

Analyse und Design

Verhalten kybernetischer Vehikel
Informatikprojekt I00-2

Reto Witschi
Andreas Gafner
Lukas Reusser
Pascal Fleury

HTA Bern I00-2



Analyse und Design

Inhaltsverzeichnis

1	Hardware Analyse.....	3
1.1	Sensoren	3
1.2	Motoren.....	4
1.3	Mechanische Genauigkeit	4
2	Analyse der Umgebung (Welt).....	4
3	Softwaredesign der Simulation	5
3.1	Sensor Anordnung und Richtungsänderung.....	5
3.2	Bewegung des Roboters	6
3.3	Klassendiagramm	7
3.3.1	Übersicht	7
3.3.2	robi.sim.gui	8
3.3.3	robi.sim.objects.....	9
3.3.4	robi.sim.code	10
3.3.5	josex.platform.rcx	11

1 Hardware Analyse

1.1 Sensoren

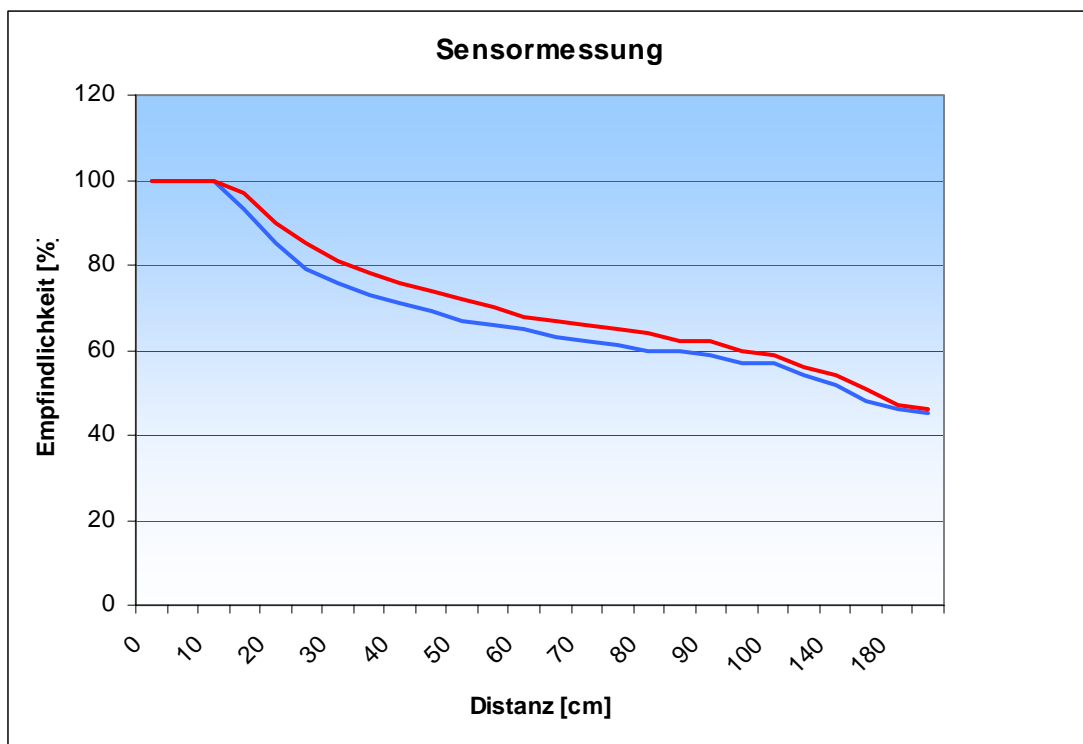
Messwerte der Sensoren:

Das Verhalten der Sensoren ist massgebend für eine genaue Reaktion des Roboters für die Richtungsänderung. Die Messung machten wir für eine Bestimmung der Ungenauigkeit.

