Initialization

EECS 367
Intro. to Autonomous Robotics

ROB 320
Robot Operating Systems

Winter 2022

Agenda

- Introduction
- So, where is my robot?
- Course administrative overview
- Action items: what I need from you now
  - Student workflow survey, Join autorob Slack and GitHub Classroom
- Assignment 1 (Path Planning) released, due January 21, 11:59pm

Course Staff

- Instructor: Anthony Opipari (topipari)
  - OH: Wednesday 3-5pm, Friday 3:30-5:30pm

- IA: Tommy Cohn (cohnt)

- IA: Ana Warner (aswarner)

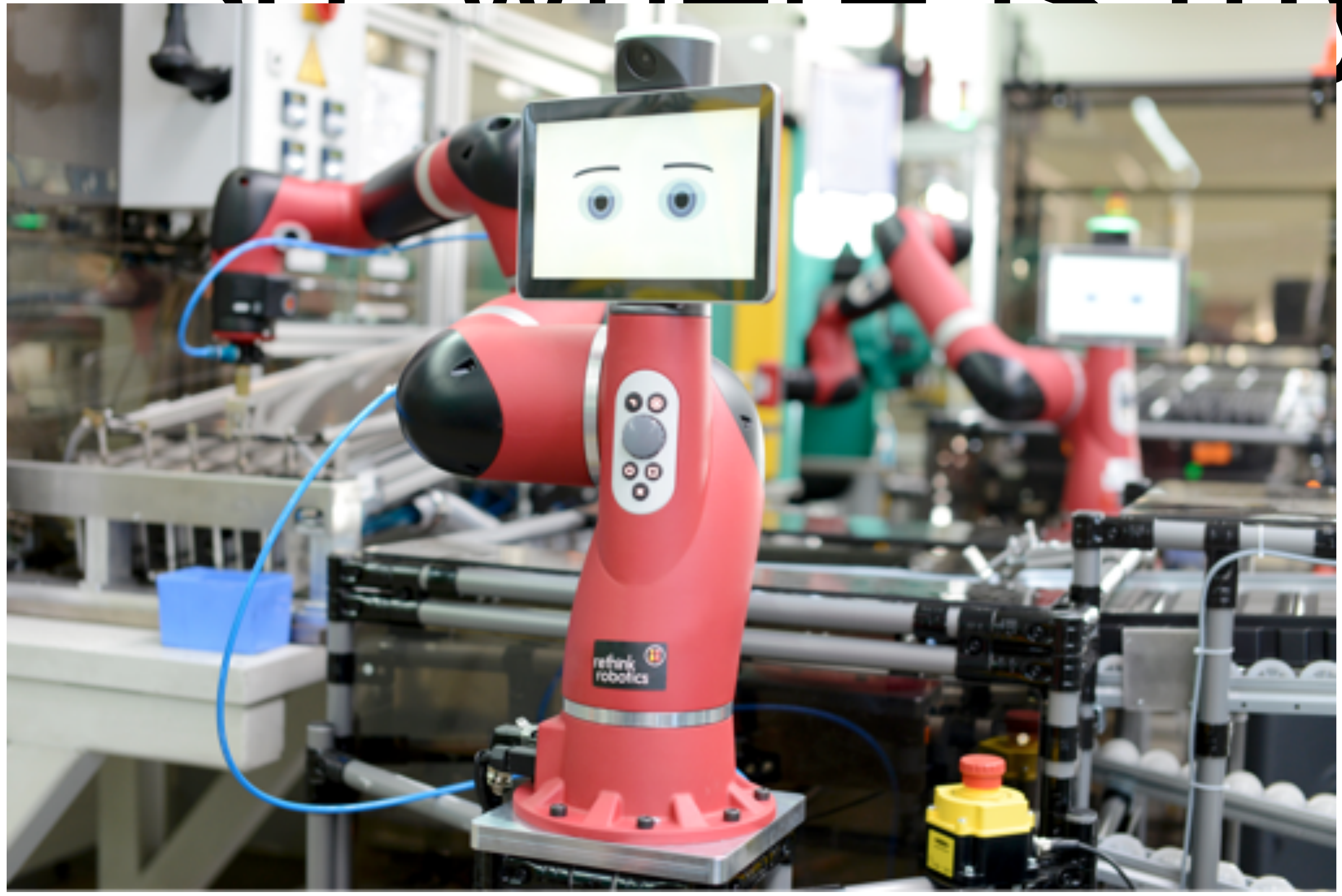
- Faculty Advisor: Chad Jenkins (ocj)

About me

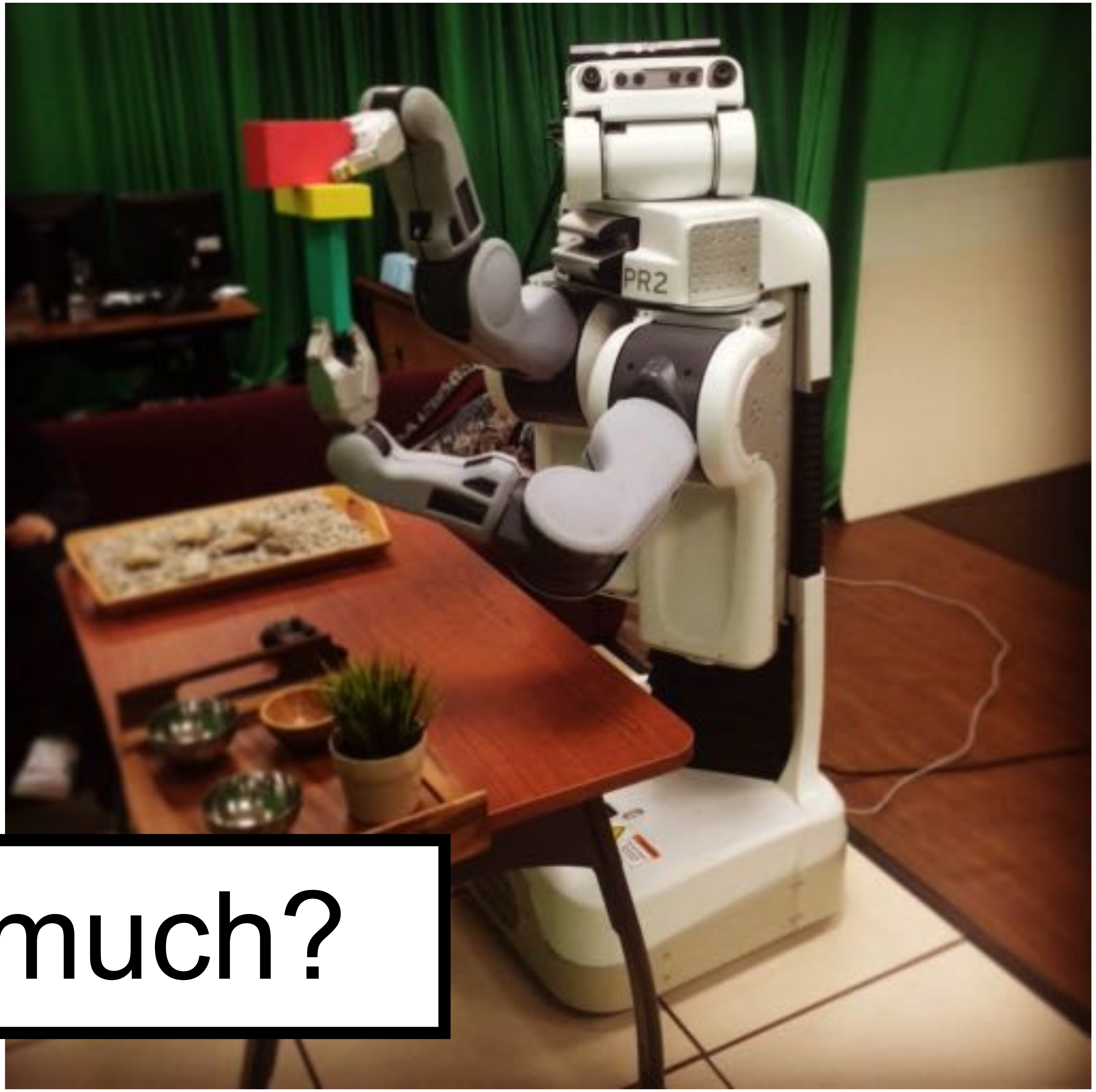
- Born and raised in Ann Arbor
- BSE and MSE degrees in computer science from UM
- PhD Student, CSE program
  - Research interests in probabilistic reasoning and perception for robotics
  - Excited about robotics as the frontier for computing in the physical world

So, where is my robot?

