



UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR TECHNISCHE INFORMATIK

Direktor: Prof. Dr.-Ing. Mladen Berekovic

Studienbegleitende Fachprüfung im Rahmen der
Bachelorprüfung im Studiengang

Robotik und Autonome Systeme / Informatik

Leistungszertifikat Typ A

SS 2018

Lehrmodul: Mobile Roboter

Prüfer: Prof. Dr.-Ing. Heiko Hamann

10. Oktober 2018

Name: _____

Matrikelnummer: _____

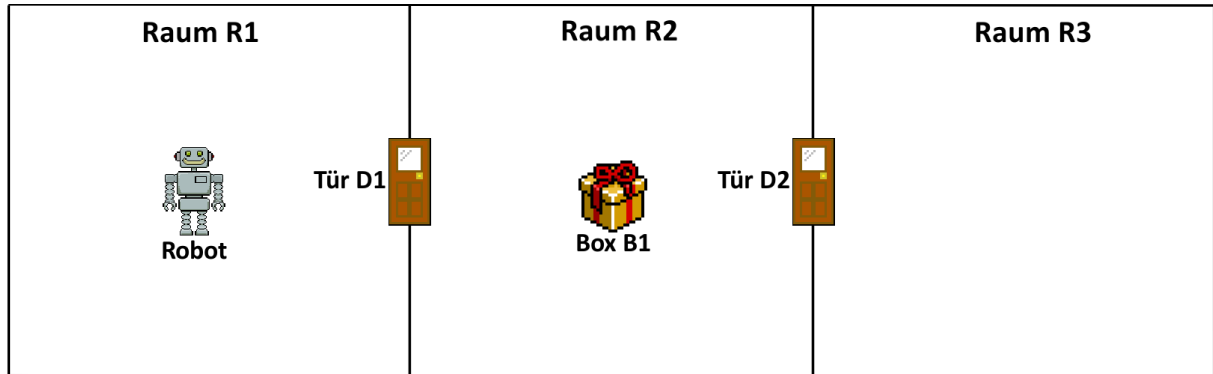
1	2	3	4	5	6	Σ
						/100

Name: _____

Matrikelnummer: _____

Aufgabe 1: Planung mit Strips**(20 Punkte)**

Gegeben sei folgende Repräsentation eines Weltmodels. In der Welt befinden sich die Räume R1, R2 und R3. Die Räume sind durch die Türen D1 und D2 miteinander verbunden. Im Raum R1 befindet sich ein Roboter, in Raum R2 die Box B1 und der Raum R3 ist initial leer. Der Roboter kann Türen durchschreiten und Boxen durch Türen in einen anderen Raum schieben. Das Ziel des Roboters soll es sein die Box B1 in den Raum R3 zu schieben.



- a) Definieren Sie in einem ersten Schritt geeignete Eigenschaften bzw. Prädikate für eine Planung mit Strips.

Name: _____

Matrikelnummer: _____

- b) Überlegen Sie sich sinnvolle Operationen für den Roboter und vervollständigen Sie die folgende Tabelle mit preconditions, der add-list und der delete-list.

operator	preconditions	add-list	delete-list

- c) Formulieren Sie das Ziel mit den bereits definierten Eigenschaften/Prädikaten und den Operationen.

Goal	
-------------	--

Name: _____

Matrikelnummer: _____

- d) Geben Sie eine Folge von Operationen an, die der Planer generieren könnte, um das spezifizierte System in den in c) definierten Zielzustand zu überführen.

Operator	Zustand
INIT	

- e) Was ist der Nachteil an einer Plangenerierung mit Strips?

Aufgabe 2: Roboterkonzeption**(15 Punkte)**

Ihre Aufgabe ist die Konzeption eines autonomen Service-Roboters, der innerhalb eines Museums zwischen festgelegten Stationen navigieren soll und bei Bedarf den Besuchern zusätzliche Informationen zu der aktuellen Stationen auf einem Display präsentiert. Der Roboter darf dabei nicht mit seiner Umgebung kollidieren, insbesondere sollen Kollisionen mit den Besuchern verhindert werden.