Messbedingungen:

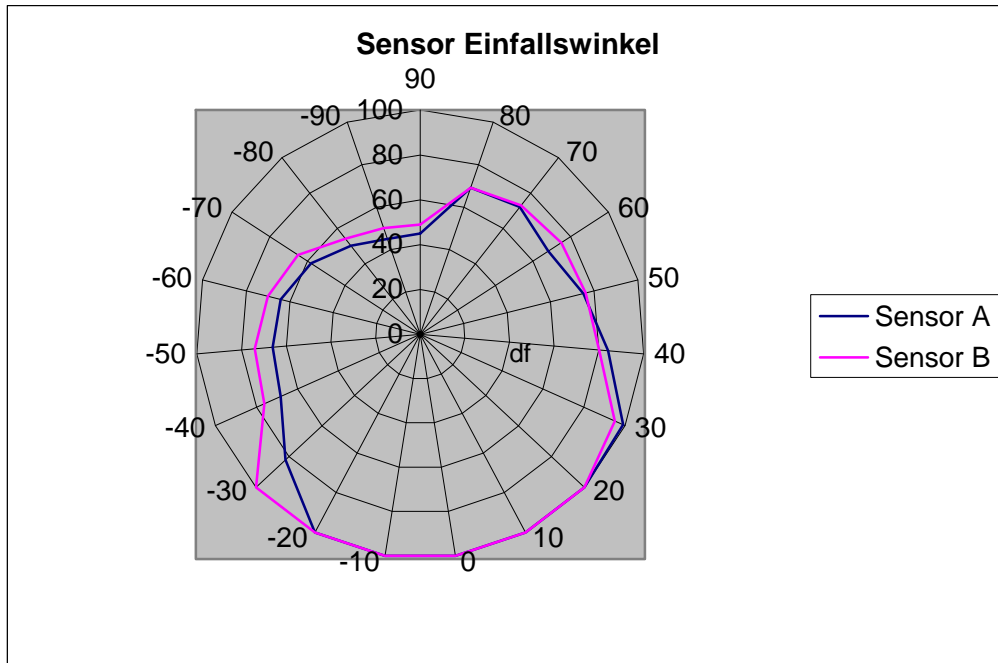
Die folgenden Messungen wurden in einem verdunkelten Raum durchgeführt. Es konnten störende Lichtquellen nicht vollständig ausgeschlossen werden.

Als Lichtquelle wurde eine handelsübliche Lichtquelle verwendet.



Wie man in dem Diagramm oben sehen kann, sind die Werte beider Sensoren (rot, blau) nicht exakt übereinstimmend, jedoch weitaus genügend für unsere Zwecke. Die beste Empfindlichkeit der Sensoren liegt zwischen 15 cm und 35 cm Distanz. Wünschenswert wäre jedoch ein linearer Verlauf über die ganze Mess-Distanz.

Zusätzlich zu der Distanz zwischen Lichtquelle und Sensor spielt auch der Einfallswinkel eine Rolle.



Der Einfallswinkel ist ab einer Distanz von > 20 cm bemerkbar. Je grösser der Winkel weg von der Sensorachse desto kleiner der Messwert.

Leider haben wir kein lineares Verhalten der Sensoren beobachtet. Den Messversuch führten wir einmal mit einem gerichteten Lichtstrahl (Taschenlampe) und zusätzlich mit einer rundum abstrahlenden Glühbirne durch. Die Messwerte waren ähnlich. Gut ersichtlich ist, dass je 20° weg von der Sensorachse immer einen Wert von 100% gemessen wurde, was auf ein stark gedämpftes Arbeiten der Sensoren hinweist.

1.2 Motoren

Anfangsschwierigkeiten hatten wir mit den ungleichen Geschwindigkeiten der beiden Motoren. Die Motoren zeichneten immer eine leichte Rechtskurve auf. Nach Anfrage bei Lego erhielten wir zwei ungefähr gleich schnell drehende Motoren. Trotzdem ist es nicht befriedigend und die Simulation weicht, auf grund solcher „Fehler“ sehr schnell von der Realität ab.

1.3 Mechanische Genauigkeit

Da das Wesen mit Lego-Bausteinen aufgebaut ist kann es zu mechanischen Ungenauigkeiten kommen. Das Spornrad z.B. kann die Bahn des Wesens bei Drehbewegungen und Richtungsänderungen geringfügig beeinflussen.

Diese Abweichungen werden durch das Verhalten unseres Roboters (gegeben durch das Programm, Wesen) fortlaufend durch entsprechende Richtungsänderungen automatisch kompensiert. Somit können wir diese Abweichungen in Kauf nehmen.

In der Simulation werden diese mechanischen Ungenauigkeiten nicht berücksichtigt.

