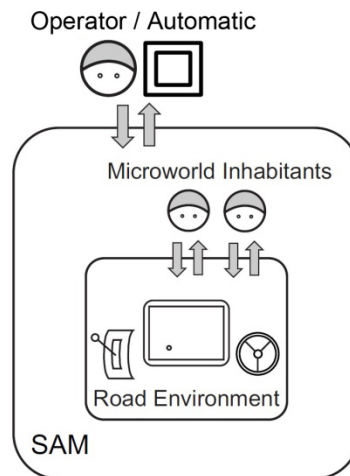


Abbildung 3 - Der Zusammenhang von SAM, den MWB sowie dem Operateur bzw. den Automatiken (nach [2]).

Dieses Konzept von SAM wurde u.a. in Smalltalk umgesetzt. Die entsprechende Software wird als **Socially Augmented Microworld in Smalltalk** (SAMs) bezeichnet und ist Gegenstand dieses Dokuments.

#### 4.1.2 LEHRE

Dieses Produkt wird zudem als Fallstudie für die Entwicklung einer Software in einem interdisziplinären Team herangezogen werden, welche es den Studenten ermöglicht, Einblicke in die Vorgehensweisen und Probleme einer solchen Software-Entwicklung zu gewinnen. Darüberhinaus können die Studenten durch die praktische Anwendung von softwaretechnischen Methoden ihr Wissen bezüglich des Software Engineerings vertiefen und festigen.

#### 4.1.3 ART DES EINSATZES

##### SAM UND DAS ATEO-SYSTEM

Die Komponente SAM ist hierbei ein Bestandteil des ATEO-Systems, welches zudem aus den Komponenten **Operateursarbeitsplatz** (OAP) und **Automatiken** (AM) besteht (siehe *Abbildung 4*).

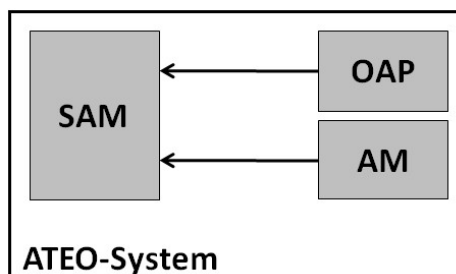


Abbildung 4 - Das ATEO-System mit seinen Komponenten SAM, OAP und AM, wobei die Pfeile die Zugriffsrichtung angeben.

Neben der bereits erwähnten Simulation des Tracking-Prozesses ermöglicht es SAM den anderen beiden Komponenten, harte und weiche Eingriffe (siehe *Abschnitt 5.3*) in den Prozess vorzunehmen und zudem auf einige Daten der Simulation zuzugreifen. Zudem werden sämtliche Simulationsdaten während der Simulation in eine Protokolldatei geschrieben, so dass diese in der Retrospektive ausgewertet werden können.

Der OAP ermöglicht es dem Operateur komfortabel und übersichtlich den Tracking-Prozess zu beobachten und zudem Eingriffe zu tätigen, falls dies nötig sein sollte. Für die Überwachung des Tracking-Prozesses kommen eine Streckenansicht einschließlich des Tracking-Objekts, ein Videobild der MWB und eine Darstellung der Joystickeingaben zum Einsatz. Darüberhinaus werden Leistungsdaten wie die Genauigkeit und die Geschwindigkeit des Trackings integriert am Tracking-Objekt in Form eines Schweißes dargestellt. Der Operateur hat zudem die Möglichkeit auditive wie visuelle Hinweise an die MWB auszugeben oder gar in den Steuerungsprozess einzugreifen, indem er die maximale Geschwindigkeit festlegt, das Lenken in eine bestimmte Richtung blockiert oder die Steuergewaltaufteilung zwischen den MWB manipuliert. Weitere Details zum OAP sind in [5] nachzulesen.

Die AM besitzen hierbei die gleiche Aufgabe wie der Operateur, nämlich das Tracking zu überwachen und zu führen. Allerdings werden sie von den Entwicklern des Systems entworfen und implementiert. Die Qualität der AM und deren Wirkung auf die Prozessüberwachung und -führung stellt ein Leistungsmaß hinsichtlich der Arbeit der Entwickler dar. Somit ist es möglich die Leistungen des Operateurs mit den Leistungen der Entwickler zu vergleichen. Mit diesem Vergleich ist es dann möglich, zu untersuchen, ob eine bestimmte Aufgabe durch einen Operateur bewältigt werden sollte oder ob es günstiger ist, für diese Aufgabe ein AM durch die Entwickler konzipieren und implementieren zu lassen. Darüberhinaus wird so eine Optimierung der jeweiligen Leistung in der Prozessüberwachung und -führung möglich. Die Details zu den AM werden in [6] ausführlich beschrieben.

## VERSUCHSAUFBAU

Der Versuchsaufbau ist in *Abbildung 5* zu sehen. Der Aufbau erstreckt sich über zwei Räume, was durch einen senkrechten Strich angedeutet wird. In einem Raum befinden sich die MWB, welche die Tracking-Aufgabe mit der SAM-Komponente, ein bis zwei Joysticks, einem 19" Monitor und dem Rechner ATEO 1 durchführen. Die Maus wird lediglich vom Versuchsleiter für die Versuchsvorbereitung benutzt. Darüberhinaus wird das Verhalten der MWB mit einer Webcam an den OAP übertragen. Dieser befindet sich in einem separaten Raum und besteht aus dem Rechner ATEO 2, einem 30" Monitor und der Komponente OAP. Neben den MWB ist auch der Operateur eine Versuchsperson, deren Verhalten untersucht wird. Daher wird auch dieses durch eine Webcam aufgezeichnet. Aus Gründen der Performance wird hierfür ein weiterer Rechner benutzt (ATEO 3). Für die Bedienung des OAP stehen dem Operateur eine Maus und eine Tastatur zur Verfügung.

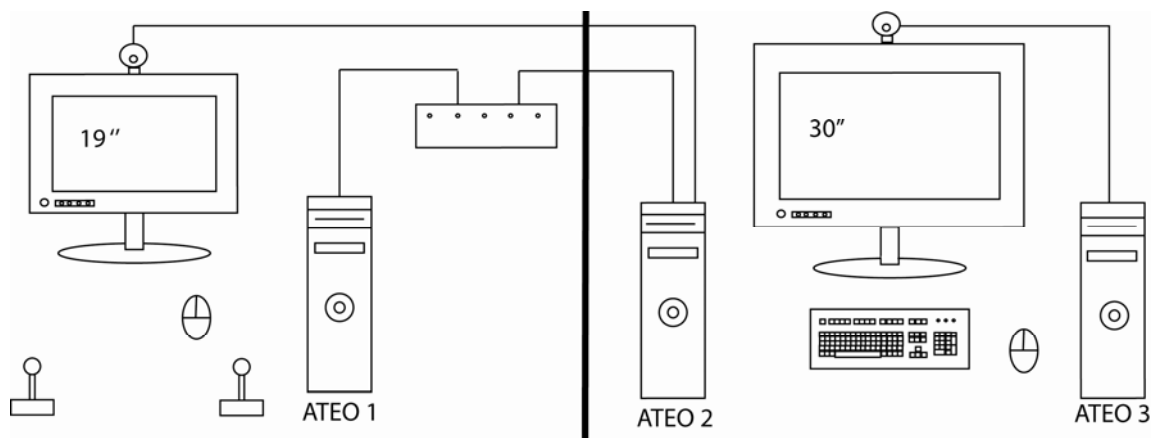


Abbildung 5 - Versuchsaufbau der psychologischen Untersuchungen.

## VERSUCHSABLAUF IM ÜBERBLICK

Der Versuchsablauf kann in drei Phasen aufgeteilt werden. Die erste Phase ist die Versuchsplanung und -vorbereitung durch den Versuchsleiter. In dieser Phase werden mehrere Vorbereitungen getroffen. Es werden die Haupt- und Nebenaufgaben, welche von den Versuchspersonen auszuführen sind ermittelt. Zudem werden die Versuchsgruppen und -teams gebildet (siehe *Abschnitt 5.4.3*). Dies geschieht unter der Verwendung von



Fragebögen, welche von den Versuchspersonen ausgefüllt werden. Hierbei findet die Bildung der Versuchsgruppen nach einer Beschreibung von bestimmten Merkmalen wie Handlungsstrategie, Expertise oder ähnlichem statt. Weiterhin gilt es den Versuchsablauf zu planen. Ein Versuch besteht aus mehreren Schritten, wobei für jeden einzelnen Schritt eine Strecke erstellt wird und zudem festgelegt werden muss, welcher MWB das Steuern übernimmt und ob es sich um ein kooperatives oder Einzel-Tracking handelt (vgl. *Abschnitt 5.6*).

In der zweiten Phase werden vor dem Beginn des Experiments durch den Versuchsleiter noch einige Parameter und versuchsrelevante Daten (siehe *Abschnitt 5.4*) festgelegt bzw. in SAMs eingegeben. Diese Daten erscheinen dann später auch in der jeweiligen Protokolldatei und dienen der Zuordnung der während der Versuchsdurchführung ermittelten Daten zu den einzelnen Versuchspersonen. Für die Eingabe dieser Daten steht ein einfaches Menü zur Verfügung, welches zudem den Start des Versuchs ermöglicht.

Die Durchführung des Versuchs ist die dritte Phase, in welcher nun die Probanden aktiv werden. Hierzu gehören einerseits die MWB, welche das Tracking-Objekt in mehreren Versuchsschritten kooperativ oder aber einzeln auf verschiedenen Strecken entlang steuern (vgl. *Abschnitt 5.2*) und andererseits der Operateur, welcher das Tracking überwacht und ggf. unterstützend eingreift. Der Operateur kann hierbei weiche Eingriffe oder harte Eingriffe vornehmen. Bei ersteren handelt es sich um das Geben von auditiven und visuellen Hinweisen und bei letzteren um Eingriffe in die Steuerung des Tracking-Objekts, wie z.B. die Manipulation der möglichen maximalen Geschwindigkeit (vgl. *Abschnitt 5.3*).

Der Versuch endet, wenn alle für diesen Versuch vorgesehenen Strecken bzw. Versuchsschritte durch die MWB absolviert wurden. Für den Versuchsleiter liegen nun für jeden einzelnen Schritt des Versuchs Protokolldateien mit den entsprechenden Simulationsdaten vor.

## 4.2 ZIELGRUPPEN

Für SAMs gibt es drei verschiedene Nutzergruppen: die MWB, den Operateur und den Versuchsleiter.

Während des Versuchs wird das Tracking-Objekt durch die MWB unter der Verwendung von Joysticks über die verschiedenen Strecken gesteuert. Die geforderten Fähigkeiten im Umgang mit den Joysticks hängen hierbei von der aktuellen Fragestellung des Versuchs ab und werden in Fragebögen und vorbereitenden Tests überprüft.

Der Operateur benutzt SAMs nur mittelbar, indem er das Tracking unter Verwendung des OAP überwacht und führt. Dennoch stellt er auch eine Zielgruppe dar, da die Möglichkeiten des OAP bezüglich der Überwachung und des Eingreifens davon abhängig sind, inwiefern SAMs die entsprechenden Daten bereitstellt bzw. die entsprechenden Eingriffe ermöglicht (vgl. *Abschnitt 5.3*). Für die Bedienung des OAP werden die Operateure vor dem Experiment noch einem Training unterzogen, welches in die Bedienung einführt.

