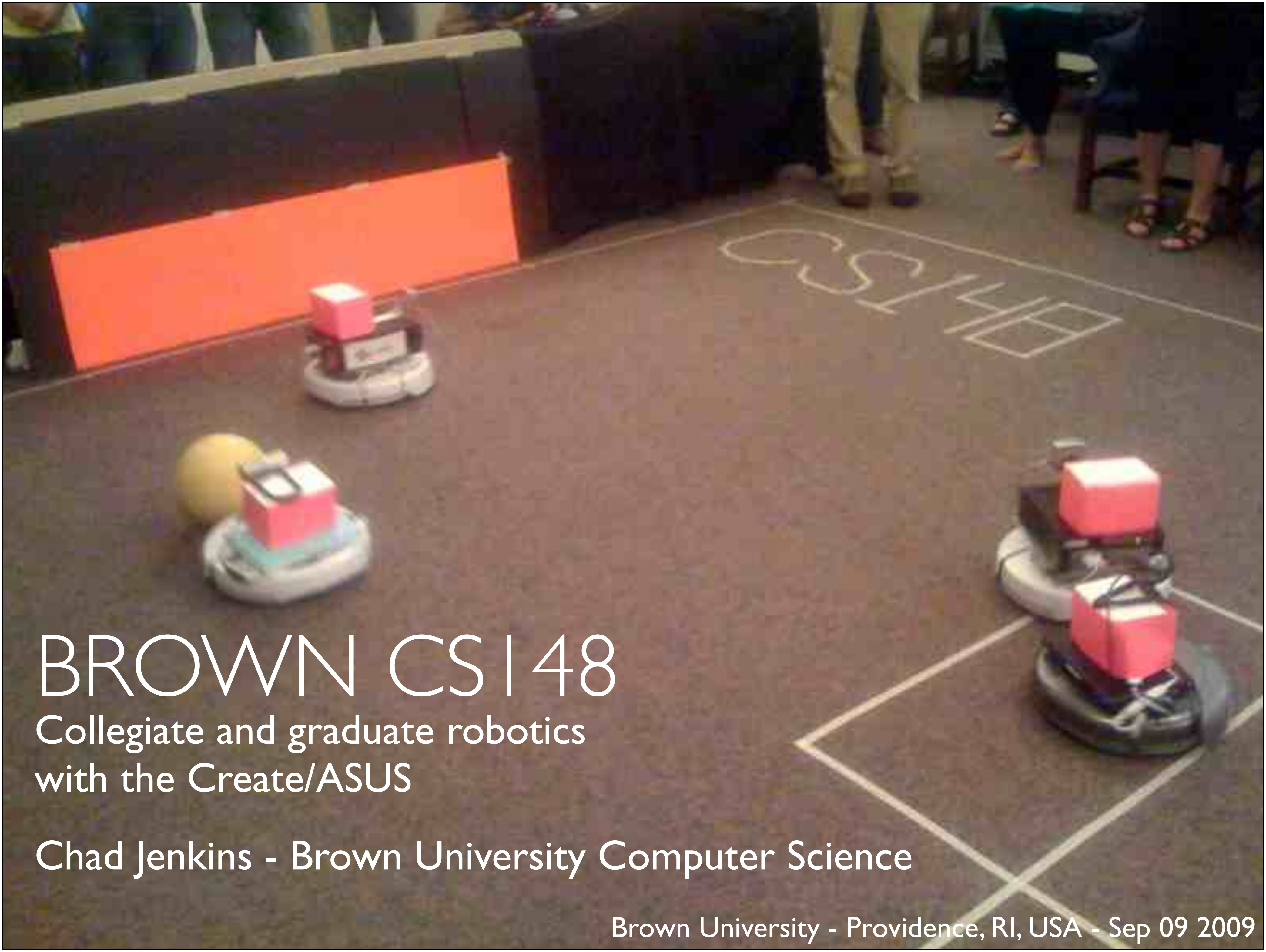


BROWN CS148

Collegiate and graduate robotics with the Create/ASUS

Chad Jenkins - Brown University Computer Science

Brown University - Providence, RI, USA - Sep 09 2009

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/will:
  - developing tools to enhance physical human productivity
    - healthcare, exploration, service, manufacturing, ...
    - 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

