

BROWN CS148

Collegiate and graduate robotics
with the Create/ASUS

Chad Jenkins - Brown University Computer Science

CS 148 website (2009 design)

CSci1480: Building Intelligent Robots



Note: my iPhone bricked, Apples rot

<http://www.cs.brown.edu/courses/cs148>

everything CS 148-related will be accessible from here

CSci1480: Building Intelligent Robots



<http://www.cs.brown.edu/courses/cs148>

ADMINISTRIVIA

- **Class meetings:** F hour (1-1:50pm)
 - Lectures in CIT 368 (typically MW)
 - Hands-on sessions/labs in CIT 404 (typically Fridays)
- **Collaborative projects**
 - implemented in groups of 2-3
 - individual written reports
 - git repositories: submission and collaborative work space
 - post-project code sharing between groups
- **Collaboration policy:** must be turned in to HTA
- Stay updated via **mailing list**, MOTD, and calendar
- **Reproducibility:** Robot Operating System (**ROS**)

CS 148

- Study of autonomous robotics
 - given robot hardware, program it to autonomously perform task
 - algorithms and architectures for autonomous robot control
 - robots are not sentient, but their programmers are
- Engineering/building of robots is a separate class
 - kinematics, dynamics, control theory

WHY ROBOTICS?

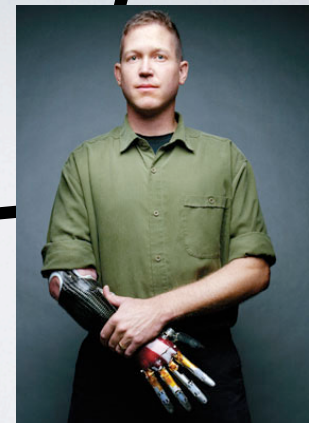
- Robotics:
 - is developing tools to enhance physical human productivity
 - healthcare, exploration, service, manufacturing, ...
 - will complement (not replace) human capability
 - an extension of computing and the Internet beyond digital worlds into the physical world that we inhabit
 - enable human users to supervise teams of robots
 - similar to how individuals supervise collections of computers

<http://www.us-robotics.us/>

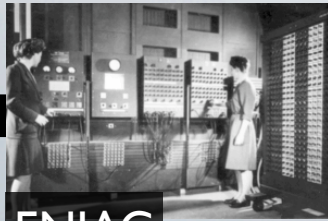
Personal computing has enabled individuals to manage digital information by supervising teams of heterogeneous computing devices



Autonomous robotics will enable individuals to manage physical environments by supervising teams of heterogeneous robotic devices



Personal Computing



ENIAC



Apple II



Laptop



OLPC

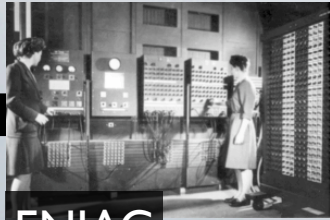
“technology exponentials”, e.g., Moore’s Law;
mentioned by Brooks and others

Research

Novelty tech

Pervasive tools

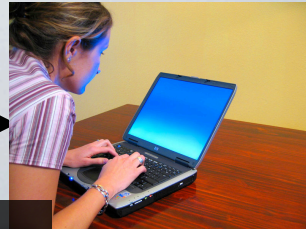
Personal Computing



ENIAC



Apple II

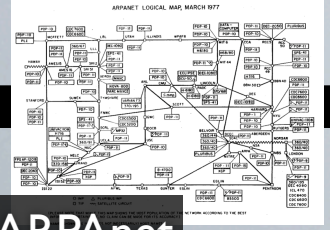


Laptop

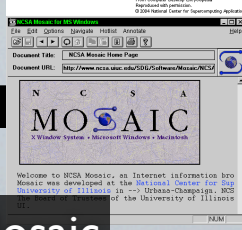


OLPC

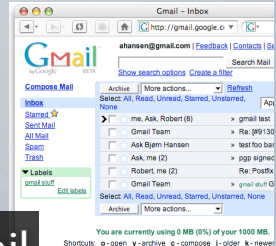
Internet



ARPANet



Mosaic

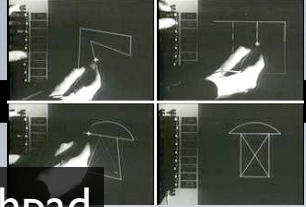


Gmail

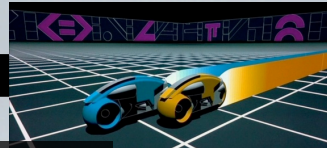


YouTube

Graphics



Sketchpad



Tron



Final Fantasy



Madden

Research

Novelty tech

Pervasive tools

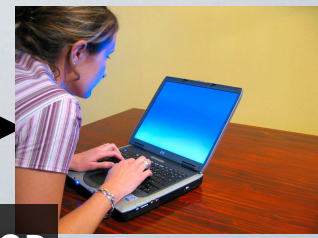
Personal Computing



ENIAC



Apple II

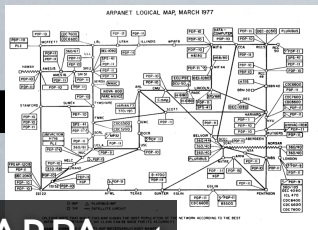


Laptop

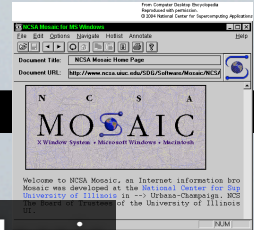


OLPC

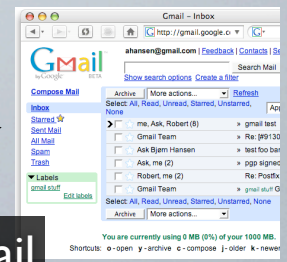
Internet



ARPANet



Mosaic

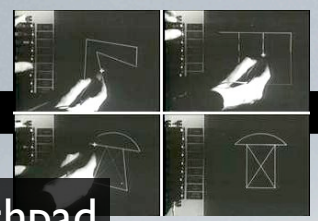


Gmail

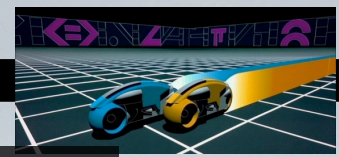


YouTube

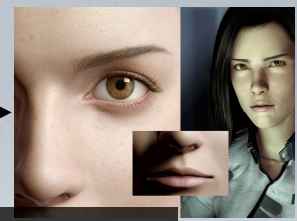
Graphics



Sketchpad



Tron



Final Fantasy

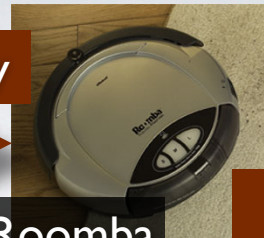


Madden

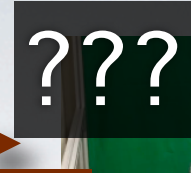
Robotics



Shakey



Roomba



???