Der Versuchsleiter sollte mit dem System ausreichend vertraut sein und gute PC-Kenntnisse besitzen, um auch vorbereitende Aufgaben auszuführen. Dazu gehören neben dem Wissen über den Programmablauf und die benötigten Tastenkombinationen auch Kenntnisse zu den Parametern im Versuchsleiter-Menü sowie zum Aufbau der Strecken und den damit verbundenen Dateien.

## 5 BESCHREIBUNG DER ANWENDUNGSFÄLLE

### 5.1 ANWENDUNGSFÄLLE IM ÜBERBLICK

Die Anwendungsfälle von SAMs sind in *Abbildung 6* in einem Anwendungsfalldiagramm dargestellt. In einem solchen Diagramm werden die Anwendungsfälle als Ovale dargestellt, Akteure als Strichmännchen und das zu modellierende System als ein Rechteck. Eine Linie zwischen einem Anwendungsfall und einem Akteur gibt an, dass es zu einer Interaktion kommt. Es kommt oft vor, dass verschiedene Anwendungsfälle die gleichen Teilanwendungsfälle besitzen. Um diese nicht redundant beschreiben zu müssen, werden sie als eigenständige Anwendungsfälle modelliert und mit den entsprechenden Anwendungsfällen über eine «include»-Beziehung verknüpft. Die Darstellung dieser Beziehung erfolgt als gestrichelter Pfeil, wobei die Spitze des Pfeils zum eingeschlossenen Anwendungsfall zeigt (vergleiche [1]).

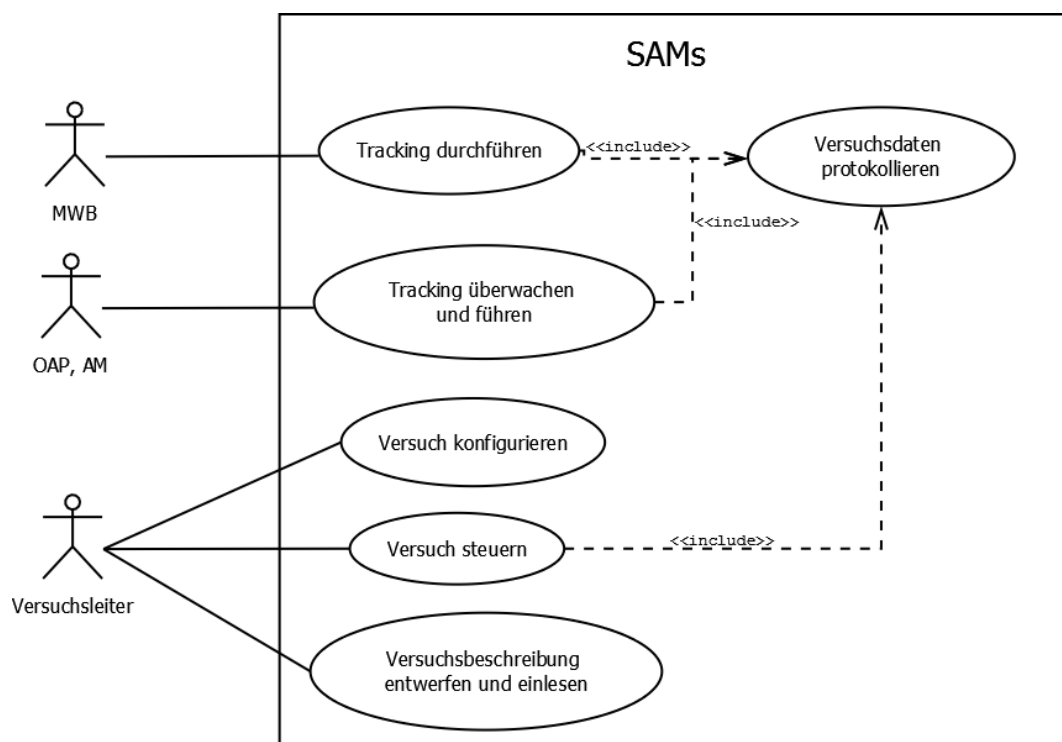


Abbildung 6 - Die Anwendungsfälle von SAMs.

Insgesamt sind sechs Anwendungsfälle in *Abbildung 6* zu sehen, welche in den nächsten Abschnitten im Detail beschrieben werden. Es gibt drei Arten von Akteuren. Zum einen die MWB, welche das Tracking durchführen, die Software-Komponenten OAP und AM, welche das Tracking überwachen und führen sowie den Versuchsleiter, welcher den Versuch entwirft, konfiguriert und steuert.

### 5.2 TRACKING DURCHFÜHREN

#### 5.2.1 ÜBERBLICK

Der Anwendungsfall *Tracking durchführen* beschreibt die Benutzung von SAMs durch die MWB. Diese führen eine Tracking-Aufgabe aus, bei der es darum geht, ein Tracking-Objekt auf einer Spur entlang zu steuern und auf Hindernisse und Eingriffe des Operators zu reagieren, insofern diese auftreten. Das Tracking beginnt automatisch, sobald Anfangsinstruktionen ausgewählt und quittiert wurden und ein Countdown abgelaufen ist. Das Tracking kann kooperativ durchgeführt werden, so dass jeder der MWB einen gewissen Einfluss auf die Steuerung ausüben kann. Initial besitzen beide MWB 50% der Steuergewalt, dieses Verhältnis kann jedoch

durch den Operator beeinflusst werden (siehe *Abschnitt 5.3*). Die MWB betrachten dabei das Geschehen aus der Vogelperspektive (vgl. *Abbildung 2*).

Damit die MWB ein Feedback bezüglich der aktuellen Geschwindigkeit erhalten, wurde in das Tracking-Objekt eine Geschwindigkeitsanzeige integriert. Diese wurde als Kreisdiagramm implementiert, welche gegen den Uhrzeigersinn zunimmt, wenn die Geschwindigkeit erhöht wird und mit dem Uhrzeigersinn abnimmt, wenn die Geschwindigkeit abnimmt. Bei maximaler Geschwindigkeit ist das Kreisdiagramm komplett schwarz, bei minimaler Geschwindigkeit komplett grün.

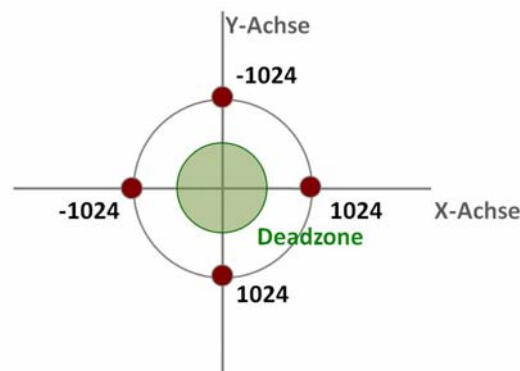
Eine Strecke wurde absolviert, wenn das Tracking-Objekt die farbige Zielmarkierung auf der Strecke überfahren oder (falls diese verfehlt wurde) das Ende der Strecke erreicht hat. Nach Abschluss einer Strecke erscheinen Versuchshinweise auf dem Bildschirm, die von den MWB durch Klicken mit der Maus bestätigt werden müssen.

### 5.2.2 ABFRAGE UND VERARBEITUNG DER STEUEREINGABEN

Für diese Steuerung stehen zwei Möglichkeiten zur Verfügung. Einerseits können Joysticks für die Steuerung benutzt werden und andererseits die Tastatur.

#### JOYSTICKSTEUERUNG

Wird der Joystick als Eingabegerät verwendet, so führen die Auslenkungen auf der horizontalen Achse des Joysticks zu einer horizontalen Bewegung des Fahrzeuges auf dem Bildschirm. Wird der Joystick auf der vertikalen Achse nach vorn bewegt, so wird die Geschwindigkeit des Tracking-Objekts erhöht, zieht man den Joystick zurück, so wird die Geschwindigkeit des Tracking-Objekts reduziert. Die vom Joystick abgefragten Werte liegen innerhalb des Intervalls  $[-1024, 1024]$ . *Abbildung 7* verdeutlicht diesen Sachverhalt grafisch.



*Abbildung 7* - Der Wertebereich eines Joysticks. Wird der Joystick nach links oder vorn ausgelenkt, so ist der Wertebereich  $[-1024, 0]$ , wird er hingegen nach rechts oder hinten ausgelenkt, so ist der Wertebereich  $[0, 1024]$ .

Weiterhin existiert eine sogenannte *Deadzone*. Diese definiert den minimalen Ausschlag, welcher überschritten werden muss, damit der Joystick einen Wert von ungleich 0 liefert. Auf diese Weise wird sichergestellt, dass in der Ruhelage des Joysticks immer 0 vorherrscht. Standardmäßig liegt die Deadzone im Bereich  $[-30, 30]$ .

Die Ermittlung der Werte der aktuellen Auslenkungen des Joysticks erfolgt durch ein *Polling*, also durch ein periodisches Abfragen der Input-Werte des Joysticks.

#### TASTATURSTEUERUNG

Weiterhin wurde eine Tracking-Objekt-Steuerung per Tastatur integriert. Diese eignet sich nicht für reale Anwendungen im Experiment, da die Tastatureingaben zunächst vom Betriebssystem abgefragt werden und somit von der internen Einstellung von Wiederholfrequenz und Wiederholverzögerung abhängig sind, was sich insbe-

sondere beim permanenten Drücken einer Taste negativ bemerkbar macht. Diese Steuerungsart dient allerdings dem Testen des Programms.

Die Belegung der Tasten für die Steuerung des Fahrzeugs per Tastatur ist in *Tabelle 1* dargestellt.

**Tabelle 1 - Tastaturbelegung für die Steuerung des Tracking-Objekts.**

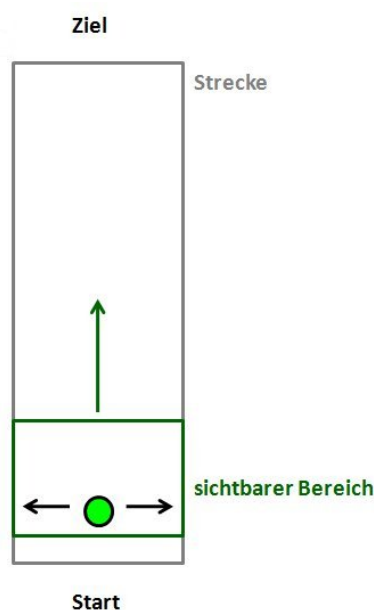
Taste	Wirkung
ö	Tracking-Objekt nach links bewegen
#	Tracking-Objekt nach rechts verschieben
ü	Geschwindigkeit des Tracking-Objekt erhöhen
ä	Geschwindigkeit des Tracking-Objekt verringern

Die Tastatursteuerung ist naturgemäß eine ereignisgesteuerte Steuerung, da das Betätigen einer Taste ein Ereignis erzeugt, welches von der jeweiligen Anwendung verarbeitet werden kann. Somit können die Eingaben der Tastatur nicht direkt periodisch abgefragt werden.