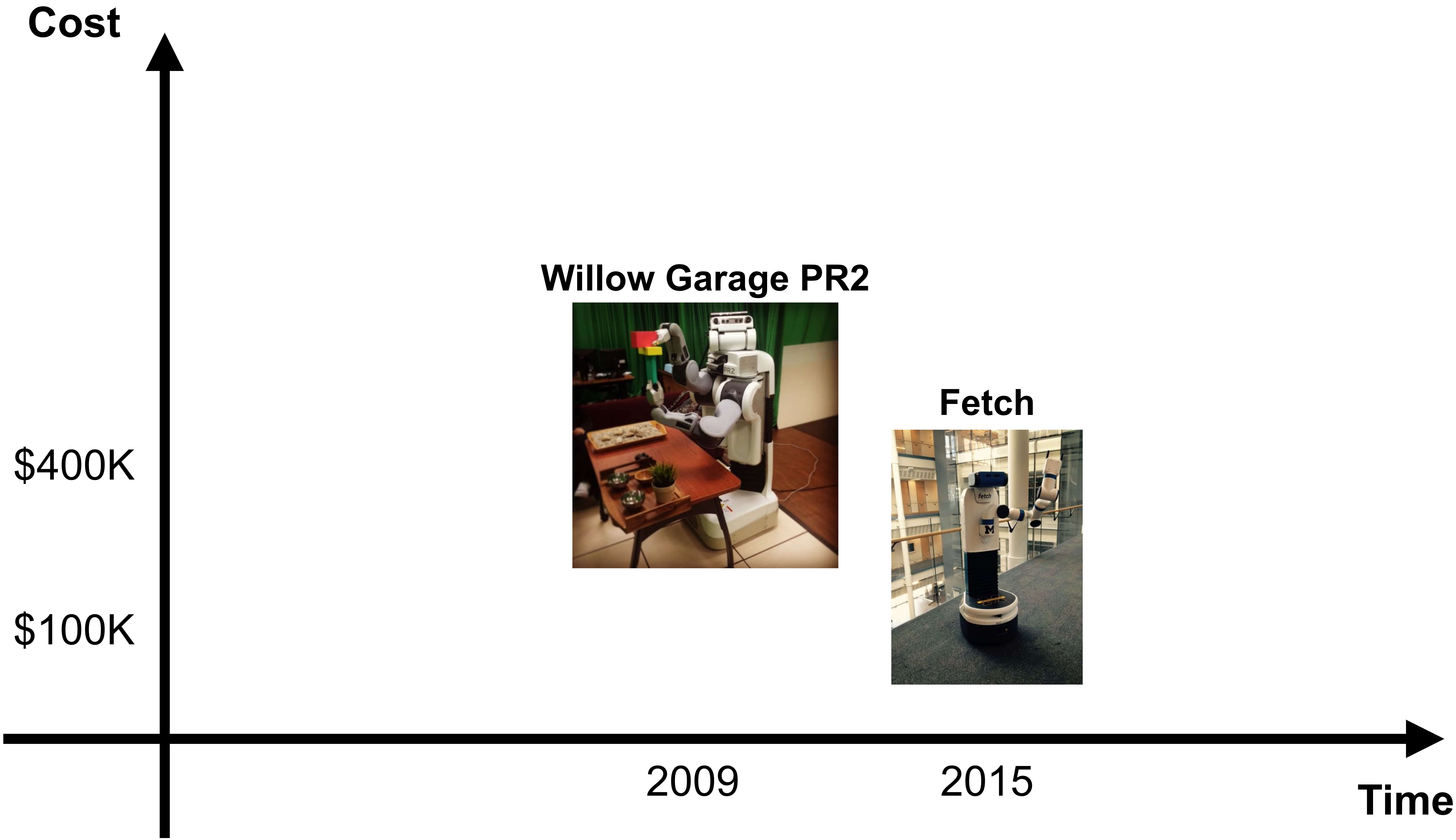
So, where is my robot?



Mobile Manipulation Robots



How much?


