



Initialization

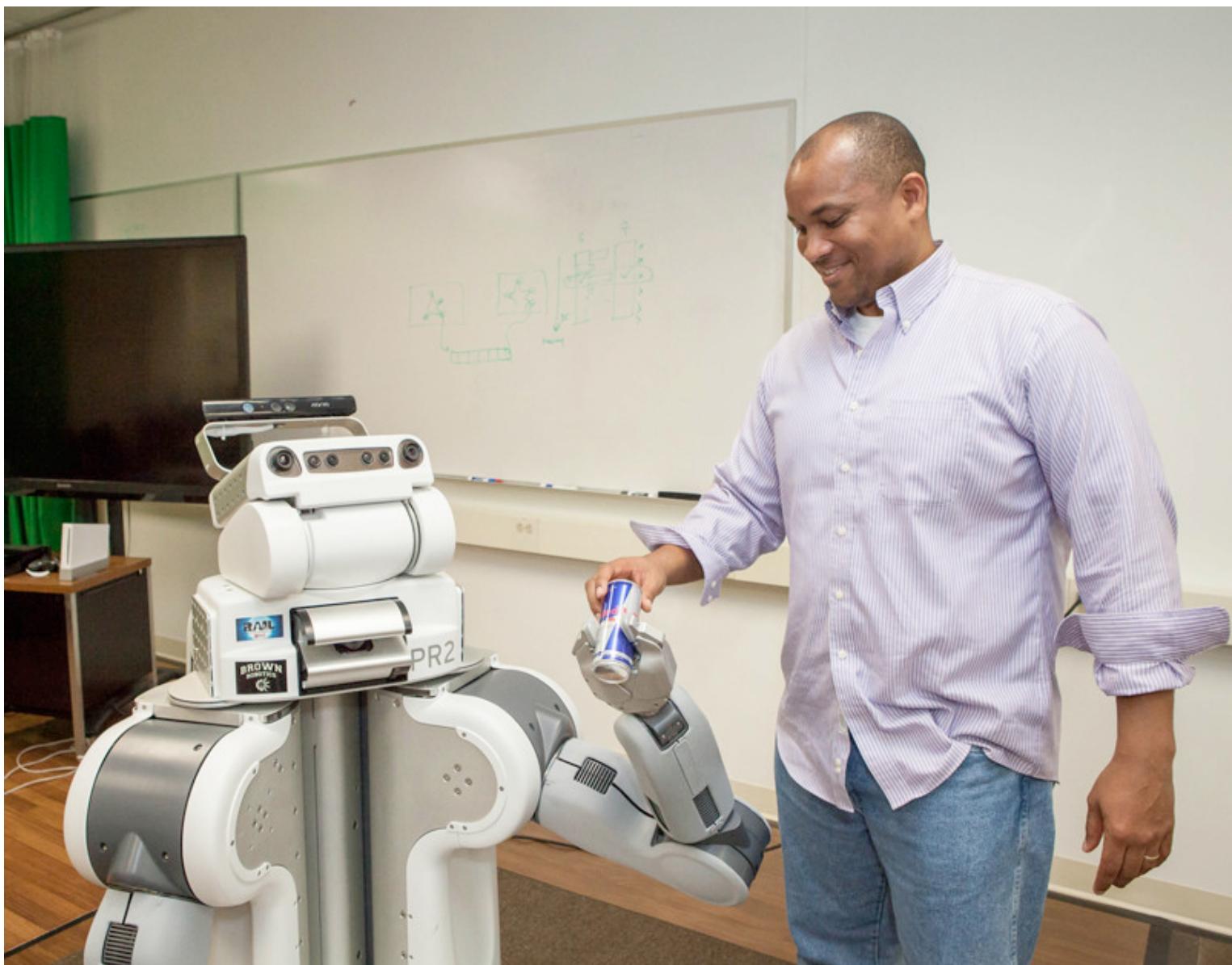
EECS 398
Intro. to Autonomous Robotics

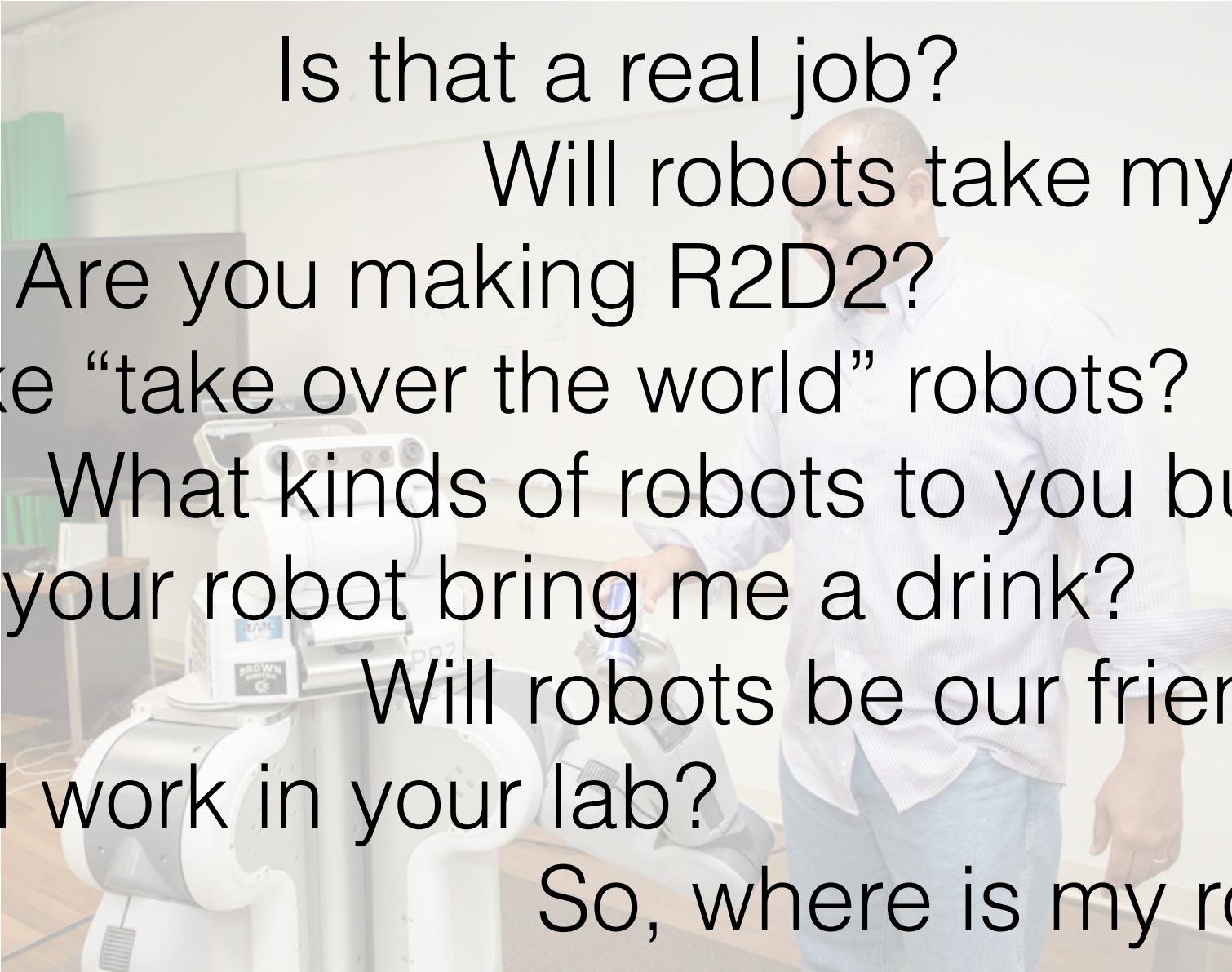
ME/EECS 567 ROB 501
Robot Modeling and Control

Fall 2018

Agenda

- So, where is my robot?
- Roadmap for autonomous robotics
- Course administrative overview
- Assignment 1 (Path Planning) assigned Monday
 - JavaScript/HTML5 and git covered this and next lecture
- Action items: what I need from you now





Is that a real job?

Will robots take my job?

Are you making R2D2?

Like “take over the world” robots?

What kinds of robots do you build?

Can your robot bring me a drink?

Will robots be our friends?

Can I work in your lab?

So, where is my robot?

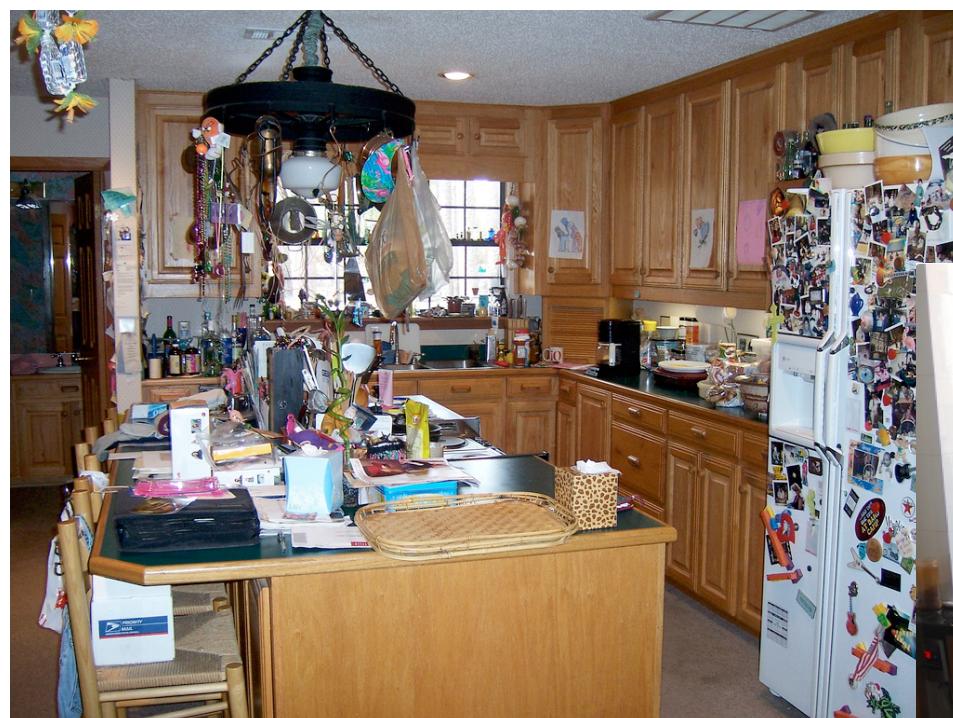
So, where is my robot?



UM EECS 398/567 - autorob.github.io

Mobile Manipulation Robots

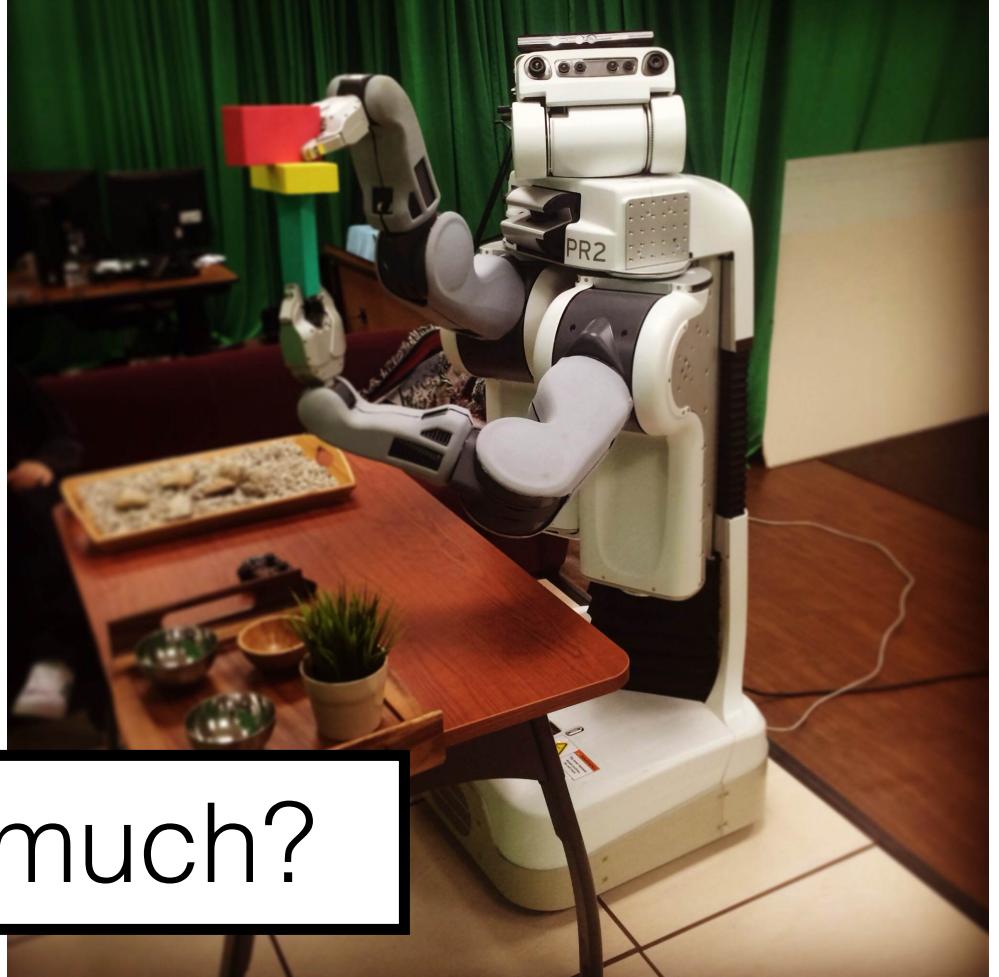




UM EECS 398/567 - autorob.github.io



How much?