PR2

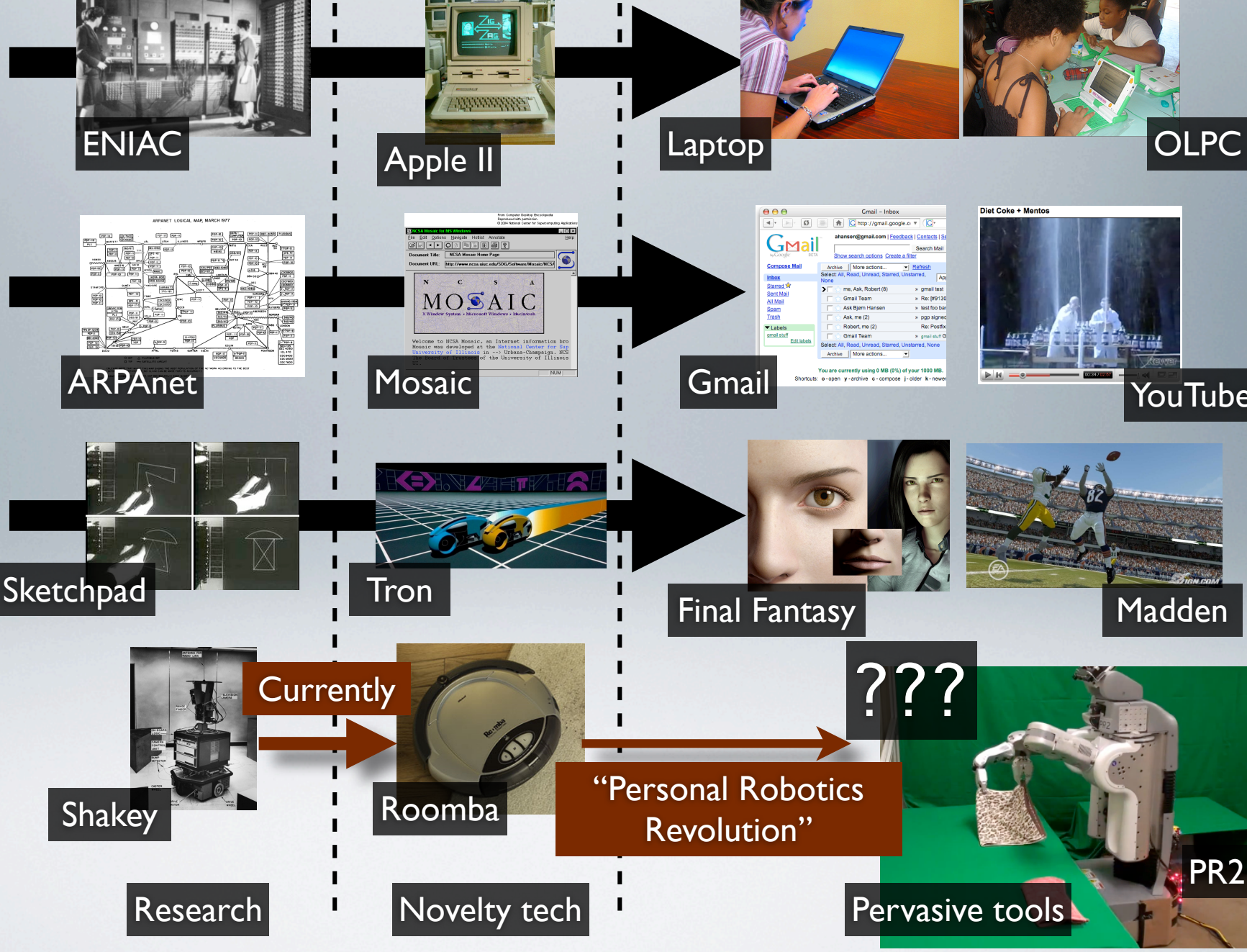
Research

Novelty tech

Pervasive tools

Currently

"Personal Robotics Revolution"



WHY ROBOTICS?

- “Personal robotics revolution”: robotics in a similar stage as
 - personal computing in the 1970s
 - computer graphics in the 1980s
 - the Internet in the early 1990s
- Robotics is a “technology exponential” in the making:
http://fora.tv/2009/05/30/Rodney_Brooks_Remaking_Manufacturing_With_Robotics
- This is the time to get in! Innovators can shape the technology... and get rich and/or famous

Autonomous robotics will enable individuals to manage physical environments by supervising teams of heterogeneous robotic devices

Challenges

Building robotic devices

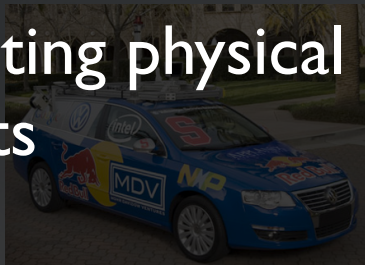
Programming robots

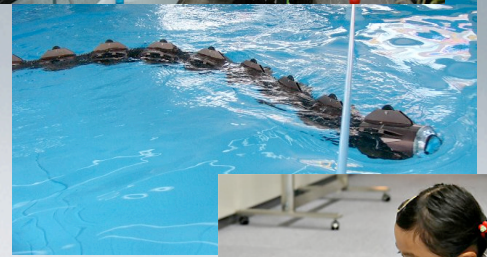
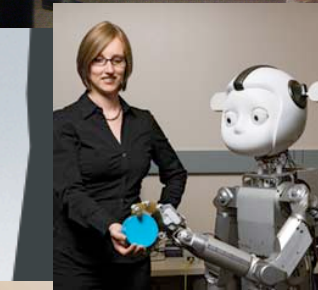
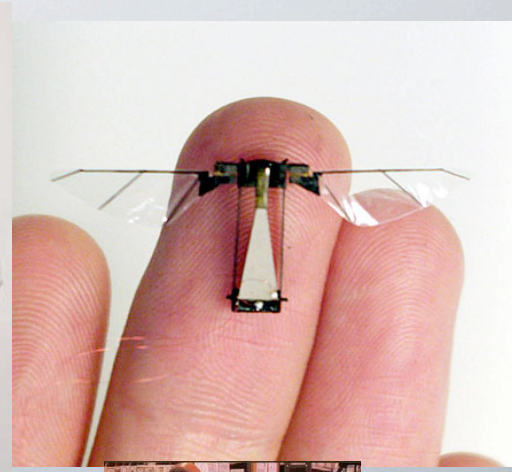
Manipulating physical environments

Coordinating robot teams

Reproducible systems

Usability and more ...





Autonomous robotics will enable individuals to manage physical environments by supervising teams of heterogeneous robotic devices

Challenges

Building robotic devices
(let's assume: "off the shelf")

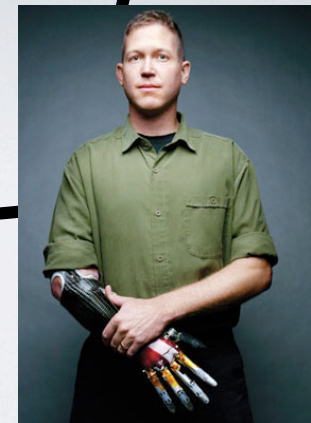
Programming robots

Manipulating physical environments

Coordinating robot teams

Reproducible systems

Usability and more ...



Autonomous robotics will enable individuals to manage physical environments by supervising teams of heterogeneous robotic devices

Challenges

Building robotic devices
(let's assume)

Programming robots

Manipulating physical environments

Coordinating robot teams

Reproducible systems

Usability and more ...

create controllers that accord w/ user's intended behavior



Autonomous robotics will enable individuals to manage physical environments by supervising teams of heterogeneous robotic devices

Challenges

Building robotic devices
(let's assume)

Programming robots

Manipulating physical environments

Coordinating robot teams

Reproducible systems

Usability and more ...



create controllers that accord w/ user's intended behavior



achieve desired physical effects, real world uncertainty



Autonomous robotics will enable individuals to manage physical environments by supervising teams of heterogeneous robotic devices

Challenges

Building robotic devices
(let's assume)

Programming robots

Manipulating physical environments

Coordinating robot teams

Reproducible systems

Usability and more ...



create controllers that accord w/ user's intended behavior



achieve desired physical effects, real world uncertainty

robots self-organize to achieve user's high-level goal



Autonomous robotics will enable individuals to manage physical environments by supervising teams of heterogeneous robotic devices

Challenges

Building robotic devices
(let's assume)

Programming robots

Manipulating physical environments

Coordinating robot teams

Reproducible systems

Usability and more ...

create controllers that accord w/ user's intended behavior

achieve desired physical effects, real world uncertainty

robots self-organize to achieve user's high-level goal

independent validation, avoid reinvention



Autonomous robotics will enable individuals to manage physical environments by supervising teams of heterogeneous robotic devices

Challenges

Building robotic devices
(let's assume)

create controllers that accord w/ user's intended behavior

Programming robots

achieve desired physical effects, real world uncertainty

Manipulating physical environments

robots self-organize to achieve user's high-level goal

Coordinating robot teams

independent validation, avoid reinvention

Reproducible systems

can society use our systems?

Usability and more ...