Cost
Willow Garage PR2
Fetch
$400K
$100K
2009
2015
Time

Cost

2002

Willow Garage PR2

Fetch

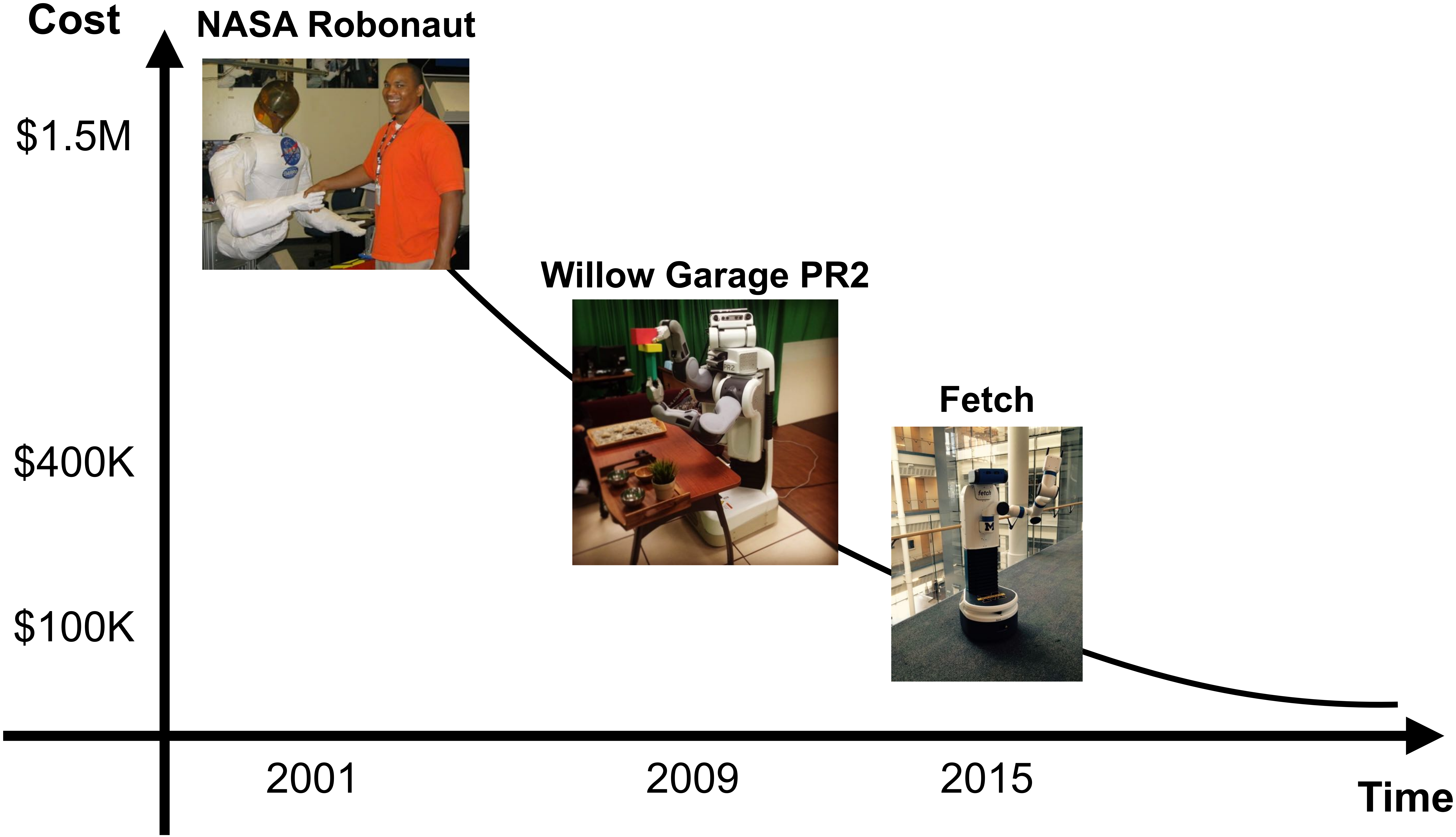
$400K

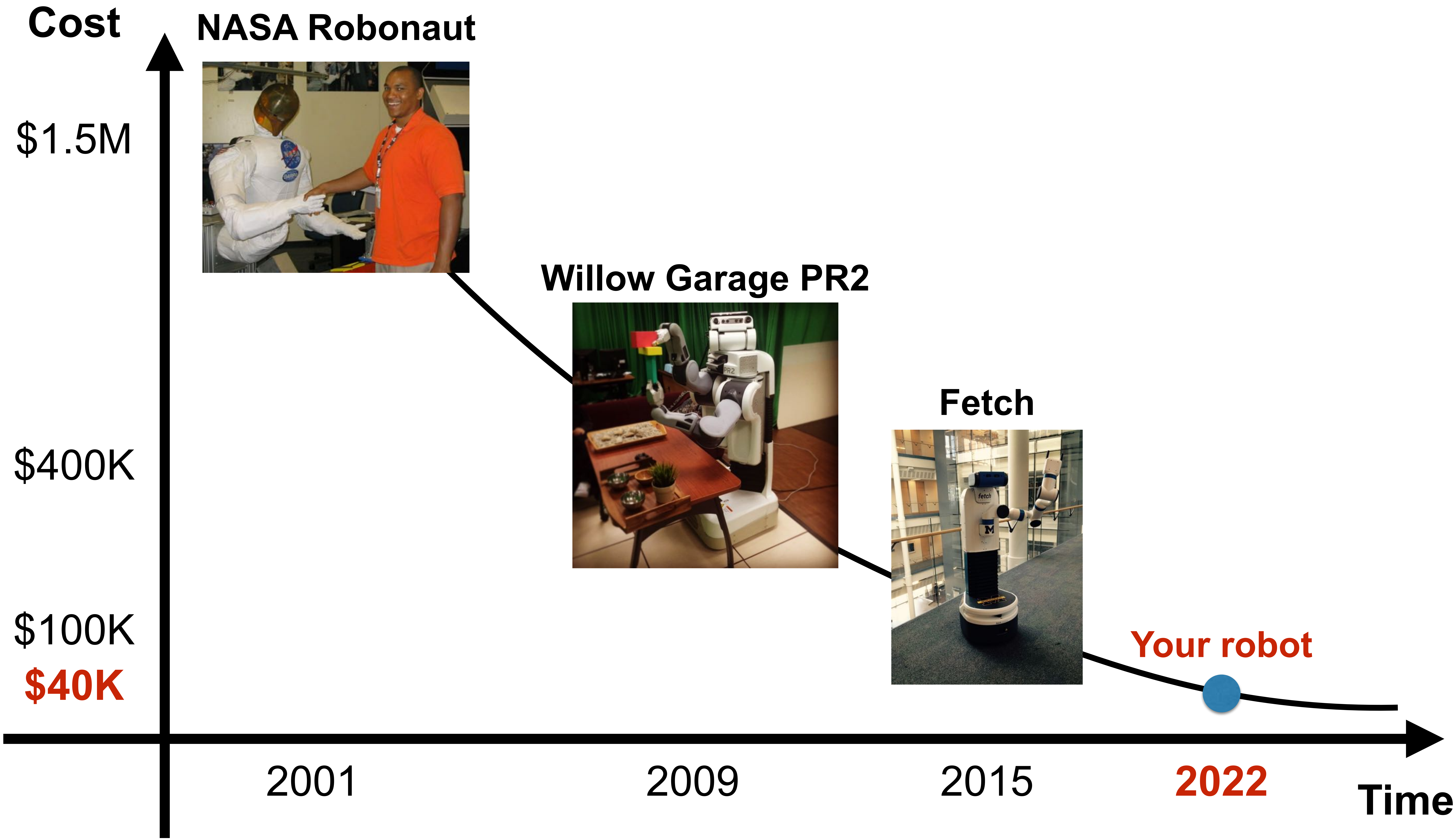
$100K

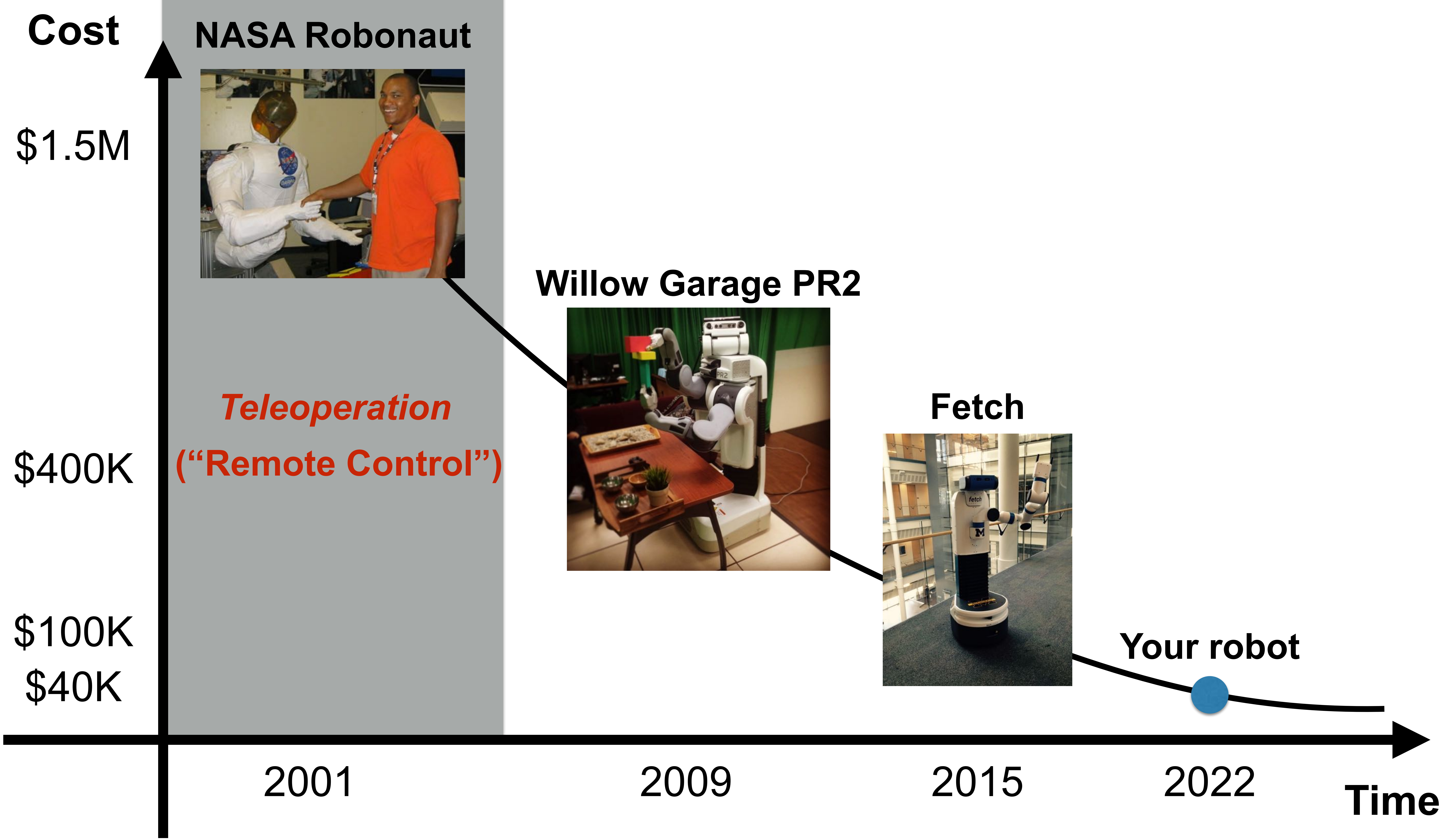
2009

2015

Time


Cost
NASA Robonaut
$1.5M
Willow Garage PR2
Fetch
$400K
$100K
2001
2009
2015
Time


Cost
NASA Robonaut
$1.5M
Willow Garage PR2
$400K
Fetch
$100K
$40K
Your robot
2001
2009
2015
2022
Time


Cost
NASA Robonaut
$1.5M
Teleoperation
("Remote Control")
$400K
$100K
$40K
Willow Garage PR2
Fetch
Your robot
2001
2009
2015
2022
Time