Cost

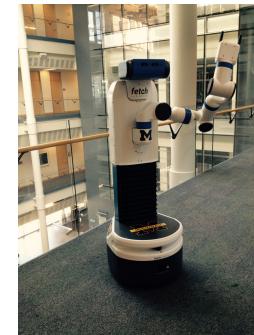
\$400K

\$100K

Willow Garage PR2



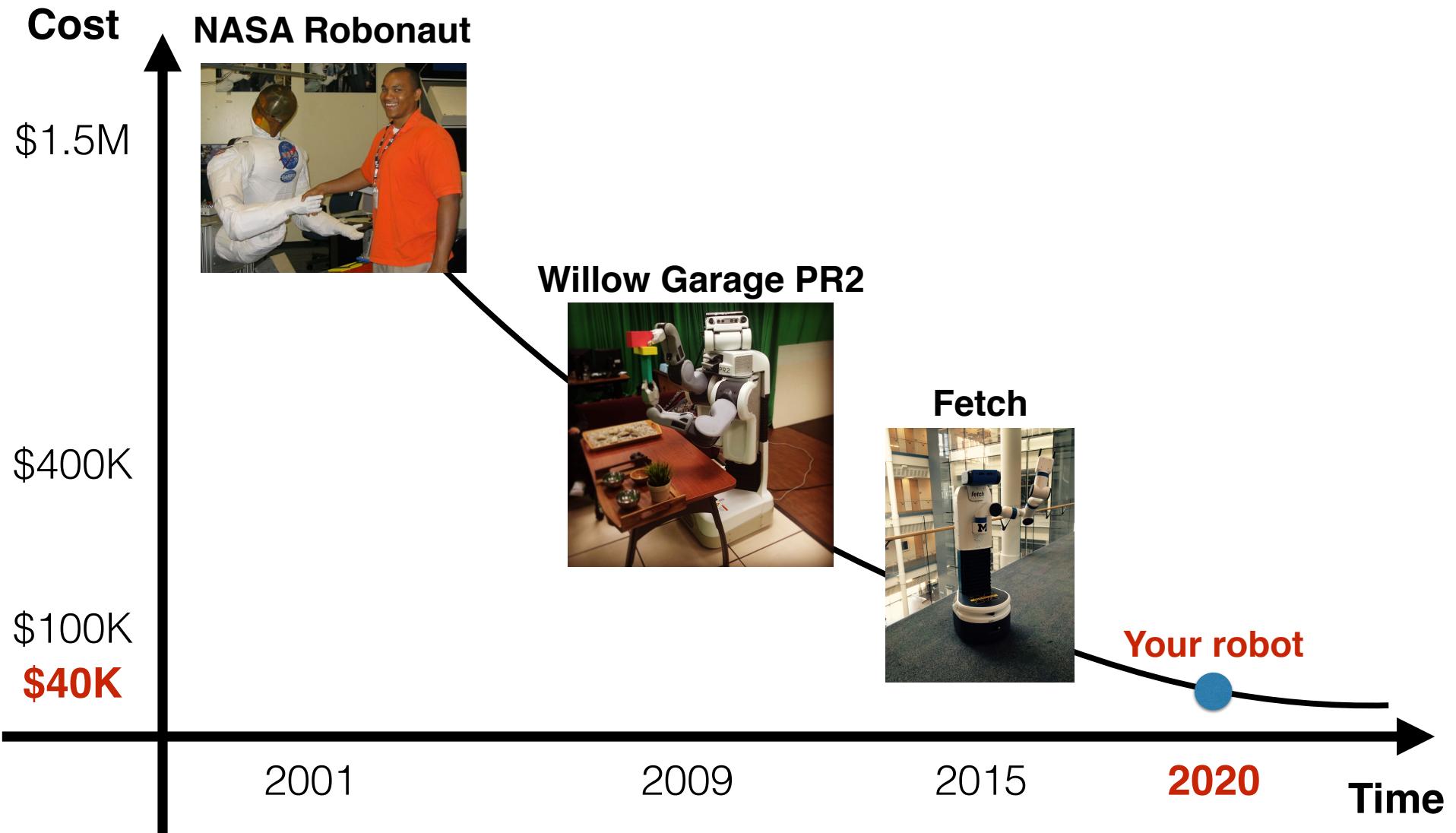
Fetch

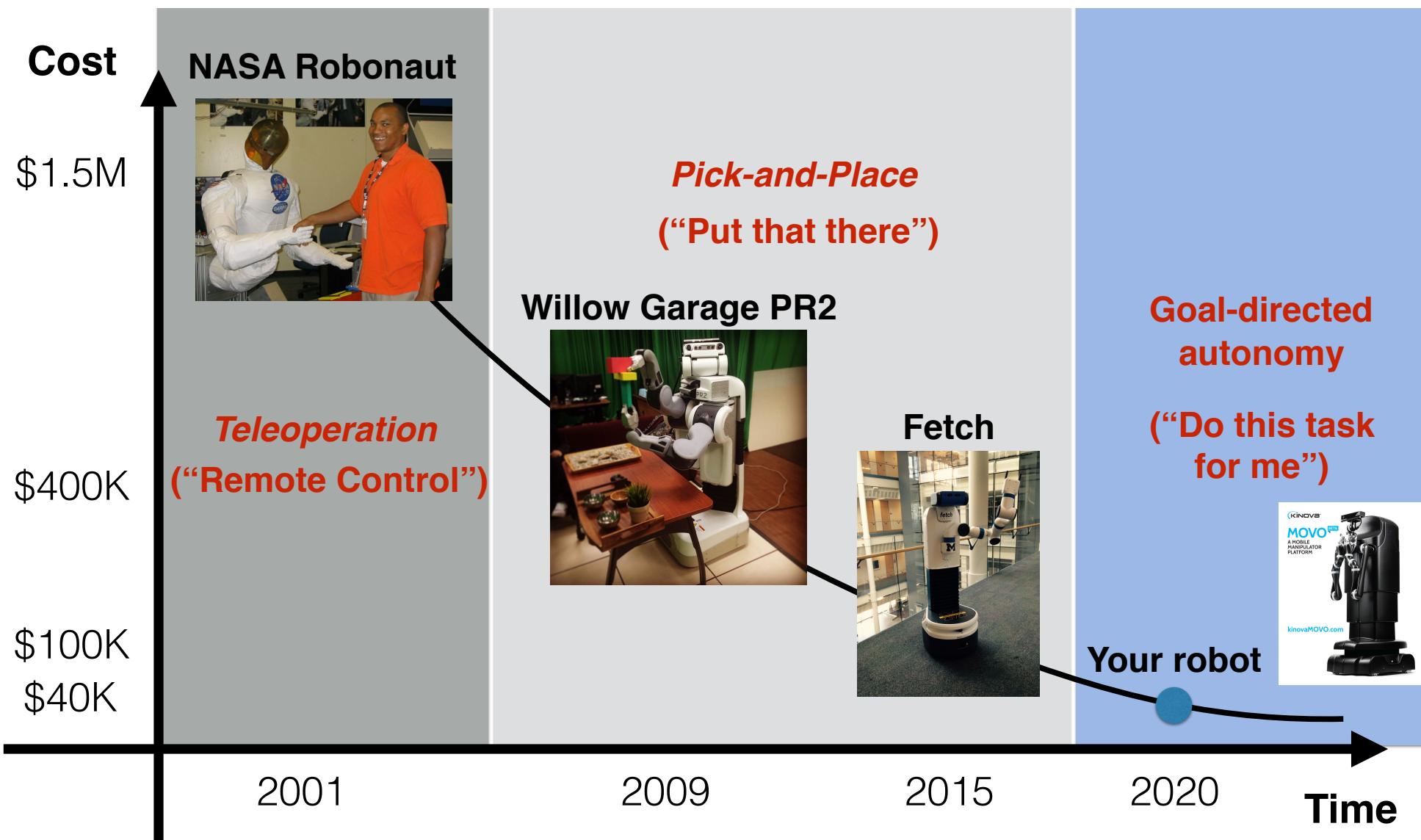


2009

2015

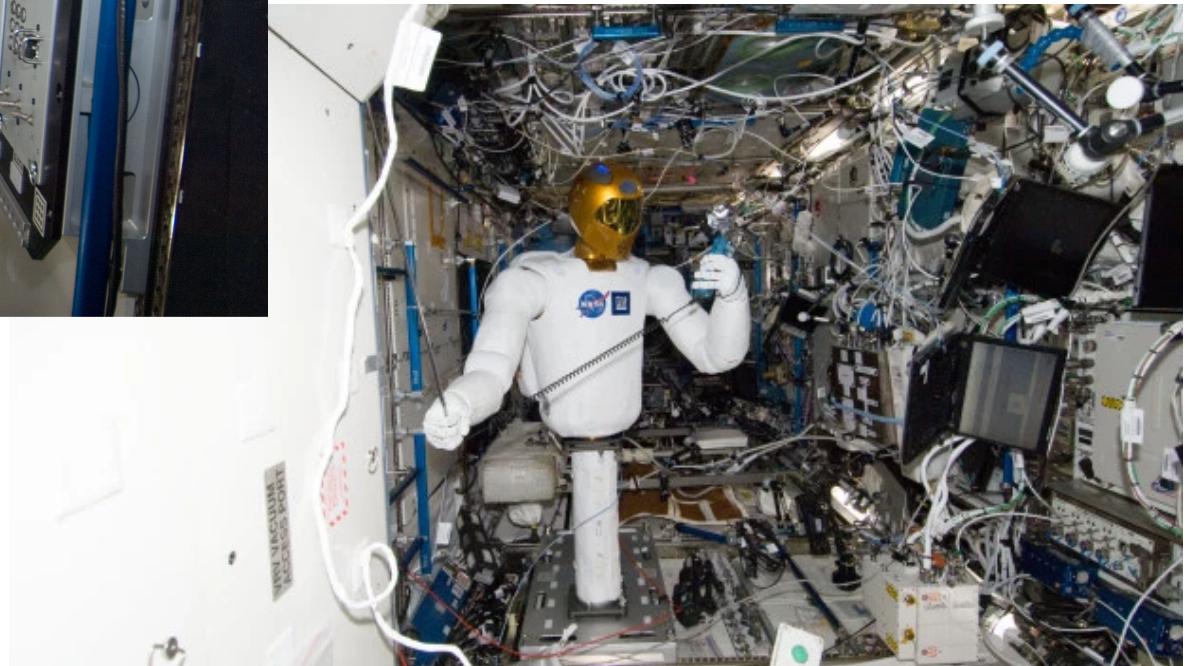
Time



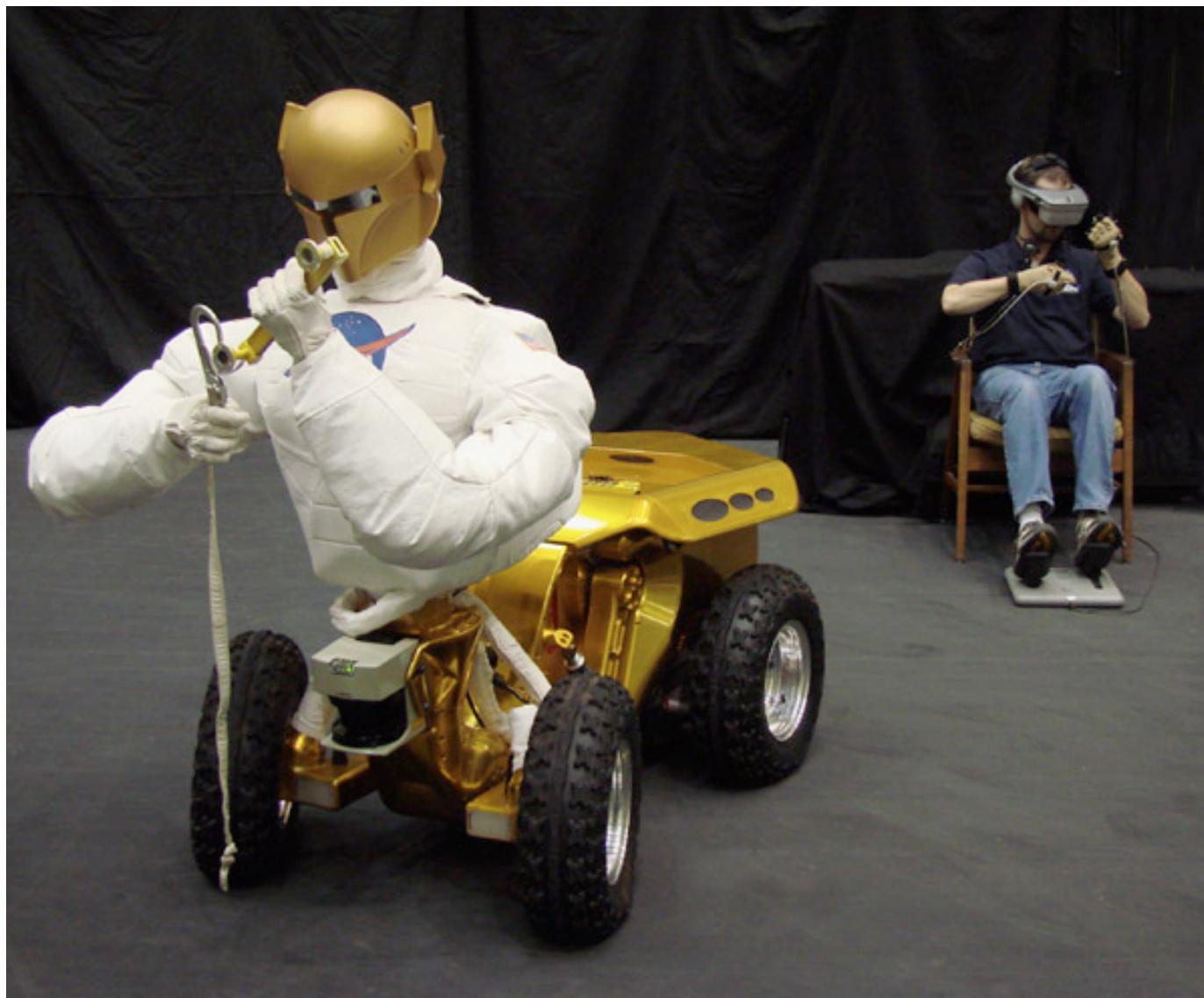




**NASA Robonaut 2 on the
International Space Station**



UM EECS 398/567 - autorob.github.io



UM EECS 398/567 - autorob.github.io

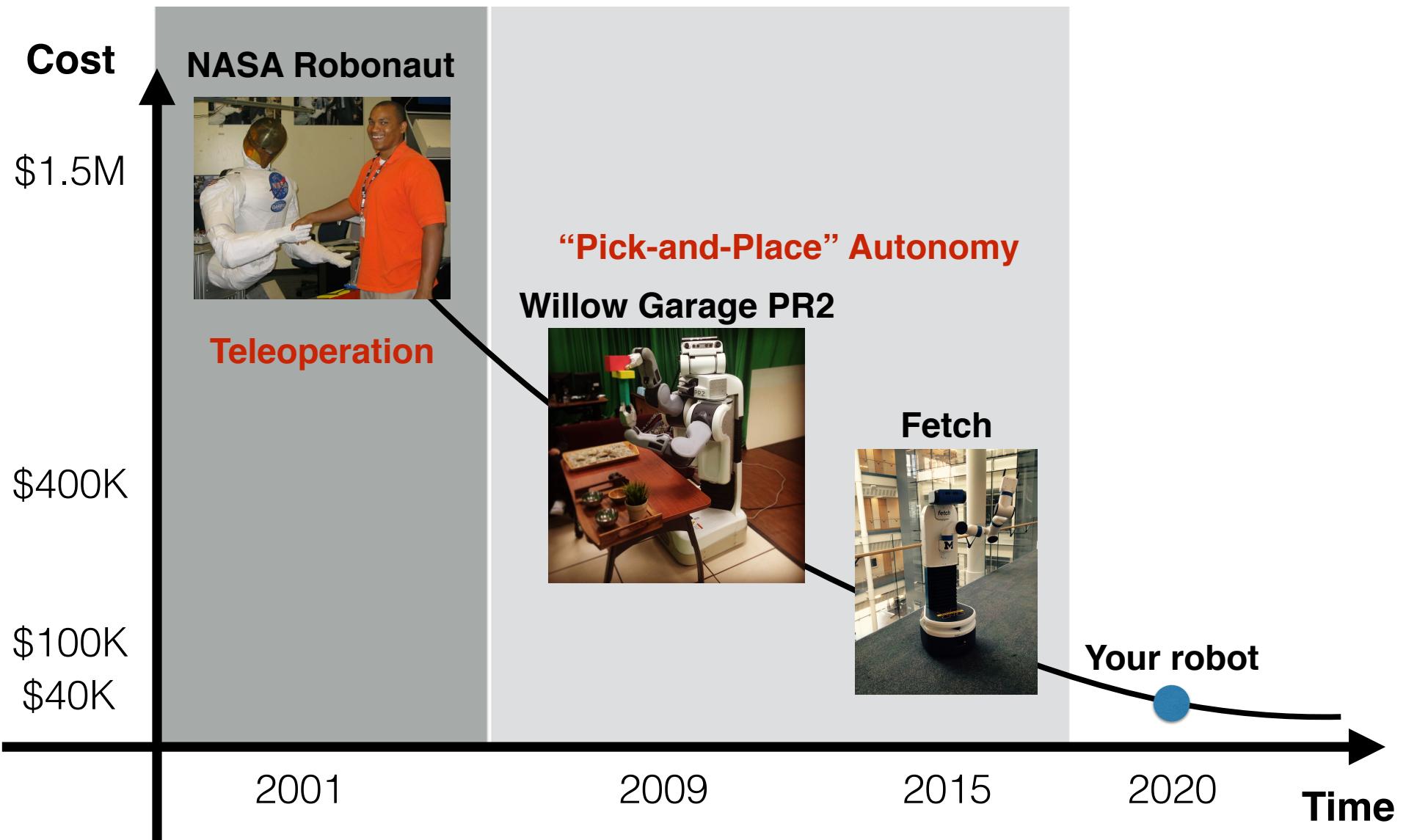
Wearable wireless motion capture for robot teleoperation



[Miller, Jenkins, et al., 2004, "Motion Capture from Inertial Sensing for Untethered Humanoid Teleoperation"]



UM EECS 398/567 - autorob.github.io



Autonomous Robotics in 3 words

Sense.
Plan.
Act.