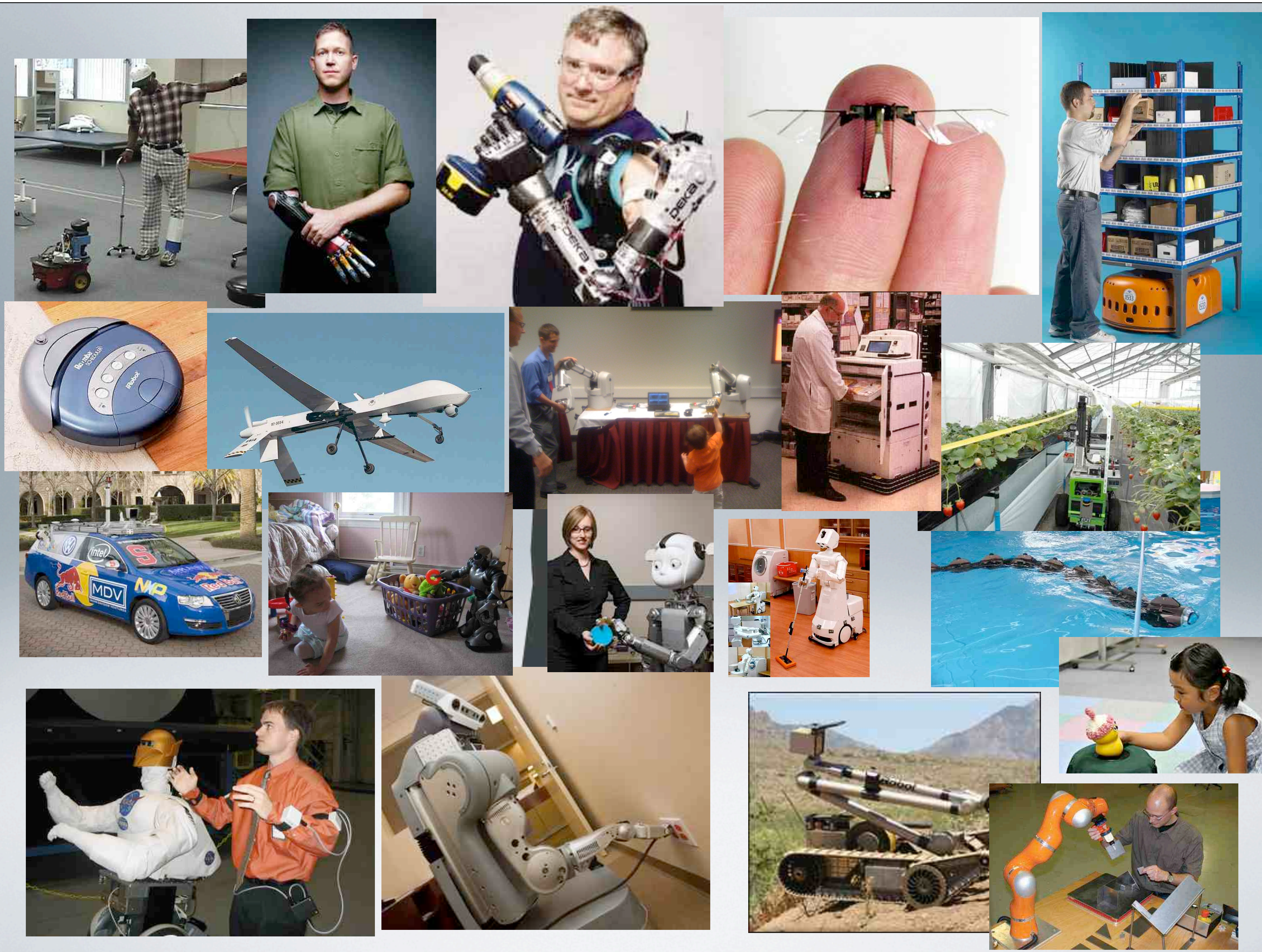
Autonomous robotics will enable individuals to manage physical environments by supervising teams of heterogeneous robotic devices

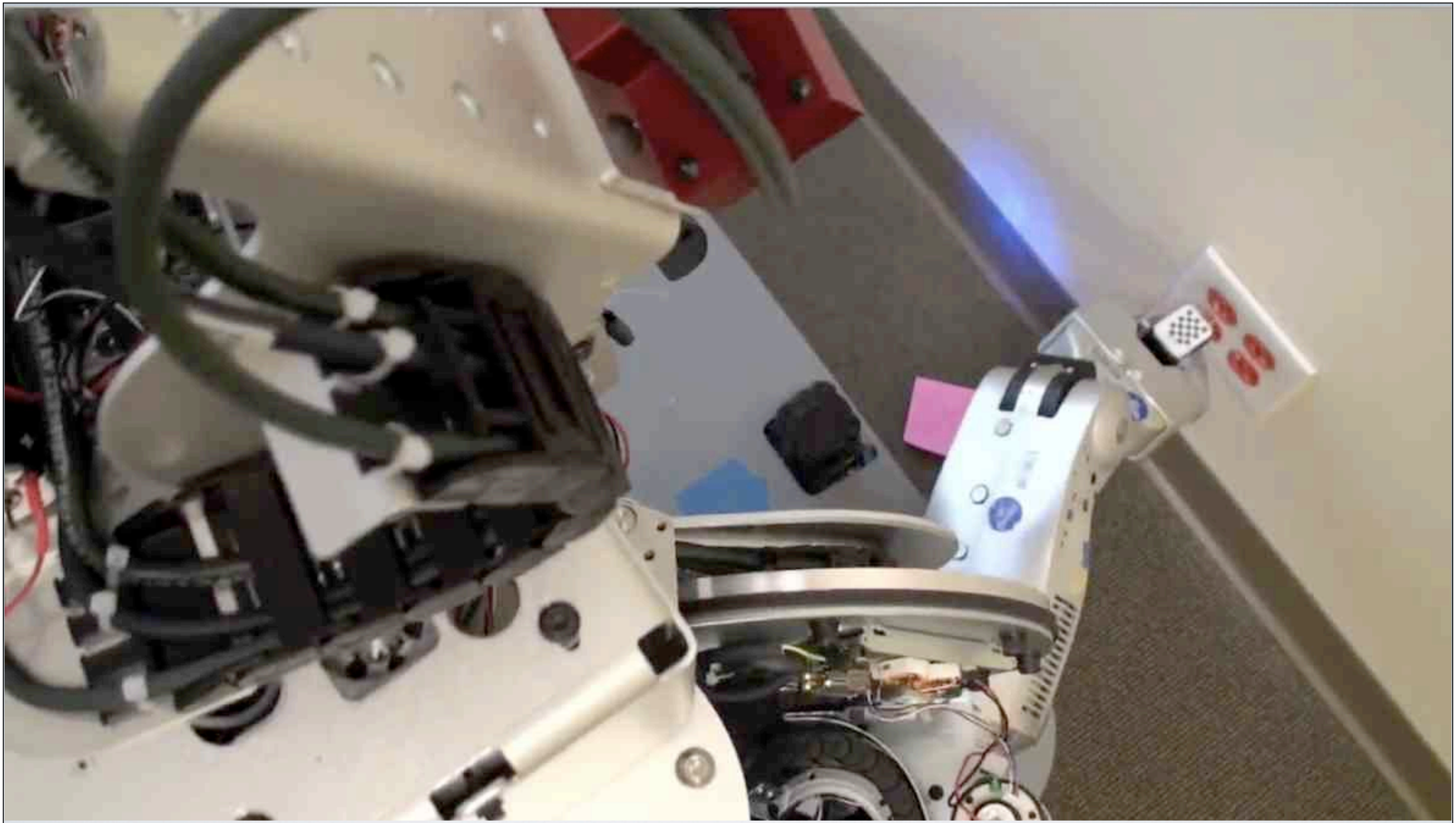
Autonomous robotics will enable individuals to manage physical environments by supervising teams of heterogeneous robotic devices