- a) Für welches der drei klassischen Paradigmen würden Sie sich entscheiden (SPA,SA,P-SA)? Begründen Sie für jedes Paradigma, warum dieses Ihrer Ansicht nach geeignet bzw. ungeeignet ist.

/4

Name: _____

Matrikelnummer: _____

- b) Sie haben bei dem Antrieb des Roboters die Wahl zwischen einem Differentialantrieb und einem holonomen Antrieb mit Omniwheels. Begründen Sie Ihre Wahl und beschreiben oder skizzieren Sie für beide Möglichkeiten wie eine Fahrtplanung bestimmt werden kann.

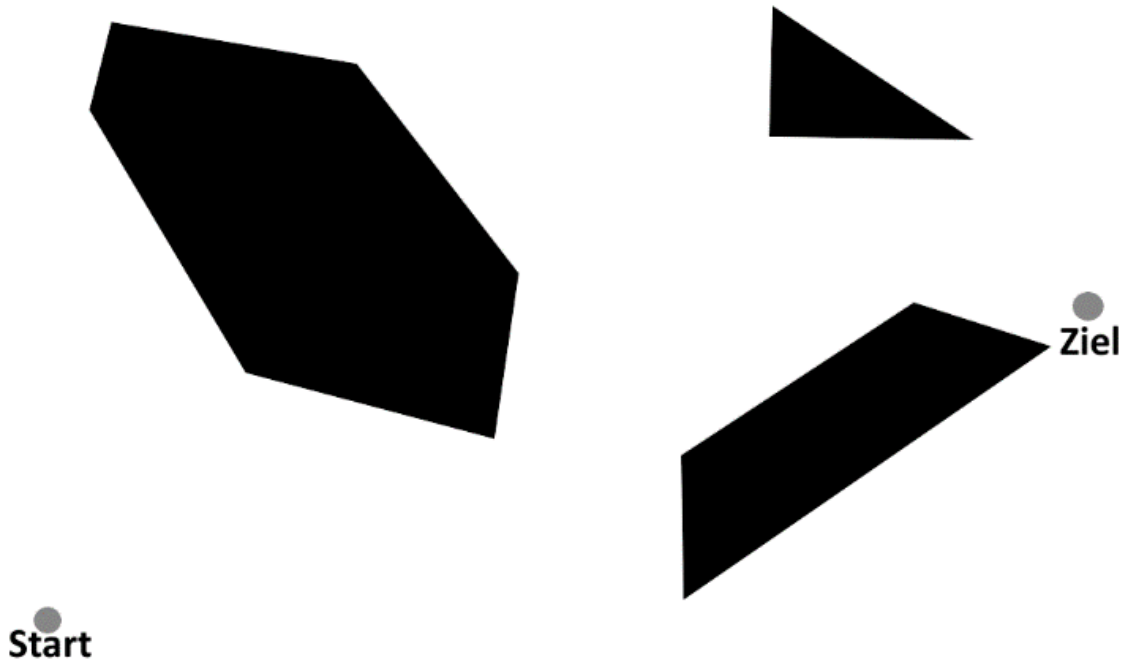
Name: _____

Matrikelnummer: _____

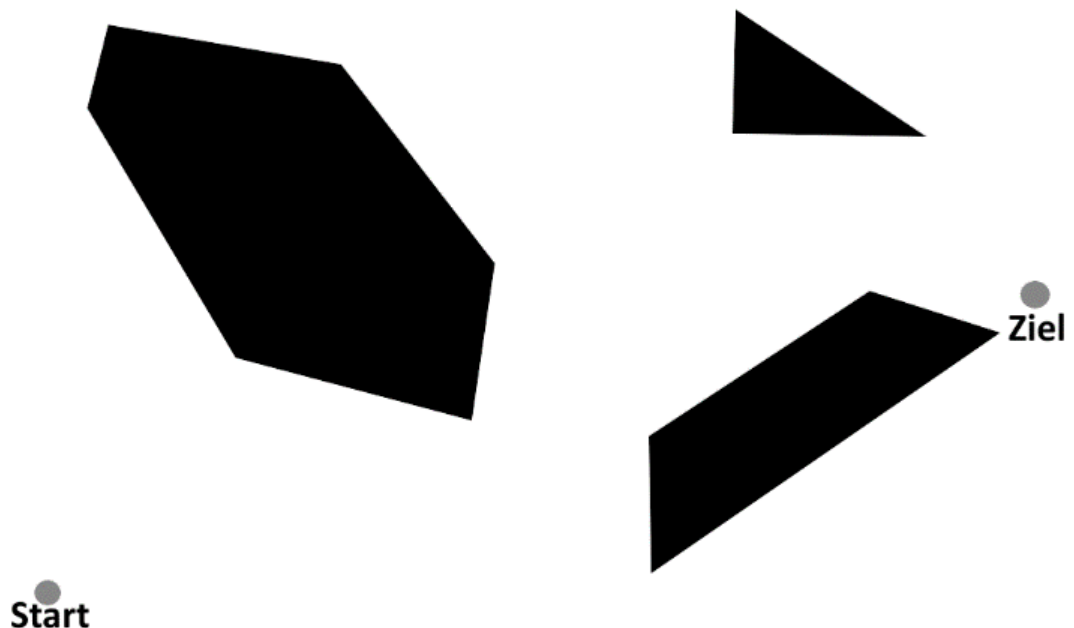
- c) Wählen Sie drei Sensoren aus um in dem Museum zu navigieren (bekannte Karte) und auch auf dynamische Hindernisse wie Besucher zu reagieren. Begründen Sie ihre Wahl.

Aufgabe 3: Karten**(20 Punkte)**

- a) Zeichnen Sie in der folgenden Abbildung den Sichtbarkeitsgraphen ein. Markieren Sie anschließend einen möglichst kurzen Pfad vom Start zum Ziel.



- b) Zeichnen Sie in der folgenden Abbildung nun die exakte Zellzerlegung (jede Zelle ist entweder komplett frei oder belegt) ein. Führen Sie die Zerlegung nur in der Vertikalen durch. Konstruieren Sie anschließend einen Graphen und markieren Sie den kürzesten Weg.



Name: _____

Matrikelnummer: _____

Aufgabe 4: Robotik-Definitionen

(15 Punkte)

- a) Nennen Sie die Eigenschaften eines "Personal Robot".

- b) Nennen Sie 5 Sensortypen, die ein Transportroboter zur Lokalisierung nutzen kann.

Name: _____

Matrikelnummer: _____

- c) Begründen Sie, warum ein Exoskelett ein Roboter ist oder warum nicht.

- d) Nennen und erklären Sie die Vor- und Nachteile, die einen Laufroboter (z.B. OSCAR) gegenüber einem rein radbetriebenen Roboter (z.B. HECTOR) hat.

Name: _____

Matrikelnummer: _____

- e) Vergleichen Sie das deliberative mit dem reaktiven Robotik-Paradigma.