Autonomous Robotics in 3 words

Sense. **Perceive** a model of the current world state.

Plan. **Search** over actions towards a goal state.

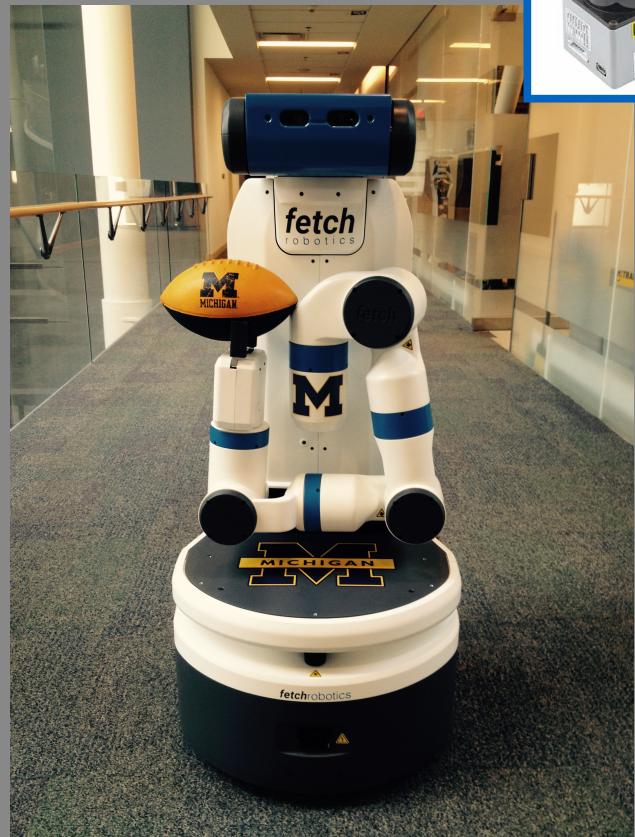
Act. **Execute** actions through forces at robot's motors



Color+Depth Camera

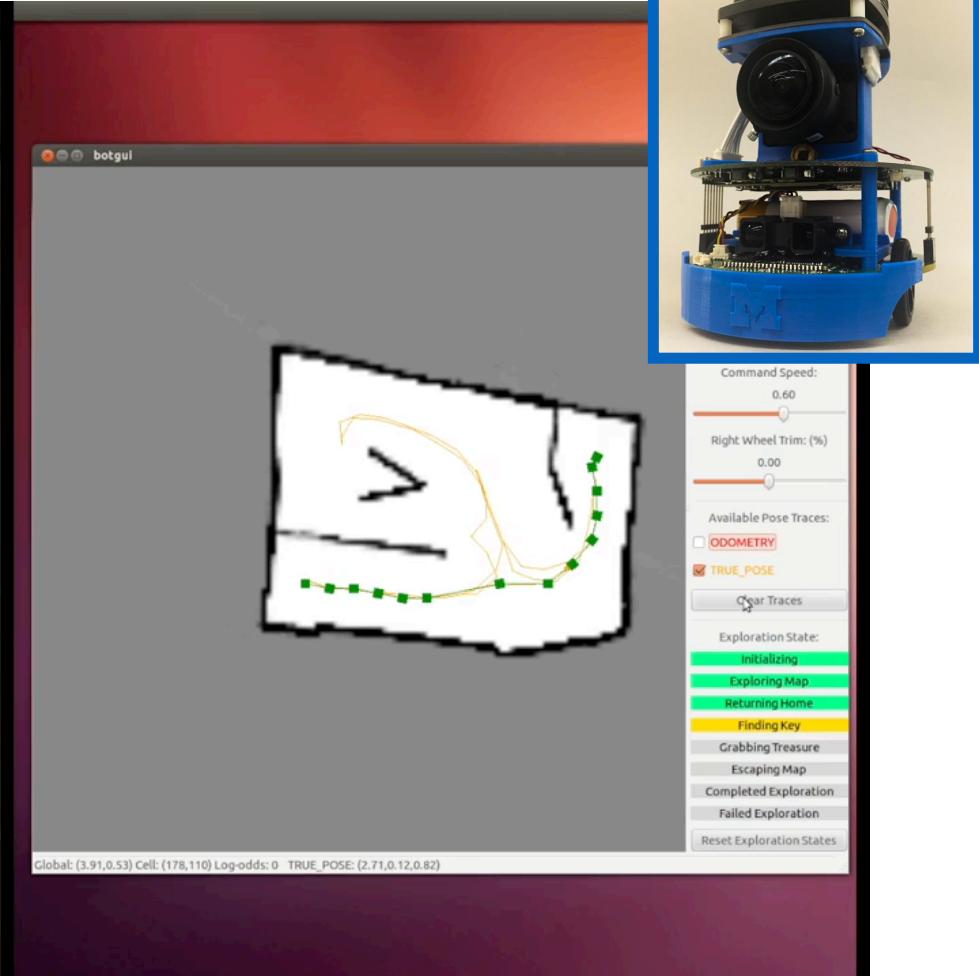
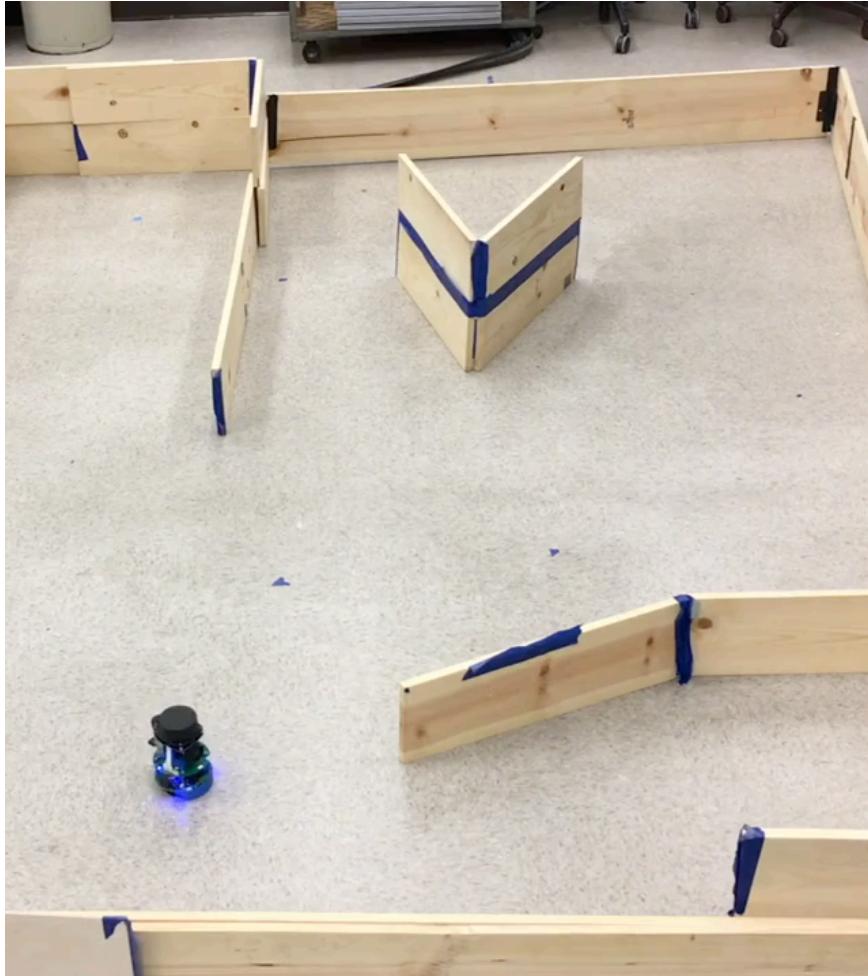


Laser Rangefinder
UM EECS 398/567 - autorob.github.io





ROB 550 BotLab / EECS 467 Escape Challenge 2016-17



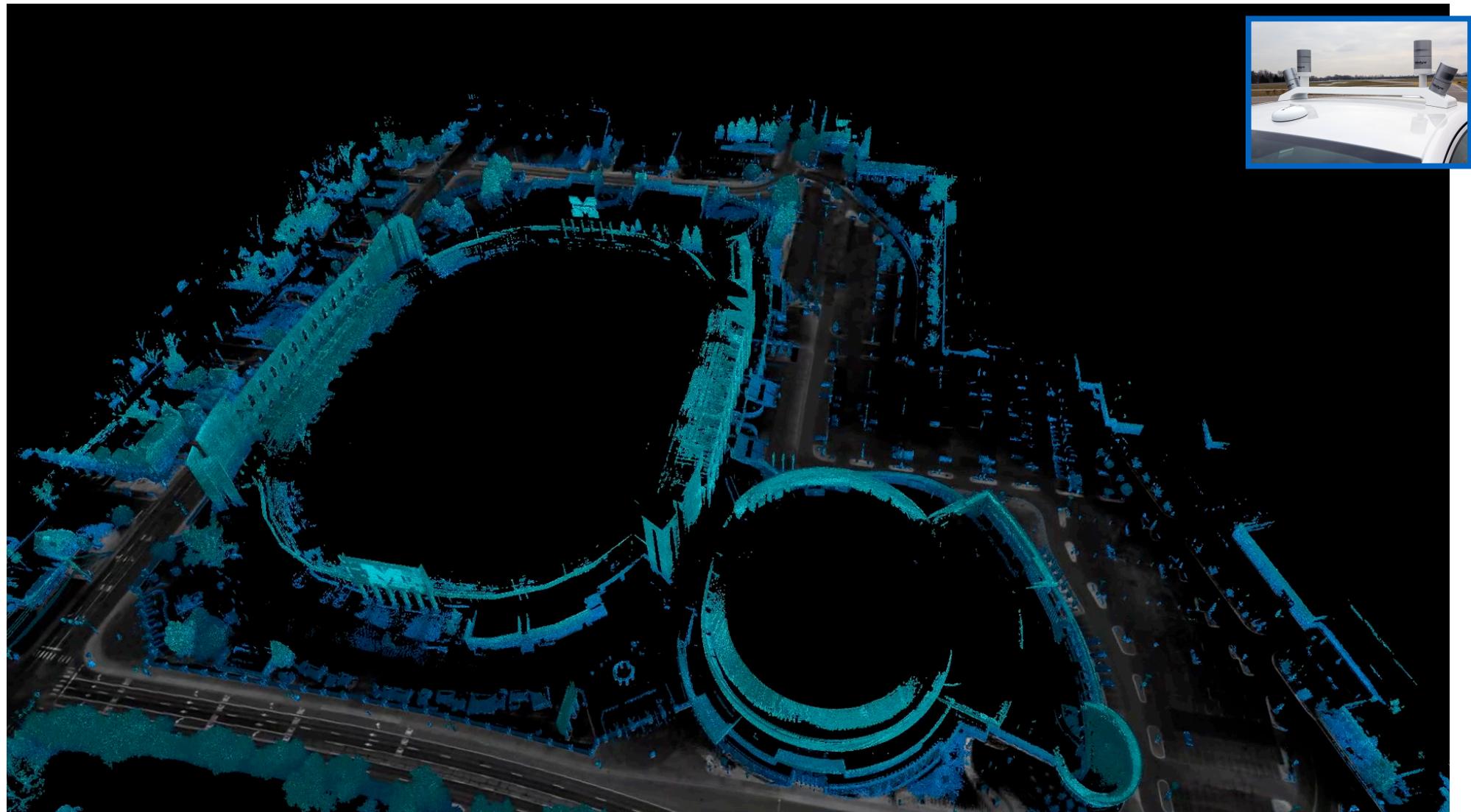
UM EECS 467 autobot.github.io Jasmine Liu et al.



Michigan Next Generation Vehicle (Olson, Eustice, et al.)

CMU ECE 698/567

autordb.github.io



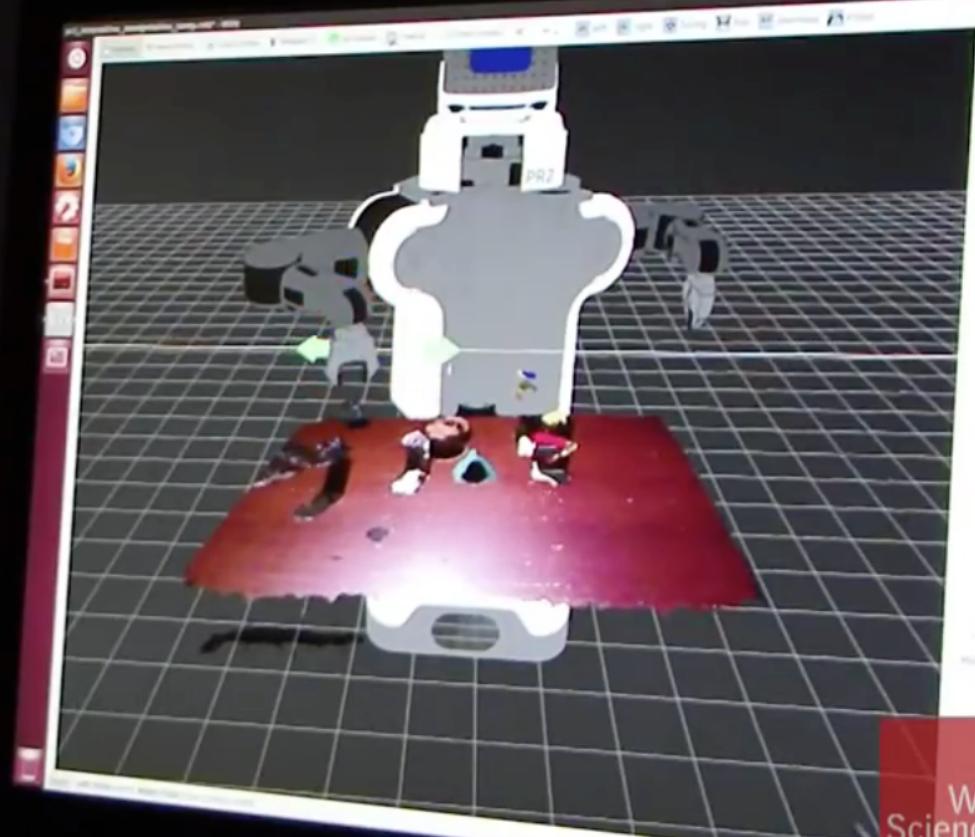
Michigan Next Generation Vehicle (Olson et al.)