2 Analyse der Umgebung (Welt)

Es steht uns kein absolut dunkler Raum ohne Reflektionen und Störlichtquellen zur Verfügung. Dies bedeutet, dass die Wesen durch diese Störeinflüsse beeinträchtigt

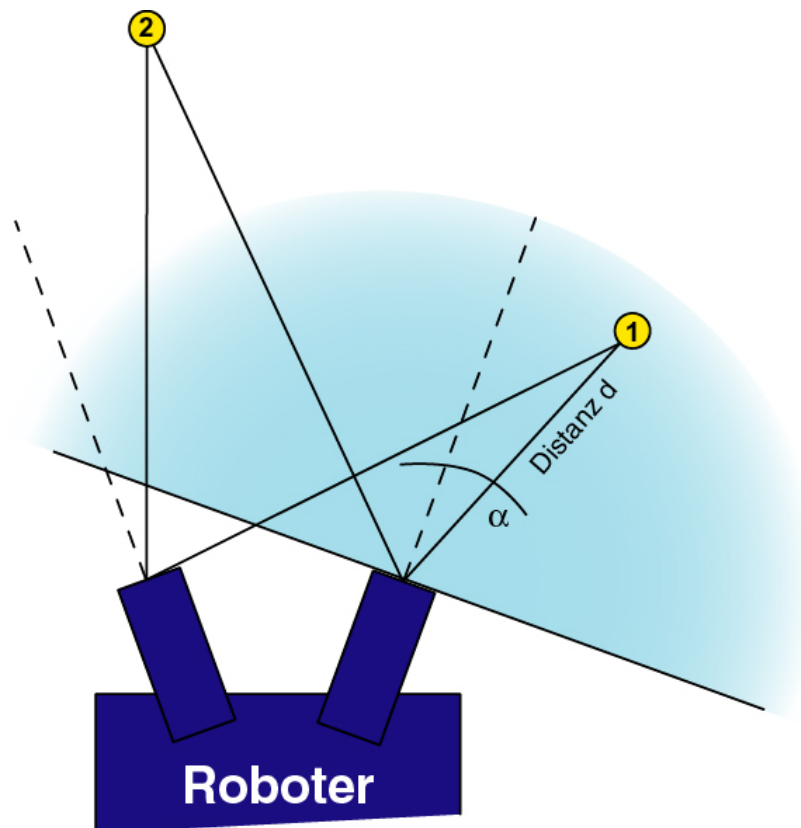
werden können. Dadurch entstehen Differenzen zwischen dem Verhalten der Wesen in der Simulation und der Wirklichkeit.

3 Softwaredesign der Simulation

3.1 Sensor Anordnung und Richtungsänderung

Auf unserem Roboter sind zwei Lichtsensoren montiert. Die Sensoren, in der Zeichnung dunkelblau eingefärbt, bestimmen eine eventuelle Richtungsänderung des Roboters. Der Roboter sucht seinen Weg durch seine Welt indem er sich immer an der stärksten Lichtquelle orientiert. Unser hier vorgestelltes Modell funktioniert auch für mehrere Lichtquellen.

Da beide Sensoren nur einen einzigen Wert, die Lichtstärke empfangen, entscheidet der höhere Wert über eine Richtungsänderung. Die Messung für eine Richtungsänderung wird fortlaufend entschieden. Der Wert welcher der Sensor 1 empfängt ist eine resultierende Summe aus beiden Lichtquellen 1 und 2, in der Zeichnung als gelber Kreis dargestellt. Das bedeutet, da die Distanz zur Lichtquelle 1 kürzer ist und demzufolge stärker als die Lichtquelle 2, wird die Drehung in Richtung der Stärkeren Lichtquelle (1) gemacht bis beide resultierenden Werte gleich sind. Der resultierende Wert für den Roboter als Gesamtes, ist eine Summe beider Sensoren und beider Lichtquellen.