Aufgabe 5: Thymio-Programmierung**(15 Punkte)**

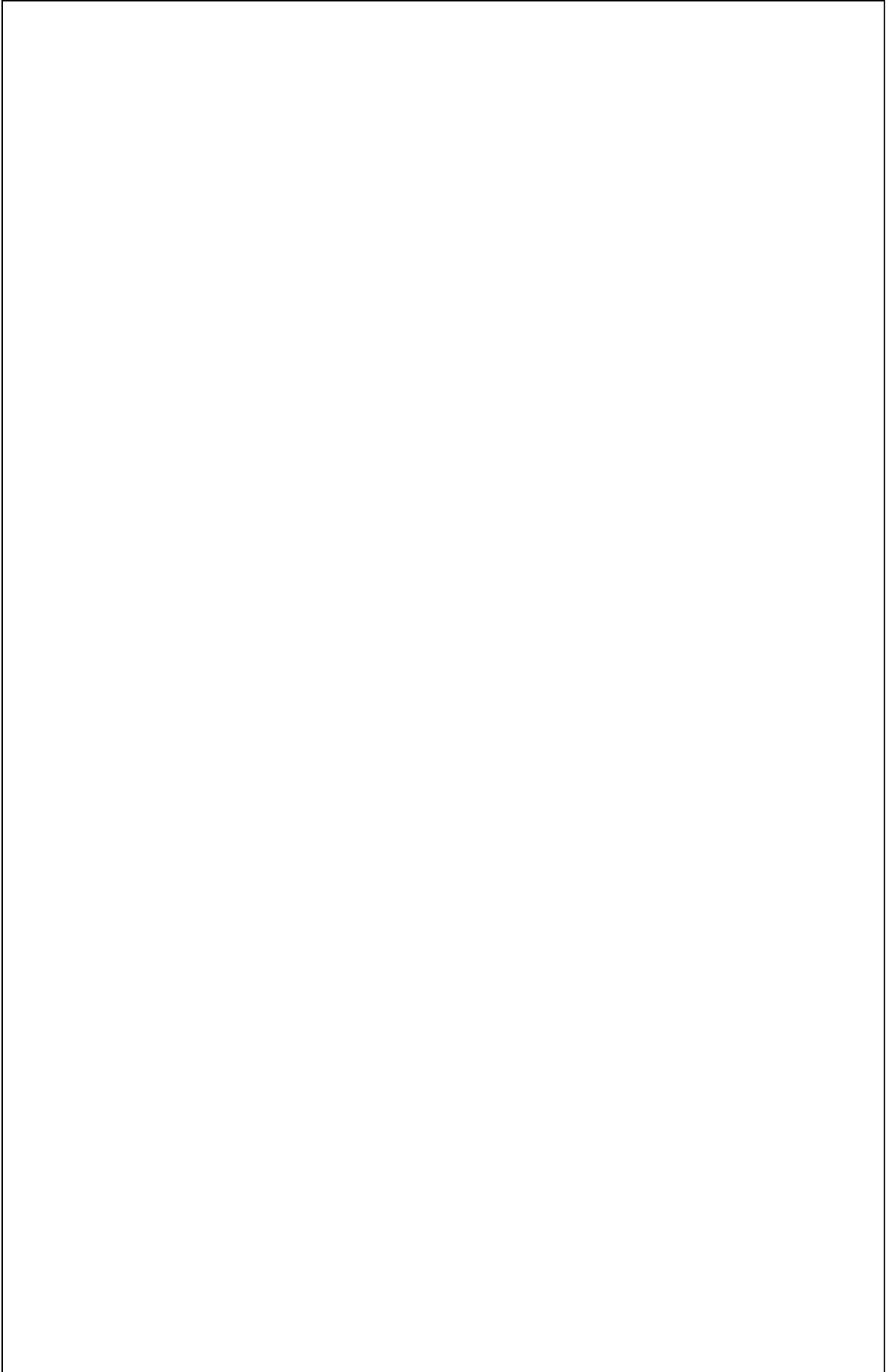
Der beiliegende Code erkennt weiße Linien auf dem Boden (eine virtuelle Mauer) und bewegt die Roboter davon weg.

- a) Nun lassen wir die Roboter auf einem Tisch fahren, so dass die Tischkante die Umrandung des befahrbaren Bereichs markiert statt einer weißen Linie. Bitte verändern und ergänzen Sie die nötigen Codezeilen, damit die Roboter nicht vom Tisch fallen.
- b) Bitte fügen Sie die nötigen Codezeilen hinzu, so dass die Roboter für 5 Sekunden anhalten, wenn sie sich treffen.
- c) Nachdem die Roboter angehalten haben, sollen sie sich 2 Sekunden lang in eine zufällige Richtung drehen, bevor sie wieder vorwärts fahren.

Bitte am Anfang immer die Codezeile angeben, an der eine Ergänzung im Original-Code stattfindet!