ROBOT PROGRAMMING

Teleoperation

- remote human control

Programming

- explicitly coded controller

Learning

- controller implicit in experience/ demonstration

ROBOT PROGRAMMING

Teleoperation

- remote human control

Programming

- explicitly coded controller

Learning

- controller implicit in experience/
demonstration

[Miller, Jenkins, et al. 2004]



- lacks autonomy, fancy RC device

ROBOT PROGRAMMING

Teleoperation

- remote human control

- + Established development paradigm
- + “Write local, run global”
- Uncertainty across tasks, users, and environments
- Limited to technically skilled programmers

Programming

- explicitly coded controller

Learning

- controller implicit in experience/ demonstration

- + Programming skills not required
- + User customization to tasks, environments
- Significantly less matured
- Algorithmically, limited range of automata
- Data-wise, collection is difficult

ROBOT PROGRAMMING

Teleoperation

- remote human control

Programming

- explicitly coded controller

Learning

- controller implicit in experience/ demonstration

Focus of this course

- decision making
- perception
- motion control

WARNING!!!

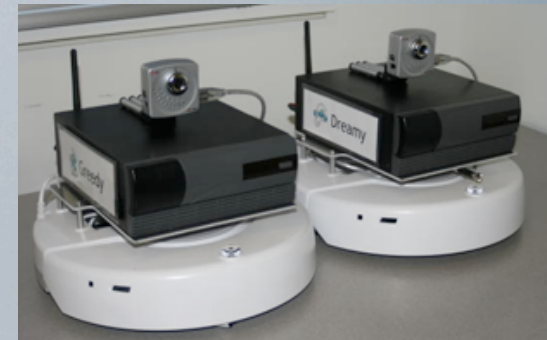
ROBOTICS IS HARD

- Robotics is unlike traditional deterministic comp. sci.
 - state is physical world not bits in memory
 - uncertainty, relatively nondeterministic behavior
- Robotics is a “systems science”
 - integrated implementation of various concepts and systems
- Experimentation and writing is essential for validating your work
- You have been warned



Version 2 -
iRobot Create,
Asus EEE subnotebook,
Player middleware

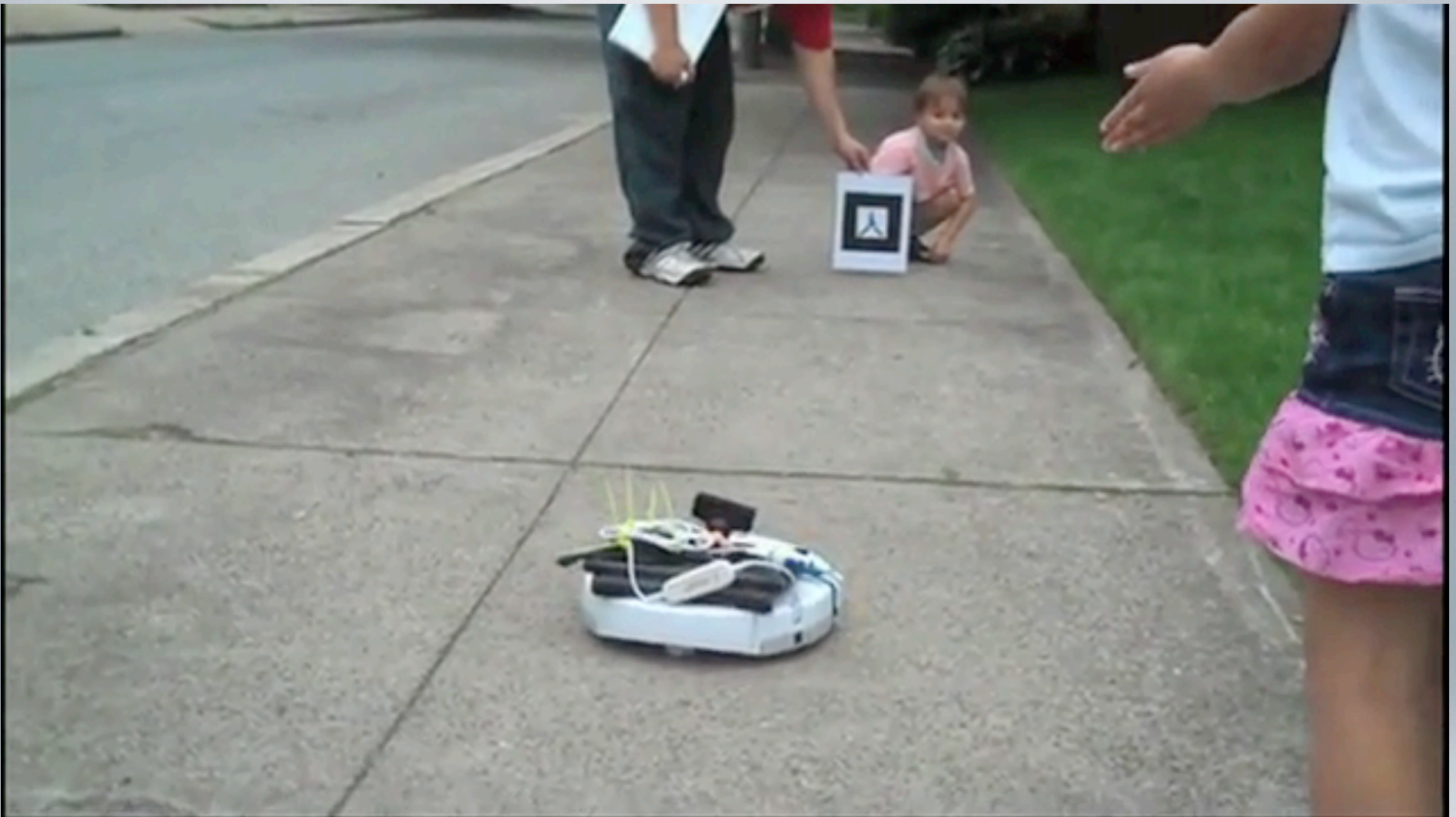
Current version -
iRobot Create,
PS3 cameras
Lenovo subnotebooks,
ROS middleware



Version 1 - iRobot
Create, Mini ITX

ROBOT PLATFORMS

<http://www.youtube.com/watch?v=88zR6IC7S0g>



AR FOLLOWING IN ROS

ROS primarily supports Python and C++
Javascript available

<http://www.youtube.com/watch?v=mKmqgVUbQQM>

Software installation: <http://code.google.com/p/brown-ros-pkg/>



The screenshot shows a web browser window with the URL http://robotics.cs.brown.edu/projects/player_icreate/. The page title is "Setting up Player 2.1.1: Asus eeePC and iCreate Robot". The author information lists Lisa Miller, Aysun BascetinCelik, and Odest Chadwicke Jenkins. Below the title is a "Table of Contents" with the following items:

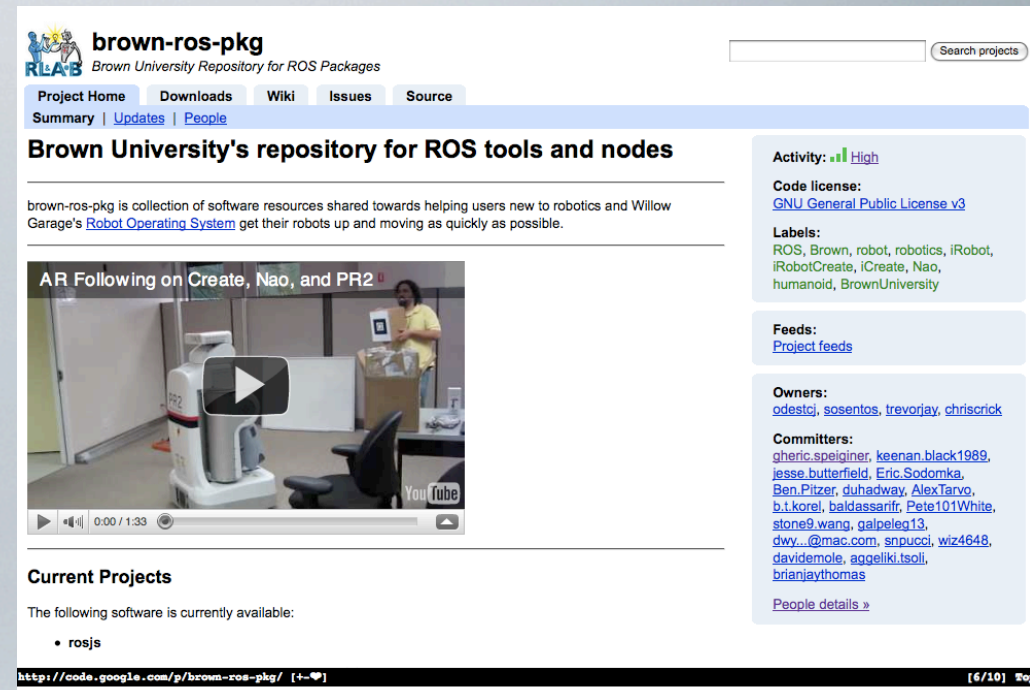
- Getting Player onto the eeePC
 - Install Standard OS
 - Install Player 2.1.1
- Running eeePC/Player on iCreate
 - Hardware Setup
 - Running and Testing Player

The browser's search bar contains the text "rein".