Cost

$1.5M

$400K

$100K

$40K

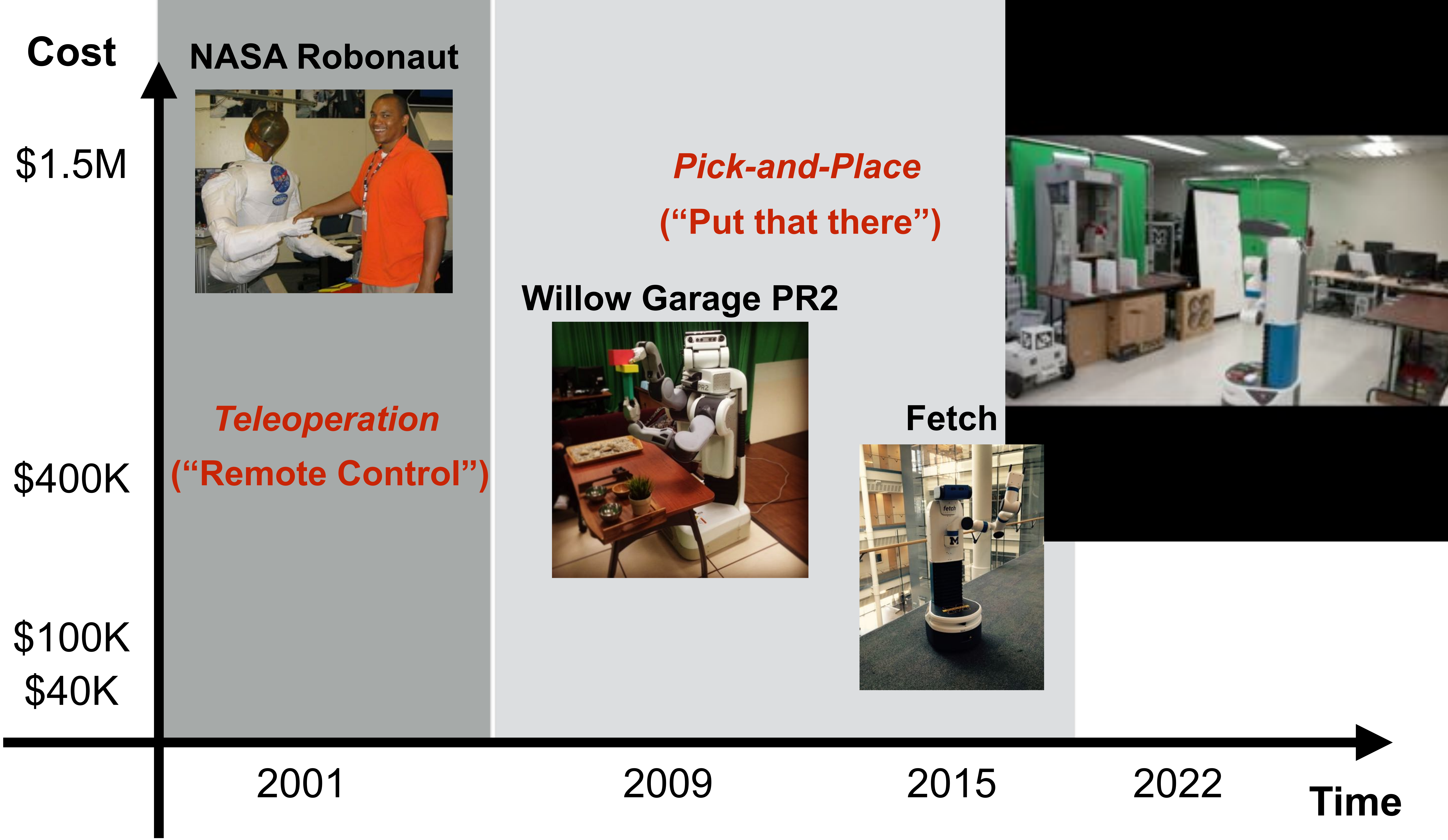
NASA Robonaut

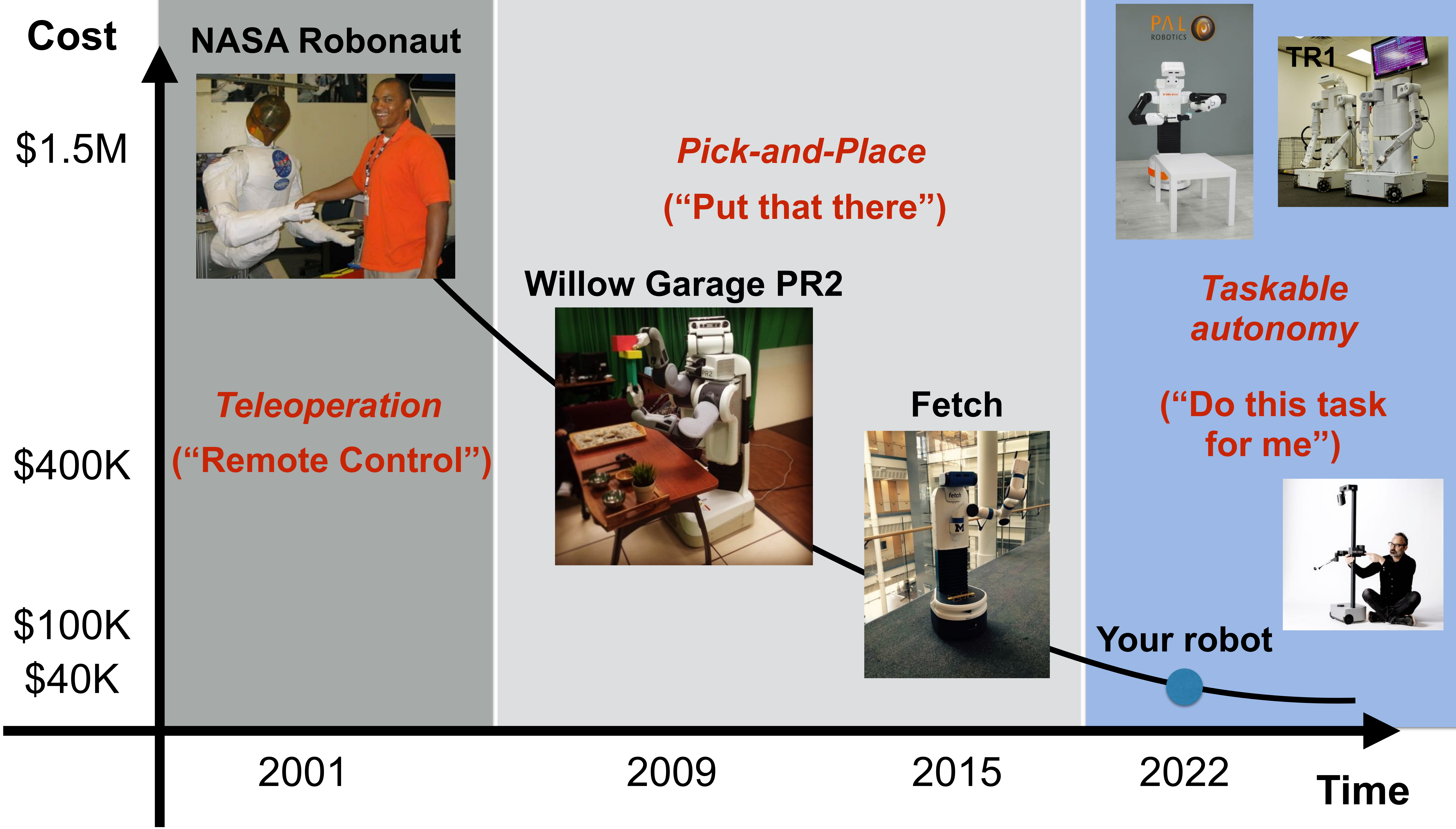
Teleoperation

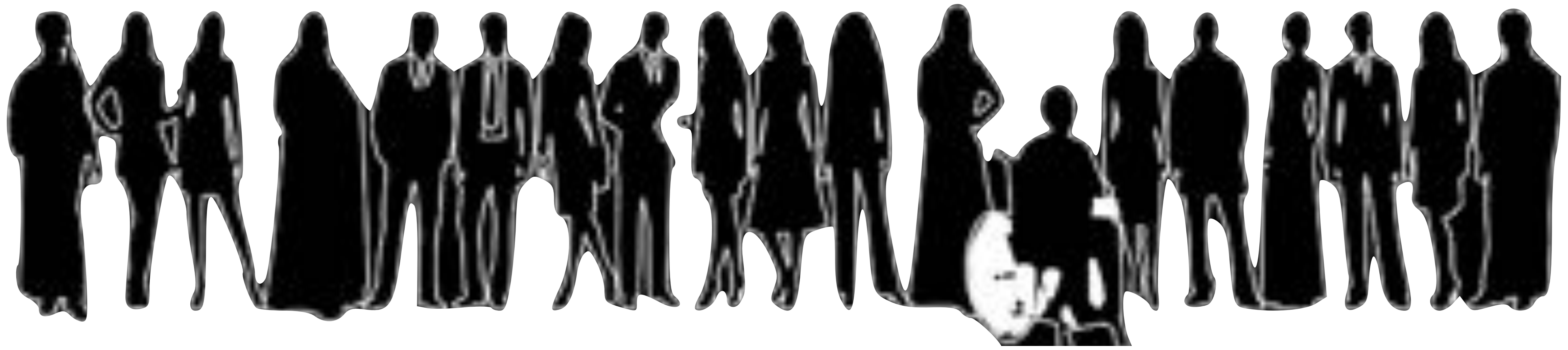
(“Remote Control”)