Es sei darauf hingewiesen, dass die Steuerung mittels Tastatur nur zu Testzwecken implementiert wurde.

## BEWEGUNGSMODELL

Der befahrbare Raum während der Tracking-Simulation wird in *Abbildung 8* dargestellt. Das graue Rechteck bezeichnet die gesamte Strecke, welche während eines Versuchsschrittes absolviert wird, vom Start bis zum Ziel. Der auf dem Bildschirm sichtbare Bereich ist als grünes Rechteck dargestellt. In diesem befindet sich das Tracking-Objekt und kann sich horizontal bis an dessen Ränder bewegen. Die vertikale Position des Tracking-Objekts auf dem Bildschirm ist jedoch fest, jedoch ändert sich diese relativ zur Strecke. Dies liegt darin begründet, dass während des Trackings der sichtbare Bereich über die Gesamtstrecke geschoben wird, um so den Eindruck einer Fahrt zu erzeugen (vgl. *Abschnitt 5.6*).



**Abbildung 8 - Der befahrbare Bereich von SAMs.**

Die Eingaben von den Steuereingaben der MWB bewirken eine Änderung der Position des Tracking-Objekts relativ zur Gesamtstrecke. Im Weiteren wird davon ausgegangen, dass  $n \in \mathbb{N}$  an einem Versuch teilnehmen, wobei  $\mathbb{N}$  die Menge der natürlichen Zahlen bezeichnet.<sup>3</sup>

Die Änderung der horizontalen Position des Tracking-Objekts  $xpos_{TO}^A$  wird wie folgt berechnet:

$$xpos_{TO}^A = \frac{\sum_{i=1}^n input_i^x \cdot alloc_i}{50 \cdot scale_x},$$

wobei  $input_i^x$  das aktuelle Input von MWB  $i$ ,  $alloc_i$  die prozentuale Steuergewaltszuweisung für MWB  $i$  und  $scale_x$  den Dämpfungsfaktor<sup>4</sup> des horizontalen Steuerinputs darstellen. Die neue horizontale Position des Tracking-Objekts  $xpos_{TO}^{neu}$  ergibt sich dann aus der Summe der alten X-Position  $xpos_{TO}^{alt}$  sowie der ermittelten Änderung  $xpos_{TO}^A$ :

$$xpos_{TO}^{neu} = xpos_{TO}^{alt} + xpos_{TO}^A.$$

Nahezu die gleiche Formel wird für die vertikale Positionsänderung des Tracking-Objekts benutzt. Hier gibt es eine kleine Besonderheit zu beachten. Die Y-Koordinate der Tracking-Objekt-Position ist zwar relativ zum sichtbaren Bereich (also auf dem Bildschirm) konstant, relativ zur Gesamtstrecke verändert sich jedoch diese Y-Koordinate. Dies liegt daran, dass die Animation der Strecke durch die Verschiebung der Spielbretter erreicht wird. Weitere Details werden in *Abschnitt 5.6* dargestellt. Es wird die folgende Formel verwandt:

$$ypos_{TO}^A = \frac{v_{max}}{100} \cdot \frac{\sum_{i=1}^n (1024 - input_i^y) \cdot alloc_i}{50 \cdot scale_y},$$

wobei  $ypos_{TO}^A$  die Anzahl der Pixel darstellt, um welche sich das Tracking-Objekt relativ zur Gesamtstrecke vorwärts bewegt hat.  $v_{max} \in [0,120]$  ist die prozentuale maximale erreichbare Geschwindigkeit<sup>5</sup> des Tracking-Objekts,  $input_i^y$  ist die vertikale Auslenkung des Steuergerätes von MWB  $i$ ,  $alloc_i$  ist die prozentuale Steuergewaltszuweisung für MWB  $i$  und  $scale_y$  stellt einen Dämpfungsfaktor des vertikalen Steuerinputs dar. Die Details zum Streckenaufbau sind *Abschnitt 5.6* zu entnehmen.

Aus den obigen Formeln lassen sich zwei Dinge ableiten. Zum ersten bewegt sich das Tracking-Objekt mit der mittleren Geschwindigkeit vorwärts, wenn sich das Steuergerät in der Ruheposition befindet, also keine Eingabe erfolgt und zum zweiten ist kein Rückwärtsfahren möglich, da bei maximaler rückwertiger Auslenkung der Wert  $input_i^y = 1024$  erreicht wird (siehe *Abbildung 7*) und somit  $ypos_{TO}^A = 0$  gilt.

### 5.2.3 ERKANNT FEHLER UND PROBLEME IM REALEN SYSTEM

Auf Grund der provisorischen Implementation der Tastatursteuerung ergeben sich Inkonsistenzen zur normalen Steuerung per Joystick. So können Daten wie die momentane Auslenkung nicht ermittelt werden. Weiterhin wird die Geschwindigkeit bei Erreichen des Ziels auch gleich als Startgeschwindigkeit bei dem nächsten Durchlauf festgelegt. Da diese Steuerungsmöglichkeit jedoch nur zu Testzwecken entwickelt wurde, beeinflussen diese Fehler nicht die normale Anwendung.

<sup>3</sup> Gegenwärtig nehmen immer 1-2 MWB am Versuch teil, es gilt also  $n = 1,2$ .

<sup>4</sup> Dieser Faktor hat die Funktion das Steuerinput zu dämpfen, damit diese nicht zu empfindlich auf die Handbewegungen der MWB reagieren.

<sup>5</sup> Unter Geschwindigkeit wird hierbei die Anzahl der Pixel verstanden, welche pro Simulationsschritt zurückgelegt werden. Siehe hierzu *Abschnitt 5.6*.

## 5.3 TRACKING ÜBERWACHEN UND FÜHREN

### 5.3.1 ÜBERBLICK

Bei diesem Anwendungsfall handelt es sich bei den Akteuren um den OAP und die AM – also um weitere Software-Systeme. SAMs stellt hierbei einige Simulationsdaten bereit und ermöglicht es, weiche und harte Eingriffe in den Tracking-Prozess zu tätigen. Es sei angemerkt, dass in SAMs 2.0 lediglich die Vernetzung mit dem OAP implementiert und getestet wurde. Würden die AM derart angepasst, dass sie das gleiche Interface besäßen wie der OAP, so wäre auch hier eine Vernetzung möglich. Weitere Details zu den anderen beiden Software-Komponenten und zur Vernetzung sind in [5], [6] und [7] zu finden. Der Entwurf des OAP nach H. Schwarz ist in *Abbildung 9* zu sehen.

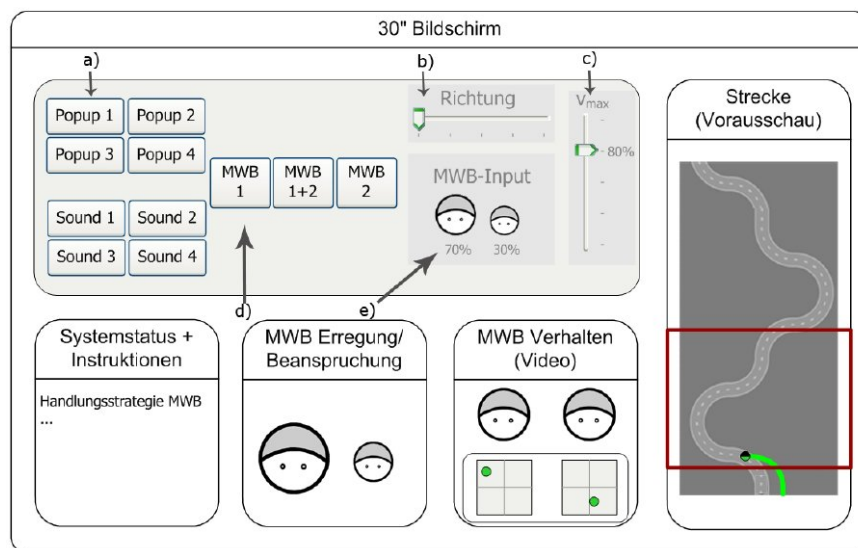


Abbildung 9 - Operateursarbeitsplatz nach H. Schwarz.

### 5.3.2 ZUGRIFF AUF DIE SIMULATIONS DATEN

SAMs 2.0 stellt den anderen Komponenten einige Daten über TCP und UDP Sockets bereits, welche in jedem Simulationsschritt in die Sockets geschrieben werden. Bei diesen Daten handelt es sich um die X-Koordinate des Tracking-Objekts und die Änderung der Y-Koordinate relativ zur Gesamtstrecke (siehe *Abschnitt 5.2*). Darüberhinaus wird dem OAP mitgeteilt, wann die Streckenkonfiguration und die dazugehörigen Grafiken (siehe *Abschnitt 5.6*) zu laden sind, wann das Tracking beginnt und welcher MWB gerade aktiv ist.

### 5.3.3 EINGRIFFE

Von SAMs werden zwei Arten von Eingriffen unterstützt: weiche und harte Eingriffe. Unter ersteren versteht man das Geben von Hinweisen, welche auditiver oder visueller Natur sein können. Letztere sind Eingriffe in den Tracking-Prozess wie z.B. das Manipulieren der Steuergewaltaufteilung zwischen MWB oder der maximal erreichbaren Geschwindigkeit.

Die möglichen visuellen Hinweise sind in *Abbildung 10* dargestellt. Zudem können auditive Hinweise<sup>6</sup> gegeben werden, wobei zunächst zu wählen ist, ob nur MWB1, nur MWB2 oder aber beide MWB den Hinweis erhalten sollen. Dies wird dadurch erreicht, dass nur der linke, nur der rechte oder aber beide Stereokanäle benutzt werden. *Tabelle 2* listet die möglichen auditiven Hinweise auf.

<sup>6</sup> In SAMs 2.0 können zwar nur 4 der 8 möglichen auditiven Hinweise gegeben werden, wobei SAMs 2.0 als auch der OAP leicht erweitert werden können, um diesem Umstand zu beseitigen.

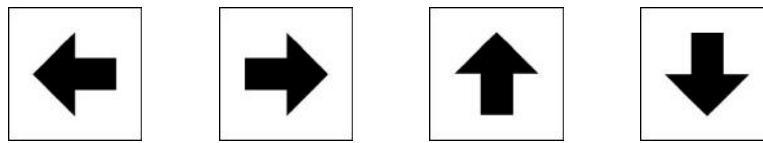


Abbildung 10 - Mögliche visuelle Hinweise (v.l.n.r.): "Links halten", "Rechts halten", "Geschwindigkeit erhöhen" und "Geschwindigkeit verringern".

Insgesamt stehen vier Arten von harten Eingriffen zur Verfügung. Es kann die maximal erreichbare Geschwindigkeit des Tracking-Objekts manipuliert werden, wobei Werte von 0% bis 120% möglich sind. Dann ist es möglich das Lenken nach links oder rechts blockiert werden, so dass das Lenken in die blockierte Richtung effektiv auf die Position des Tracking-Objekts ist. Weiterhin ist es möglich, die Steuergewaltaufteilung in 25% zwischen den MWB zu verändern, so dass das Steuerinput des einen MWB stärker gewichtet wird als das Steuerinput des anderen. Hierbei ist auch möglich, einem MWB 100% und dem anderen 0% der Steuergewalt zuzuweisen.