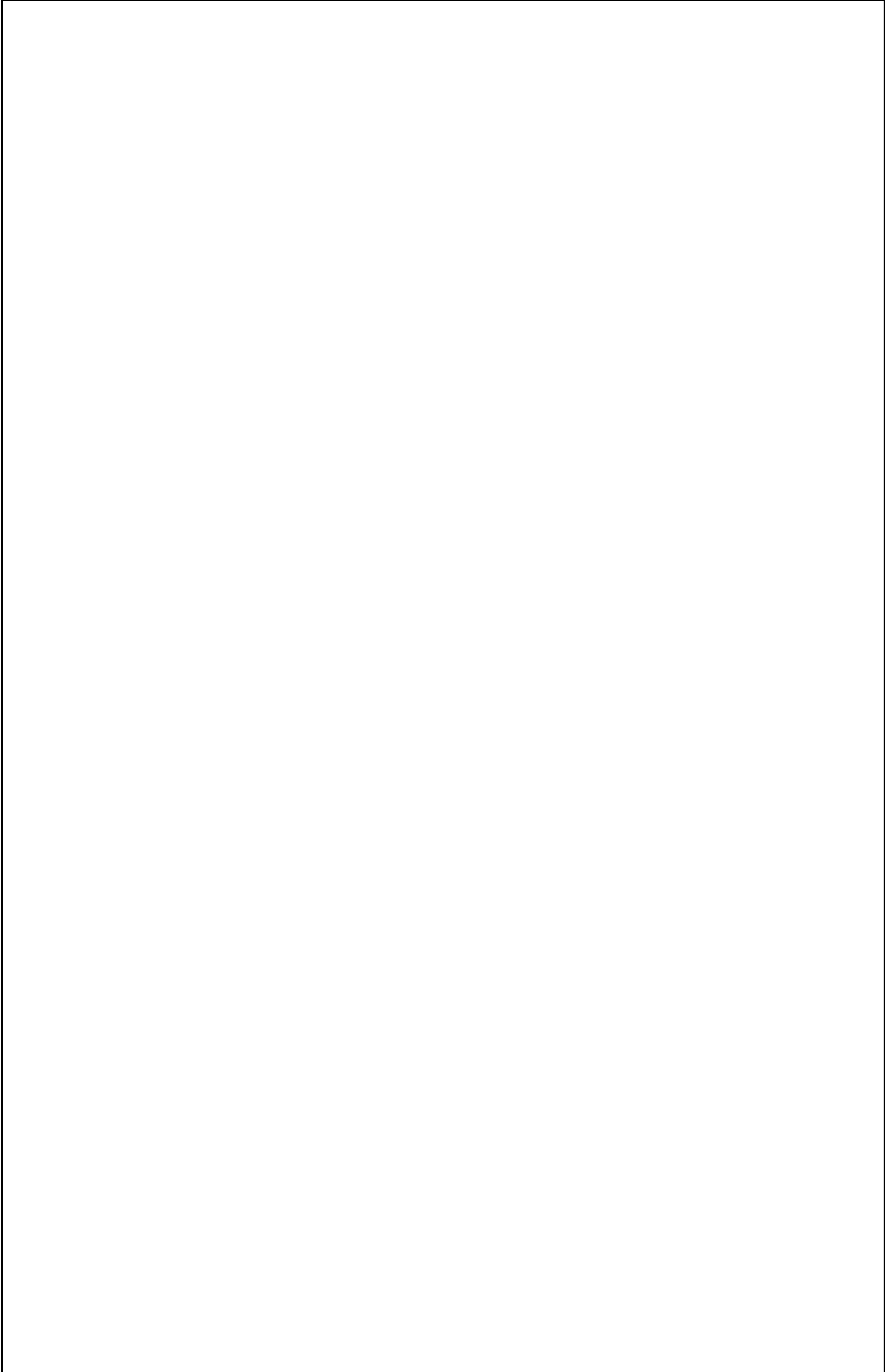
Name: _____

Matrikelnummer: _____

A large, empty rectangular box with a thin black border, occupying the central portion of the page. It is intended for a drawing or a detailed written response.

Name: _____

Matrikelnummer: _____

A large, empty rectangular box with a thin black border, occupying the central portion of the page. It is intended for a drawing or a detailed answer.

```
1 import dbus
2 import dbus.mainloop.glib
3 import gobject
4 import time
5 from random import randint, random
6 from optparse import OptionParser
7
8 proxSensorsVal=[0,0,0,0,0]
9 groundSensorsVal=[0,0]
10 forward_button=0
11 center_button=0
12 robotState = 1
13 turnDirection = "left"
14 ....
15
16 def BEECLUST():
17     global robotState
18     global turnDirection
19     ...
20
21     network.GetVariable("thymio-II", "prox.horizontal",
22         reply_handler=get_variables_reply,error_handler=get_variables_error)
23     network.GetVariable("thymio-II", "prox.ground.reflected",
24         reply_handler=get_ground_sensors,error_handler=get_variables_error)
25     network.GetVariable("thymio-II", "button.forward",r
26         eply_handler=get_forward_button,error_handler=get_variables_error)
27     network.GetVariable("thymio-II", "button.center",
28         reply_handler=get_center_button,error_handler=get_variables_error)
29
30     if forward_button == 1:
31         robotState = 1
32     elif center_button == 1:
33         robotState = 0
34
35     if groundSensorsVal[0] > 900 or groundSensorsVal[1] > 900:
36         if robotState != 2:
37             turnDirection = "left" if random() > 0.5 else "right"
38             robotState = 2
39             ...
40             turn(turnDirection)
41
42     if robotState == 0:
43         network.SetVariable("thymio-II", "motor.left.target", [0])
44         network.SetVariable("thymio-II", "motor.right.target", [0])
45     elif robotState == 1:
46         network.SetVariable("thymio-II", "motor.left.target", [100])
47         network.SetVariable("thymio-II", "motor.right.target", [100])
48     elif robotState == 2:
49         if ...: #time is not up keep turning
50             turn(turnDirection)
51         else:
52             robotState = 1
53
54     print "state = " + str(robotState)
55
56     return True
57
58 def turn(direction = "left"):
59     if direction == "left":
60         network.SetVariable("thymio-II", "motor.left.target", [-200])
61         network.SetVariable("thymio-II", "motor.right.target", [200])
62     else:
63         network.SetVariable("thymio-II", "motor.left.target", [200])
64         network.SetVariable("thymio-II", "motor.right.target", [-200])
65
66 def get_variables_reply(r):
67     global proxSensorsVal
68     proxSensorsVal=r
69
```


Name: _____

Matrikelnummer: _____

```
70 def get_ground_sensors(r):
71     global groundSensorsVal
72     groundSensorsVal=r
73
74 def get_forward_button(r):
75     global forward_button
76     forward_button=r[0]
77 def get_center_button(r):
78     global center_button
79     center_button=r[0]
80
81 def get_variables_error(e):
82     print 'error:'
83     print str(e)
84     loop.quit()
85
86 if __name__ == '__main__':
87     parser = OptionParser()
88     parser.add_option("-s", "--system", action="store_true", dest="system", default=False, help="use the sy:
89     (options, args) = parser.parse_args()
90     dbus.mainloop.glib.DBusGMainLoop(set_as_default=True)
91
92     if options.system:
93         bus = dbus.SystemBus()
94     else:
95         bus = dbus.SessionBus()
96
97     network = dbus.Interface(bus.get_object('ch.epfl.mobots.Aseba', '/'), dbus_interface='ch.epfl.mobots.A:
98
99     print network.GetNodesList()
100    print 'starting loop'
101    loop = GObject.MainLoop()
102    handle = GObject.timeout_add (100, BEECLUST) #every 0.1 sec
103    loop.run()
```

Aufgabe 6: Roboter-Lernen**(15 Punkte)**

- a) In Bezug auf künstliche neuronale Netze:
- i. Definieren und erklären Sie die Funktionsweise der Komponenten eines künstlichen Neurons: Input, Transferfunktion und Aktivierungsfunktion.

- ii. Erklären Sie, auf welche Weise die lineare, die Stufen- und die sigmoide Aktivierungsfunktion qualitative Unterschiede in ihrer Wirkung haben.

Name: _____

Matrikelnummer: _____

b) Man nehme an, dass wir einen Thymio-Roboter nutzen und dieser lernen soll, einer weißen Linie auf schwarzem Grund zu folgen (line following). Erklären Sie Ihren Ansatz für diese Lernaufgabe, wenn Sie jeweils

- i. Q-learning,
- ii. Hebbsches Lernen,
- iii. künstliche Evolution (z.B. einen genetischer Algorithmus)

verwenden würden. Was muss für jeden dieser Algorithmen im Allgemeinen definiert und als Eingabe bereitgestellt werden? Wie würden Sie diese Eingabe dann hier für das Linienfolgeverhalten definieren?