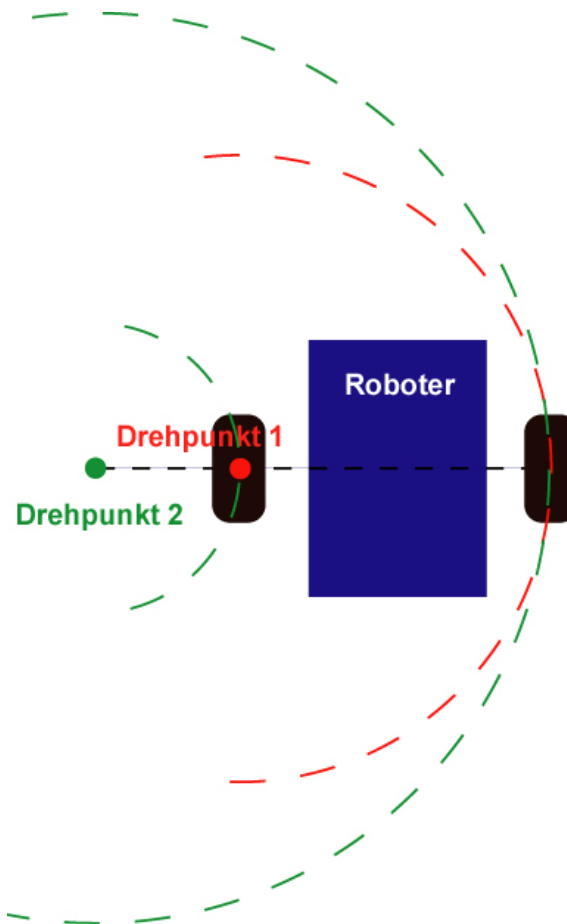
3.2 Bewegung des Roboters

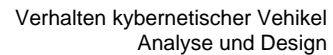
Nachfolgendes Bild beschreibt das Drehverhalten des Roboters. Die Drehbewegung in der Skizze ist beispielsweise nur auf zwei Drehpunkte gesetzt. Der effektive Drehpunkt ist in Wirklichkeit variabel, resultierend von der Geschwindigkeit beider separat angesteuerten Motoren.

Die Differenz zwischen der maximalen und minimalen Geschwindigkeit welche von lejos zur Verfügung gestellt werden (1-7), ist sehr gering. Nach Braitenberg wird eine Drehung nie mit einem still stehenden, oder sogar rückwärts laufenden Rad durchgeführt. Die Drehung erfolgt somit mit einem zu grossen Radius, je nach Nähe der Lichtquelle unbrauchbar.

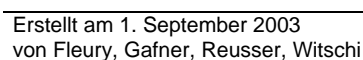
Wir versuchen diese Unschönheit zu umgehen indem wir in der Simulation nicht exakt nach Braitenberg vorgehen.

Wird die Kurve zu eng muss das eine Rad vollständig blockiert werden und die Drehung wird um den Drehpunkt 1 ausgeführt (Schwellenwert von einem Sensor wird unterschritten).