Tabelle 2 - Mögliche auditive Hinweise.

Sound	Hinweis
Sound1.mp3	Ihr Partner wird jetzt die Führung übernehmen!
Sound2.mp3	Ihr Partner lenkt zu heftig! Bitte gleichen Sie das aus!
Sound3.mp3	Ihr Partner fährt etwas schnell! Bitte gleichen Sie das aus!
Sound4.mp3	Bitte fahren Sie etwas genauer!
Sound5.mp3	Bitte übernehmen Sie die Führung!
Sound6.mp3	Ihr Partner lenkt zu schwach! Bitte gleichen Sie das aus!
Sound7.mp3	Ihr Partner fährt etwas langsam. Bitte gleichen Sie das aus!
Sound8.mp3	Bitte fahren Sie etwas schneller!

Außer dem Setzen der maximal erreichbaren Geschwindigkeit werden die gewünschten Eingriffe über Codes vom OAP bei SAMs angefordert, welche in *Tabelle 4* dargestellt werden.

Tabelle 3 - Codes für das Auslösen von weichen und harten Eingriffen.

Code	Bedeutung
1	Joystickblockade links
2	Keine Joystickblockade
3	Joystickblockade rechts
4	Rechter Stereokanal gewählt
5	Linker Stereokanal gewählt
6	Beide Stereokanäle gewählt
7	Auditiver Hinweis 1
8	Auditiver Hinweis 2
9	Auditiver Hinweis 3
10	Auditiver Hinweis 4
20	Steuergewalt MWB1: 0%, MWB2: 100%
21	Steuergewalt MWB1: 25%, MWB2: 75%
22	Steuergewalt MWB1: 50%, MWB2: 50%
23	Steuergewalt MWB1: 75%, MWB2: 25%
24	Steuergewalt MWB1: 100%, MWB2: 0%
41	Visueller Hinweis "Links halten"
42	Visueller Hinweis "Rechts halten"
43	Visueller Hinweis "Geschwindigkeit erhöhen"
44	Visueller Hinweis "Geschwindigkeit verringern"

### 5.3.4 PROTOKOLLIERUNG VON FEHLERN DES OPERATEURS

Etwaige Versuche des Operateurs, trotz bereits aktiver Einblendungen oder Soundausgaben, in das Geschehen einzugreifen, werden protokolliert, indem der eigentlich zu protokollierenden Zahl eine '0' angehängt wird. So bedeutet '70' in der Spalte „Sound“, dass gerade versucht wurde Sound7 abzuspielen, obwohl momentan ein anderer präsentiert wird.

Audioeinspielungen können nur durchgeführt werden, indem direkt vor dem Drücken der Zahl für die gewählte Datei, eine Taste zum Auswählen der Versuchspersonen gedrückt wird. Eine etwaige Fehlbedienung durch Drücken der Taste zum Einspielen ohne vorherige Versuchspersonenauswahl, wird in der Protokolldatei mit einer angefügten '5' vermerkt. Die Protokolleinträge werden durch kleine Zahlen dargestellt, um eine postoperative Auswertung durch Werkzeuge zu vereinfachen.

## 5.4 VERSUCH KONFIGURIEREN

### 5.4.1 ÜBERBLICK

Die Versuchsvorbereitung besteht zum einen aus der Konfiguration der versuchsrelevanten Daten und zum anderen aus der Vergabe der Ziele, welche die einzelnen MWB während des Versuchs verfolgen sollen. Für ersteres steht hierbei ein Interface zur Verfügung, welches im *Abschnitt 5.4.3* näher erläutert wird. Die Vorbereitung wird allein vom Versuchsleiter übernommen.

Mögliche Ziele für die Versuchspersonen können u.a. folgende sein:

- beide Versuchspersonen sollen den Parcours so schnell wie möglich durchfahren
- beide Versuchspersonen sollen den Parcours möglichst ohne Verlassen der Straße durchfahren – Zeit spielt dabei keine Rolle
- eine Versuchsperson soll die Geschwindigkeit regulieren und die andere übernimmt die Lenkung

Die jeweiligen Ziele hängen vom Versuchsdurchlauf ab und der Kreativität des Versuchsleiters sind hier ansonsten keine Grenzen gesetzt. Es sei noch angemerkt, dass den MWB verschiedene Ziele zugewiesen werden können.

Sind alle Daten konfiguriert und die Ziele mit den Versuchspersonen besprochen so wird nach Drücken des Weiter-Buttons in den Instruktionsfenstern die Fahraufgabe gestartet. Anschließend erscheint der Countdown auf dem Bildschirm, nach dessen Ablauf die Fahrt erst beginnt.

### 5.4.2 FUNKTIONALE BESCHREIBUNG

Dem Versuchsleiter steht zu Beginn des Versuches ein Menü zur Verfügung, in dem verschiedene Einstellungen getätigt werden können. *Abbildung 11* zeigt den Aufbau des Konfigurationsmenüs mit dem die folgenden Einstellungen möglich sind:

1. **Eingabe des Operateurs (Element a)**  
Es wird dem Operateur, der den aktuellen Versuch überwachen soll, eine eindeutige Zahl zugeordnet.
2. **Eingabe der Versuchsgruppe (Element b)**  
Es wird eine eindeutige Zahl zu dem jeweiligen Personenkreis zugeordnet, der beim aktuellen Versuch beteiligt ist.
3. **Eingabe der Teamnummer (Element c)**  
In einer Gruppe können mehrere Teams sein, die aus jeweils zwei Versuchspersonen bestehen.
4. **Eingabe Versuchspersonennummer von MWB 1 und 2 (Element d und e)**  
Jeder Versuchsperson wird eine eindeutige Zahl zugeordnet.



5. **Eingabe der Versuchspersonennummer des beginnenden MWB (Element g)**  
Es wird angegeben, welcher der beiden Versuchspersonen mit dem Versuch beginnt.
6. **Eingabe des Geschlechts der Versuchsgruppe (Element g)**  
Es wird festgehalten, welches Geschlecht die Versuchspersonen der aktuellen Versuchsgruppe haben. Da es nur gleichgeschlechtliche Gruppenzusammensetzungen gibt, sind zwei Werte definiert: 0 für männlich und 1 für weiblich.
7. **Initialisieren der Anwendung (Element h)**  
Durch Betätigen dieses Buttons wird die Anwendung mit den eingegebenen Werten initialisiert und die Fahrstrecke geladen.
8. **Angabe des Steuerungs-Modus (Element i)**  
Auswahl zwischen Tastatur - und Joysticksteuerung.

-	+	1	= Operateur	a
-	+	9	= Versuchsgruppe	b
-	+	9	= Teamnummer	c
-	+	1	= vpMB1	d
-	+	2	= vpMB2	e
-	+	1	= startMB	f
-	+	1	= Geschlecht	g
<div> <div>h</div> <div>i</div> <div>Los</div> <div>Joystick</div> </div>				

Abbildung 11 - Konfigurationsmenü für den Versuchsleiter.

### 5.4.3 BESCHREIBUNG DER DATEN

#### VERSUCHSGRUPPENNUMMER

Jeder Versuchsgruppe wird eine natürliche Zahl zugeordnet. Diese dient in der Auswertung der Versuchsdaten der Zuordnung von bestimmten Testbedingungen zu den einzelnen Versuchsgruppen, wie z.B. der Einfluss von Musik während des Trackings.

#### TEAMNUMMER

Jedem Team wird eine natürliche Zahl zugeordnet. Nachdem die Versuchsgruppennummer die groben Testbedingungen zuordnet, gibt die Teamnummer noch weitere Bedingungen an.

Beispiel:

- 1 = Versuchspersonen verfolgen gleiches Ziel
- 2 = Versuchspersonen verfolgen verschiedenen Ziele
- 3 = Leichte Streckenzusammensetzung

### VERSUCHSPERSONENNUMMER

Jeder der MWB bekommt eine natürliche Zahl zugeteilt, um die entstandenen Logfile-Daten eindeutig dem richtigen MWB zuordnen zu können.

### STARTPERSONENNUMMER

Die Versuchspersonennummer des MWB, welcher mit dem ersten Versuchsschritt beginnt.

### STEUERUNGSMODUS

Es kann zwischen verschiedenen Eingabegeräten gewählt werden. Allerdings stehen bis jetzt nur die Eingabegeräte Joystick und Tastatur zur Verfügung, wobei letzteres nur für Testzwecke gedacht ist.

### VERSUCHSANWEISUNGEN

Zu Beginn und zum Ende eines jeden Versuchsschritts werden Hinweise bzw. Anweisungen auf dem Bildschirm angezeigt. Diese sind zum gegenwärtigen Zeitpunkt in externen Textdateien gespeichert und können außerhalb von Squeak geändert werden.

### COUNTDOWN

Vor dem Start der Fahrt wird ein Countdown angezeigt, welcher mittels vier hintereinander gezeigten JPG-Dateien erzeugt wird.

## 5.5 VERSUCH STEuern

Der Versuchsleiter hat die Möglichkeit anhand einiger Tastenkombinationen den Ablauf des Versuchs zu steuern. Ihm stehen dabei die folgenden Kombinationen zur Verfügung:

1. **Start einer Fahrt (Tastenkombination „Alt + l“)**  
Nachdem mittels Los-Button (Element f) die Anwendung initialisiert wurde, kann die Fahrt gestartet werden. Dies ist aber nicht mehr nötig, da der Versuch durch das Drücken des Weiter-Buttons im Instruktionsfenster gestartet wird.
2. **Wiederholung einer Fahrt (Tastenkombination „Alt + r“)**  
Die aktuelle Fahrt kann mittels der Tastenkombination „wiederholt“ werden, d.h. die aktuelle Fahrt startet von vorn und es wird ein neues Log-File angelegt. Die Instruktionen zu Beginn der Fahrt werden erneut eingeblendet.
3. **Vorzeitiges Beenden einer Fahrt (Tastenkombination „Alt+e“)**  
Die aktuelle Fahrt kann jederzeit beendet werden. Dies führt dazu, dass zur nächsten Fahrt (Versuchsschritt) übergegangen wird. Die Log-Datei wird hierbei bis zur letzten Position gespeichert.
4. **Beenden des gesamten Versuchsdurchlaufes (Tastenkombination „Alt+q“)**  
Der Versuch kann komplett abgebrochen werden. Hierbei gelangt man dann wieder zu dem anfänglichen Konfigurationsdialog.
5. **Neustart des Versuches (Tastenkombination „Alt+a“)**  
Der Versuch kann komplett neugestartet werden. Hierbei gelangt man wieder zur ersten Probefahrt.

## 5.6 VERSUCHSBESCHREIBUNG ENTWERFEN UND EINLESEN

### 5.6.1 ÜBERBLICK

Im Versuch haben die MWB eine Strecke (Parcours) zu absolvieren, welche bereits vor dem Versuchen konstruiert wurde. Diese Strecke wird durch die Aneinanderreihung von Streckenteilen, den sogenannten Spielbrettern, konstruiert. Diese Spielbretter liegen in Form von GIF-Grafiken vor und besitzen alle eine Breite von 800 Pixeln und einer variablen Länge.