Hardware instructions:

http://robotics.cs.brown.edu/projects/player_icreate/

“CREATE” YOUR OWN



The screenshot shows the project page for "brown-ros-pkg" on Code.google.com. The page title is "Brown University's repository for ROS tools and nodes". The description states: "brown-ros-pkg is collection of software resources shared towards helping users new to robotics and Willow Garage's [Robot Operating System](#) get their robots up and moving as quickly as possible." Below the description is a video player with the title "AR Following on Create, Nao, and PR2". The video player shows a person standing next to a robot. The page also includes a "Current Projects" section with the following software currently available:

- rosjs

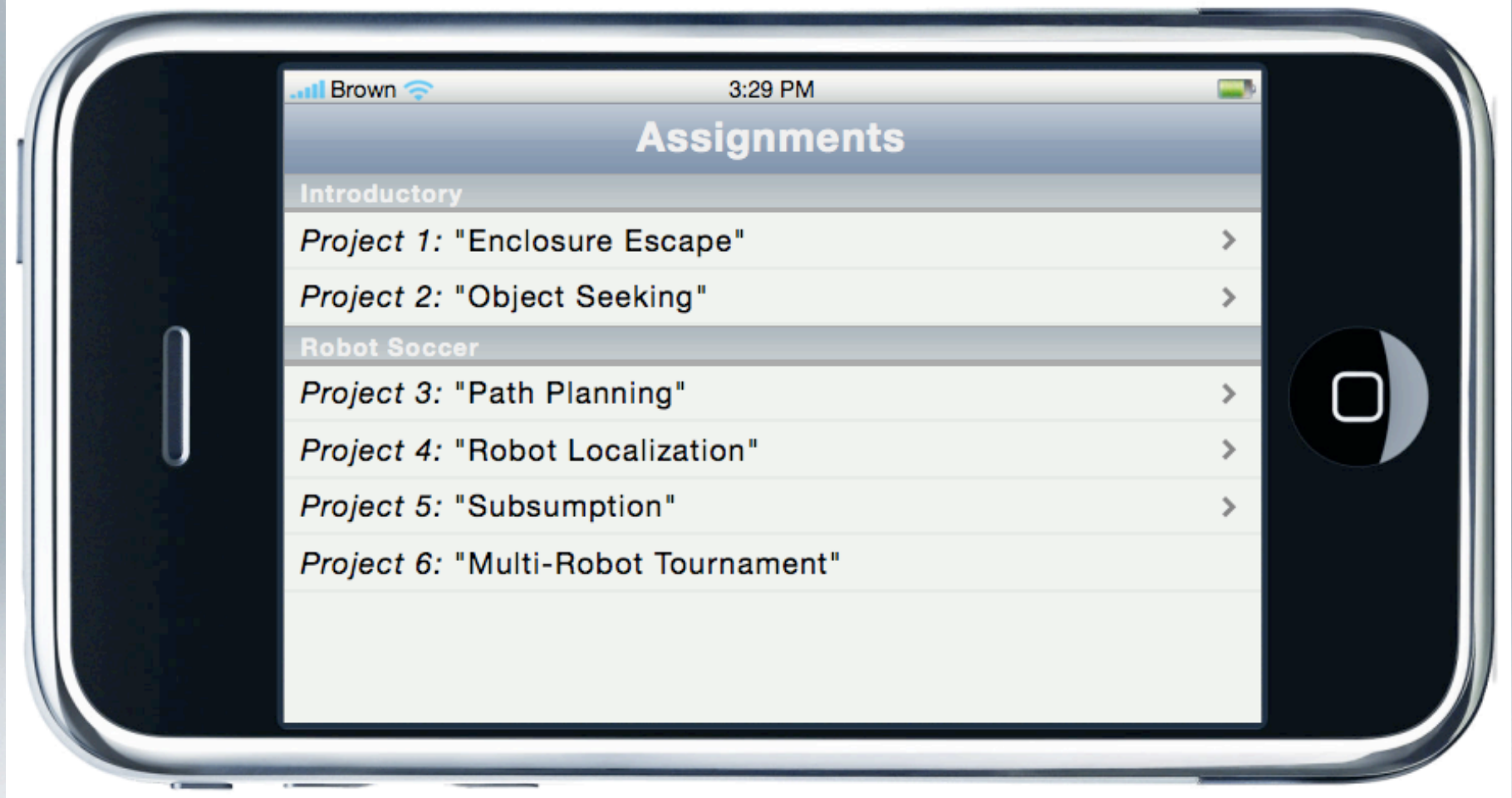
The page footer shows the URL <http://code.google.com/p/brown-ros-pkg/> and a page number of 6/10.

CSci1480: Building Intelligent Robots



<http://www.cs.brown.edu/courses/cs148>

CSci1480: Building Intelligent Robots

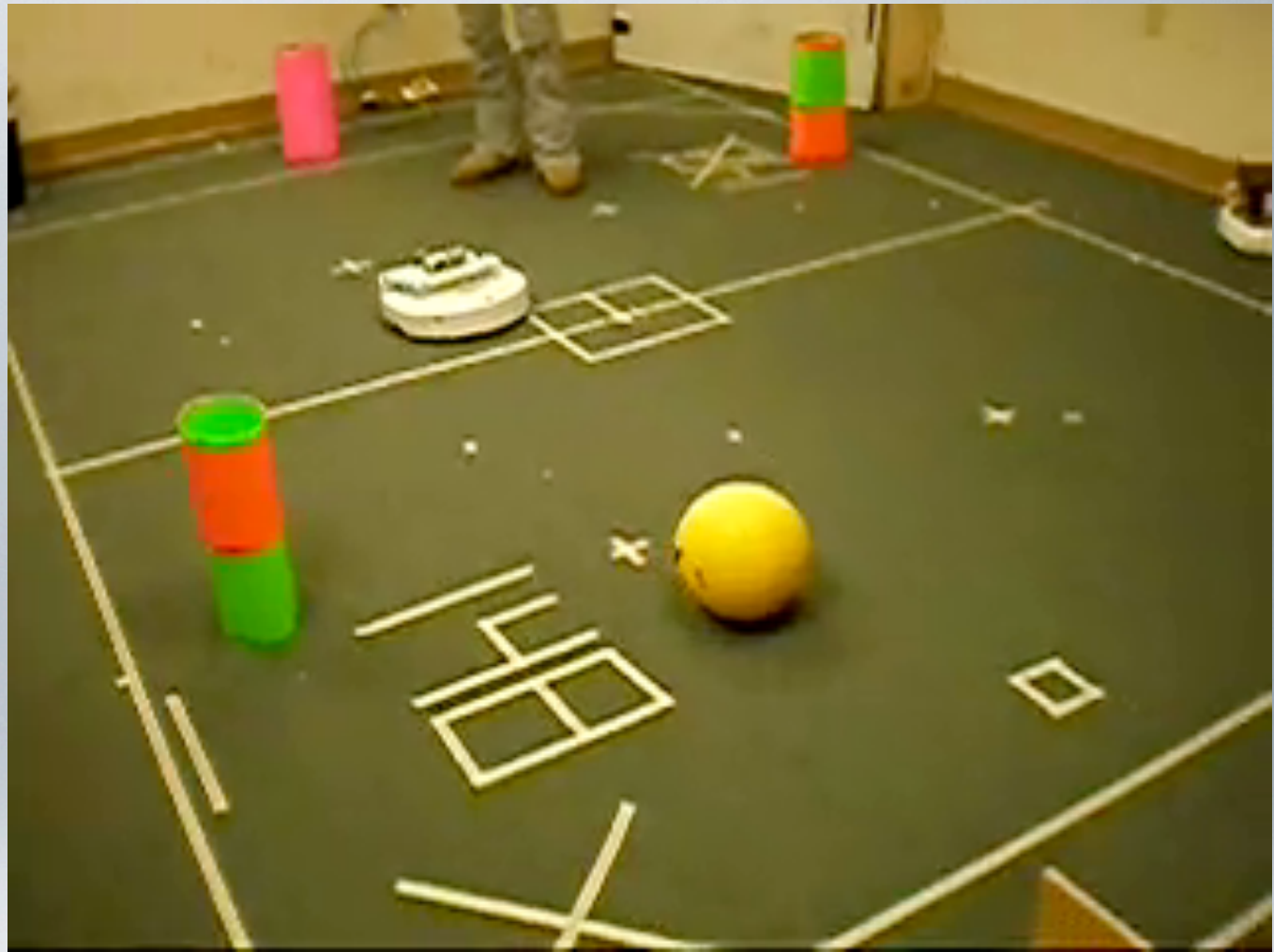
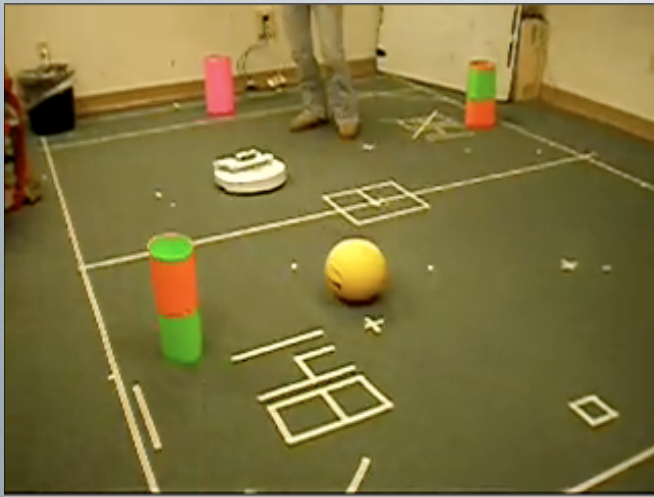