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
- ☐ and more ...


iRobot
DEKA
intel
MDV
Red Bull



PR2 MILESTONE 2

http://www.willowgarage.com/blog/2009/06/03/watch-milestone-2

PR2 MILESTONE 2

http://www.willowgarage.com/blog/2009/06/03/watch-milestone-2

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
- ☐ and more ...

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

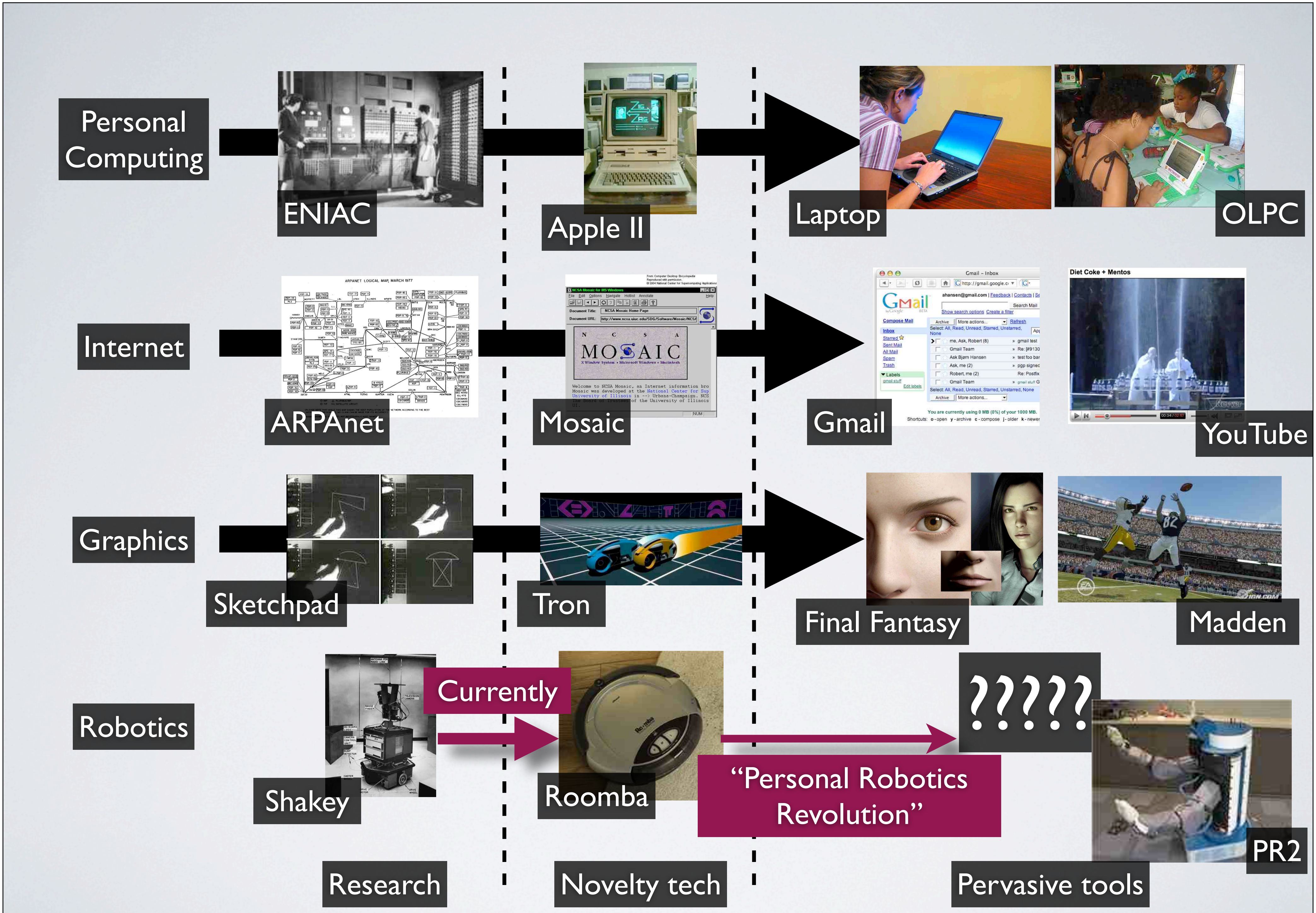
Challenges

- ✓ Building robotic devices (let's assume)
- ☐ Programming robots (this course)
- ☐ Manipulating physical environments
- ☐ Coordinating robot teams
- ☐ and more ...