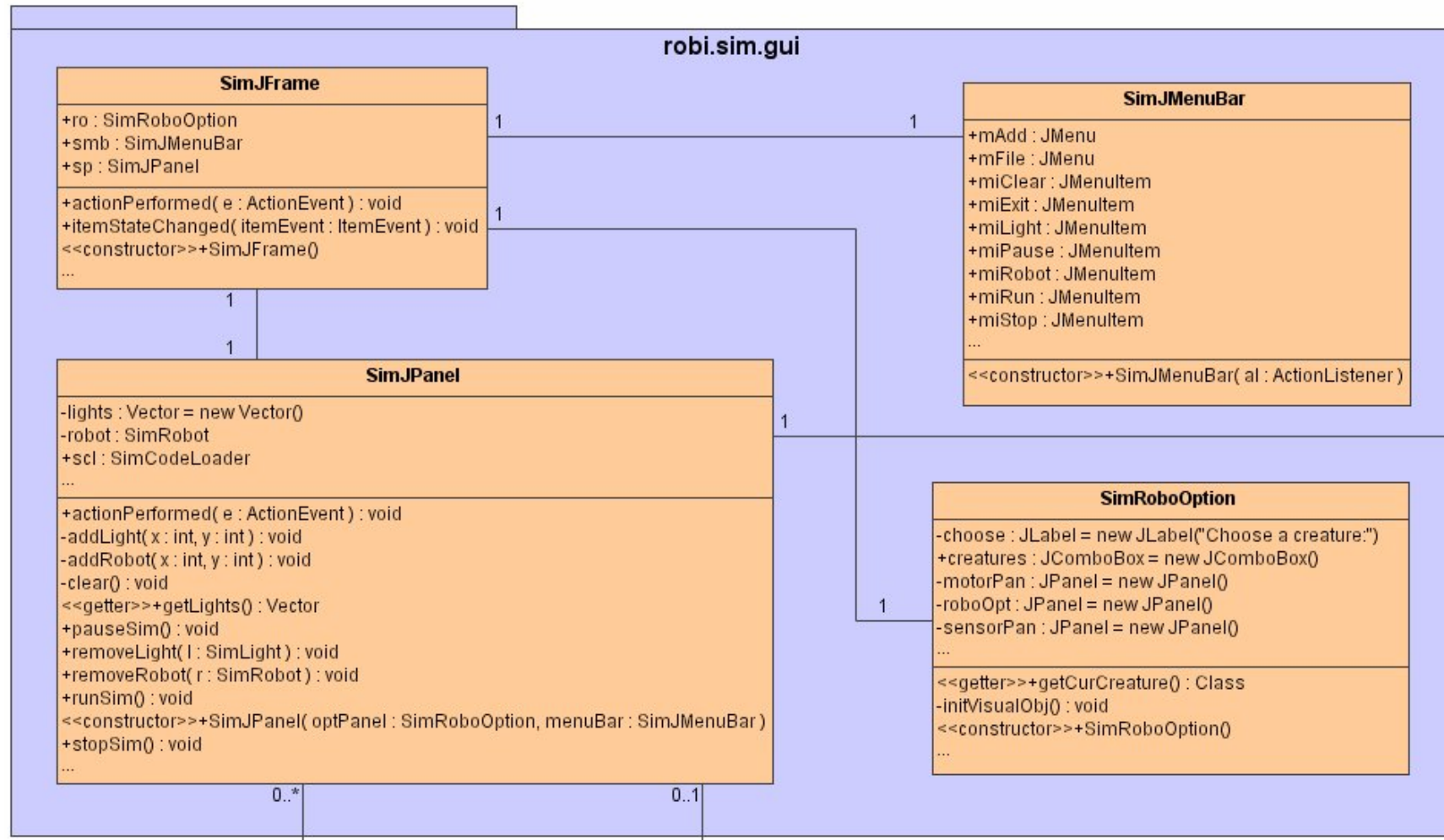




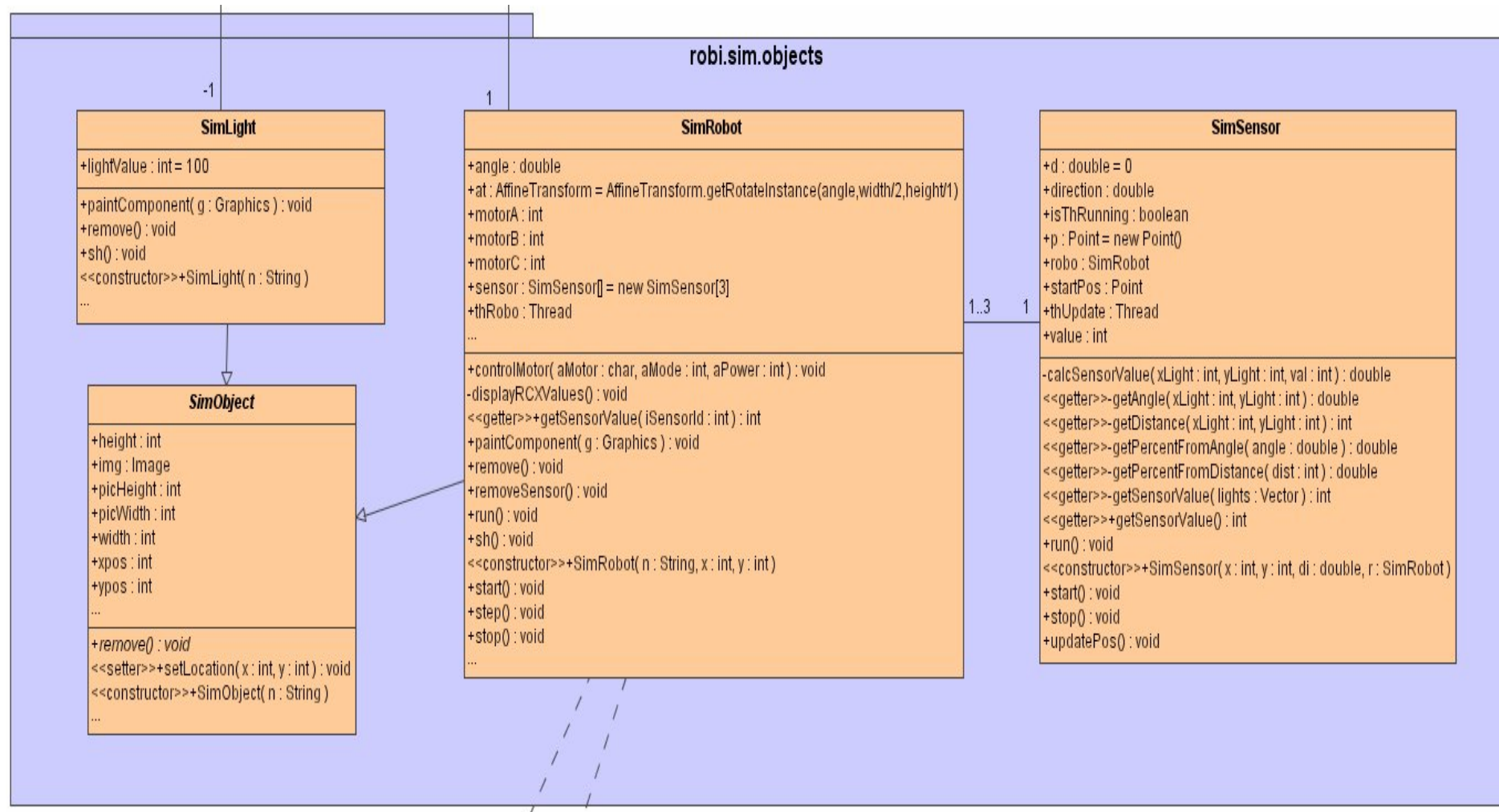
3.3.1 Übersicht