Color+Depth Camera

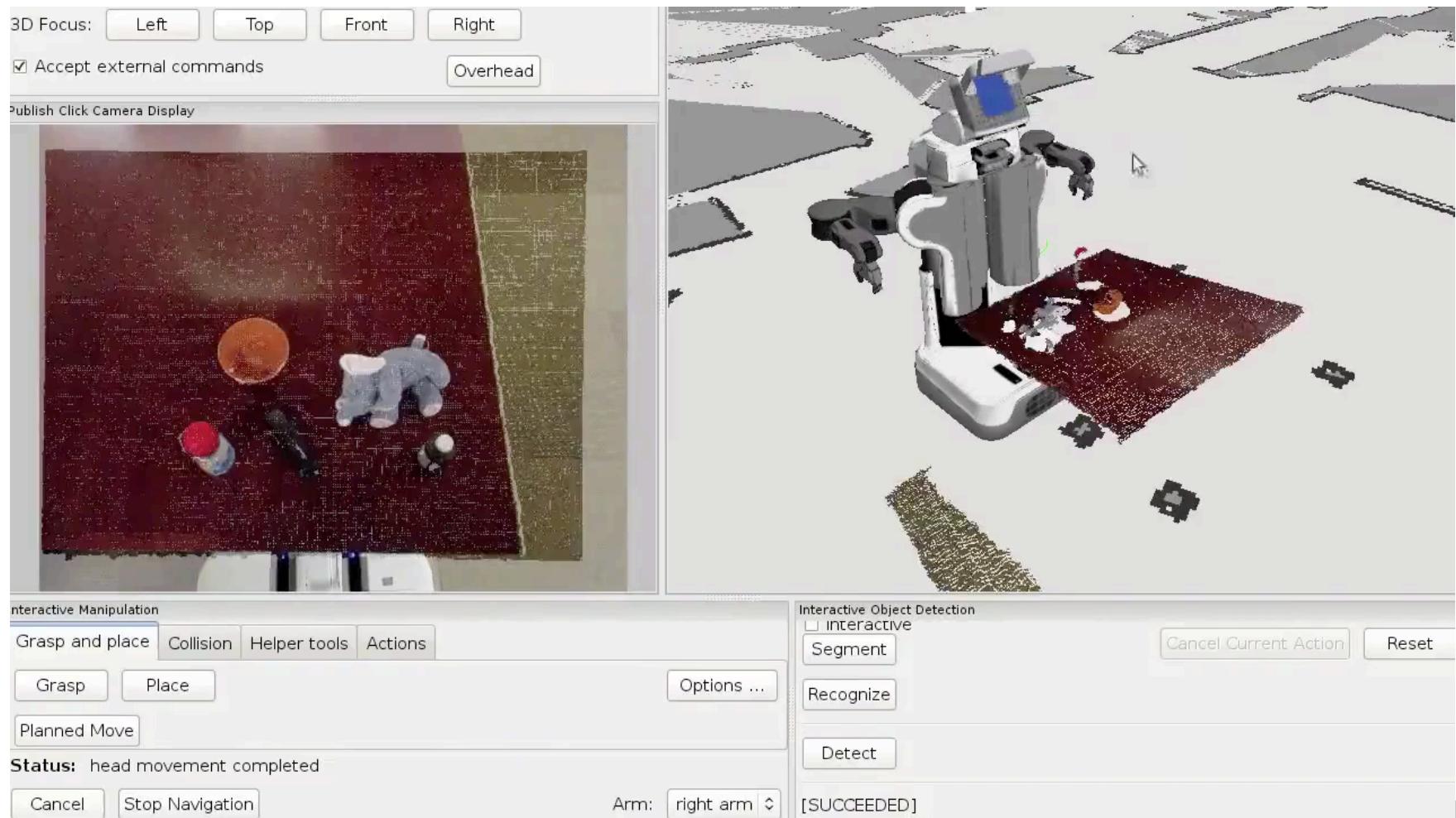


Laser Rangefinder
UM EECS 398/567 - autorob.github.io



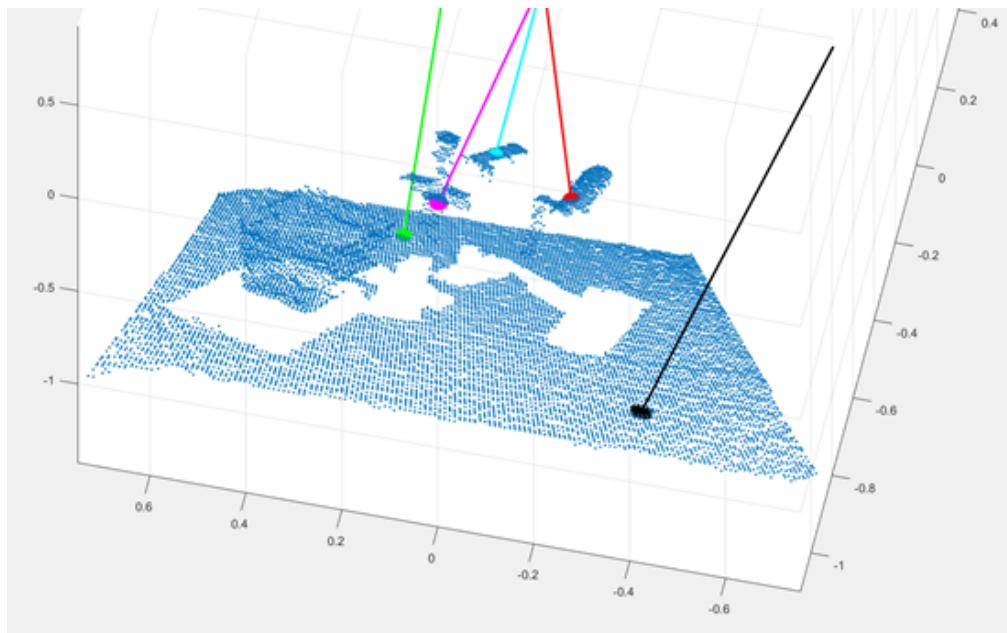
Object Manipulation

Willow Garage



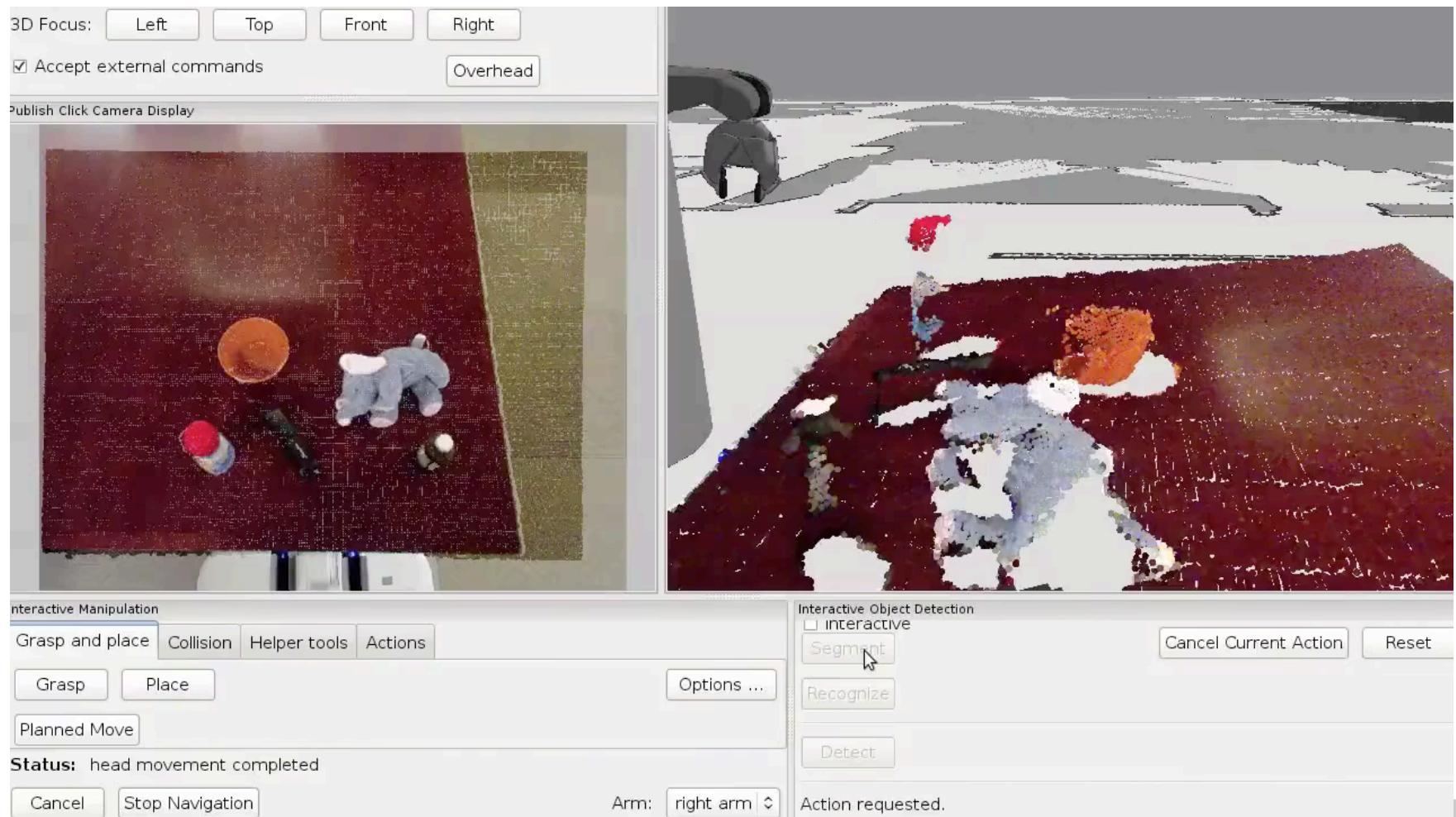
Point Cloud Processing (briefly)

- For every point:
 - compute nearest neighbors
 - compute principal components in neighborhood
`eig(cov(nbhd(:, 1:3)))`
 - surface normal is eigenvector for smallest eigenvalue
- Cluster points based on direction similarity of normal



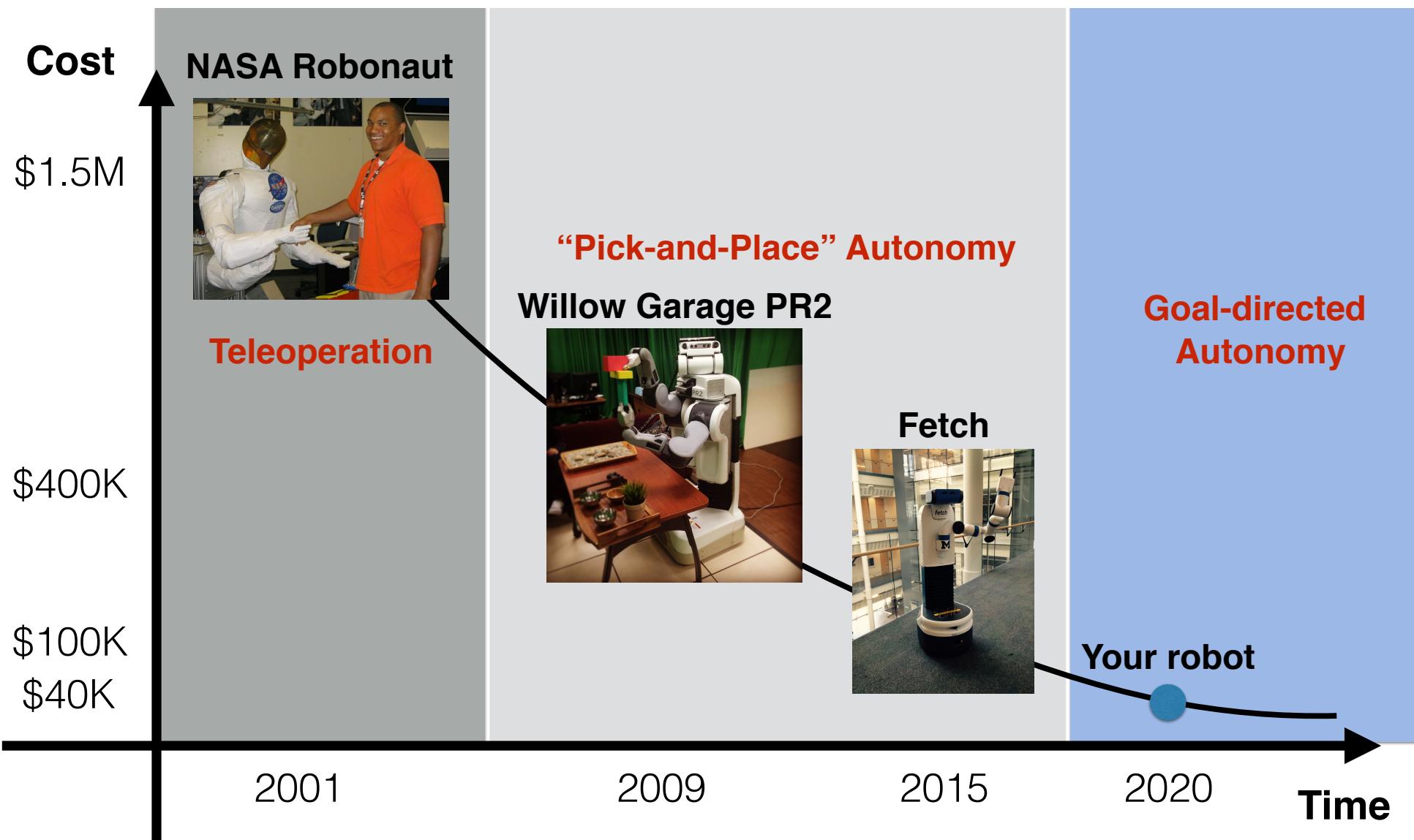
Object Manipulation

Willow Garage



[Sui, Jenkins, et al. IROS 2017]





Goal-directed Manipulation



Initial Scene → Goal Scene

“put small objects in bin”

“put small objects in bin”

[Sui, Xiang, Jenkins, Desingh, IJRR 2017]

Semantic Robot Programming

[Zeng, Jenkins, et al. ICRA 2018]



Enable natural user programming of robots
by demonstration of intended goal scenes



Goals shown in upper left corners

8x

Agenda

- So, where is my robot?
- Roadmap for autonomous robotics
- **Course administrative overview**
- Assignment 1 (Path Planning) assigned today
 - JavaScript/HTML5 and git covered this and next lecture

Course Staff

- Instructor: Chad Jenkins (ocj)
 - Office hours (Beyster 3644)
 - Monday 3-5pm, Tuesday 1-3pm
- GSI: Maithili Varang Patel (maithili)
 - Office hours (Beyster 1637 A)
 - Monday 3-5pm, Tuesday 1-3pm (tentative)

Administrivia

- Meeting time/place
 - MW 1:30-2:40pm MMT, EECS 1500
- Website
 - <http://autorob.github.io/>
- Discussion channel
 - <https://autorob.slack.com/>

Slack via web

The screenshot shows a Slack interface in a web browser. The left sidebar lists channels and direct messages. The #general channel is selected and active.

#general
Company-wide announcements and work-based matters

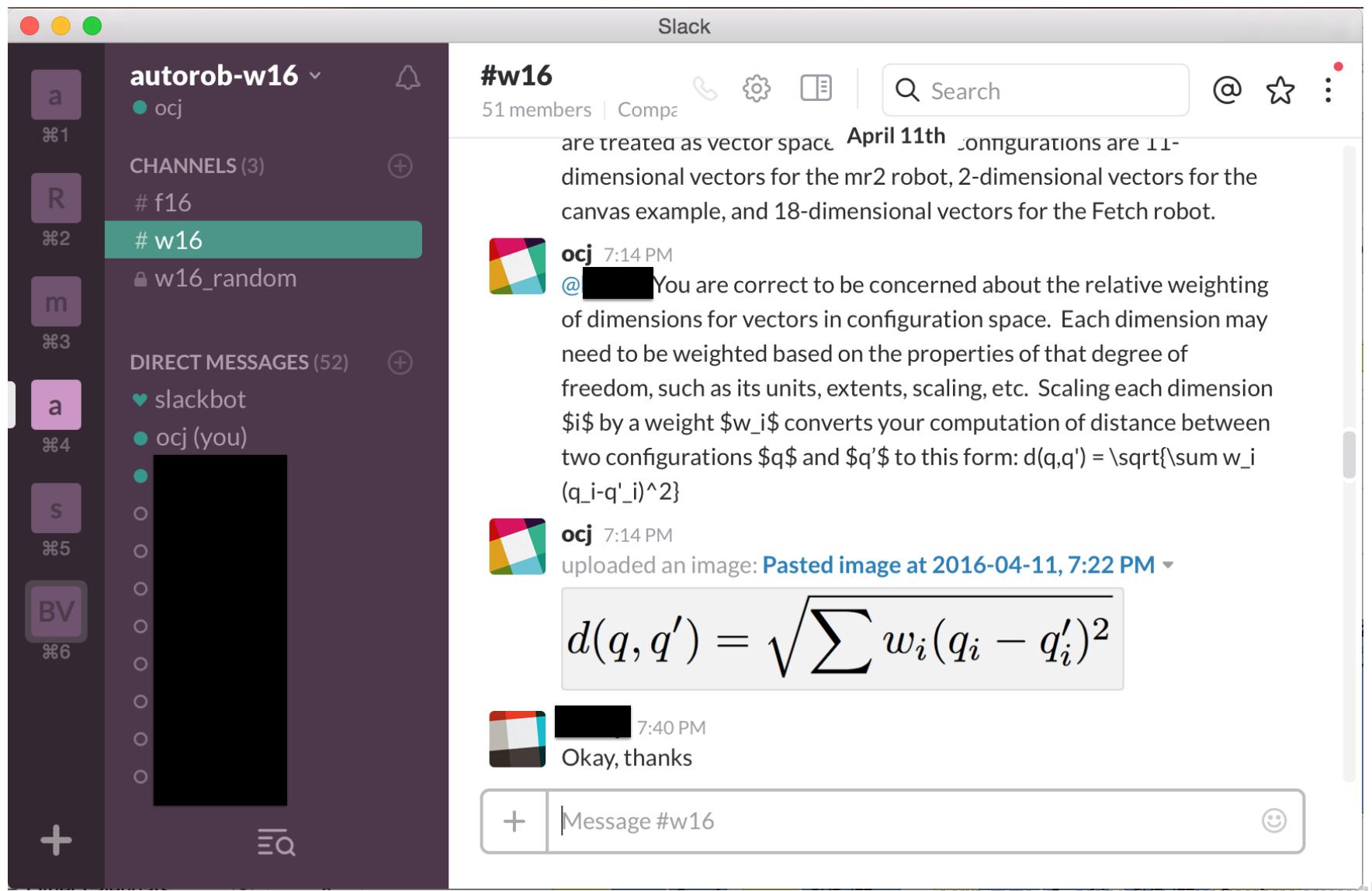
Channels: # advanced-extensions, # asgn1-pathplan, # asgn2-pendulum, # asgn3-fk, # asgn4-dance-fsm, # asgn5-ik, # asgn6-rrt, # asgn7-best-use, # general (selected), # random, # staff.

Direct Messages: slackbot, Chad Jenkins (you), Maithili.

Recent Activity: Chad Jenkins joined #general along with Maithili at 3:16 PM on Tuesday, August 28th.

Message input field: Message #general

Slack OS X app

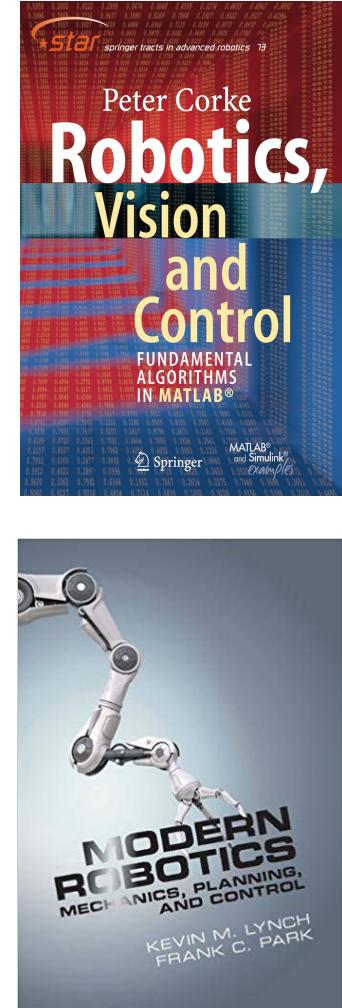


Course Structure

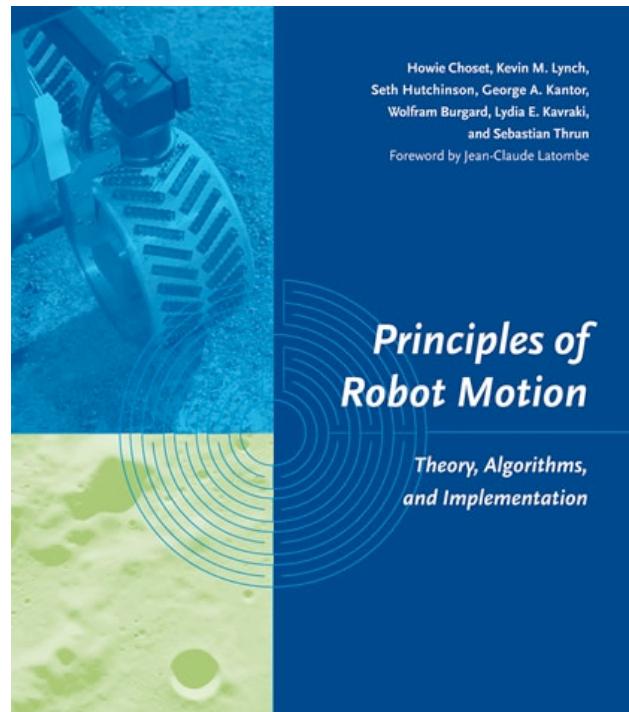
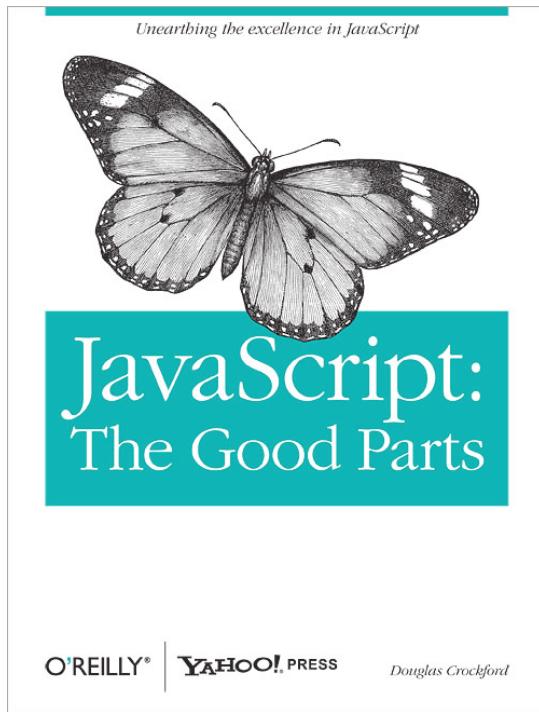
- Autonomous robot modeling and control
 - Objective: Give you the computational skills (and code) to model and control any mobile manipulator
- Project-focused class
 - 7 individual projects: from single joint control up to articulated motion planning
- Computing-friendly introduction to robotics: projects in JavaScript

Course Textbook

- Robot Modeling and Control (Spong, Hutchinson, Vidyasagar)
- Alternative: Robotics, Vision, and Control (Corke)
- Suggested but unsupported: Modern Robotics (Lynch and Park)
- In-depth coverage of concepts and math contained in textbooks
- Additional handouts and links will appear on the course website



Optional reading



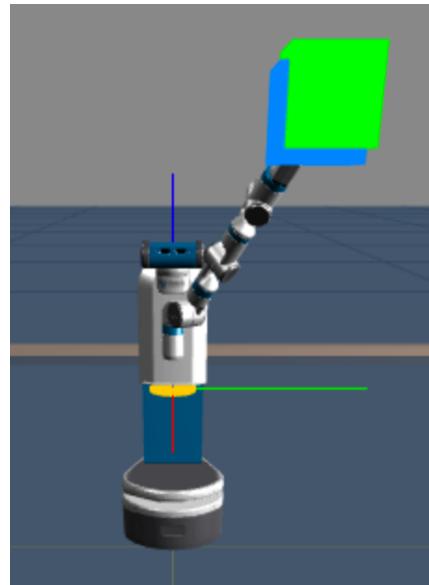
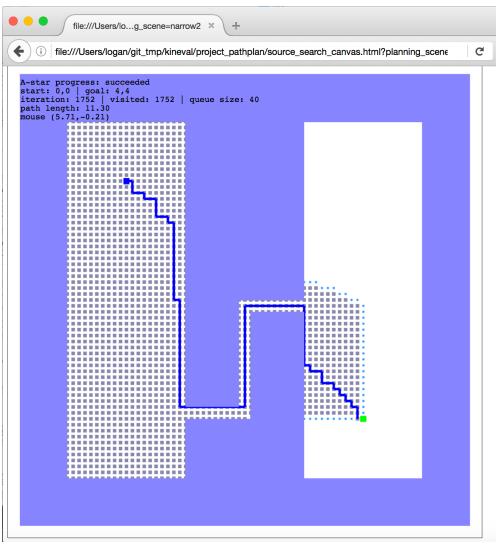
Projects

- Projects implemented in JavaScript/HTML5 using KinEval stencil
 - Projects submitted and tracked through git (gitlab|github|bitbucket)
 - Instructor (ohseejay|ocj) needs admin access
- 7 projects
 - 6 Programming, 1 Written/Oral
- Grading: projects are broken down into features that are “checked”
 - points are earned through successful implementation of features

Projects

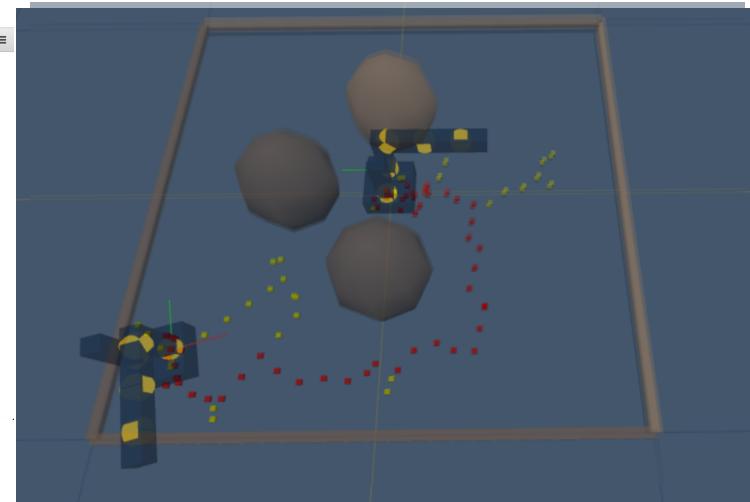
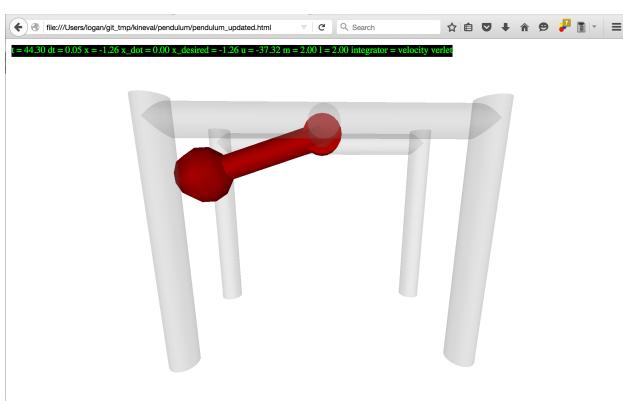
- **Path Planning** A-star search in 2D world
- **Pendulum** physical simulation and PID control of 1 DoF robot
- **Forward Kinematics** convert robot configuration to 3D space
- **Dance Contest** control of robot joints to do a dance
- **Inverse Kinematics** control gripper of a robot to reach a point in 3D
- **Motion Planning** collision-free planning over robot configurations
- **Best Use of Robotics** what will you do with all of this power?

- Path Planning



- FK/IK

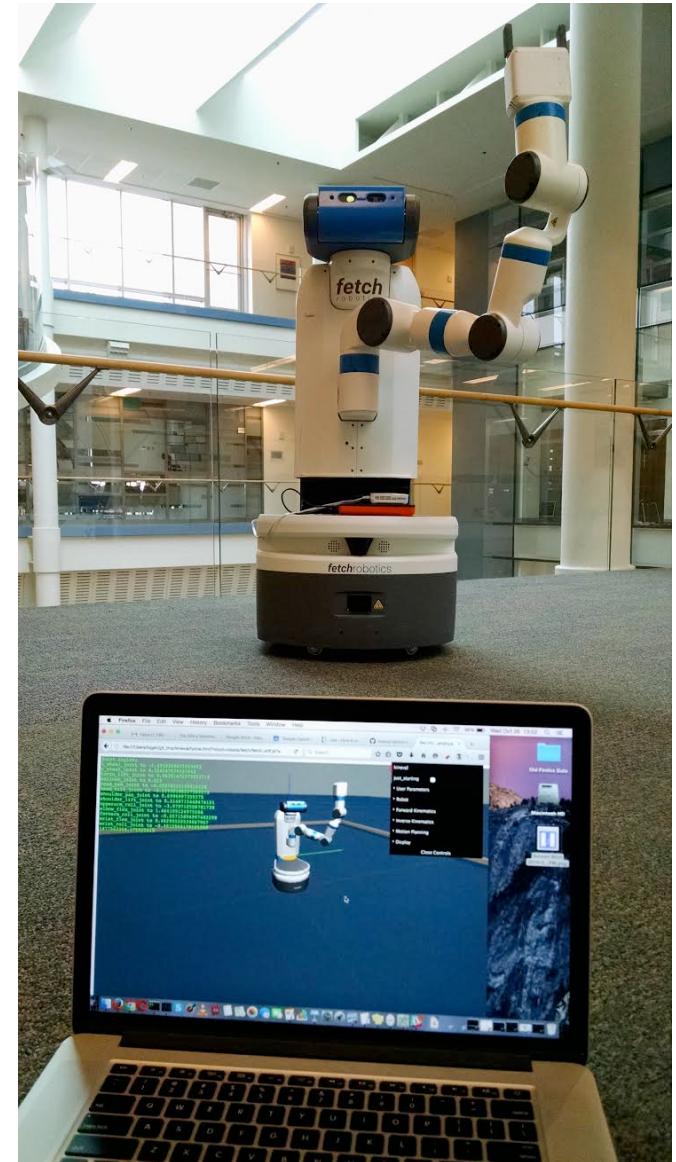
- Pendularm



- RRT

Will you work with a real robot?

Yes, at least once
using rosbridge/ROS



Grading Summary

EECS 398: Introduction to
Autonomous Robotics

- 7 projects (12 points each)
- 5 quizzes (4 points each)

A: 93+ points

B: 83+ points

C: 73+ points

ME/EECS 567 ROB 510:
Robot Kinematics and
Dynamics

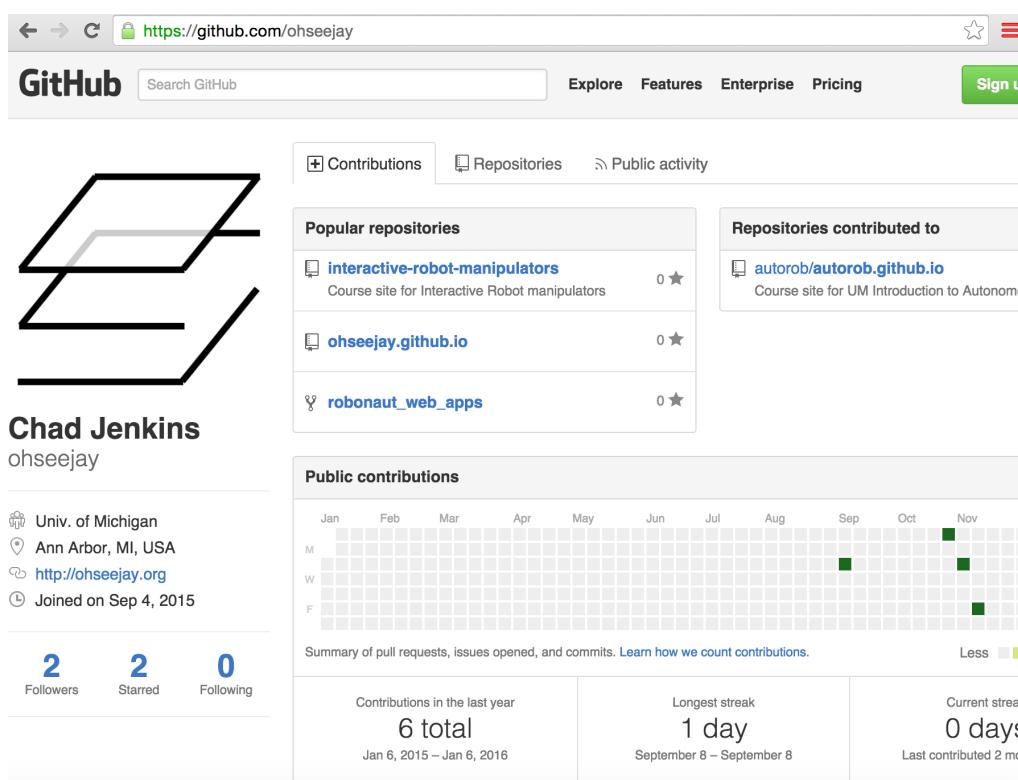
- 7 projects (18 points each)
- 5 quizzes (4 points each)
- Advanced features (4 points)

A: 135+ points

B: 120+ points

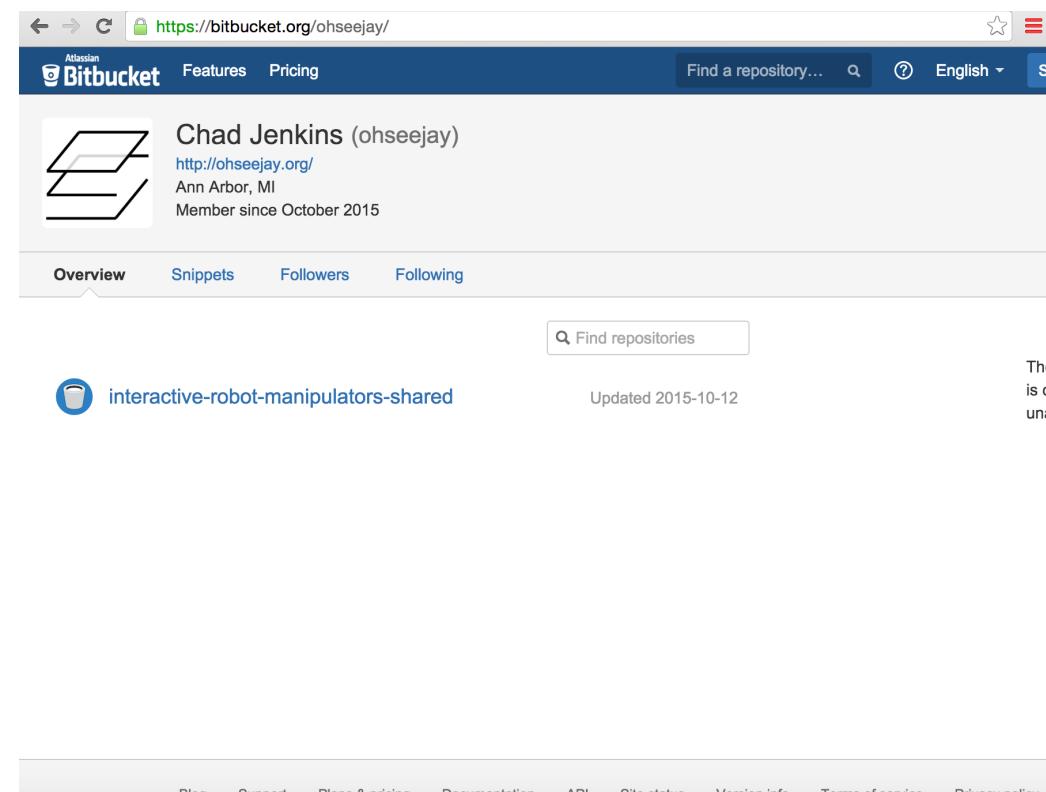
C: 105+ points

github.com/ohseejay



A screenshot of a GitHub user profile page for 'ohseejay'. The profile picture is a stylized drawing of a robotic arm. The top navigation bar includes links for Explore, Features, Enterprise, Pricing, and Sign up. Below the header, there are sections for Popular repositories (listing 'interactive-robot-manipulators', 'ohseejay.github.io', and 'robonaut_web_apps') and Repositories contributed to (listing 'autorob/autorob.github.io'). A large section titled 'Public contributions' shows a grid of activity points from January to November. Summary statistics at the bottom indicate 6 total contributions in the last year, a longest streak of 1 day (from September 8 to September 8), and a current streak of 0 days (last contributed 2 months ago). On the left sidebar, there are links for Univ. of Michigan (Ann Arbor, MI, USA), the personal website (<http://ohseejay.org>), and the date joined (Sep 4, 2015). At the bottom, social metrics show 2 Followers, 2 Starred, and 0 Following.

bitbucket.org/ohseejay



A screenshot of a Bitbucket user profile page for 'ohseejay'. The profile picture is the same as on GitHub. The top navigation bar includes links for Features, Pricing, and a search bar. Below the header, it shows 'Chad Jenkins (ohseejay)' with the URL <http://ohseejay.org/>, location Ann Arbor, MI, and member since October 2015. There are tabs for Overview, Snippets, Followers, and Following. A repository card for 'interactive-robot-manipulators-shared' is displayed, showing it was updated on 2015-10-12. The bottom navigation bar includes links for Blog, Support, Plans & pricing, Documentation, API, Site status, Version info, Terms of service, and Privacy policy.

UM EECS 398/567 - autorob.github.io

gitlab.eecs.umich.edu/ocj

The screenshot shows a GitLab user profile for 'ocj'. The top navigation bar includes the GitLab logo, the URL 'https://gitlab.eecs.umich.edu/u/ocj', and a notification badge 'Odest' with a warning icon. Below the header is a sidebar with the name 'ocj' and a red fox icon. The main content area features a large orange fox logo, the name 'ocj', and the text '@ocj · Member since Jul 16, 2016 7:29pm'. A horizontal navigation bar below the bio includes tabs for 'Activity' (which is active), 'Groups', 'Contributed projects', 'Personal projects', and 'Snippets'. The activity timeline shows a grid from September to June, with days labeled M, W, F. Below the grid is a legend with four colored squares (light blue, medium blue, dark blue, black) followed by the text 'Summary of issues, n'. Two recent activity items are listed:

- ocj pushed to branch master at ocj / testing**
e7e73105 · added a new file about 4 hours ago
- ocj pushed new branch master at ocj / testing**
9be3d074 · Add new file about 4 hours ago

Collaboration Policy

- All work submitted must be your own
 - All code submitted must comply with Michigan Honor License
- No code can be communicated, including verbally
 - Explicit use of external sources must be clearly cited
 - Repositories must be **private** for proper compliance
- Free flow of discussion and ideas is encouraged

Michigan Honor License

- 3-Clause BSD License + Michigan Honor Code + proper citation
- Assert the compliance of your code with the MHL
 - Append your name to the end of LICENSE in your repository
- Submitted code will not be graded without asserting LICENSE

Late Policy

- Projects submitted after deadline may not be graded (zero credit)
- If a late submission is allowed, it can receive at most
 - 80% credit if pushed within 2 weeks of the deadline
 - 60% credit if pushed within 4 weeks of the deadline
 - 50% credit if pushed anytime before final grading

Regrading policy

- Projects features are graded with:
 - “CHECK” (sufficiently completed)
 - “DUE” (insufficiently completed)
 - “PENDING” (not due yet)
- A project feature can be regraded for partial credit for at most
 - 80% credit if pushed within 2 weeks of the last returned grading
 - 60% credit if pushed anytime before final grading

 **sdnt / repository name**

Private

 Unwatch ▾ 2

 Star 0

 Fork 0

 Code

 Issues 0

 Pull requests 0

 Projects 0

 Wiki

Insights ▾

Projects for EECS398 Intro to Autonomous Robotics at the University of Michigan

 71 commits

 7 branches

 0 releases

 3 contributors

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

 sdnt Updated motion planning to disallow rotations about x-z axis

Latest commit b14c770b Jan 10

 js	Fall 2016 release	a year ago
 kineval	Disallow rotations about x-z axis. Fixed return val. Can now trace path	8 months ago
 project_pathplan	2d rrt-connect complete using drawHighlightedPath. Had to change give...	8 months ago
 project_pendulum	pendulum seems complete	11 months ago
 robots	Fall 2016 release	a year ago
 tutorial_heapsort	Added heap increase key function to heap.js to help with A* algorithm	10 months ago
 tutorial_js	Playing with Javascript samples	a year ago
 worlds	Deleted all source_*.js files and added honor.txt	10 months ago
 .gitignore	Implemented and tested matrix multiply and matrix transpose	9 months ago
 README.md	Fall 2016 release	a year ago
 grading.txt	added grading for Final grading	9 months ago
 home.html	Fall 2016 release	a year ago

0.io

Branch: master ▾

repository name / grading.txt



odescj added grading for Final grading

1 contributor

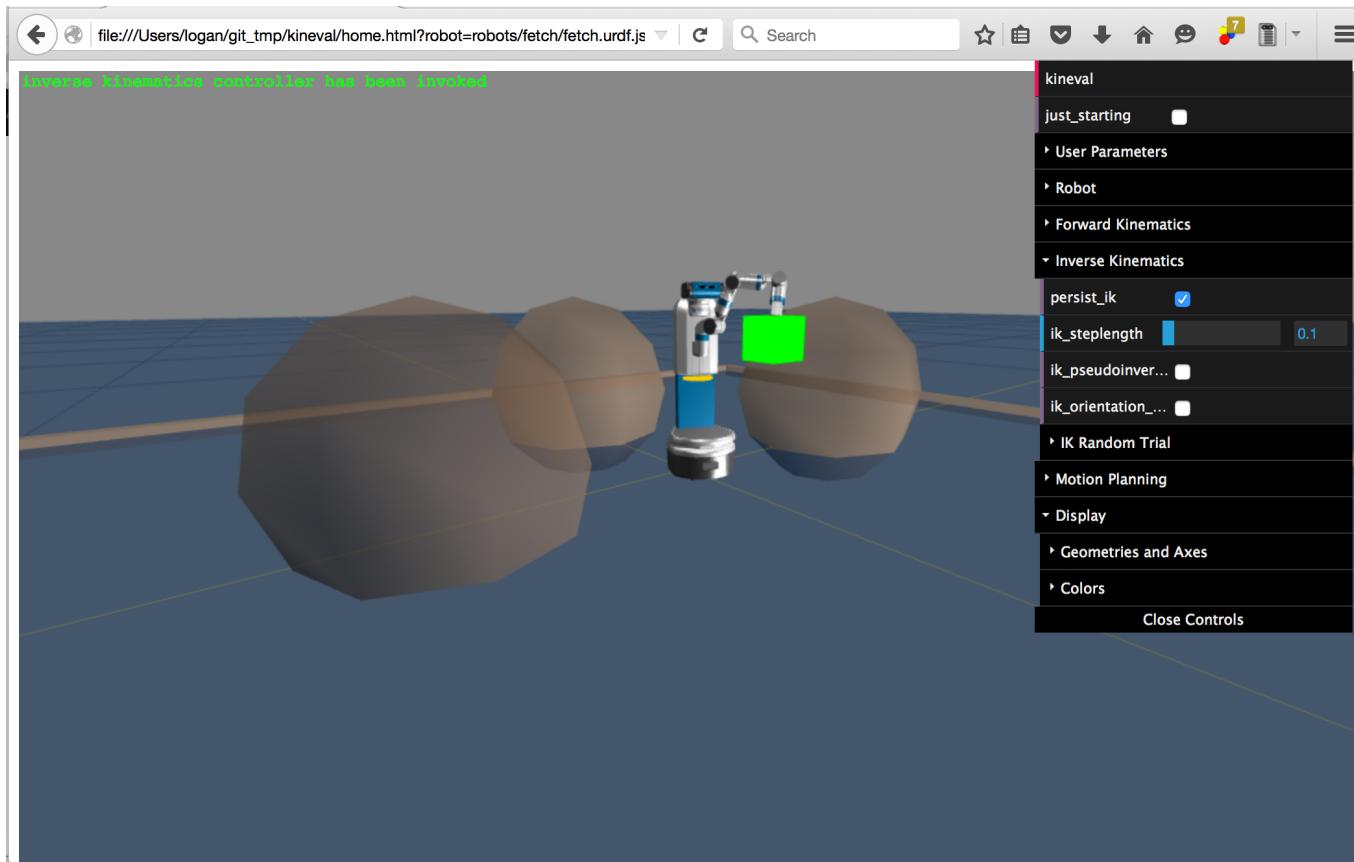
60 lines (58 sloc) | 2.26 KB

[Raw](#) [Blame](#)

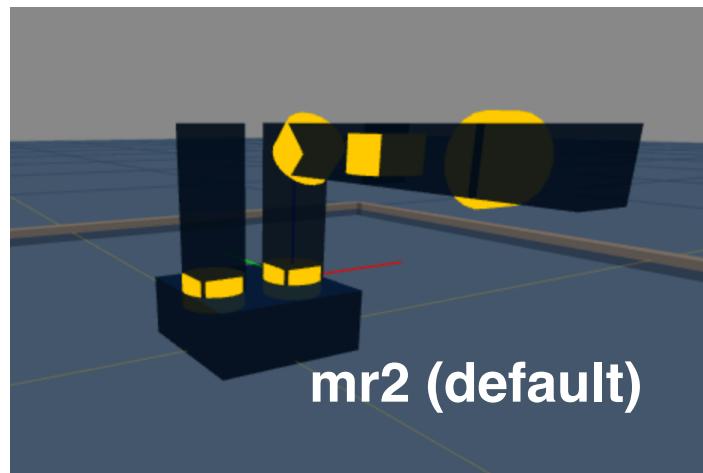
```
1 53: student name
2 EECS 398-004 F16
3 Assignment 1 feature 1 (4.00/4): PathPlan_Heap: CHECK
4   comment: good work
5 Assignment 1 feature 2 (8.00/8): PathPlan_AStar: CHECK
6   comment: good work
7 Assignment 2 feature 3 (4.00/4): Pendularm_Euler: CHECK
8   error: Euler integrator is incorrect
9   comment: why is this integrator stable?
10  error: integration of dynamics is incorrect
11  comment: double check computation of pendulum acceleration
12  regrade: servo converges after several oscillations, borderline control performance
13 Assignment 2 feature 4 (4.00/4): Pendularm_VelocityVerlet: CHECK
14   error: integration of dynamics is incorrect
15   comment: double check computation of pendulum acceleration
16   regrade: working
17 Assignment 2 feature 5 (4.00/4): Pendularm_PID: CHECK
18   error: integration of dynamics is incorrect
19   comment: modulo correction is not needed
20 Assignment 3 feature 6 (2.00/2): FK_MatrixRoutines: CHECK
```

ob.github.io

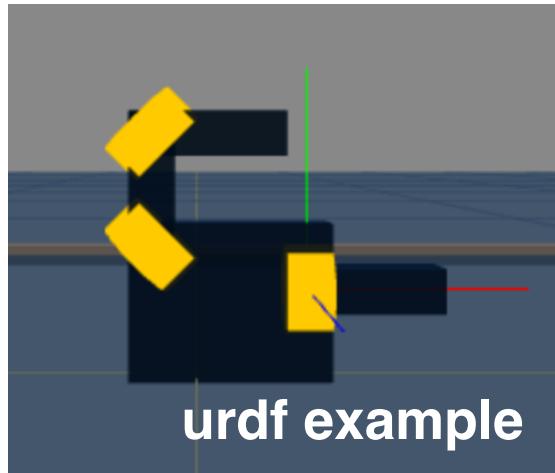
KinEval code stencil



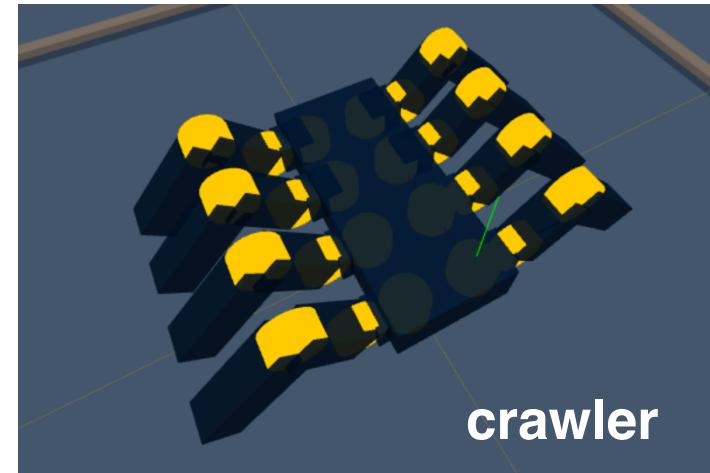
UM EECS 398/567 - autorob.github.io



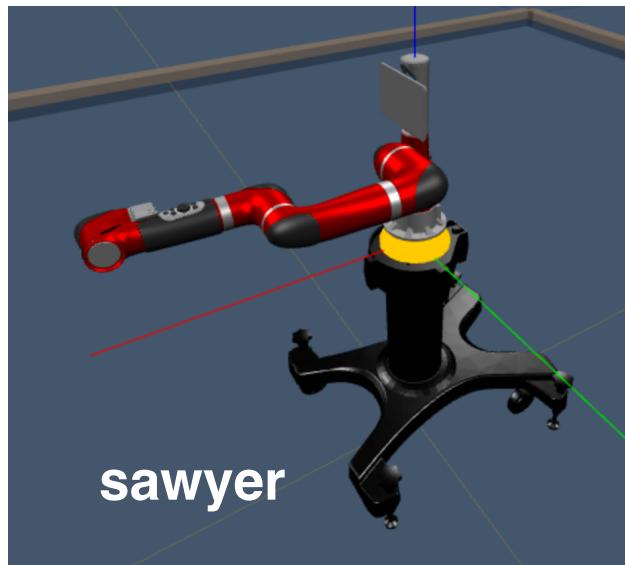
mr2 (default)



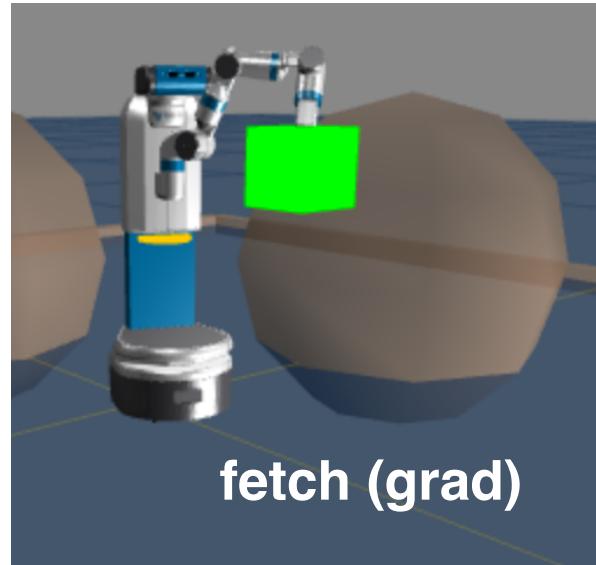
urdf example



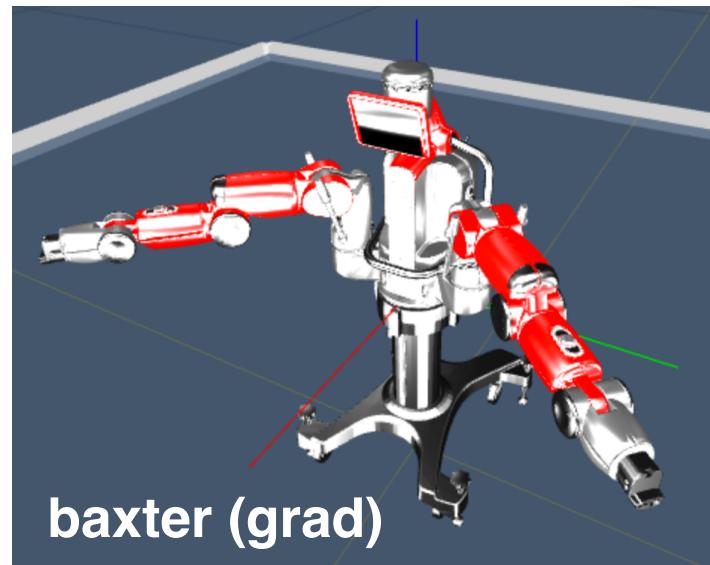
crawler



sawyer



fetch (grad)



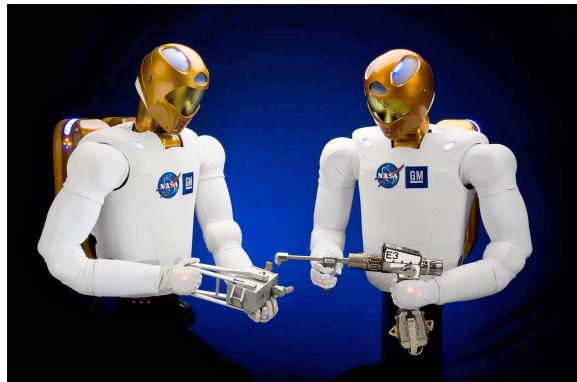
baxter (grad)

KINNOVA

MOVO^{BETA}

A MOBILE
MANIPULATOR
PLATFORM

kinovaMOVO.com

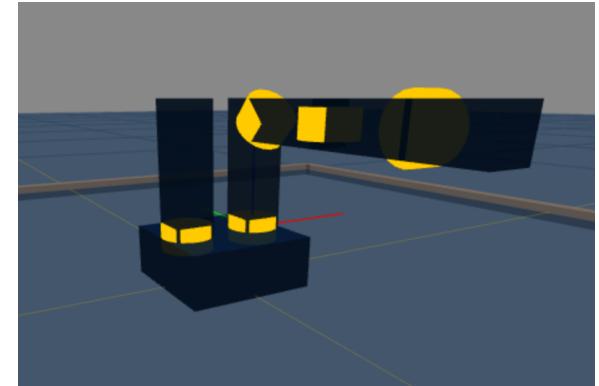


Maybe also?

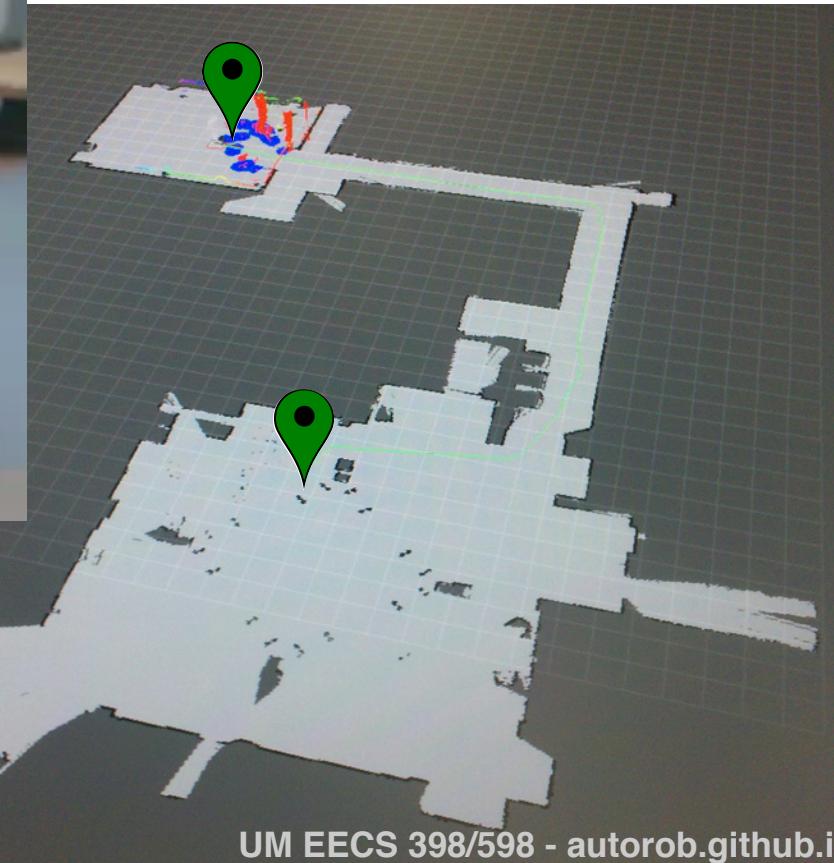


KinEval code stencil

- Code stencil for AutoRob projects in 3D
- Uses threejs 3D rendering library and WebGL
- URDF-like robot description
- Usable, but not perfect, camera and UI controls
- AABB collision detection provided for planning
- Warning: professor-level coding



Let's start with base navigation

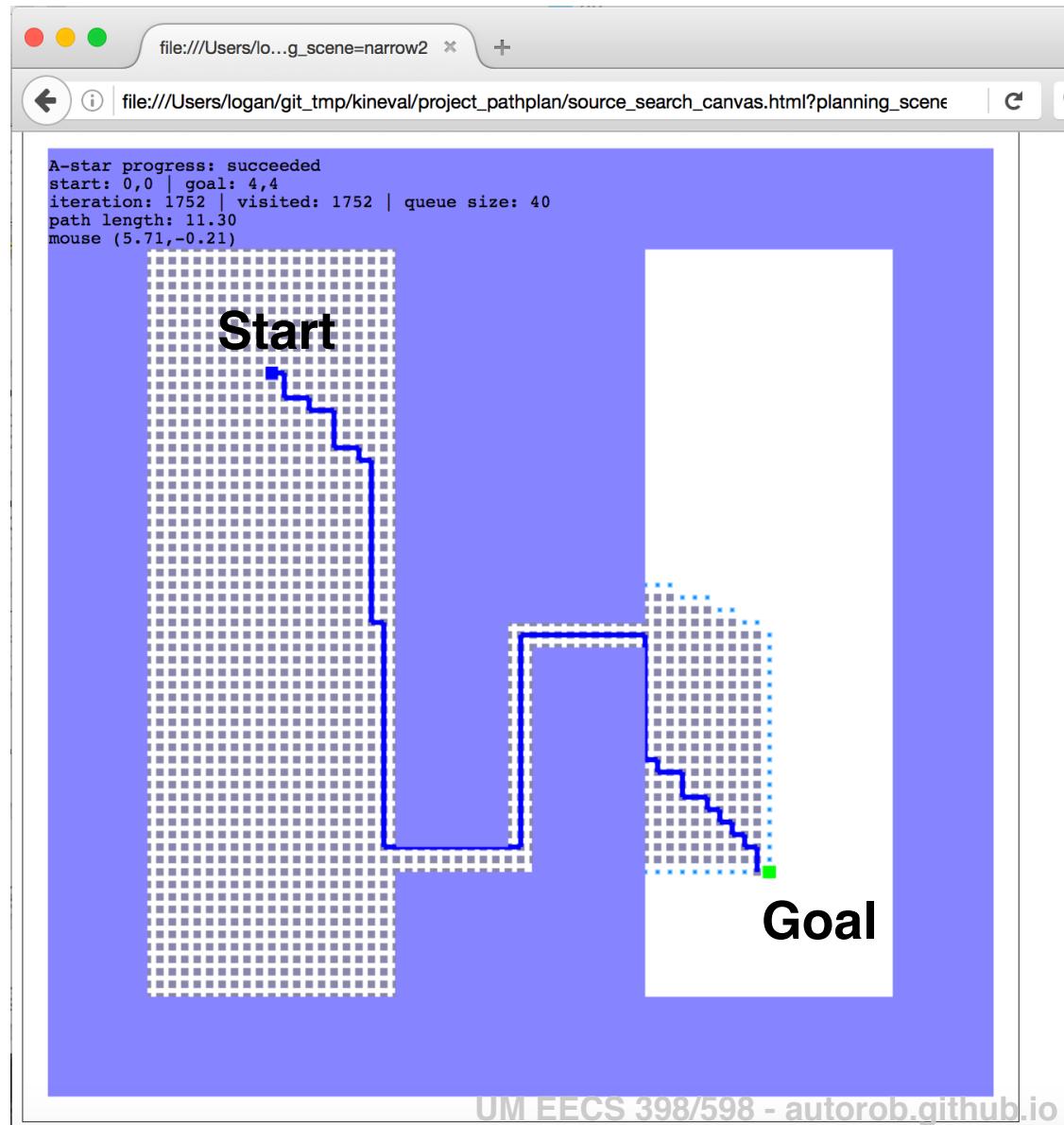


Let's start with base navigation

How to get from Location A to
Location B?