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


Personal Computing
ENIAC
Apple II
Laptop
OLPC
Internet
ARPAnet
Mosaic
Gmail
YouTube
Graphics
Sketchpad
Tron
Final Fantasy
Madden
Robotics
Shakey
Currently
Roomba
"Personal Robotics Revolution"
?????
PR2
Research
Novelty tech
Pervasive tools

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

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

everything CS148-related will be accessible from here

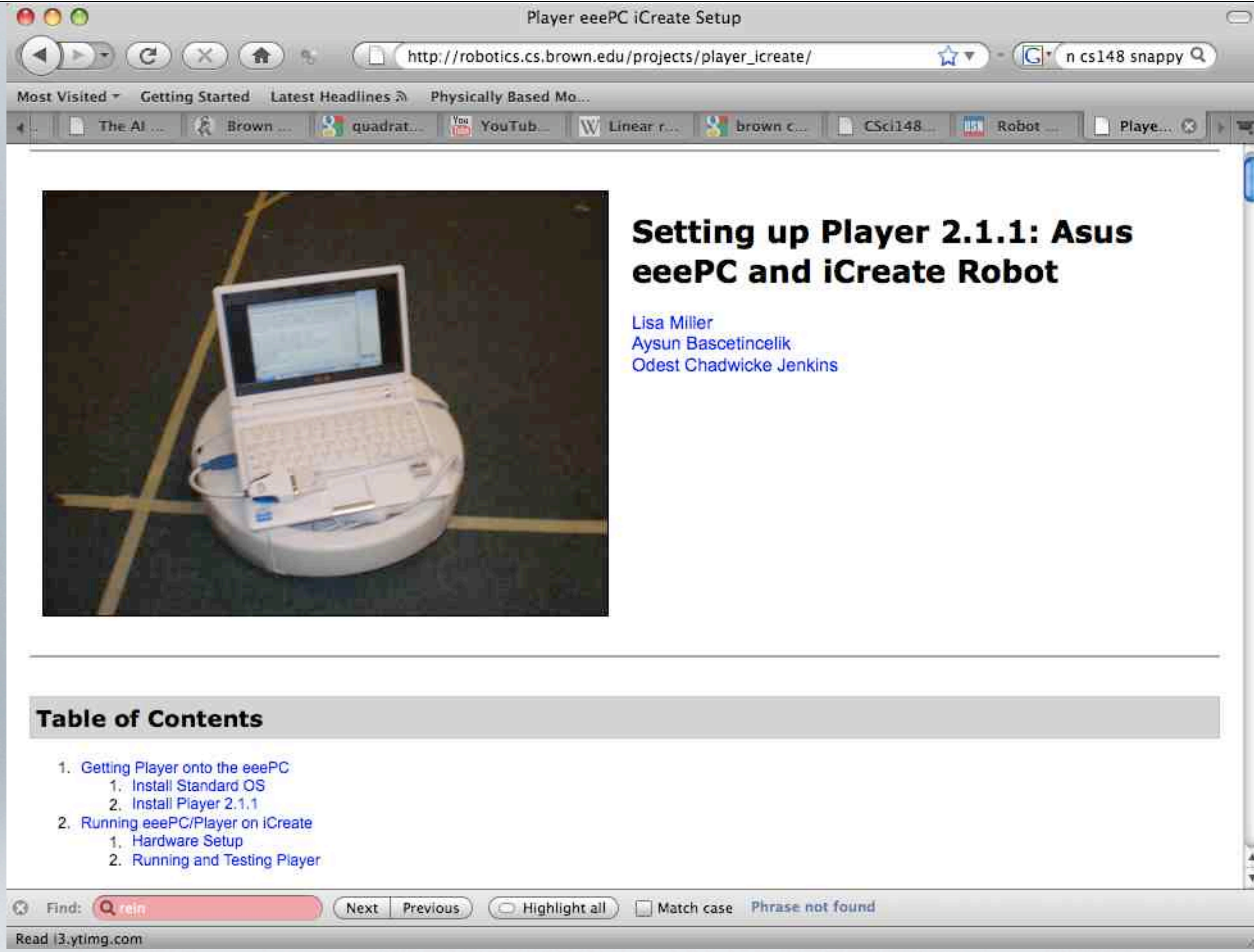
Version 1 - iRobot Create, Mini ITX

Current version - iRobot Create, Asus EEE subnotebook

Player/Stage/Gazebo middleware (client/server)