#### SPIELBRETT

Insgesamt stehen 31 verschiedene Spielbretter zur Verfügung, die frei kombiniert werden können, diese variieren von „leicht kurvig“ über Gabelungen bis hin zu „fast zickzack“ (vgl. *Abbildung 12*, *Abbildung 13* und *Abbildung 14*). Die Strecke endet jedoch stets mit dem Ziel-Spielbrett, welches in *Abbildung 2* zu sehen ist. Damit nun die Strecke nicht einfach nach dem Ziel-Spielbrett aufhört, gibt es zwei weitere komplett grüne Spielbretter die zum Auffüllen für den Bereich hinter dem Ziel-Spielbrett benutzt werden können.

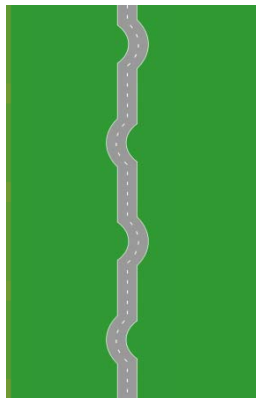


Abbildung 12 - Streckenteil "leicht kurvig".

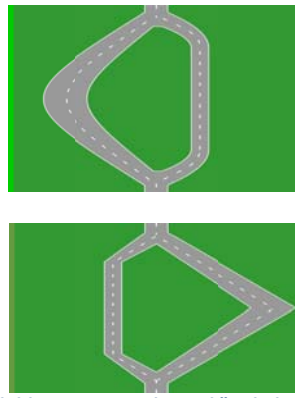


Abbildung 13 - Streckenteil "Gabelung".

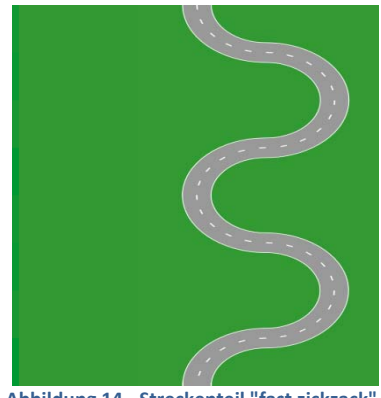


Abbildung 14 - Streckenteil "fast zickzack".

Um auch die verschiedenen Teile eines Spielbrettes unterscheiden zu können, wurden diese Informationen am linken Rand der Spielbretter in Form von verschiedenen Rot-Werten platziert. Diese Rot-Werte besagen z.B. ob es sich um eine Kurve oder aber einen geraden Spielbrettteil handelt. Eine Übersicht über alle verfügbaren Spielbretter und die jeweiligen Rot-Werte der Elemente dieser Spielbretter ist in [8] nachzulesen.

Die Versuchsstrecke entsteht durch Aneinanderreihung von Spielbrettern, was unter Verwendung einer Text-Datei geschieht. Wie sich dies genau gestaltet ist in *Abschnitt Fehler! Verweisquelle konnte nicht gefunden werden.* dargestellt.

#### HINDERNISSE

Auf der Strecke können statische und auch dynamische Hindernisse erscheinen, wobei im Moment festgelegt ist, auf welchem Spielbrett dies geschieht. Hindernisse sind demnach nur möglich auf dem Spielbrett SbQx260806.gif bzw. Bild Nummer 26.

Die statischen Hindernisse sind in einem Slalom angeordnet und unterscheiden sich in dem Grad der Straßenabdeckung, der 25% oder 50% betragen kann. Weiter kann ein statischer Slalom mit dem ersten Hindernis auf der rechten oder linken Seite beginnen. Es sind also vier Varianten eines Slaloms von statischen Hindernissen möglich.

Zusätzlich kann ein dynamisches Hindernis von links nach rechts über das Bild bewegt werden, wobei die Geschwindigkeit des Hindernisses auf das Fahrverhalten der MWBs so angepasst ist, dass ohne Änderung des

Fahrverhaltens, sprich Beschleunigen oder Bremsen, eine Kollision unausweichlich ist. In einem früheren Versuchsschritt wird dafür die Geschwindigkeit der MWBs auf dem relevanten Streckenbild, welches dreimal in einem kooperativ gefahrenen Versuchsdurchlauf erscheint, gemessen, gemittelt und gespeichert.

## STRECKENANIMATION

Die Tracking-Simulation geschieht in mehreren gleichartigen Simulationsschritten, welche wiederholt ausgeführt werden bis das Ende der Tracking-Simulation erreicht ist. Ein Simulationsschritt wird alle 30 bis 49 ms ausgeführt. In einem solchen Schritt werden die Steuereingaben der MWB abgefragt, die neue Position von Tracking-Objekt, der Strecke und der Hindernisse ausgerechnet und auf dem Bildschirm gezeichnet. Darüberhinaus werden die Eingaben durch den Operateur oder die Automaten umgesetzt. Sollte es zu Kollisionen kommen, so werden auch diese behandelt.

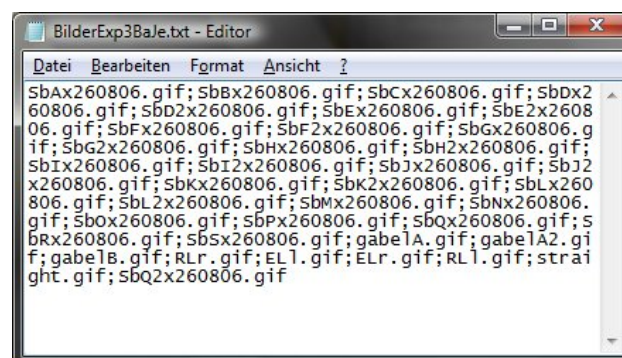
Durch das Verschieben der Spielbretter in den einzelnen Simulationsschritten entsteht der Eindruck einer Fahrt. Hierbei werden stets zwei der Grafiken der Spielbretter übereinander angeordnet. Wurde ein Spielbrett komplett über den Bildschirm geschoben, so wird das nächste Spielbrett innerhalb der Streckenkonfiguration geladen und über dem gerade sichtbaren Spielbrett platziert, also im nicht sichtbaren Bereich des Bildschirms. Besteht beispielsweise eine Strecke aus den Spielbrettern A, B, C, so wird Spielbrett A auf dem Bildschirm dargestellt und Spielbrett B darüber (im nichtsichtbaren Bereich) platziert. Während der Fahrt werden nun A und B kontinuierlich (nach unten) über den Bildschirm geschoben. Wurde Spielbrett A auf diese Weise komplett über den Bildschirm geschoben, wird Spielbrett C über dem gerade sichtbaren Spielbrett B platziert. Dies wird fortgesetzt bis keine weiteren Spielbretter in der Streckenkonfiguration vorgesehen sind.

Hierbei werden die Spielbretter um den Betrag von  $ypos_{To}^A$  (siehe *Abschnitt 5.2.2*) verschoben. Das Tracking-Objekt selber verharrt an seiner Position auf dem Bildschirm. Dennoch ändert sich die Position des Tracking-Objekts relativ zur absolvierenden Strecke.

### 5.6.2 VERSUCHS- UND STRECKENKONFIGURATION

Die Zusammenstellung der Strecke vollzieht sich in drei Schritten. Zunächst muss eine Auswahl derjenigen Spielbretter erfolgen, welche in der Strecke vorkommen sollen. Die Datei-Namen dieser Streckenteile werden dann in eine Text-Datei eingetragen, wobei jeder Eintrag durch ein Semikolon abgeschlossen werden muss (siehe *Abbildung 15*).

Die Anzahl und die Reihenfolge der Spielbretter sind frei wählbar, da die Straße auf jedem Spielbrett auf der gleichen Position der Bildschirmhorizontalen beginnt und endet, so dass die Übergänge zwischen allen Spielbrettern fließend sind.



**Abbildung 15 - Der erste Schritt der Streckenkonstruktion ist die Auswahl der zu verwendenden Streckenteile.**

Jedes Spielbrett wird dabei nur einmal aufgelistet, auch wenn es mehrmals verwendet werden soll. Die Reihenfolge der Teilstrecken in der Text-Datei ist beliebig und hat keinen Einfluss auf die Reihenfolge in der Strecke.

In einem *zweiten Konfigurationsschritt* wird in einer weiteren Text-Datei die gewünschte Reihenfolge der ausgesuchten Spielbretter in Form einer Liste notiert, deren Einträge wieder durch Semikolon getrennt werden (siehe *Abbildung 16*). Um eine Spielbrett zu identifizieren, benutzt man den Index des Spielbretts (beginnend bei 1) in der Liste der vorkommenden Spielbretter, welche in *Abbildung 15* dargestellt ist.

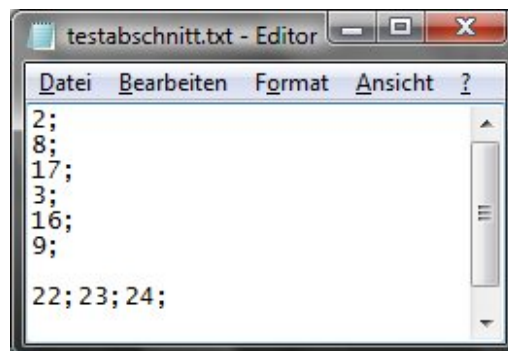


Abbildung 16 - Im zweiten Schritt wird die Reihenfolge der Spielbretter festgelegt.

Soll z. B. der Streckenabschnitt testabschnitt.txt mit der Spielbrett SbBx260806.gif beginnen, so muss der erste Eintrag in der Text-Datei „2“ lauten.

Zusätzlich können einzelne Komplexitätserweiterungen wie Hindernisse oder Inputmanipulation aktiviert werden. *Tabelle 4* listet alle möglichen Werte auf. Hierbei ist zu beachten, dass diese Erweiterungen an der Stelle im Parcours erscheinen, an der sie auch in dieser Textdatei platziert ist, und zwar immer auf dem nachfolgenden Streckenbild.

Tabelle 4 - Aktivierungscodes für Erweiterungen der Strecke.

Wert	Bedeutung
1100	Erstellung eines statischen Slaloms mit 25% Fahrbahnabdeckung und dem ersten Hindernis rechts.
1110	Erstellung eines statischen Slaloms mit 25% Fahrbahnabdeckung und dem ersten Hindernis links.
1101	Erstellung eines statischen Slaloms mit 50% Fahrbahnabdeckung und dem ersten Hindernis rechts.
1111	Erstellung eines statischen Slaloms mit 50% Fahrbahnabdeckung und dem ersten Hindernis links.
111	Erstellung eines dynamischen Hindernisses.
112	Entfernen aller Hindernisse von den entsprechenden Streckenbildern.
200	Aktivierung der adaptiven Geschwindigkeitsmessung.

Sollen im Versuch in verschiedenen Versuchsschritten auch verschiedene Strecken verwendet werden, so ist für jede dieser Strecken eine separate Datei mit der entsprechenden Reihenfolgeauflistung zu erstellen. Es gilt: gleiche Strecken in verschiedenen Versuchsteilen referenzieren auf die gleiche Reihenfolgeauflistung, verschiedene Strecken in verschiedenen Versuchsteilen referenzieren auf verschiedene Reihenfolgeauflistungen.