2001 2009 2015 2022

Time


Cost
$1.5M
$400K
$100K
$40K
NASA Robonaut
Teleoperation
("Remote Control")
Pick-and-Place
("Put that there")
Willow Garage PR2
Fetch
2001
2009
2015
2022
Time

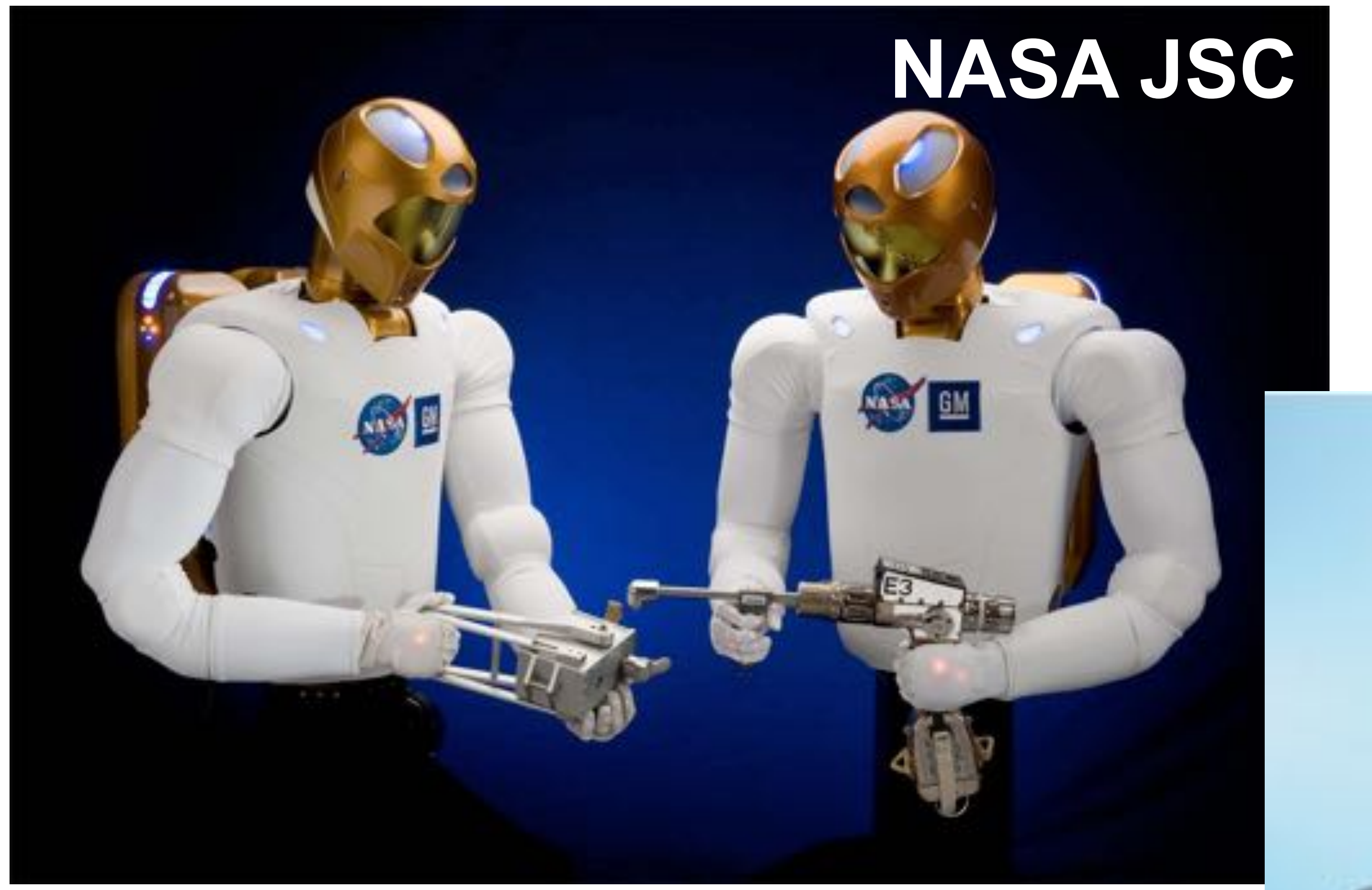

Cost
$1.5M
$400K
$100K
$40K
NASA Robonaut
Teleoperation
("Remote Control")
Pick-and-Place
("Put that there")
Willow Garage PR2
Fetch
PAL ROBOTICS
TR1
Taskable autonomy
("Do this task for me")
Your robot
2001
2009
2015
2022
Time