<http://www.cs.brown.edu/courses/cs148>



ENCLOSURE ESCAPE

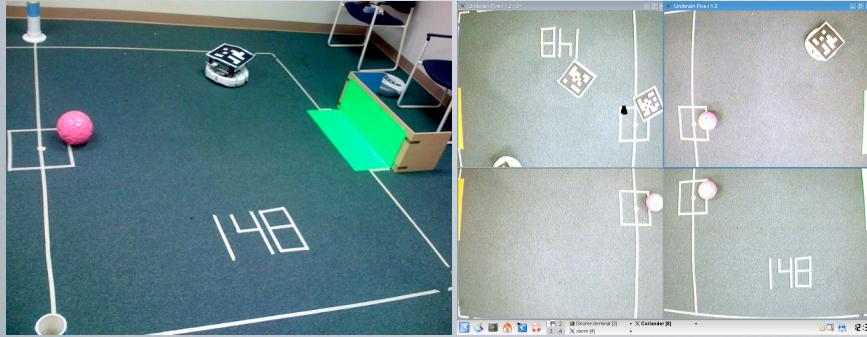
<http://www.youtube.com/watch?v=VBzXDrz8rMI>



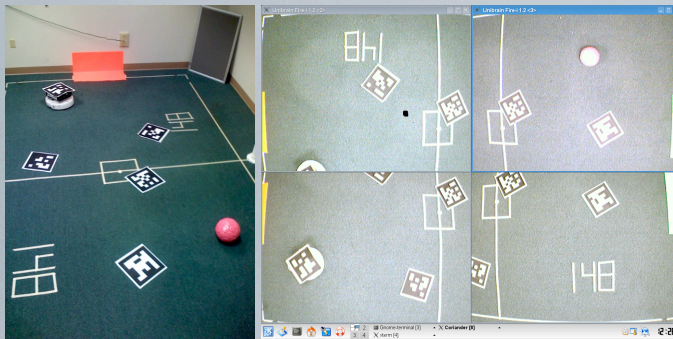
OBJECT SEEKING

<http://www.youtube.com/watch?v=-hOA0jMUggg>

score goal



seek
ball



challenges (graded)

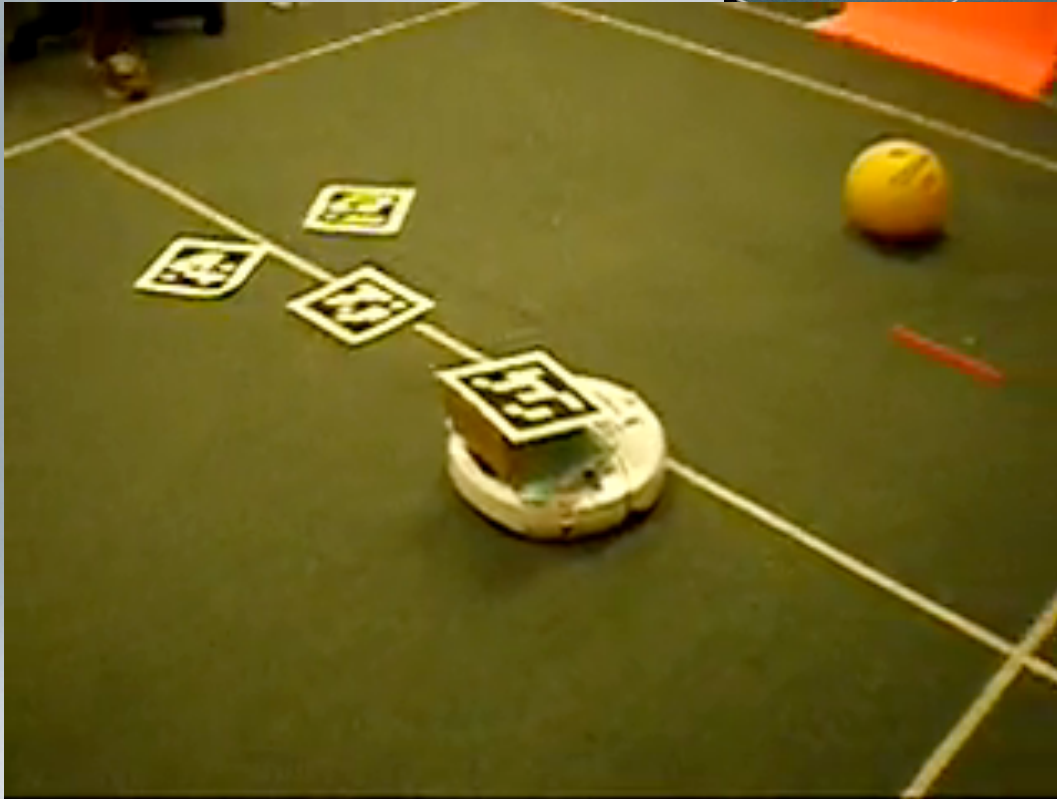
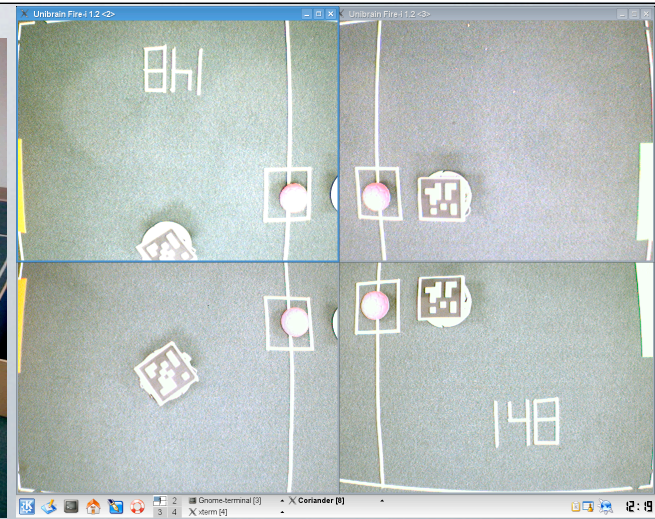


head-to-head
competition
(ungraded)