Im *dritten Konfigurationsschritt* wird festgelegt, welche der konstruierten Strecken in welchem Versuchsschritt abgefahren werden soll (siehe *Abbildung 17*). Das erwartete Format für die Versuchsbeschreibung wird in Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.** näher erläutert.

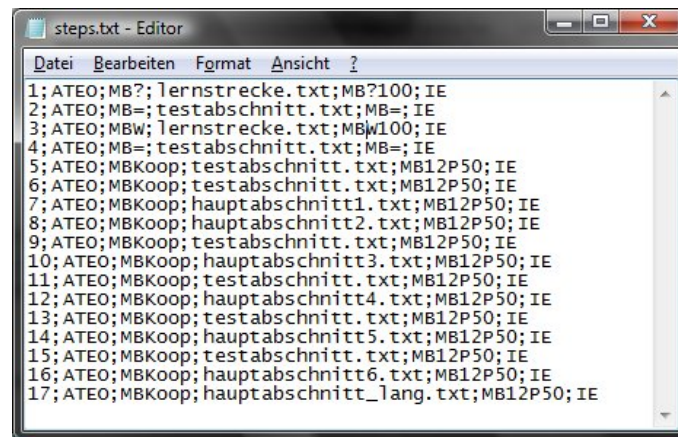


Abbildung 17 - Im dritten Schritt werden die erstellten Strecken den einzelnen Versuchsschritten zugewiesen.

### 5.6.3 ZUWEISUNG DER STRECKEN ZU DEN VERSUCHSSCHRITTEN

Nachdem die Strecken erstellt wurden, muss noch festgelegt werden, in welchem Versuchsschritt die jeweilige Strecke abgefahren werden soll. Dies passiert in der Test-Datei steps.txt. Dieser Name ist fest vorgegeben und wird auch so von dem ATEO-System referenziert. *Abbildung 17* zeigt wie eine solche Datei aussehen kann.

Ein Eintrag in die Datei steps.txt hat dabei die folgende Struktur:

*StepNr ; TrackingTyp ; Steuergewalt ; Strecke ; SteuerInput ; Instruktion*

wobei folgendes gilt:

- *StepNr*: die Nummer des Versuchsschritts.
- *TrackingTyp*: die Art des Trackings. Legacy Code, Datum wird nicht mehr vom System ausgewertet. Der Wert beträgt immer „ATEO“.
- *Steuergewalt*: bestimmt, welcher MWB die Steuergewalt besitzt. Mögliche Werte sind
  - *MB?*: Es beginnt derjenige MWB, welcher im Konfigurationsmenü als startender MWB eingetragen wurde (startMB).
  - *MB=*: Der im letzten Versuchsschritt aktive MWB steuert das Tracking-Objekt auch im aktuellen Versuchsschritt.
  - *MBW*: Der im letzten Versuchsschritt inaktive MWB steuert das Tracking-Objekt im aktuellen Versuchsschritt.
  - *MBKOOP*: Es wird kooperatives Tracking betrieben, d.h. das Tracking-Objekt wird durch beide MWB gesteuert.
- *Strecke*: der Name derjenigen Textdatei, welche den Aufbau der Strecke festlegt. Folgende Default-Strecken stehen u.a. zur Verfügung:
  - lernstrecke.txt : immer Einzelstrecke zum Erlernen der Steuerung etc.
  - testabschnitt.txt : ebenfalls kurze Strecke zum Test.
  - hauptabschnitt.txt : Beispiel für kooperatives Tracking.
- *SteuerInput*: Verteilung des Steuer-Inputs. Folgende Werte sind hier möglich:
  - *MB? 100*: der startende MWB (startMB) erhält 100% der Steuergewalt.
  - *MBW100*: Es fährt der zuvor inaktive MWB mit 100% der Steuergewalt.
  - *MB=*: Es fährt der MWB aus dem vorherigen Versuchsschritt mit derselben Steuergewalt.
  - *MB1P100*: Der erste MWB erhält 100% der Steuergewalt.
  - *MB2P100*: Der zweite MWB erhält 100% der Steuergewalt.
  - *MB12P50*: Beide MWB erhalten 50% der Steuergewalt.

- *Instruktion:* Dieser Wert ist immer auf „IE“ gesetzt. Eine Auswertung im Programm findet auch hier nicht statt. (Legacy Code)

Die zulässigen Kombinationen für *SteuerInput* und *Steuergewalt* lassen sich [10] entnehmen.

#### 5.6.4 FEHLER UND PROBLEME IM REALEN SYSTEM

Wünschenswert ist eine Performancesteigerung des Programms, da in bisherigen Versuchen Schwankungen und insbesondere Einbrüche der Geschwindigkeit zu verzeichnen waren, was sehr wahrscheinlich auf die großen, ständig zu verschiebenden Bitmaps zurückzuführen ist.

Falls die Startanweisungen zu Beginn eines Versuchsschrittes nicht mit „OK“ bestätigt werden und mit „Alt+I“ der Versuch gestartet wird, beginnt die Tracking-Aufgabe dennoch mit den Anweisungen im Vordergrund.

### 5.7 VERSUCHSDATEN PROTOKOLLIEREN

#### 5.7.1 ÜBERBLICK

Die bei einem Versuch aufgezeichneten Daten dienen zur Verhaltensauswertung der Versuchspersonen und in Zukunft auch des Operators. Der ATEO-Versuch ist in Schritte. Je Versuchsschritt wird eine Datei im XML-Format erzeugt. Diese Datei wird als Log-File bezeichnet. Die Versuchsaufzeichnung beginnt mit dem Starten der Fahrt und wird periodisch wiederholt. Die Zeitspanne zwischen zwei Messungen beträgt hierbei 30 bis 49 ms.

#### 5.7.2 FUNKTIONALE BESCHREIBUNG

Für jeden Versuchsschritt wird eine eigene Protokolldatei angelegt. Startet man eine Fahrt neu, so wird eine weitere Protokolldatei angelegt. Dies spiegelt sich auch in der Benennung der Protokolldateien wider. Wie diese genau aussieht wird weiter unten beschrieben.

Zu Beginn eines jeden Versuchsschritts wird zuerst die Kopfzeile der Protokolldatei erzeugt und dann werden die Konfiguration, die vom Versuchsleiter eingestellt wird, und die Variablen, die während der Fahrübung erzeugt werden, in der Protokolldatei vermerkt.

Es gibt fünf Daten-Quellen für die Protokolldatei: Daten von der Steuerung durch die Versuchspersonen, Daten vom Versuchsleiter, Daten von den Hinweisen des Operators, Daten von einem Unterstützungssystem, Daten von einem Messgerät für physiologischen Beanspruchungsmaße.

Hierbei wird ein kompletter Datensatz, welcher die Daten aus allen fünf Quellen enthält, alle 30 ms in die Protokolldatei geschrieben.

#### BENENNUNG DER PROTOKOLLDATEN

Der Name der Protokolldateien setzt sich wie folgt zusammen

*'log', VTNr, 'S', StepNr, 'v', D, '.xml'*

wobei

- *'log', 'S', 'v'* und *'.xml'* Zeichenketten darstellen, welche direkt in den Dateinamen eingehen. Das Prefix *'log'* deutet an, dass es sich um eine Protokolldatei handelt, *'S'* gibt an, dass die Nummer des



Versuchsschritt folgt, 'v' deutet an, dass nun die Anzahl der Durchläufe für einen Versuchsschritt folgt und, '.xml' ist das typische Suffix für XML-Dateien.

- *VTNr* ist die Nummer des aktuellen Versuchsteams.
- *StepNr* ist die Nummer des aktuellen Versuchsschritts.
- *D* gibt an, zu welchem Durchlauf eines Versuchsschritts die Protokolldatei gehört.

### 5.7.3 BESCHREIBUNG DER DATEN

Die geloggtten Daten lassen sich in die Kategorien Versuchssteuerung, Systemzustände, Ereignisse, Assistenzsysteme, physiologische Beanspruchung und Leistungsmaße einteilen. Im Folgenden beschreiben wir jede Kategorie der Daten in einem Abschnitt. Diese Darstellung erfolgt daher nicht in der Reihenfolge wie die Daten in der Protokolldatei notiert werden. Daher wird in einem weiteren Abschnitt auf die Struktur der Protokolldatei eingegangen, welche diese Reihenfolge im Detail erläutert.

#### VERSUCHSSTEUERUNG

Die Daten dieser Kategorie dienen der Beschreibung des Versuchs im Großen.

Variable	Datentyp	Definitionsbereich	Beschreibung
ExpNr	Integer	Beliebig	Nummer der Versuchsgruppe.
TeamNr	Integer	beliebig	Nummer des Teams.
SubjNr1	Integer	beliebig	Nummer des Mikroweltbewohners 1.
SubjNr 2	Integer	beliebig	Nummer des Mikroweltbewohners 2.
OperatorNo	Integer	beliebig	Nummer des Operateurs
TeamSex	Integer	[0..1]	Geschlecht der Versuchsgruppe
ShareMI1	Real	[0..1]	prozentualer Anteil an der Fahrzeugsteuerung von Mikroweltbewohner 1.
ShareMI2	Real	[0..1]	prozentualer Anteil an der Fahrzeugsteuerung von Mikroweltbewohner 2.
StepNr	Integer	beliebig	Nummer des aktuellen Versuchsschritts.
AssistanceImpl	Integer	{0, 1, 2, 3}	Nummer der implementierten Automatik.
AssistanceAct	Integer	{0, 1, 2, 3, 01, 02, 03, 12, 13, 23}	Nummer der aktiven Automatik(en) 0 := keine Automatik aktiv 1 := 1. Automatik aktiv 2 := 2. Automatik aktiv 3 := 3. Automatik aktiv 01, 02, 03, 12, 13, 23 := Kombinationen von zwei aktiven Automaten
SupervisorSetShareMI1	Real	[0..1]	vom Versuchsleiter eingestellte Inputmanipulation für MWB1
SupervisorSetShareMI2	Real	[0..1]	vom Versuchsleiter eingestellte Inputmanipulation für MWB2

#### SYSTEMZUSTÄNDE

Die Daten dieser Kategorie beschreiben Systemzustände während des Versuchs, wobei hier noch einmal zwischen Daten bezüglich der Strecke und den Daten, die sich auf das Fahrzeug beziehen unterschieden wird.