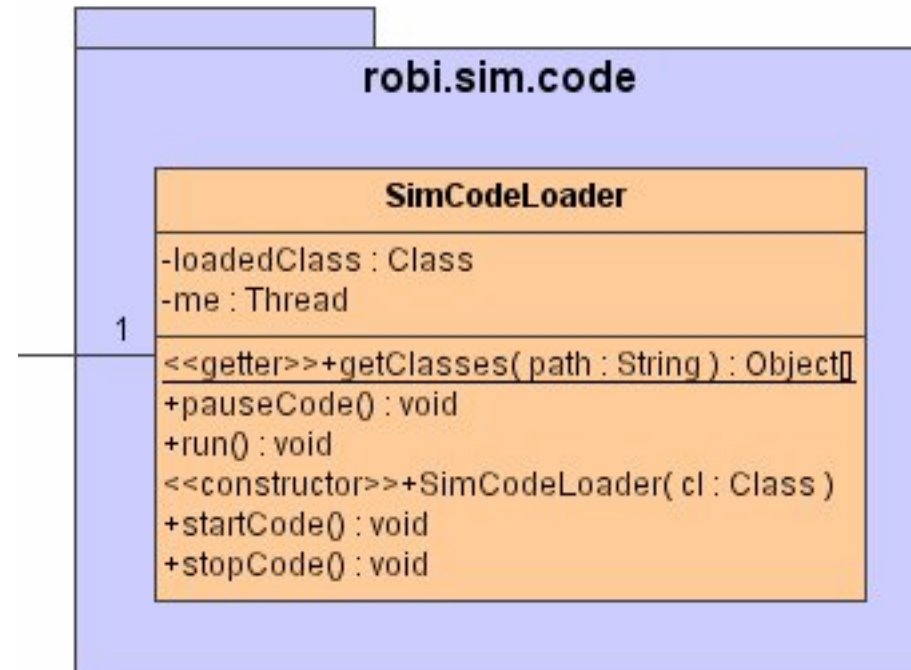
3.3.2 robi.sim.gui



3.3.3 robi.sim.objects



3.3.4 robi.sim.code



3.3.5 josx.platform.rcx