ROBOT SOCCER

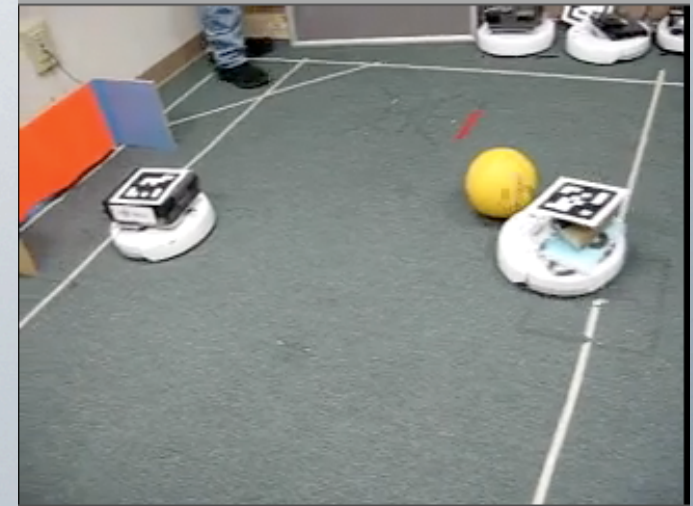
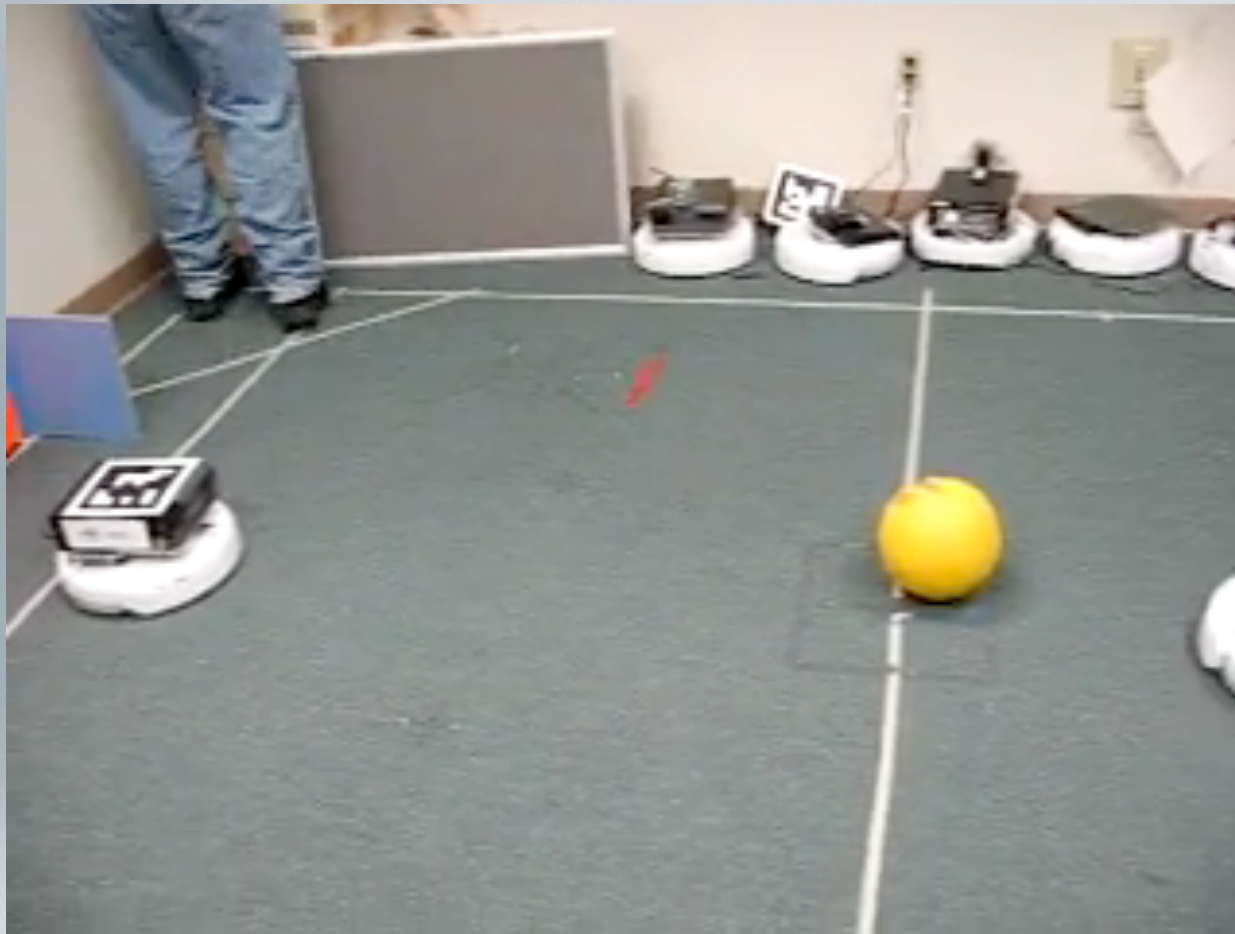
<http://www.cs.brown.edu/courses/cs148/documents/missive.pdf>

[toonami.avi](#)



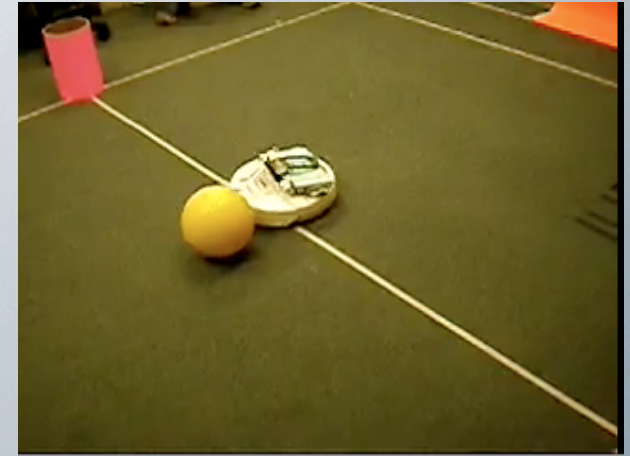
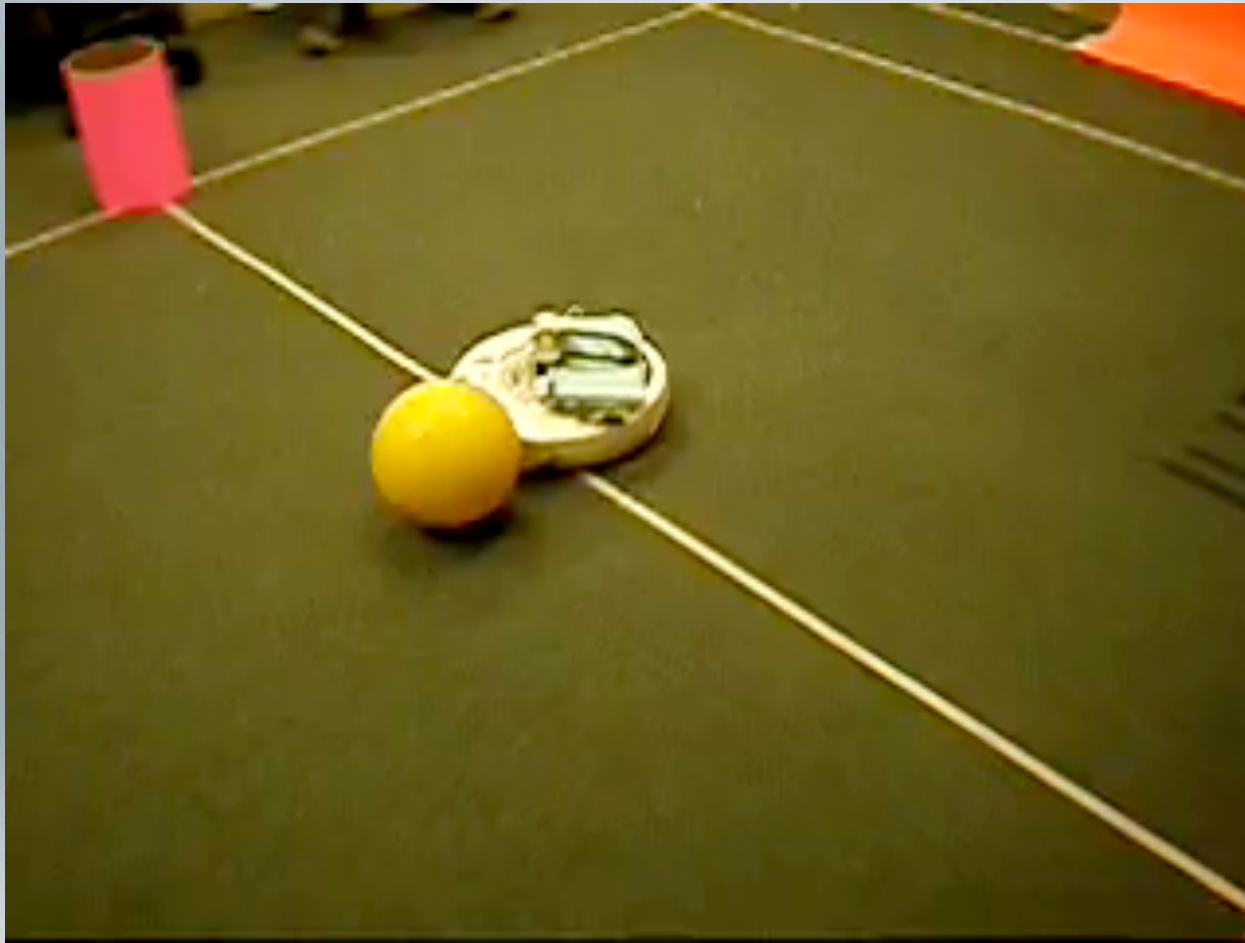
PATH PLANNING