ROBOT PLATFORMS

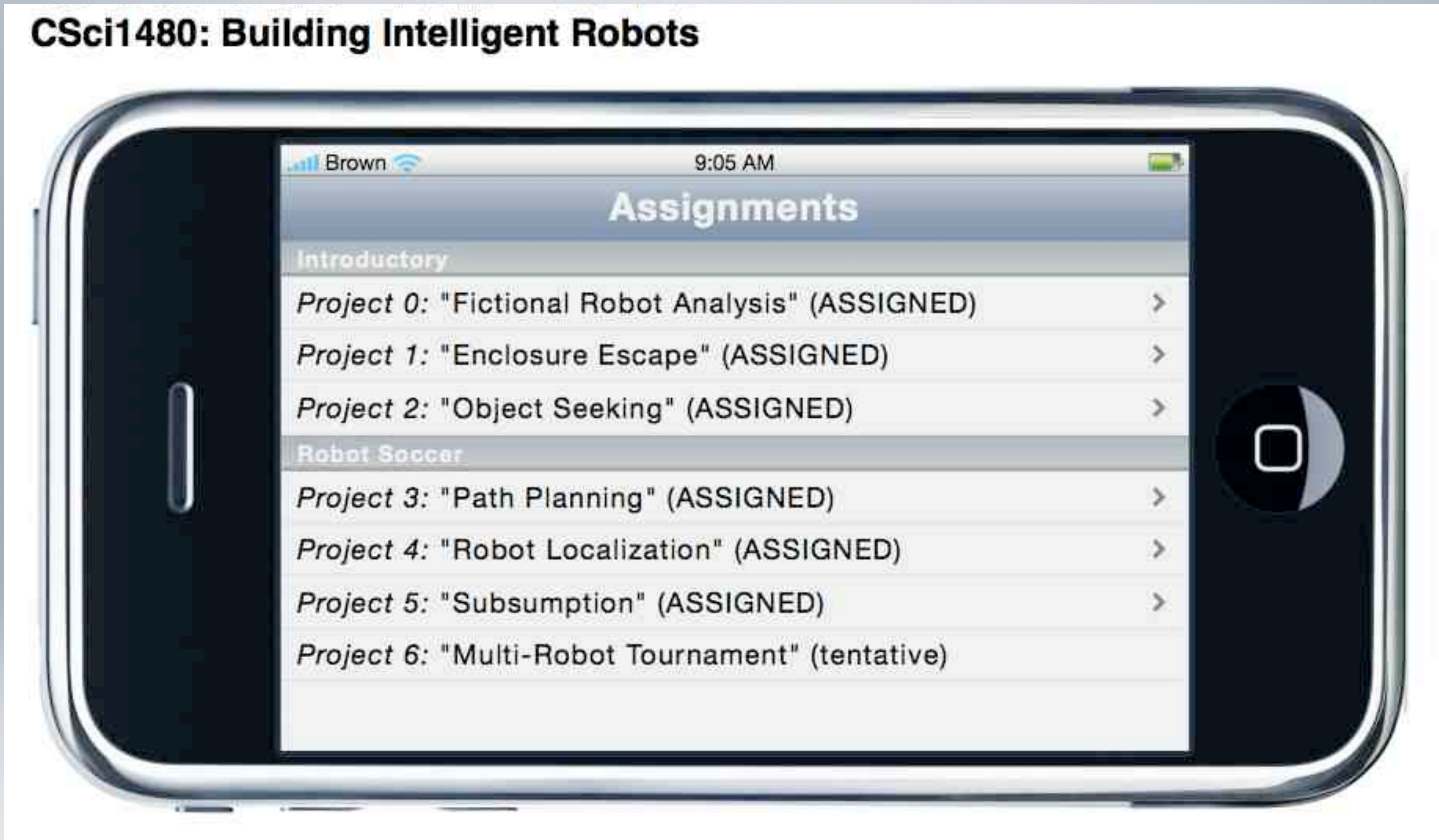
http://www.youtube.com/watch?v=88zR6IC7S0g