Taskable autonomy

Pick-and-Place

Teleoperation



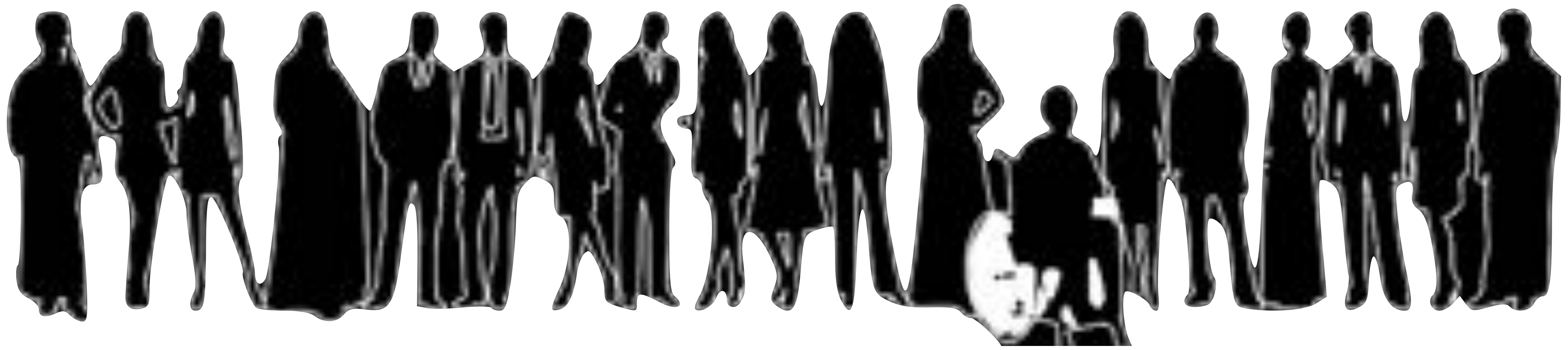
NASA JSC

TU Munich

Dexterous Manipulation

Harvard/Wyss

TRACLabs



Taskable autonomy

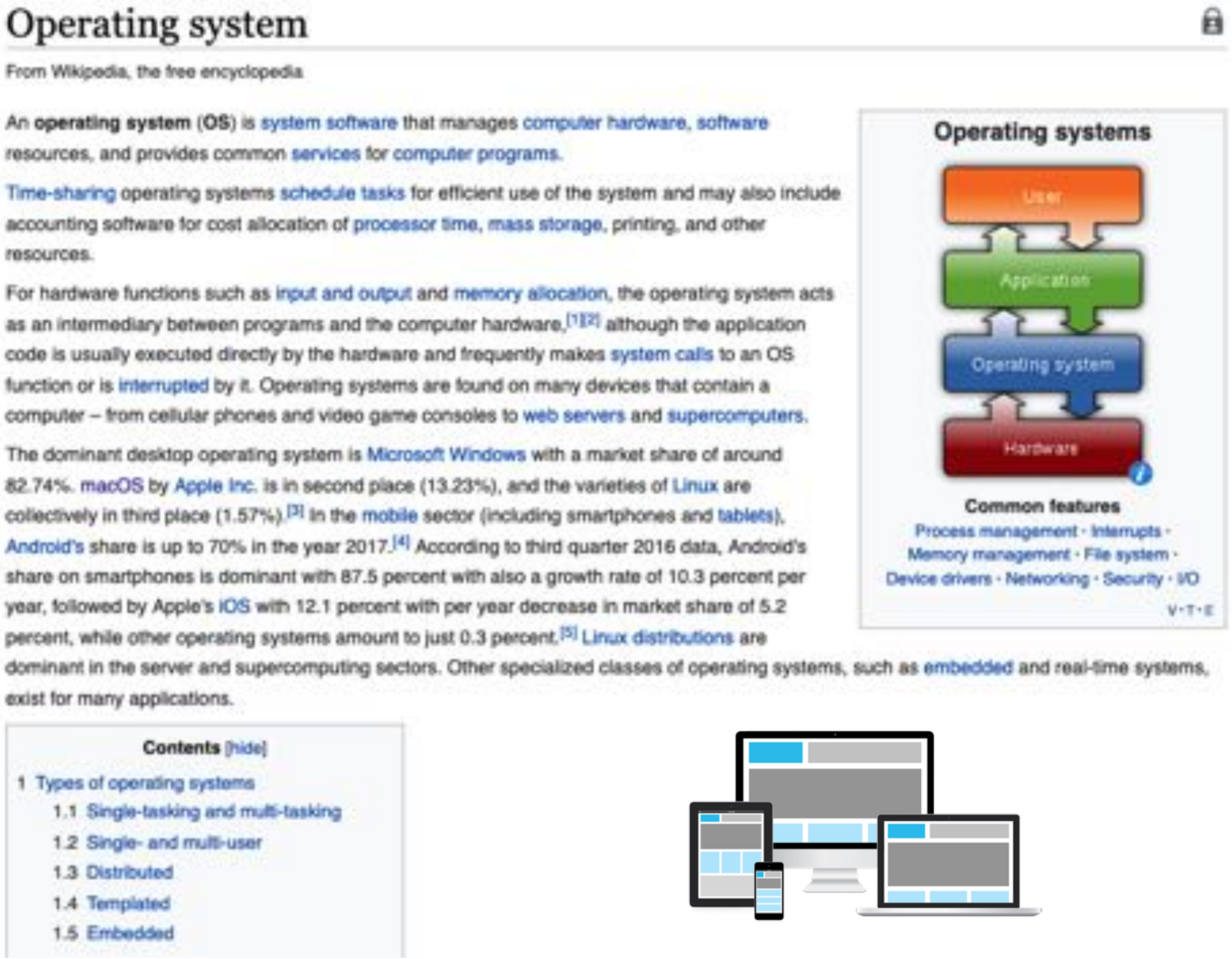
Dexterous Manipulation

Teleoperation

Taskable autonomy

Dexterous Manipulation

Teleoperation



Operating system

From Wikipedia, the free encyclopedia

An **operating system** (**OS**) is system software that manages computer hardware, software resources, and provides common services for computer programs.

Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, printing, and other resources.

For hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between programs and the computer hardware,[1][2] although the application code is usually executed directly by the hardware and frequently makes system calls to an OS function or is interrupted by it. Operating systems are found on many devices that contain a computer – from cellular phones and video game consoles to web servers and supercomputers.

The dominant desktop operating system is Microsoft Windows with a market share of around 82.74%. macOS by Apple Inc. is in second place (13.23%), and the varieties of Linux are collectively in third place (1.57%).[3] In the mobile sector (including smartphones and tablets), Android's share is up to 70% in the year 2017.[4] According to third quarter 2016 data, Android's share on smartphones is dominant with 87.5 percent with also a growth rate of 10.3 percent per year, followed by Apple's iOS with 12.1 percent with per year decrease in market share of 5.2 percent, while other operating systems amount to just 0.3 percent.[5] Linux distributions are dominant in the server and supercomputing sectors. Other specialized classes of operating systems, such as embedded and real-time systems, exist for many applications.

Operating systems

Common features

Process management · Interrupts · Memory management · File system · Device drivers · Networking · Security · I/O

V · T · E

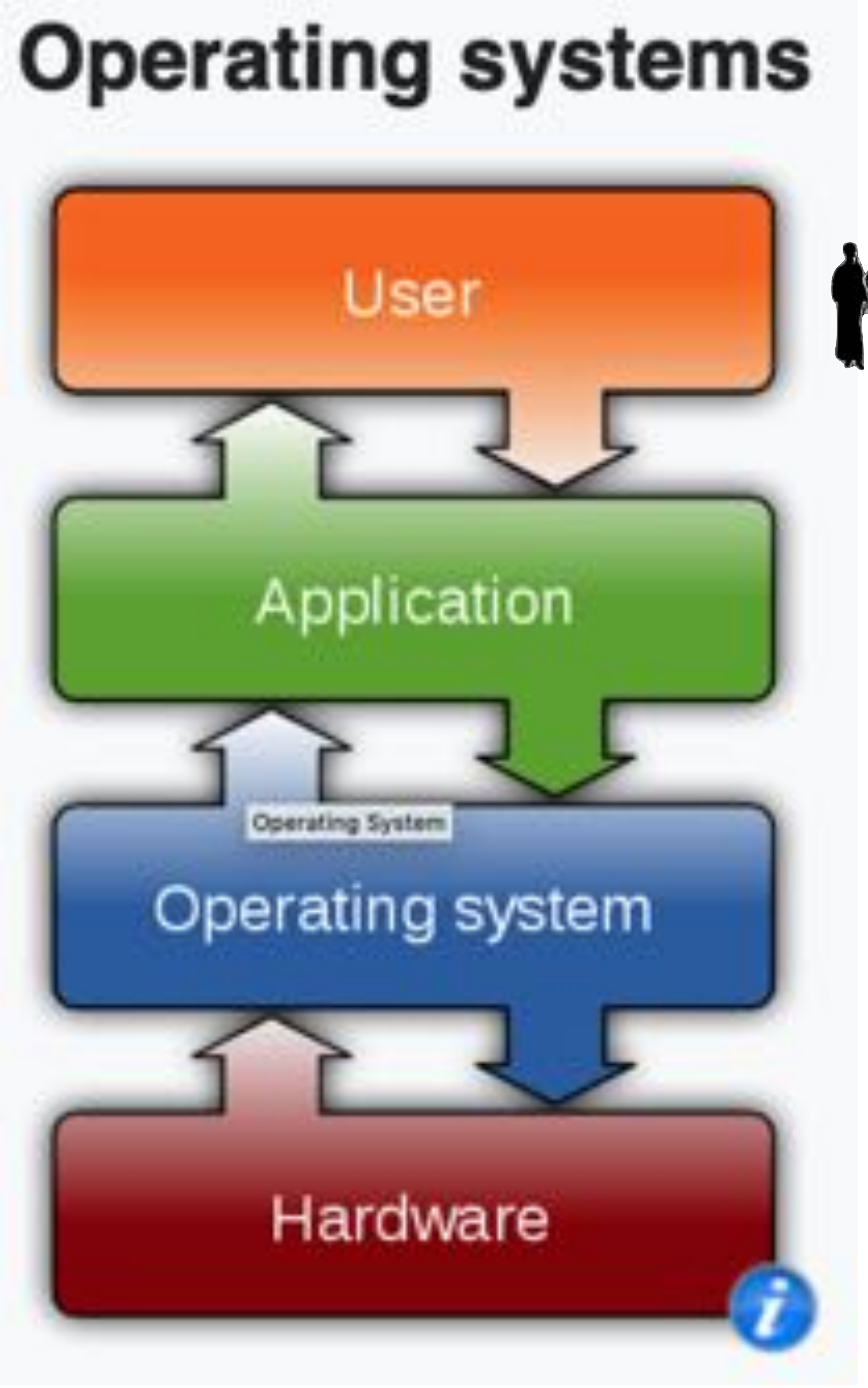
Contents [hide]

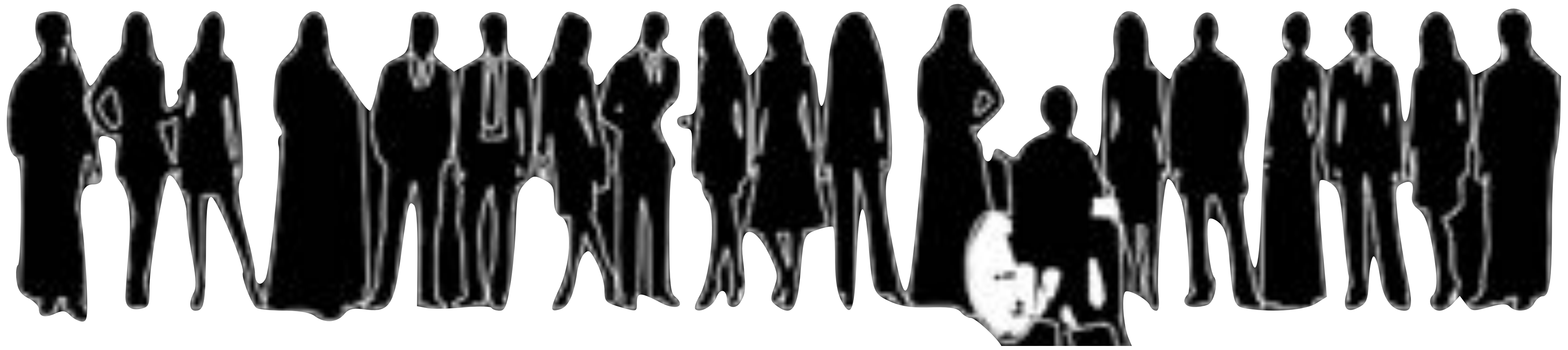
Taskable autonomy

Dexterous Manipulation

Teleoperation

An **operating system (OS)** is system software that manages computer hardware, software resources, and provides common services for computer programs.