<http://www.youtube.com/watch?v=2Z2RyeofsZg>



PATH PLANNING (MATCH)

<http://www.youtube.com/watch?v=UIoThYPW5AU>



ROBOT LOCALIZATION

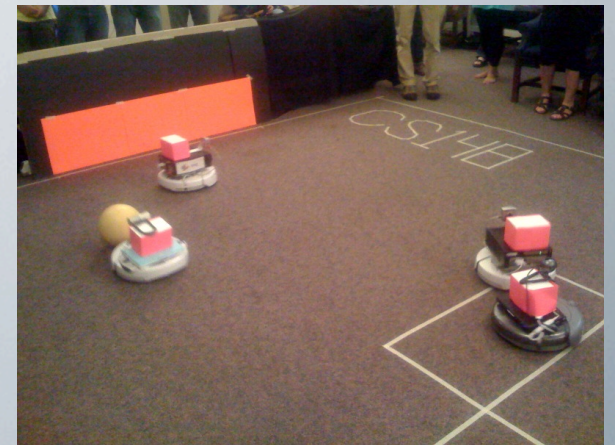
<http://www.youtube.com/watch?v=WpJrZ89IW5k>



SUBSUMPTION (MATCH)

http://www.youtube.com/watch?v=PQGSJjK9OLw&feature=channel_page

Fall 2007 - CIT Atrium



Spring 2009 - Petteruti Lounge



Fall 2009 - Andrews Hall

2-ON-2 (PUBLIC SPACE)

<http://www.youtube.com/watch?v=Df23qNAy9Jc>

assignments grading:
50% implementation, 50% written report

not just working project,
understand when it works

Assignment 1: Enclosure Escape

CS148

Erich C. Deise (edeise)

Introduction:

The task given by assignment 1 was to create a Player controller capable of directing a SmURV robot to reactively respond to obstacles in its path. Specifically, we are given the task of constructing a control program that results in the SmURV escaping from enclosures of arbitrary geometric configurations utilizing only information provided by bumper-sensors. We conjecture that by implementing a very simple *nondeterministic* reactive control structures that the Smurv will be capable of negotiating (I.e., escaping from) most *continuous* enclosed spaces.

Approach and Methods:

Our general approach was to investigate three enclosure-escape algorithms – *random walk*, *simple (deterministic) wall-following* and *simple randomized (nondeterministic) wall-following*.

Random walk is implemented as a finite state machine in which the controller directs the SmURV to enter an initial forward state. The robots proceeds linearly until its bumper sensor is activated at which time it transitions to a second state whereby the SmURV reverses for a set amount of time. Once there, it transitions to a third state in which it rotates unidirectionally for a random amount of time. Once complete the initial state is revisited.

The second algorithm considered was a *simple deterministic wall – following* procedure. Implemented as a finite state machine the controller directs the SmURV to enter an initial –*forward*– state. Once a bumper strike is recorded the machine transitions to second state whereby the robot reverses for a fixed amount of time and speed. Once complete the third state iss invoked in which

We then tested the nondeterministic wall following algorithm in a triangular shaped enclosure (an approximation of an isosceles triangle with the exit located at the corner with the narrowest angle as depicted above. To determine if the randomized wall-following algorithm could be relied upon to perform at better than chance levels, the random-walk algorithm was selected as a suitable control condition. Elapsed time required for the random-walk controlled SmURV to negotiate the enclosure were taken as the base for comparison. See table 3.a and 3.b for results.

Tables 3.a and 3.b present the results for the random-walk and randomized wall-following algorithms in the triangular enclosure space respectively.

Trial	Elapsed time (m:sec)
1	Failed to escape in < 5 minutes
2	1:25
3	1:05

Table 3.a *Random walk*

Trial	Elapsed time (m:sec)
1	0:26
2	1:14
3	2:46

Table 3.b *Randomized wall-following*

Since no modifications were made to the iRobot plant nor the mini-ITX (C.f. <http://robotics.cs.brown.edu/projects/smurv> for specifications and

assignments grading: 50% implementation, 50% written report

Assignment 1: Enclosure Escape

CS148

Erich C. Deise (edeise)

Introduction:

The task given by assignment 1 was to create a Player controller capable of directing a SmURV robot to reactively respond to obstacles in its path. Specifically, we are given the task of constructing a control program that results in the SmURV escaping from enclosures of arbitrary geometric configurations utilizing only information provided by bumper-sensors. We conjecture that by implementing a very simple *nondeterministic* reactive control structures that the Smurv will be capable of negotiating (I.e., escaping from) most *continuous* enclosed spaces.

Approach and Methods:

Our general approach was to investigate three enclosure-escape algorithms – *random walk*, *simple (deterministic) wall-following* and *simple randomized (nondeterministic) wall-following*.

Random walk is implemented as a finite state machine in which the controller directs the SmURV to enter an initial forward state. The robots proceeds linearly until its bumper sensor is activated at which time it transitions to a second state whereby the SmURV reverses for a set amount of time. Once there, it transitions to a third state in which it rotates unidirectionally for a random amount of time. Once complete the initial state is revisited.

The second algorithm considered was a *simple deterministic wall – following* procedure. Implemented as a finite state machine the controller directs the SmURV to enter an initial –*forward*– state. Once a bumper strike is recorded the machine transitions to second state whereby the robot reverses for a fixed amount of time and speed. Once complete the third state iss invoked in which

We then tested the nondeterministic wall following algorithm in a triangular shaped enclosure (an approximation of an isosceles triangle with the exit located at the corner with the narrowest angle as depicted above. To determine if the randomized wall-following algorithm could be relied upon to perform at better than chance levels, the random-walk algorithm was selected as a suitable control condition. Elapsed time required for the random-walk controlled SmURV to negotiate the enclosure were taken as the base for comparison. See table 3.a and 3.b for results.

Tables 3.a and 3.b present the results for the random-walk and randomized wall-following algorithms in the triangular enclosure space respectively.