“CREATE” YOUR OWN

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

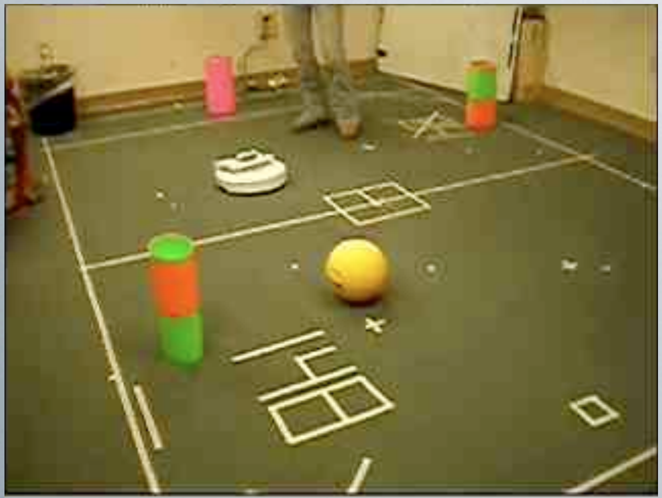
http://www.cs.brown.edu/courses/cs148



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

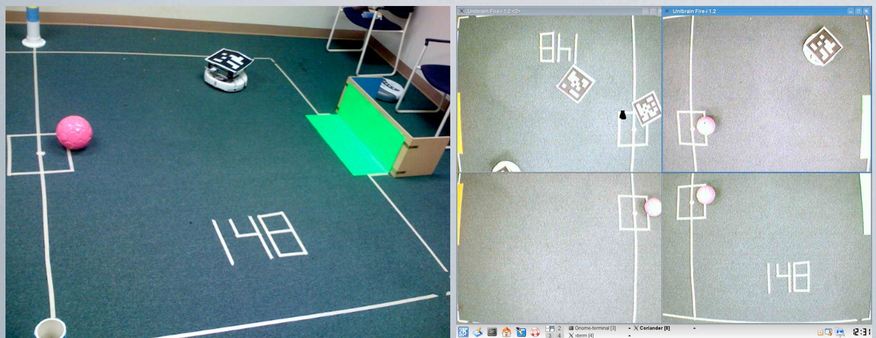
ENCLOSURE ESCAPE

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

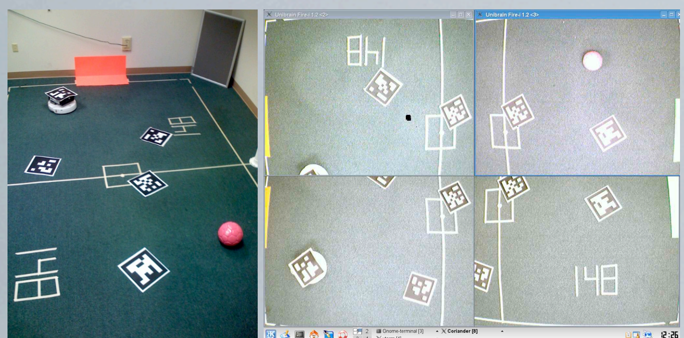


OBJECT SEEKING

http://www.youtube.com/watch?v=-hOA0jMUggg score goal



seek ball



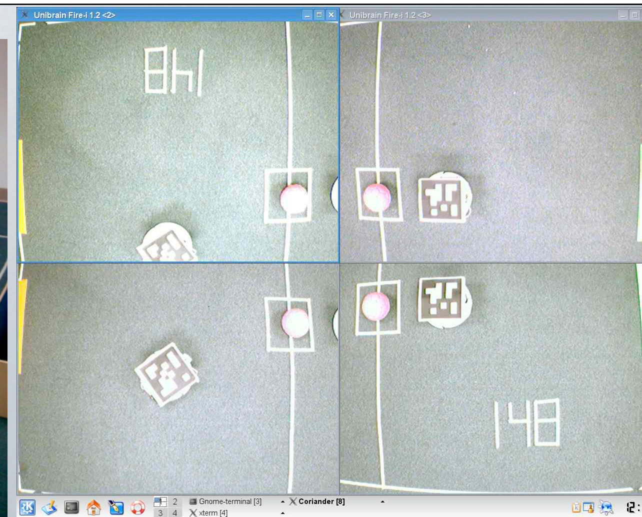
challenges (graded)

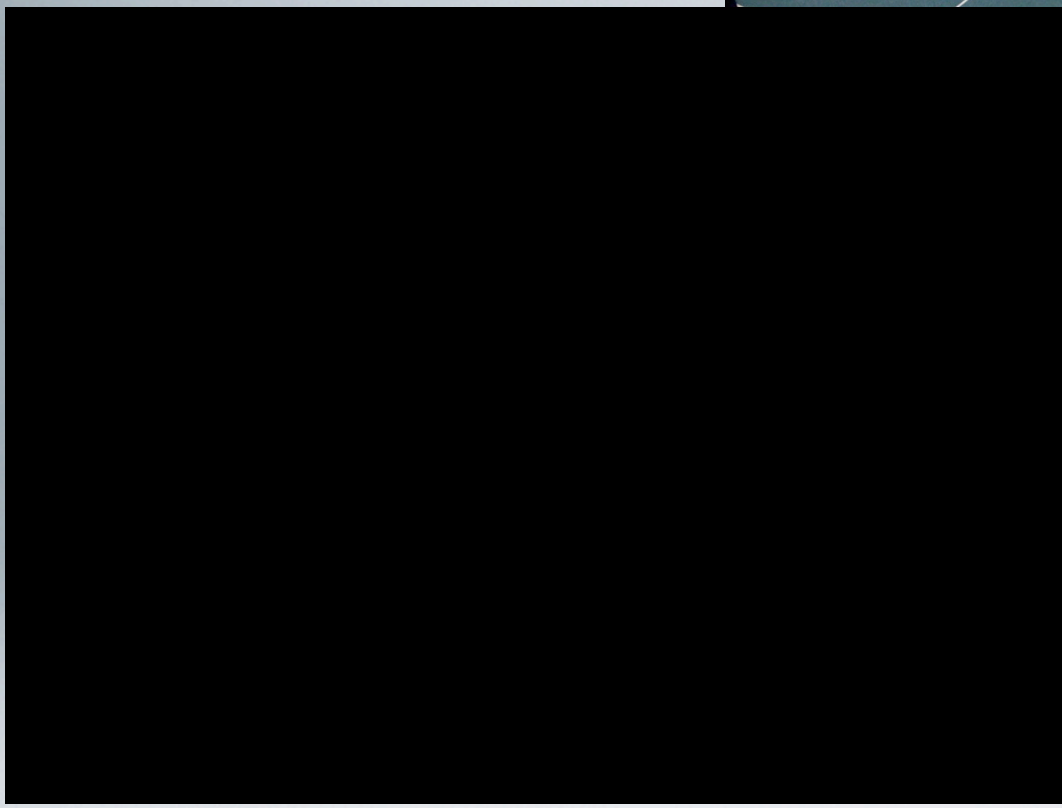
head-to-head competition (ungraded)