Variable	Datentyp	Definitionsbereich	Beschreibung
NamePic1	String	* + ".gif"	Name der Bilddatei des aktuellen Streckenabschnittes
PosPic1Top	Real	beliebig	Obere Position des aktuellen Spielbrettes



PosPic1Bot	Real	beliebig	Untere Position des aktuellen Spielbrettes
NamePic2	String	* + ".gif"	Name der Bilddatei des nächsten Streckenabschnittes
PosPic2Top	Real	beliebig	Obere Position des nächsten Spielbrettes
PosPic2Bot	Real	beliebig	Untere Position des nächsten Spielbrettes
ColorCode	Real	[0..255]	Rotfarbwert der aktuellen Streckenposition bei besonderen Streckenabschnitten zur genaueren Positionsbestimmung (RGB-Wert)
FirstStaticObstacleXCoordinate	Real	beliebig	X-Koordinate des ersten statischen Hindernisses
FirstStaticObstacleYCoordinate	Real	beliebig	Y-Koordinate des ersten statischen Hindernisses
SecondStaticObstacleXCoordinate	Real	beliebig	X-Koordinate des zweiten statischen Hindernisses
SecondStaticObstacleYCoordinate	Real	beliebig	Y-Koordinate des zweiten statischen Hindernisses
DynamicObstacleXCoordinate	Real	beliebig	X-Koordinate des dynamischen Hindernisses
DynamicObstacleYCoordinate	Real	beliebig	Y-Koordinate des dynamischen Hindernisses
BarrierVisMI	Integer	{0, 1}	0 := Hindernis ist nicht sichtbar für die MWBs 1 := Hindernis ist sichtbar für die MWBs
TrackEdit	String	{0, 1_RLl, 1_RLr, 1_ELl, 1_ELr, 3, 4_25lr, 4_25rl, 4_50lr, 4_50rl }	0 := alte Strecke, keine Erweiterung 1_RLl := Gabelung runde Kurve links 1_RLr := Gabelung runde Kurve rechts 1_ELl := Gabelung eckige Kurve links 1_ELr := Gabelung eckige Kurve rechts 3 := dynamisches Hindernis 4_25lr := statisches Hindernisslalom, erst links dann rechts 4_25rl := statisches Hindernisslalom, erst rechts dann links 4_50lr := statisches Hindernisslalom, erst links dann rechts 4_50rl := statisches Hindernisslalom, erst rechts dann links

Weiterhin fallen Daten an, welche sich auf das **Fahrzeug** beziehen. Diese sind in den folgenden beiden Tabellen beschrieben.

Variable	Datentyp	Definitionsbereich	Beschreibung
yDisKumul	Real	beliebig	aufsummierte zurückgelegte Strecke in vertikaler Richtung des Fahrobjektes (in Pixel)
yDistperTick	Real	beliebig	Schrittweite in vertikaler Richtung des Fahrobjektes (in Pixel)
PositionX	Real	beliebig	horizontale Position des Fahrobjektes (in Pixel)

Das Tracking-Objekt verfügt über neun Farbsensoren, welche wie folgt verteilt sind:

- Sensor 1 → in der Mitte des Tracking-Objekts
- Sensor 2 → links vom Tracking-Objekt
- Sensor 3 → oben links vom Tracking-Objekt
- Sensor 4 → über dem Tracking-Objekt
- Sensor 5 → oben rechts vom Tracking-Objekt
- Sensor 6 → rechts vom Tracking-Objekt
- Sensor 7 → unten rechts vom Tracking-Objekt
- Sensor 8 → unter dem Tracking-Objekt
- Sensor 9 → unten links vom Tracking-Objekt

Diese Sensoren fragen die Farben an diesen neun Stellen auf der Strecke ab. Diese Sensoren dienen beispielsweise dazu, die genaue Position des Tracking-Objekts (auf der Spur oder nicht) abzufragen. Die Werte von fünf dieser neun Sensoren werden in jedem Simulationsschritt in die Protokolldatei eingetragen:

Variable	Datentyp	Definitionsbereich	Beschreibung
RotValS1	Real	[0..255]	Rotwert von Sensor 1 (RGB-Wert)
RotValS2	Real	[0..255]	Rotwert von Sensor 2 (RGB-Wert)
RotValS3	Real	[0..255]	Rotwert von Sensor 3 (RGB-Wert)
RotValS4	Real	[0..255]	Rotwert von Sensor 4 (RGB-Wert)
RotValS5	Real	[0..255]	Rotwert von Sensor 5 (RGB-Wert)

## EREIGNISSE

Die Daten dieser Kategorie entstehen durch auftretende Ereignisse während des Versuchs. Quellen dieser Ereignisse sind sowohl die MWB als auch der Operateur.

Die Ereignisse der MWB sind ihr Lenkverhalten, welche Daten im genauen mitgeschrieben werden, beschreibt die folgende Tabelle:

Variable	Datentyp	Definitionsbereich	Beschreibung
LeftRightMB1	Integer	beliebig	horizontale Ausrichtung des Joysticks von Mikroweltbewohner 1
LeftRightMB2	Integer	beliebig	horizontale Ausrichtung des Joysticks von Mikroweltbewohner 2
UpDownMB1	Integer	beliebig	vertikale Ausrichtung des Joysticks von Mikroweltbewohner 1
UpDownMB2	Integer	beliebig	vertikale Ausrichtung des Joysticks von Mikroweltbewohner 2

Die Ereignisse des Operateurs hingegen resultieren aus den getätigten **Eingriffen** desselbigen.

Variable	Datentyp	Definitionsbereich	Beschreibung
Balance	Integer	{0, 1, 2}	0 := ausgewählte Versuchsperson 1 1 := ausgewählte Versuchsperson 1+2 2 := ausgewählte Versuchsperson 2; bei fehlerhafter Bedienung wird der Ziffer eine '5' angehängt
Popup	Integer	{1, 2, 3, 4}	1 := "schneller fahren" 2 := "rechts halten" 3 := "langsamer fahren" 4 := "links halten"
Sound	Integer	{1..8}	Name der abgespielten Audio-Datei ist "Sound[1..8].mp3"; bei fehlerhafter Bedienung wird der Ziffer eine '0' angehängt

PopupIntensity	Integer	{1, 2}	1 := schwarzer Pfeil (normaler Hinweis) 2 := roter Pfeil (dringlicher Hinweis)
----------------	---------	--------	---

Es ist nicht möglich während der Anzeigezeit eines Hinweises von 1,5 Sekunden diesen abubrechen oder einen weiteren Hinweis zu senden. Genauso verhält es sich mit den Audioeinspielungen. Etwaige Versuche des Operateurs dennoch in das Geschehen einzugreifen, werden protokolliert, indem der eigentlich zu protokollierenden Zahl eine '0' angehängt wird, so bedeutet '70' in der Spalte „Sound“, dass gerade versucht wurde Sound7 abzuspielen, obwohl momentan ein anderer präsentiert wird.

Audioeinspielungen können nur durchgeführt werden, indem direkt vor dem Drücken der Zahl für die gewählte Datei, eine Taste zum Auswählen der Versuchspersonen gedrückt wird (hier auch permanenter Druck möglich). Eine etwaige Fehlbehandlung (Drücken der Taste zum Einspielen ohne vorherige Versuchspersonenauswahl) wird in der Protokolldatei mit einer angefügten '5' vermerkt.

Die PopupIntensity ist im Moment nicht im System implementiert, der Protokolldatei-Eintrag stellt hier einen Platzhalter für zukünftige Erweiterungen dar.

### PHYSIOLOGISCHE BEANSPRUCHUNG

Im späteren Projektverlauf soll auch die physiologische Beanspruchung der Mikroweltbewohner gemessen und geloggt werden. Welche Methoden und Geräte dafür benutzt werden, wurde noch nicht definiert, weswegen auch der Datentyp und der Datenbereich ungewiss sind. Die beiden Einträge in der Log-Datei sind daher ohne Werte und als Platzhalter zu sehen.

Variable	Datentyp	Definitionsbereich	Beschreibung
WorkloadMI1	Undefined	DUMMY	physiologische Beanspruchung von Mikroweltbewohner 1
WorkloadMI2	Undefined	DUMMY	physiologische Beanspruchung von Mikroweltbewohner 2

### ASSISTENZSYSTEME

Die Assistenzsysteme in Form von Automaten werden im späteren Verlauf des Projektes untersucht. Aus diesem Grund werden auch die Eingriffe auf die Inputverteilung und das Steuern der Automaten geloggt. Genauso wird vermerkt, ob ein Hindernis im sichtbaren Bildschirmbereich zu sehen ist, damit evtl. Handlungsverzögerungen oder die Ignorierung des Hindernisses durch die Automaten erkannt werden kann, bzw. wie schnell gehandelt wurde.

Variable	Datentyp	Definitionsbereich	Beschreibung
AssistanceUDShare	Real	[0..1]	Anteil der Automaten am Beschleunigen und Bremsen in %
AssistanceLRShare	Real	[0..1]	Anteil der Automaten an Rechts/Links-Bewegungen in %
AssistanceSetShareMI1	Real	[0..1]	Anteil der Inputmanipulation (erhöhen oder erniedrigen) des MWB1 durch die Automaten in %
AssistanceSetShareMI2	Real	[0..1]	Anteil der Inputmanipulation (erhöhen oder erniedrigen) des MWB2 durch die Automaten in %
BarrierVisAssistance	Integer	{0, 1}	0 := Automaten hat kein Hindernis auf dem sichtbaren Bildschirmbereich 1 := Automaten hat ein Hindernis auf dem sichtbaren Bildschirmbereich

## LEISTUNGSMASSE

Die Daten der letzten Kategorie dienen als Grundlage für die Bestimmung von Leistungsdaten und sind daher auch so betitelt. Die folgende Tabelle erläutert diese Daten im Detail:

Variable	Datentyp	Definitionsbereich	Beschreibung
Ms	Integer	> 0	aktuelle relative Systemzeit (in ms)
TimeDelta	Integer	> 0	Differenz der aktuellen zur vorhergehenden Zeitmessung (in ms)
NrSensors	Integer	[0..9]	Anzahl der Sensoren, die sich auf dem Grün befinden befinden.
CollisionGrade	Integer	[0..9]	gibt an, welcher Sensor 1-9 eine Kollision gemeldet hat

### 5.7.4 FEHLER UND PROBLEME IM REALEN SYSTEM

## UNANGEMESSENE HARDWARE-ANFORDERUNGEN

Damit SAMs in der Lage ist innerhalb von 30 bis 49 ms einen Log-Eintrag zu schreiben, wird ein vergleichsweise neuer und schneller Rechner benötigt. Andernfalls schwankt das Protokoll-Intervall drastisch und reicht durchaus auch in den dreistelligen Millisekunden-Bereich. Dies steht in keinem Verhältnis zu dem theoretisch notwendigen Rechenaufwand für die Ausführung von SAMs, der deutlich geringer ausfallen sollte. Eine Optimierung an dieser Stelle ist dringend erforderlich.

## 6 NICHTFUNKTIONALE ANFORDERUNGEN

### 6.1 ZUVERLÄSSIGKEIT / EFFIZIENZ

Die mittels SAMs im Rahmen der Forschung gewonnenen Ergebnisse wären wertlos, wenn das System die Anforderungen der Zuverlässigkeit und Effizienz vernachlässigen würde. So ist es beispielsweise entscheidend, dass die in der Protokolldatei zu erfassenden Daten kontinuierlich und innerhalb des Intervalls [30,49] ms erhoben werden. Wird die vorgegebene Zeitspanne zwischen dem Abfragen von Daten lastbedingt erheblich überschritten oder ist diese sehr unregelmäßig, so kann es zu Ungenauigkeiten und Verzerrungen in der Protokolldatei kommen und zudem kann dies zu Verzögerungen während der Tracking-Simulation kommen, welche für die MWB erkennbar sind und somit einen ungewünschten Einfluss auf die MWB ausüben. Dies kann zu unbrauchbaren Versuchsdaten führen. Ebenso führen unplanmäßige Programmabbrüche zu unbrauchbaren Versuchsdaten und sind damit nach Möglichkeit zu vermeiden.