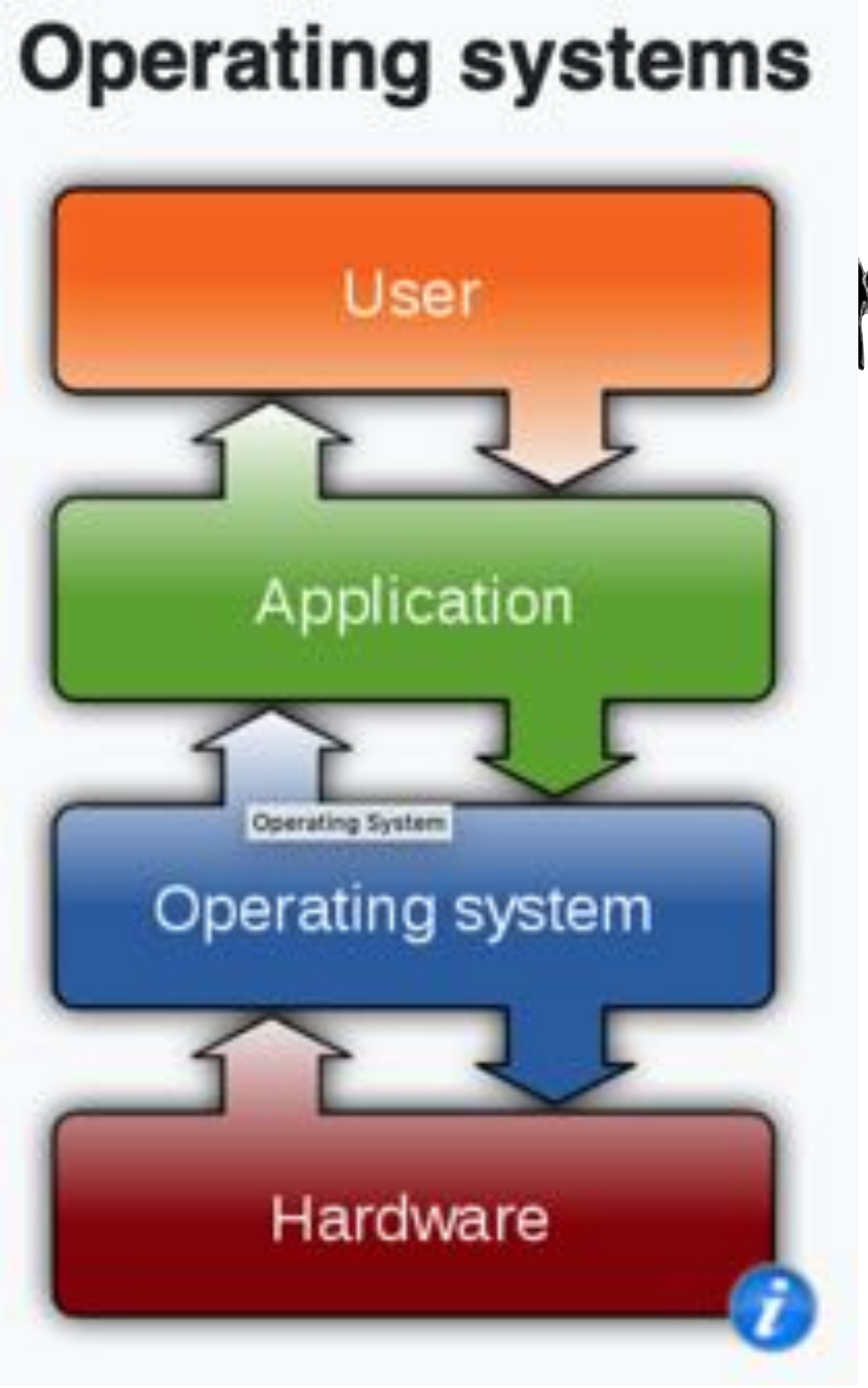
Taskable autonomy

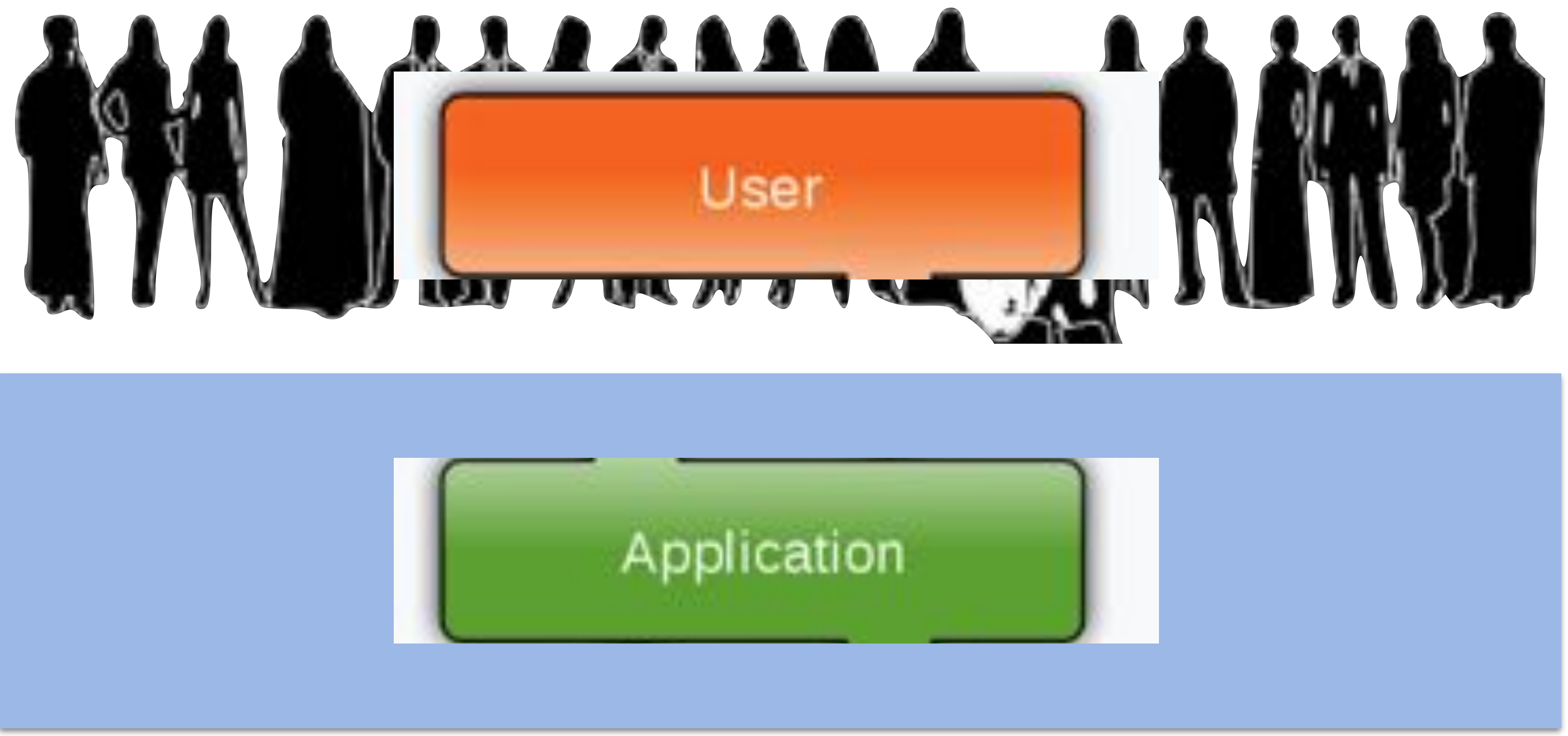
Dexterous Manipulation

Teleoperation

An **operating system (OS)** is a special program that runs on the bare machine and hides the gory details of managing processes and devices.