ROBOT SOCCER

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

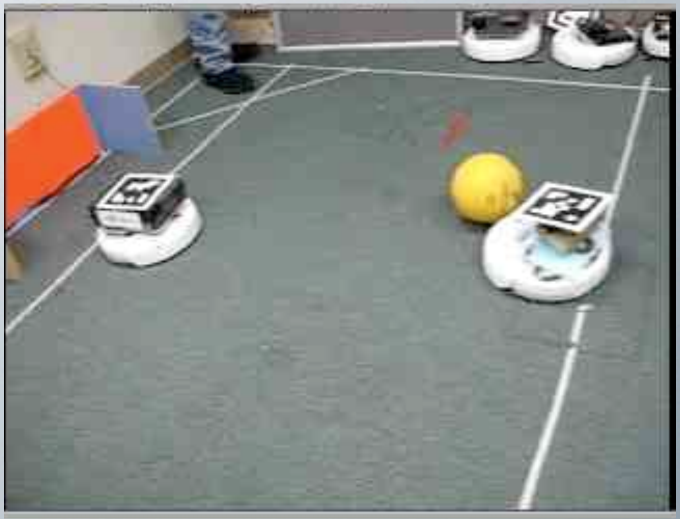
toonami.avi





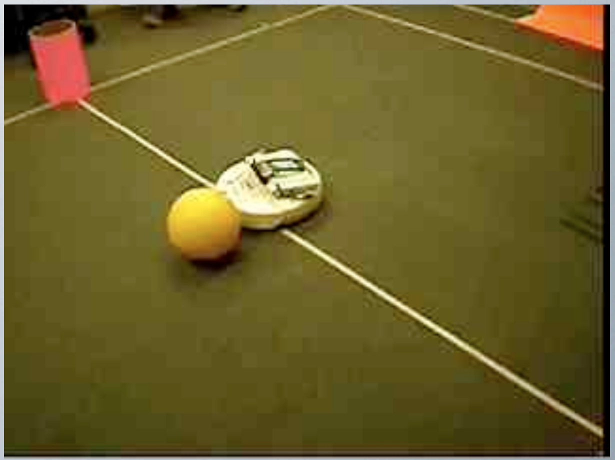
PATH PLANNING

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



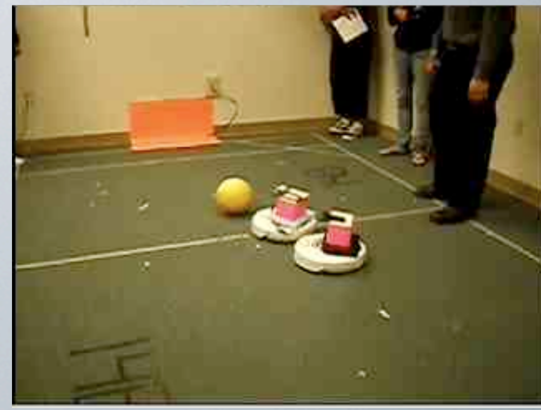
PATH PLANNING (MATCH)

http://www.youtube.com/watch?v=U1oThYPW5AU



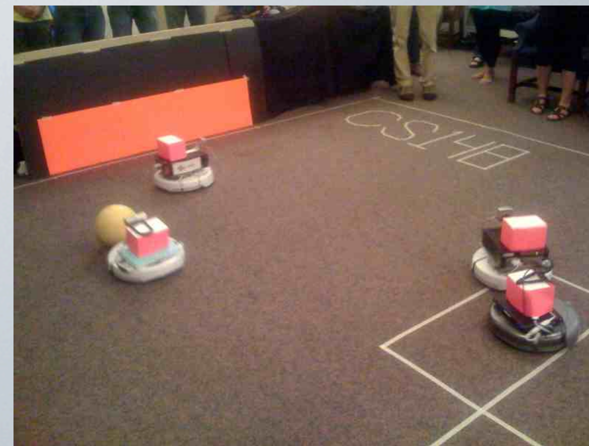
ROBOT LOCALIZATION

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



SUBSUMPTION (MATCH)