### 6.2 BEDIENBARKEIT

Da es sich bei den Probanden nicht unbedingt um PC-erfahrene Personen handelt, muss die Bedienung der Software, insbesondere die Instruktionmasken und die Steuerung der Navigationshinweise, einfach zu handhaben sein. Auch wenn die Tracking-Aufgabe keine realistische Fahrsimulation darstellt, sollte die Fahrobjektsteuerung nachvollziehbar und leicht erlernbar sein.

Auch muss dem Operateur ein möglichst bedienungsfreundliches Arbeitsumfeld geboten werden. Hierzu gehört zum einen ein übersichtliches Interface, auf dem alle relevanten Informationen, wie z.B. der Streckenverlauf und das Videobild der MWB, leicht erfasst werden können. Zum anderen, müssen die Kontrollelemente eine möglichst korrekte, eindeutige und schnelle Interaktion ermöglichen.

Nicht zuletzt soll auch der Versuchsleiter bei der Konfiguration des Versuchs von der Software unterstützt werden.

Bis jetzt werden auch nur die Eingabegeräte Joystick und Tastatur unterstützt. Hier wäre es wünschenswert eine Unterstützung für eine breitere Palette von Steuergeräten zu ermöglichen.

### 6.3 WARTBARKEIT

Das Projekt ATEO entwickelt sich ständig weiter und somit ist auch das System ATEO ständigen Änderungen ausgesetzt. Deswegen ist die Erweiterbarkeit des Systems eine ausschlaggebende Anforderung. Der Entwurf von allgemeinen Interfaces und Bibliotheken mit wieder verwendbaren Funktionen soll den Entwicklungsprozess vereinfachen.

### 6.4 VARIABILITÄT

Das Tracking ist momentan fix, d.h. die Position, Farbe, Größe, ... ist fest im Projekt verankert. Wenn man diese festen Vorgaben ebenfalls konfigurierbar macht, eröffnen sich weitere Testmöglichkeiten und Einsatzgebiete für ATEO.

## 7 TECHNISCHE SYSTEMUMGEBUNG

### 7.1 REFERENZSYSTEM

#### 7.1.1 HARDWARE

Das ATEO-System wird momentan erfolgreich auf dem folgenden System eingesetzt:

- Intel Core2 Quad Q6600 @ 2.40 Ghz (4x 2.40 Ghz)
- 4GB DDR2 800Mhz (2x2GB)
- Nvidia Gforce 9800 GTX
- Mainboard: ASUS P5N-D Nvidia Nforce 750i SLI (FSB 1333 DDR2 800)
- 200 GB Samsung HD 200HJ S-ATA-II

Die Peripherie umfasst wie bereits in *Abbildung 5* angegeben:

- 2 Bildschirme (19" und 30")
- Maus und Tastatur zur Versuchssteuerung, Operateursarbeitsplatz
- Maus für das Versuchspersonenpaar (ohne mittlere Maustaste)
- Lautsprecher zur Ausgabe von Anweisungen
- 2 Kopfhörer für das Versuchspersonenpaar
- 2 Joysticks

#### 7.1.2 SOFTWARE

SAMs läuft auf der Squeak Plattform unter Windows XP 64Bit, wobei die Portabilität auf andere Plattformen dadurch gegeben ist, dass Squeak eine virtuelle Maschine beinhaltet, welche für mehrere Plattformen entwickelt wurde.

### 7.2 SQUEAK

Squeak ist die für die ATEO-Software benutzte Entwicklungs- und Ausführungsumgebung. Es implementiert Smalltalk-80 und bietet viele vorimplementierte Klassen und Objekte zur Umsetzung zahlreicher multimedialer Wünsche. Squeak ist Open Source und wird gestützt durch eine eifrige Gemeinde, die permanent neue Komponenten entwickelt. Ziel dieses Projektes ist es unter anderem die Squeak-Gemeinschaft zu unterstützen und mit einem so noch nicht da gewesenen Produkt zu bereichern. Squeak offeriert die bekanntesten Programmierkonzepte, ohne dass diese vom Anwender in Smalltalk-Syntax bekannt sein müssen. Die Squeak Umgebung wächst durch neu implementierte Funktionen oder Klassen stetig weiter, wodurch auch zu späteren Zeitpunkten noch auf Bestandteile früherer Konzepte zurückgegriffen werden kann. Weitere Informationen zu Squeak sind beispielsweise in [9] nachzulesen.

## 8 LITERATUR

- [1] BALZERT, HELMUT: **Lehrbuch der Software-Technik, Bd. 1 Software-Entwicklung**, Spektrum Akademischer Verlag, 2000. – ISBN 3-82740480-0
- [2] NACHTWEI, JENS ; KAIN, SASKIA ; KRINNER, CORDULA ; GROSS, BARBARA: **ATEO – ein Beitrag zum Graduiertenkolleg prometei**, 2008. URL: <http://www.psychologie.hu-berlin.de/prof/ingpsy/forschung/ateo> (abgefragt am 07.10.2008).
- [3] GROSS, BARBARA ; NACHTWEI, JENS: Using the microworld to acquire knowledge for developers and operators. In: WAARD, D. DE (Hrsg.) ; HOCKEY, G.R.J. (Hrsg.) ; NICKEL, P. (Hrsg.) ; BROOKHUIS, K.A. (Hrsg.): **Human Factors Issues in Complex System Performance**. Maastricht, the Netherlands : Shaker Publishing, 2007, S. 345-350.
- [4] WANDKE, HARTMUT ; NACHTWEI, JENS: The different human factor in automation: the developer behind versus the operator in action. In WAARD, D. DE (Hrsg.) ; FLEMISCH, F. O. (Hrsg.) ; LORENZ, B. (Hrsg.) ; OBERHEID, H. (Hrsg.) ; BROOKHUIS, K.A. (Hrsg.): **Human factors for assistance and automation**. Maastricht, the Netherlands : Shaker Publishing, 2008, S. 493-502.
- [5] SCHWARZ, HERMANN: **Fenster zum Prozess: ein Operatorsarbeitsplatz zur Überwachung und Kontrolle von kooperativem Tracking**, Humboldt-Universität zu Berlin, Institut für Informatik, Unveröffentlichte Diplomarbeit, 2009.
- [6] KESSELRING, KAI: **Entwicklung einer Softwarekomponente zur Systemprozessüberwachung in einer psychologischen Versuchsumgebung**, Humboldt-Universität zu Berlin, Institut für Informatik, Unveröffentlichte Diplomarbeit, 2009.
- [7] NIESTROJ, NICOLAS: **Vernetzung im ATEO Projekt aus inhaltlicher und technischer Sicht**, Humboldt-Universität zu Berlin, Institut für Informatik, Unveröffentlichte Diplomarbeit, 2009.
- [8] Übersicht Spielbretter und Rot-Werte, Stand 31.12.07
- [9] BLACK, ANDREW ; DUCASSE, STÉPHANE ; NIERSTRASZ, OSCAR ; POLLET, DAMIEN ; CASSOU, DAMIEN ; DENKER, MARCUS: **Squeak by Example**, Square Bracket Associates, 2008. – URL <http://SqueakByExample.org/index.html>. – ISBN 978-3-9523341-0-2
- [10] GROSS, BARBARA ; SACKLOWSKI, ULRICH: **ATEO-System – Nutzerdokumentation**, 2008

## 9 GLOSSAR

Administration	Verwaltung
Akteur	Handelnde Person oder technisches System
Aktuelle relative Systemzeit	Zeit seit Start des Versuches
Anwendungsfälle (Use Cases)	Grundfunktionalitäten des Systems.
Assistenzsysteme	Elektronische Unterstützungssysteme mit geeigneten Mensch-Maschine-Schnittstellen, wie z.B. Bremshilfe und Spurhaltung.
Audioanweisungen	Sprachanweisungen (hier: über die Kopfhörer)
Automatik	Ein System, welches eine Aufgabe ohne Eingriff durch einen Menschen ausführt. (Vollautomatik)
Bitmap	Eine gebräuchliche verkürzte Bezeichnung für das Grafikformat Windows Bitmap (.bmp)
Button	Bedienelement in graphischen Benutzeroberflächen (Schaltfläche)
Countdown	Zeitspanne des Rückwärtszählens, „Herunterzählen“
Effizienz	Wirksamkeit, Leistung, Wirtschaftlichkeit
Eingabemedium	Eingabegerät
Interface	Graphische Benutzeroberfläche, auch Graphics User Interface (GUI) genannt
Joystick	Eingabegerät für Computer („Steuerknüppel“)
Konfiguration	Einstellung
Logging	Das Aufzeichnen aller versuchsrelevanten Daten, die nötig sind um die gewünschte Versuchsauswertung vorzunehmen.
Mikrowelt	In Squeak definierte „Scheinwelt“, in der die Versuchspersonen (Nutzer) die Steuerung von Objekten übernehmen
Mikroweltbewohner	Eine Versuchsperson im System ATEO, welche die Aufgabe hat das Fahrzeug (Tracking-Objekt) zu steuern.
Modus	Art und Weise, Betriebsart in einem Anwendungsprogramm eines Computers
Operateur	Eine Person oder/und ein Softwaresystem, die in den Versuchsablauf korrigierend, assistierend eingreifen kann, z.B. Hinweise zum Strecken-



	verlauf geben, Eingriff in die Steuerung.
Parameter	Ein Argument, das einem Programm oder Unterprogramm übergeben wird
Parcours	Hindernisstrecke
Performance	Leistung
Popup	Visuelles Element eines Computerprogramms
Protokolldatei	Die Datei in der alle aufgezeichneten versuchsrelevanten Daten gespeichert werden.
Simulation	Nachbildung der Realität auf dem Computer
Smalltalk	Eine objektorientierte Programmiersprache.
Sound	Ton, Schall
Squeak	Entwicklungs- und Ausführungsumgebung für die Programmiersprache Smalltalk.
Steuermodalität	Art und Weise / Ausführungsart der Steuerung
Systembrowser	Browser: Software, die Datenträger, Verzeichnisse, Dateien oder Netzwerke durchsucht hier: Systemfenster, der alle Verzeichnisse und Dateien anzeigt
Team	Eine Gruppe von Mikroweltbewohnern, welche gemeinsam das Fahrzeug steuern.
Tracking	Eine Tracking bezeichnet hier die Aufgabe, ein Objekt in einem bestimmten Bereich (z.B. einer Straße) zu halten.
Versuchsgruppe	Eine Gruppe von Versuchspersonen, welche an einem Versuch unter bestimmten Bedingungen teilnehmen.
Versuchsleiter	Person, die für die Leitung eines Versuches zuständig ist.
Versuchsperson, Proband	Person, die sich einer Prüfung unterzieht oder als Test- oder Vergleichsperson einer Prüfung unterzogen wird.
Vogelperspektive	Bezeichnung für die Betrachtung eines Gegenstandes von einem schräg darüber gelegenen Augenpunkt.
XML	Abkürzung für Extensible Markup Language. Ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdateien.