The background image shows a street intersection. On the left, a traffic light has a green pedestrian symbol lit. To the right, a pedestrian crossing signal shows a red figure. In the background, there is a modern building with red window frames and some trees with autumn-colored leaves.

Softwareprojekte mit dem Rennsimulator TORCS

8. Münsteraner Workshop zur Schulinformatik
18. Mai 2018

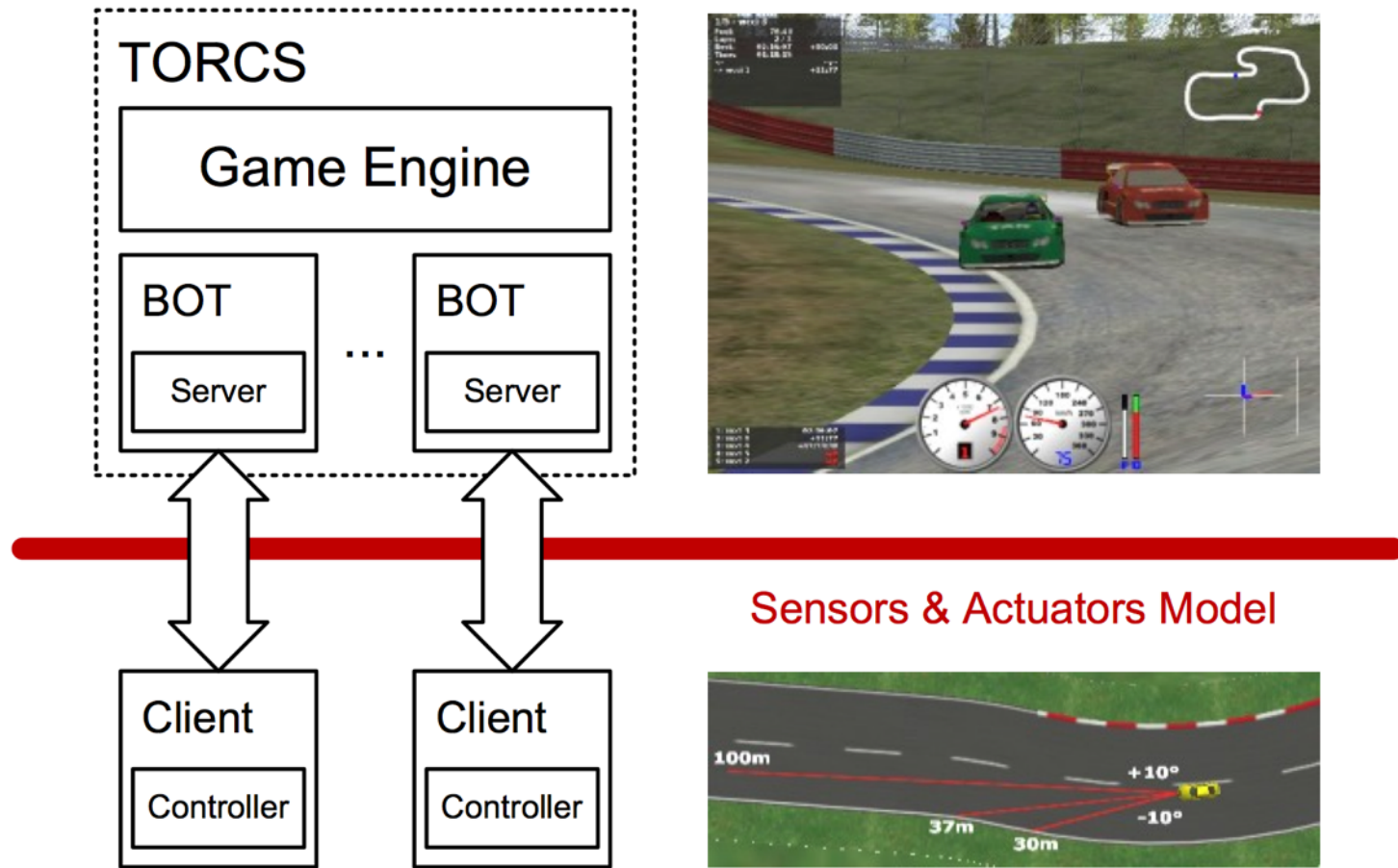
Till Zoppke, Gymnasium Tiergarten

TORCS

- Open-Source Rennsimulation
- Programmiert in C++
- Seit 1997
- Maintainer: Bernhard Wymann
- Realistische Fahrdynamik
- Plattform für Maschinelles Lernen (Genetische Algorithmen)
- 2007-2015: Simulated Car Racing (SCR) Championship
- <http://torcs.sourceforge.net>



Simulated Car Racing (SCR)



[LCL13] Loiacono, Daniele; Cardamone, Luigi; Lanzi, Pier Luca: Simulated Car Racing Championship Competition Software Manual, 2013, <https://arxiv.org/abs/1304.1672>, 8.4.2018.

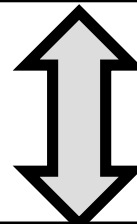
TORCS Architektur

SCR Server Patch (installiert)

- Bietet 10 neue Fahrer zur Auswahl: scr_driver_1 bis scr_driver_10

TORCS 1.3.4 (läuft nur in Windows-PC Pools)

Kommunikation über Netzwerk
(Socket)



Messages alle 20ms (1 tick):

- Server schickt Status
- Client schickt Aktionen

SCR Java Client: `package torcs.scr`

- Messagehandling, Datenklassen, abstract class Driver
- Nur benutzen, nicht verändern!

`SensorModel`

`public abstract Action`

`control(SensorModel sensors);`

`Action`

class VollgasDriver: `package torcs.vollgas`

Neues Package anlegen:

- class NameDriver: `package torcs.name`

Steuerung des Fahrzeugs

Name	Range	Description
accel	[0,1]	Virtual gas pedal (0 means no gas, 1 full gas).
brake	[0,1]	Virtual brake pedal (0 means no brake, 1 full brake).
clutch	[0,1]	Virtual clutch pedal (0 means no clutch, 1 full clutch).
gear	-1,0,1,⋯,6	Gear value.
steering	[-1,1]	Steering value: -1 and +1 means respectively full right and left, that corresponds to an angle of 0.366519 rad.
focus	[-90,90]	Focus direction (see the <i>focus</i> sensors in Table 1) in degrees.
meta	0,1	This is meta-control command: 0 do nothing, 1 ask competition server to restart the race.

[LCL13] Loiacono, Daniele; Cardamone, Luigi; Lanzi, Pier Luca: Simulated Car Racing Championship Competition Software Manual, 2013, <https://arxiv.org/abs/1304.1672>, 8.4.2018.

Umgebungsinformationen: insgesamt 19 Datenfelder

ProInformatik an der Freien Universität

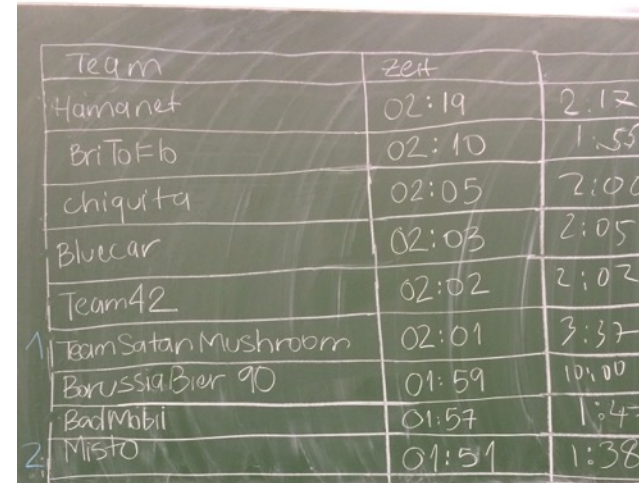
- Für Schülerinnen und Schüler der Sek II
- Zwischen Abitur und Studium
- Blockveranstaltungen während der vorlesungsfreien Zeit
- Können z.T. auf ein Studium der Informatik angerechnet werden
- In diesem Sommer 6 Vorlesungen bzw. Kurse:
<http://pro.inf.fu-berlin.de/>



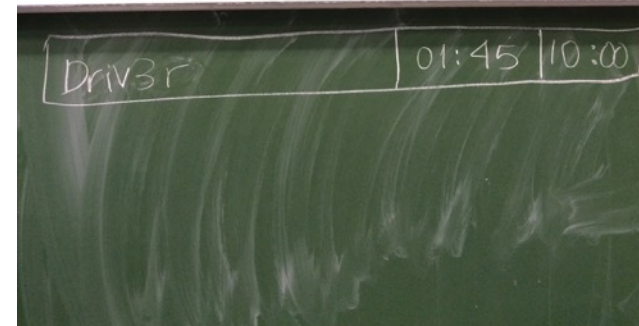
TORCS an der Freien Universität

- Abschlussprojekt für „Objektorientierte Programmierung“ (3 Iterationen)
- „Entwickelt eine Fahrer-KI für TORCS“
- Aufbauend auf einem Prototypen: VollgasDriver
- 3 bzw. 5 Tage, In 2-3er Gruppen
- Lernziele:
 - Anwendung der Java-Programmierkenntnisse
 - Motivation für das Studium
 - Gemeinsamer Abschluss (Wettbewerb im Hörsaal)

➔ Fazit: 😊



Team	zeit	
HamaNet	02:19	2:12
BriTofb	02:10	1:55
chiquita	02:05	2:00
Bluecar	02:03	2:05
Team42	02:02	2:02
1 TeamSatan Mushroom	02:01	3:37
BorussiaBier 90	01:59	10:00
BadMobi	01:57	1:47
2: Misto	01:51	1:38



Driver	01:45	10:00
--------	-------	-------

Gymnasium Tiergarten



- Schule in Berlin Mitte
- Verbreitet Migrationshintergrund
- Ca. 900 Schülerinnen und Schüler
- Bilingualer Zweig (Englisch)
- Informatik als Wahlpflichtfach ab Klassenstufe 9



TORCS am Gymnasium Tiergarten

- Softwareprojekt in Q4, 9 Schüler
- „Entwickelt eine Fahrer-KI für TORCS“
- Aufbauend auf einem Prototypen: SimpleDriver
- Ca. ~~27~~ 17 Schulstunden, in 3er Gruppen
- Lernziele:
 - Eigenständige Projektplanung
 - Vertiefungsgebiet Künstliche Intelligenz
 - Projektwerkzeuge anwenden (git/ github, Konsole)
- ➔ Fazit: 😊
 - Kurzweilige Autorennen
 - Schwächere Schüler überfordert
 - Kaum Fortschritt zum SimpleDriver

TORCS als Softwareprojekt an der Schule

Pro	Contra
<ul style="list-style-type: none">• Forschungsnahes Thema• Motivation durch Wettbewerb• Differenzierung „nach oben“• Software-Engineering (batch-Skripte)	<ul style="list-style-type: none">• Einarbeitungs- und Installationsaufwand für die Lehrkraft• Keine Differenzierung „nach unten“ (Programmierkenntnisse)• Kaum Gelegenheit für verteiltes Arbeiten

Empfehlungen

- Regelmäßige Programmieraufgaben in Q1-Q3
- Aufgabenstellung auf finalen Wettbewerb fokussieren
- Als Klausurersatzleistung
- Hilfestellungen für die Entwicklung von KI-Funktionen