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



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***

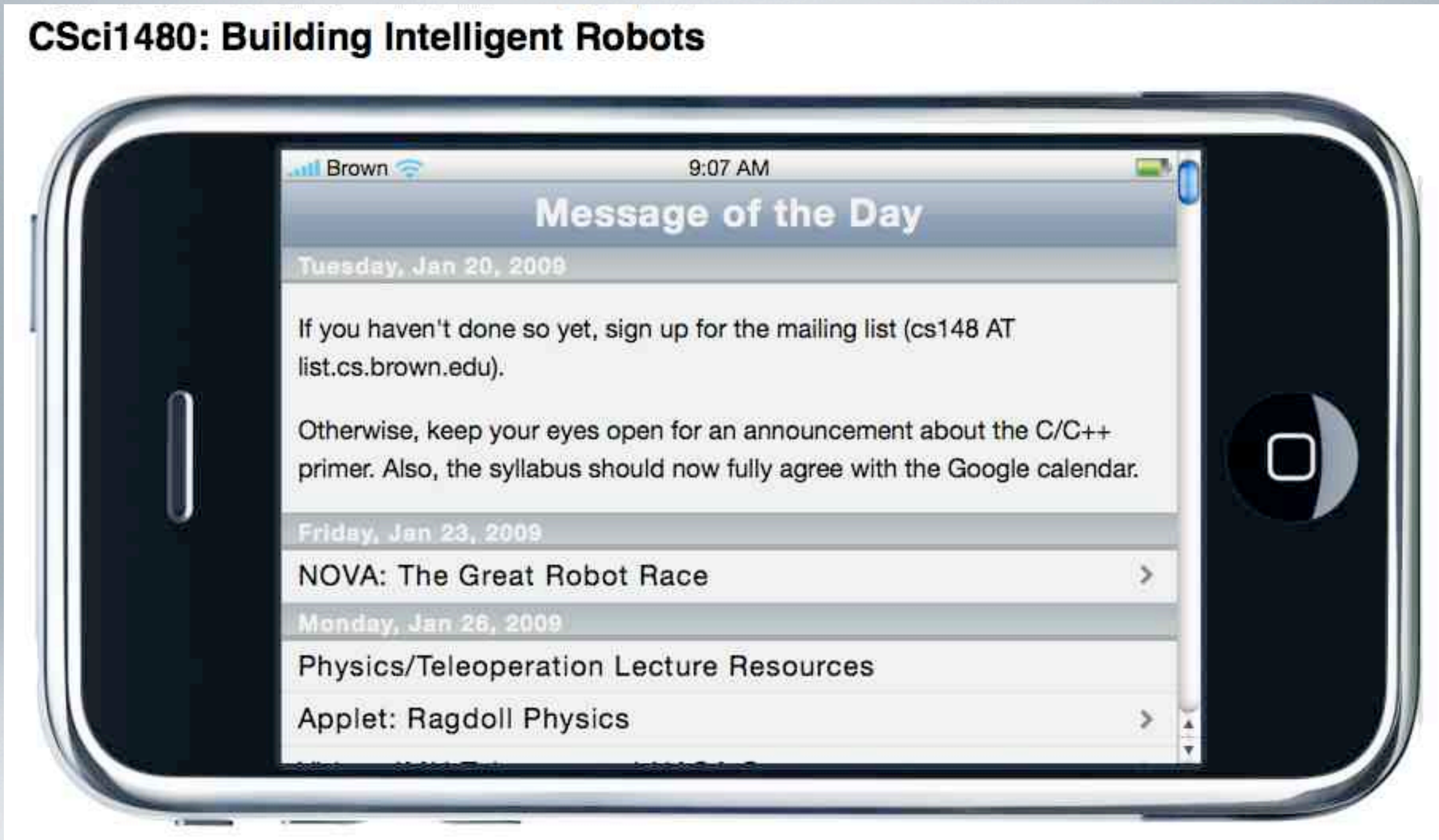
multi-trial experiments

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 essentially my lecture notes with videos

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



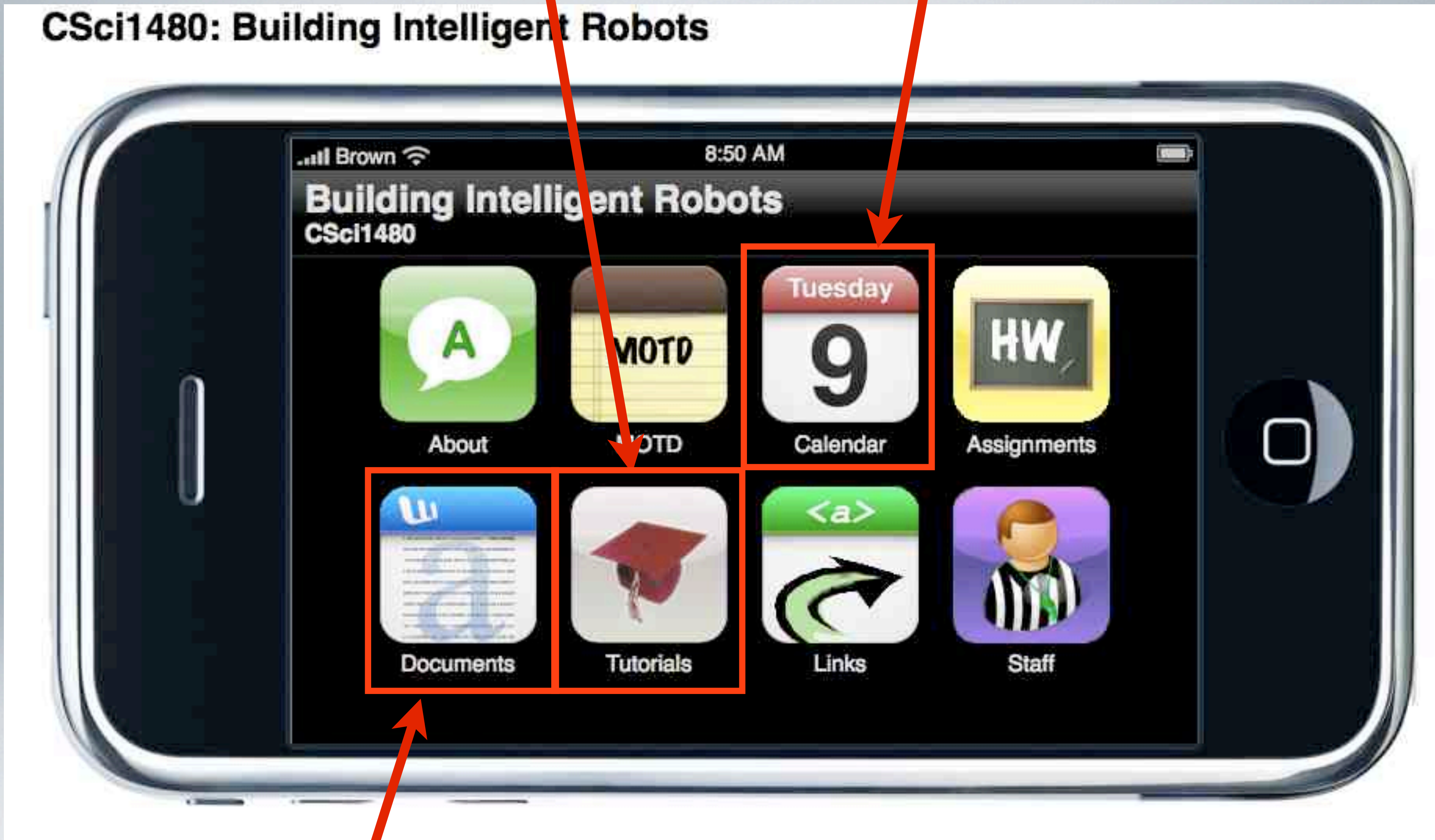
http://www.cs.brown.edu/courses/cs148

sign up for the mailing list! enjoy the links.

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

tutorials on cmake, svn, etc.

calendar, "up-to-date" syllabus



missive, syllabus, collaboration policy, etc.

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

ADMINISTRIVIA

- Subversion (SVN) version control
  - project submission and collaborative working space
- Collaboration policy: must be turned in to HTA
  - Projects performed in groups of 2-3
  - Individual reports
  - Post-project code sharing
- Class lectures in CIT 368 (typically MW)
- Hands-on sessions/labs in CIT 404 (typically Fridays)
- Stay updated via mailing list, MOTD, and calendar