Project 1: 2D Path Planning

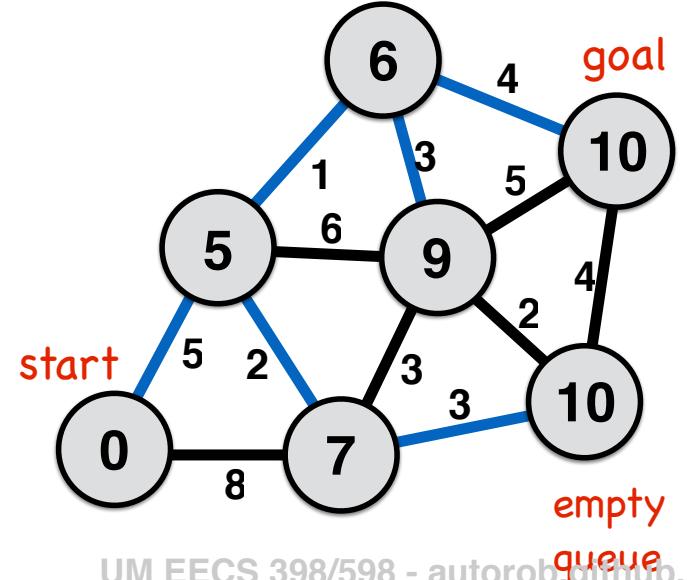
- A-star algorithm for search in a given 2D world
- Implement in JavaScript/HTML5
- Heap data structure for priority queue
- Grad: DFS, BFS, Greedy
- Submit through your git repository



Dijkstra shortest path algorithm

```
all nodes  $\leftarrow \{dist_{start} \leftarrow \text{infinity}, parent_{start} \leftarrow \text{none}, visited_{start} \leftarrow \text{false}\}$ 
start_node  $\leftarrow \{dist_{start} \leftarrow 0, parent_{start} \leftarrow \text{none}, visited_{start} \leftarrow \text{true}\}$ 
visit_queue  $\leftarrow \text{start\_node}$ 

while visit_queue != empty && current_node != goal
    cur_node  $\leftarrow \text{min\_distance(visit\_queue)}$ 
    visitedcur_node  $\leftarrow \text{true}$ 
    for each nbr in not_visited(adjacent(cur_node))
        enqueue(nbr to visit_queue)
        if distnbr > distcur_node + distance(nbr,cur_node)
            parentnbr  $\leftarrow \text{current\_node}$ 
            distnbr  $\leftarrow dist_{cur\_node} + distance(nbr,cur\_node)$ 
        end if
    end for loop
end while loop
output  $\leftarrow \text{parent, distance}$ 
```



A-star shortest path algorithm

```
all nodes  $\leftarrow \{dist_{start} \leftarrow \text{infinity}, parent_{start} \leftarrow \text{none}, visited_{start} \leftarrow \text{false}\}$   
start_node  $\leftarrow \{dist_{start} \leftarrow 0, parent_{start} \leftarrow \text{none}, visited_{start} \leftarrow \text{true}\}$   
visit_queue  $\leftarrow \text{start\_node}$ 
```

```
while (visit_queue != empty) && current_node != goal
```

```
    dequeue: cur_node  $\leftarrow f\_score(\text{visit\_queue})$  ← implement min binary heap  
for priority queue  
    visitedcur_node  $\leftarrow \text{true}$ 
```

```
    for each nbr in not_visited(adjacent(cur_node))
```

```
        enqueue: nbr to visit_queue
```

```
        if distnbr > distcur_node + distance(nbr,cur_node)
```

```
            parentnbr  $\leftarrow \text{current\_node}$ 
```

```
            distnbr  $\leftarrow dist_{cur\_node} + distance(nbr,cur\_node)$ 
```

```
            f_score  $\leftarrow distance_{nbr} + line\_distance_{nbr,goal}$ 
```

```
        end if
```

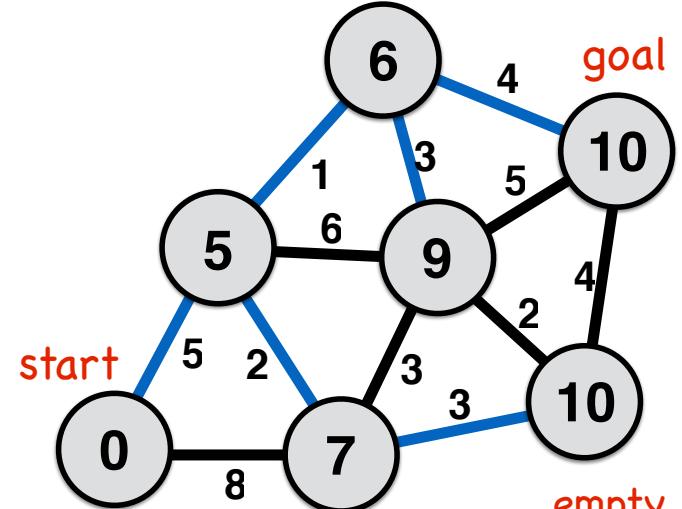
```
    end for loop
```

```
    end while loop
```

```
output  $\leftarrow \text{parent, distance}$ 
```

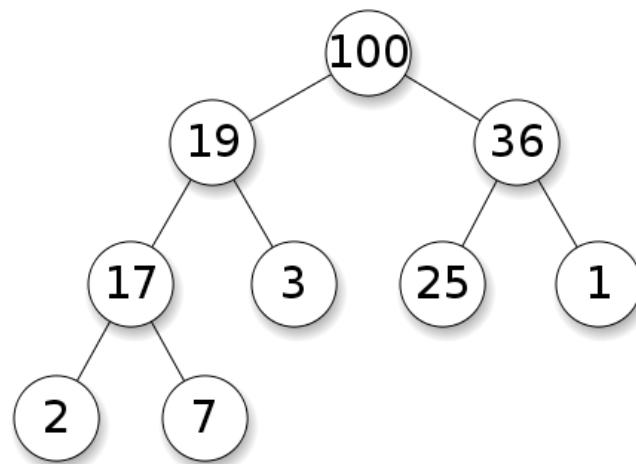
g_score:
distance along path

h_score:
best distance to goal

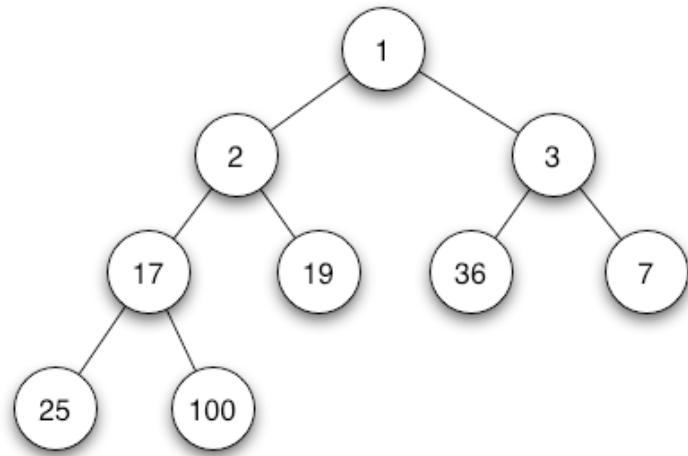


Heaps

A heap is a tree-based data structure satisfying the heap property:
every element is greater (or less) than its children



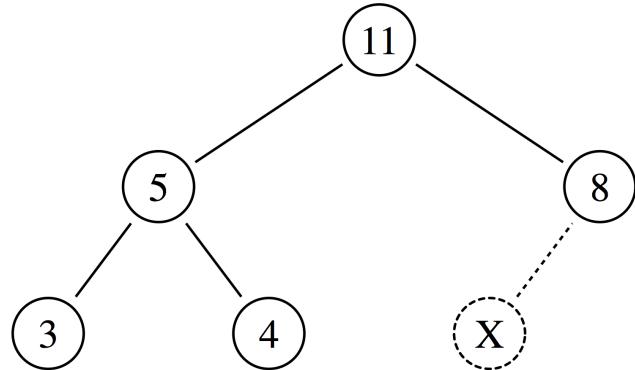
max heap



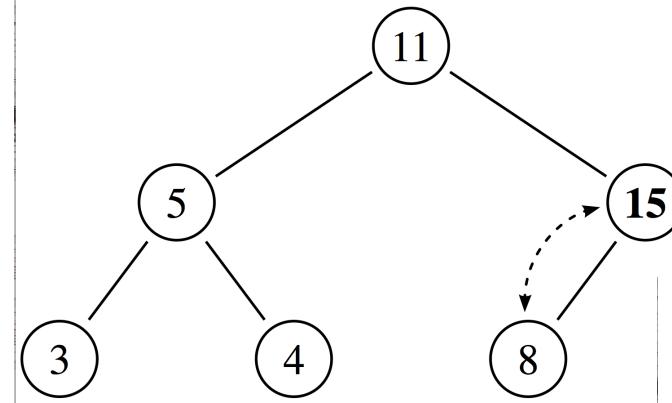
min heap

Heap operations: Insert

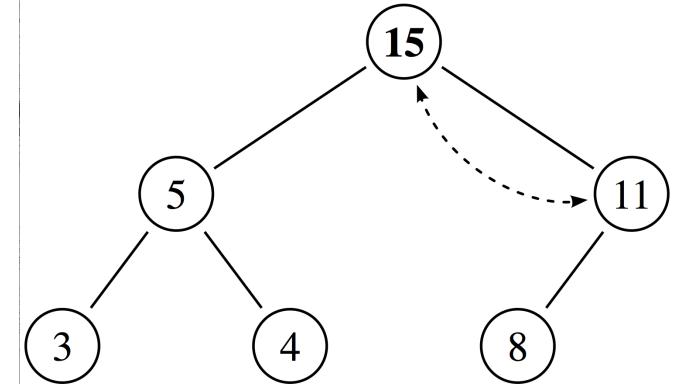
1) add new element to end of tree



2) swap with parent

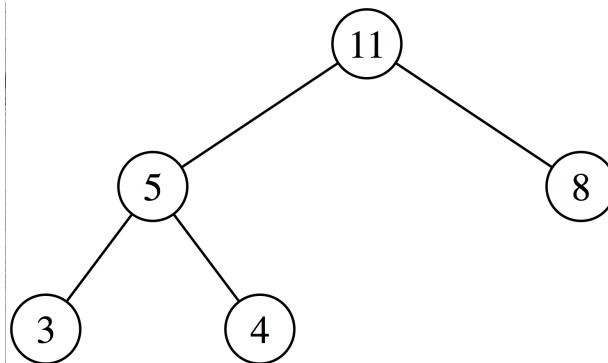


3) until heaped, do (2)

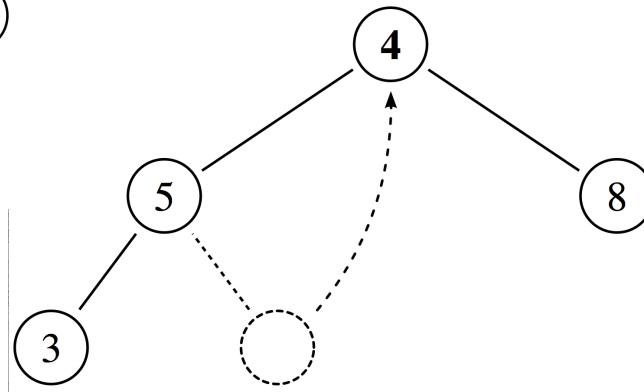


Heap operations: Extract

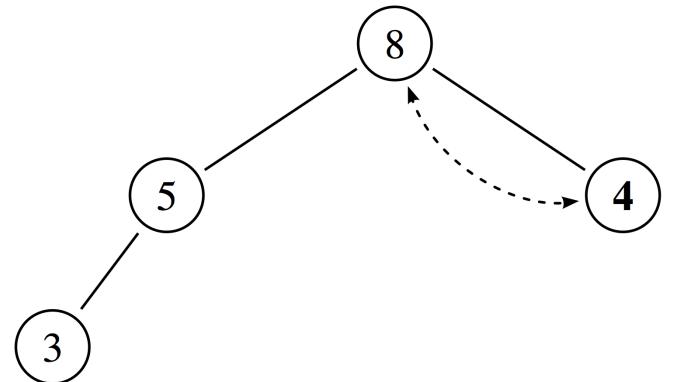
1) extract root element



2) put last element at root



3) swap with smaller child
4) until heaped, do (3)



```
<!DOCTYPE html>                                         heapsort.html  
<html>  
  <head>  
    <title>Heap Sort by Chad Jenkins</title>  
  </head>  
  <body onload=startMeUp()>                         HTML5 document will invoke  
JavaScript function at start  
    <a href="http://ohseejay.org"><h1>My Heap Sort</h1></a>  
    <canvas width=900 height=200 id="myCanvas"></canvas>  
    <div id="output">hello world</div>  
  </body>  
</html>  
  
<script>  
function startMeUp() {  
  .. executable code to perform heapsort; will depend on heap.js ...  
}  
</script>
```

git basics

- Create a git repository from gitlab, github, or bitbucket website
- Install git on your machine
 - <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
 - OSX: <https://code.google.com/p/git-osx-installer/>

git basics

- create a local copy of a repository: `git clone <repo url>`
- add files to a repository: `git add <file listing>`
- commit changes to local repository: `git commit -a -m "<msg>"`
- push local changes to a remote repository: `git push`
- pull remote changes to a local repository: `git fetch` or `git pull`
- create a code branch in a repository: `git branch <branch name>`
- checkout a code branch from a repository : `git branch <branch name>`
- merge branches in a repository:

```
git checkout <branch name>
```

```
git merge <other branch name>
```

Highly recommended tutorial

The screenshot shows a web browser window for learngitbranching.js.org/?NODEMO. On the left, a terminal window titled "Learn Git Branching" displays a sequence of git commands:

```
$ git commit  
$ git commit  
$ git branch newthing  
$ git checkout newthing  
$ git commit  
$ git commit  
$ git checkout master  
$ git commit -m "grading"  
$ git checkout newthing  
$ git commit -m "more work"  
$ git checkout master  
$ git merge newthing
```

To the right of the terminal is a visual git commit history diagram. The commits are represented by circles labeled C0 through C8. Commit C0 is at the top, followed by C1, C2, and C3. A vertical line of arrows connects C0 to C1, C1 to C2, and C2 to C3. Below this main vertical line, commit C4 is connected to C5, which is connected to C7. Commit C6 is also connected to C7. Commit C8 is connected to both C7 and C6. An orange arrow points from the text "newthing" to commit C7. A pink arrow points from the text "master*" to commit C8. A "Fork me on GitHub" button is visible in the top right corner of the diagram area.

<http://learngitbranching.js.org/>

UM EECS 398/567 - autorob.github.io

Can I enroll in this class?

- Most likely! Probably yes!
- We will try to enroll all who want to take the course
- Additional office hours to discuss enrollment issues
 - Thursday (tomorrow) 3-5pm in Beyster 3644
 - Friday 11am-12pm in Beyster 3644

What I need from you **now**

- Accept invitation to the course discussion channel (coming tonight)
 - <https://autorob.slack.com>
- Install git and setup your working environment
 - create a git repository: <https://gitlab.eecs.umich.edu/>
 - ensure you can clone, commit, and push files to your repository
- Over the discussion channel, send me:
 - informal introduction confirming your name, email, and enrollment
 - your favorite picture of yourself
 - pointer to your git repo for the course

What I need from you **next week**

- Get started on Assignment 1 (Path Planning)
 - Clone kineval-stencil-fall18 repository (release this weekend)
 - Study examples in tutorial_js subdirectory
 - Complete tutorial_heapsort (noted with “STENCIL” in files)
 - Complete project_pathplan (concepts covered next week)