Trial	Elapsed time (m:sec)
1	Failed to escape in < 5 minutes
2	1:25
3	1:05

Table 3.a *Random walk*

multi-trial experiments

Trial	Elapsed time (m:sec)
1	0:26
2	1:14
3	2:46

Table 3.b *Randomized wall-following*

Since no modifications were made to the iRobot plant nor the mini-ITX (C.f. <http://robotics.cs.brown.edu/projects/smurv> for specifications and

CSci1480: Building Intelligent Robots



cs 148 google calendar

missive, collaboration policy, guest lectures

<http://www.cs.brown.edu/courses/cs148>

CSci1480: Building Intelligent Robots

my lecture slides
with other tutorials



<http://www.cs.brown.edu/courses/cs148>

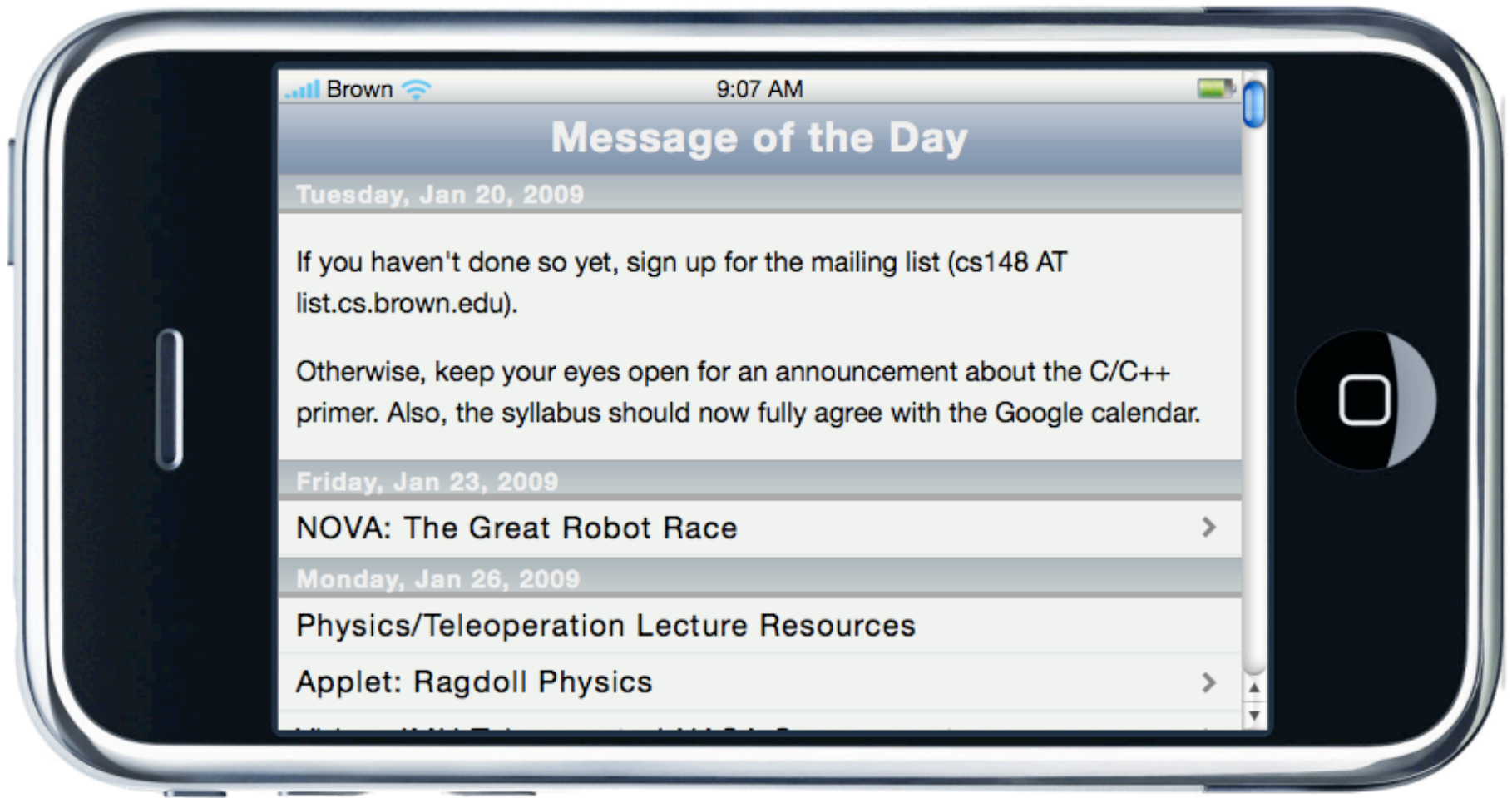
CSci1480: Building Intelligent Robots

“message of the day”
random interesting bits



<http://www.cs.brown.edu/courses/cs148>

CSci1480: Building Intelligent Robots



<http://www.cs.brown.edu/courses/cs148>

sign up for the mailing list! enjoy the links.

CSci1480: Building Intelligent Robots



<http://www.cs.brown.edu/courses/cs148>

CSci1480: Building Intelligent Robots



course staff

<http://www.cs.brown.edu/courses/cs148>

COURSE STAFF

Instructor

Chad Jenkins (cjenkins)



Head Teaching Assistant

Pete White (pjwhite)

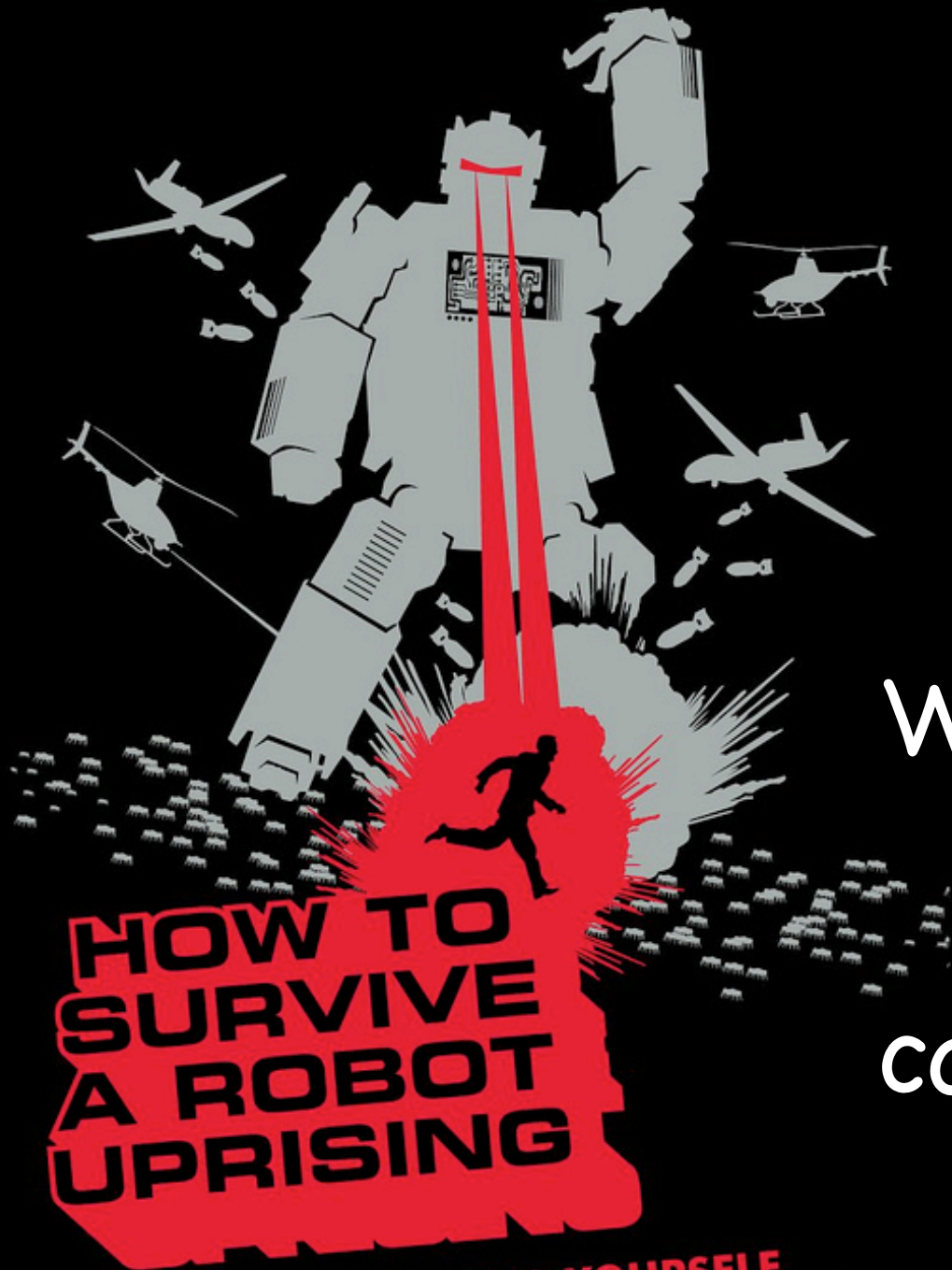


Teaching Assistants

Francois Baldassari (fbaldass)

Paul O'Leary McCann (polm)





Topic 1

What are robots?

... and are they coming to get us?

**HOW TO
SURVIVE
A ROBOT
UPRISING**

**TIPS ON DEFENDING YOURSELF
AGAINST THE COMING REBELLION**