- https://perldoc.perl.org/perlglossary.html#operating-system

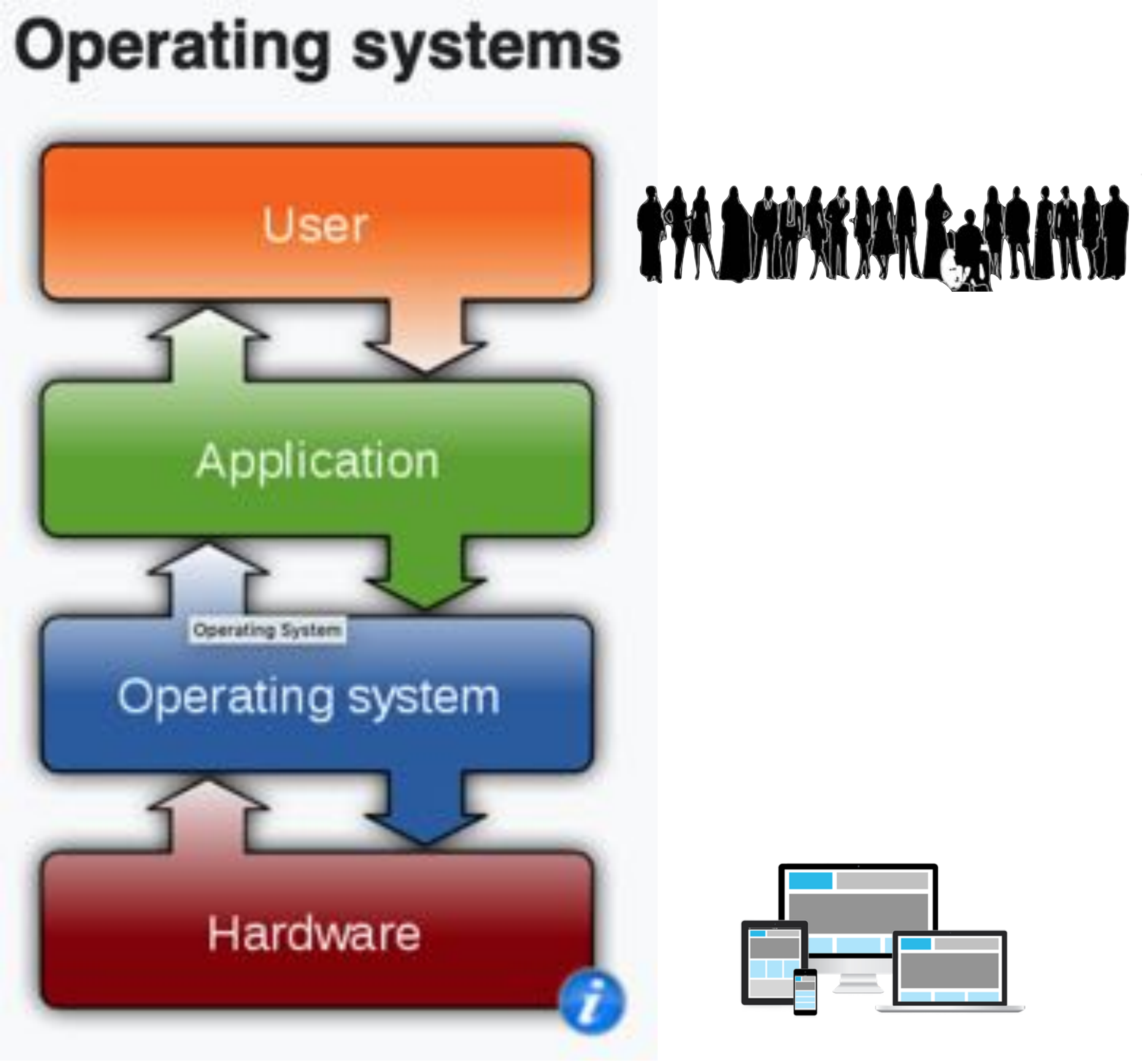




Dexterous Manipulation

An **operating system (OS)** is
a special program that runs on the bare machine and hides the gory details of managing processes and devices.

- https://perldoc.perl.org/perlglossary.html#operating-system



Users

Robot Applications

Dexterous Manipulation

Then, what is this?

Operating System

Hardware

Users

Robot Applications

Robot Operating System

Operating System

Hardware

A **robot operating system (robot OS)** is a special program that runs on the operating system and hides the gory details of controlling robot devices, autonomy processes, and sensorimotor routines.

This abstraction provides a platform for robot applications to run seamlessly across a wide variety of robots capable of mobility and/or dexterous manipulation.

Users

Robot Applications

Robot Operating System

Operating System

Hardware

Users

Robot Applications

Robot Operating System

Operating System

Hardware

Users

Robot Applications

Robot Operating System

LCM ROS Player YARP MOOS

Operating System

Hardware

Users

Robot Applications

Robot Operating System

LCM ROS Player YARP MOOS10

Operating System

Hardware

Users

Robot Applications

Then, what is this?

Robot Operating System

LCM

ROS

Player

YARP

MOOS

Operating System

Hardware

Users

Robot Applications

Someday in the Future…

"Do this task for me"

Robot Operating System

LCM ROS Player YARP MOOS

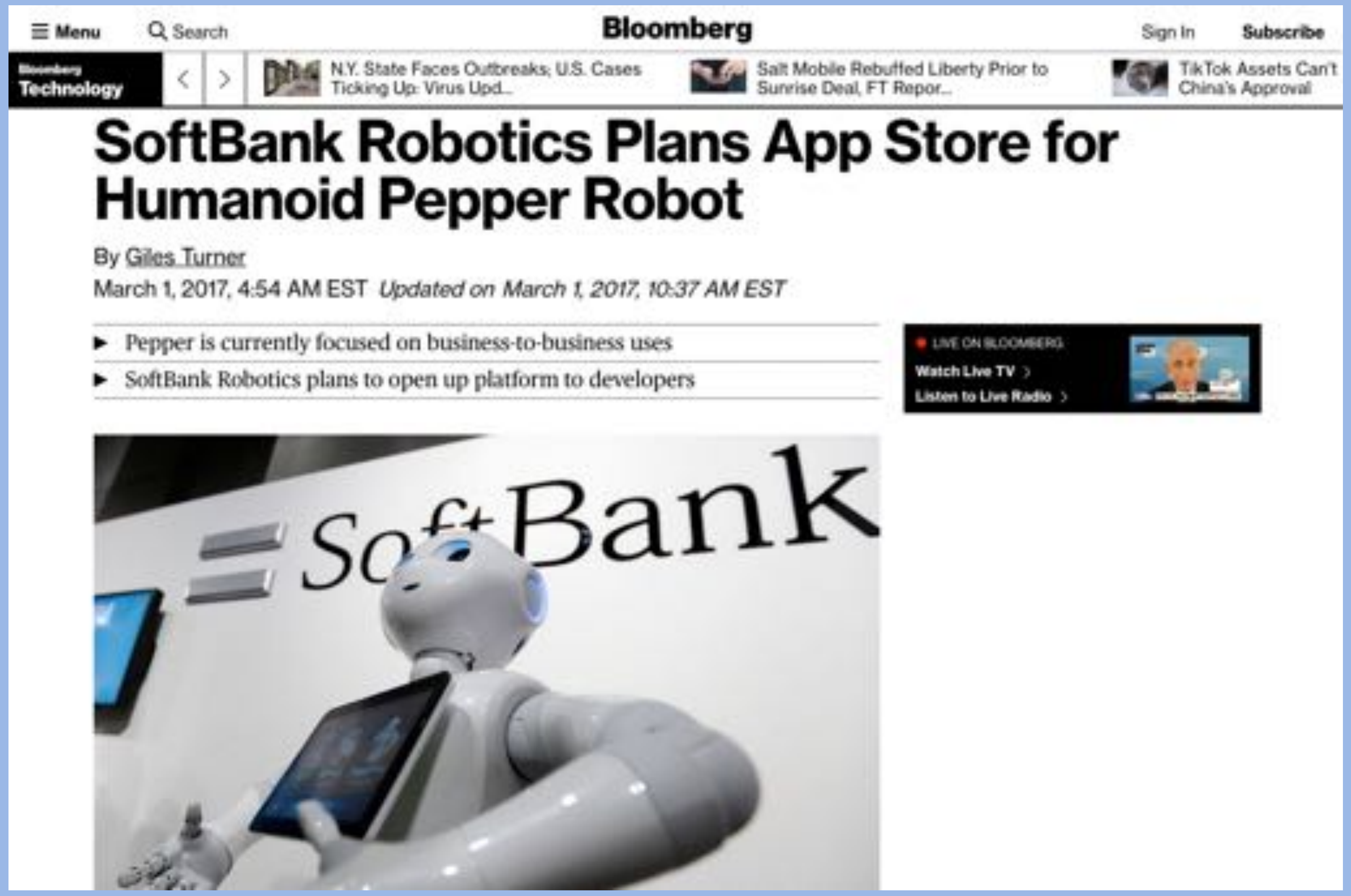
Operating System

Hardware

“Do this task for me”

Can we make your

world programmable ?



Can we make a **robot app store** ?

What's a robot app?

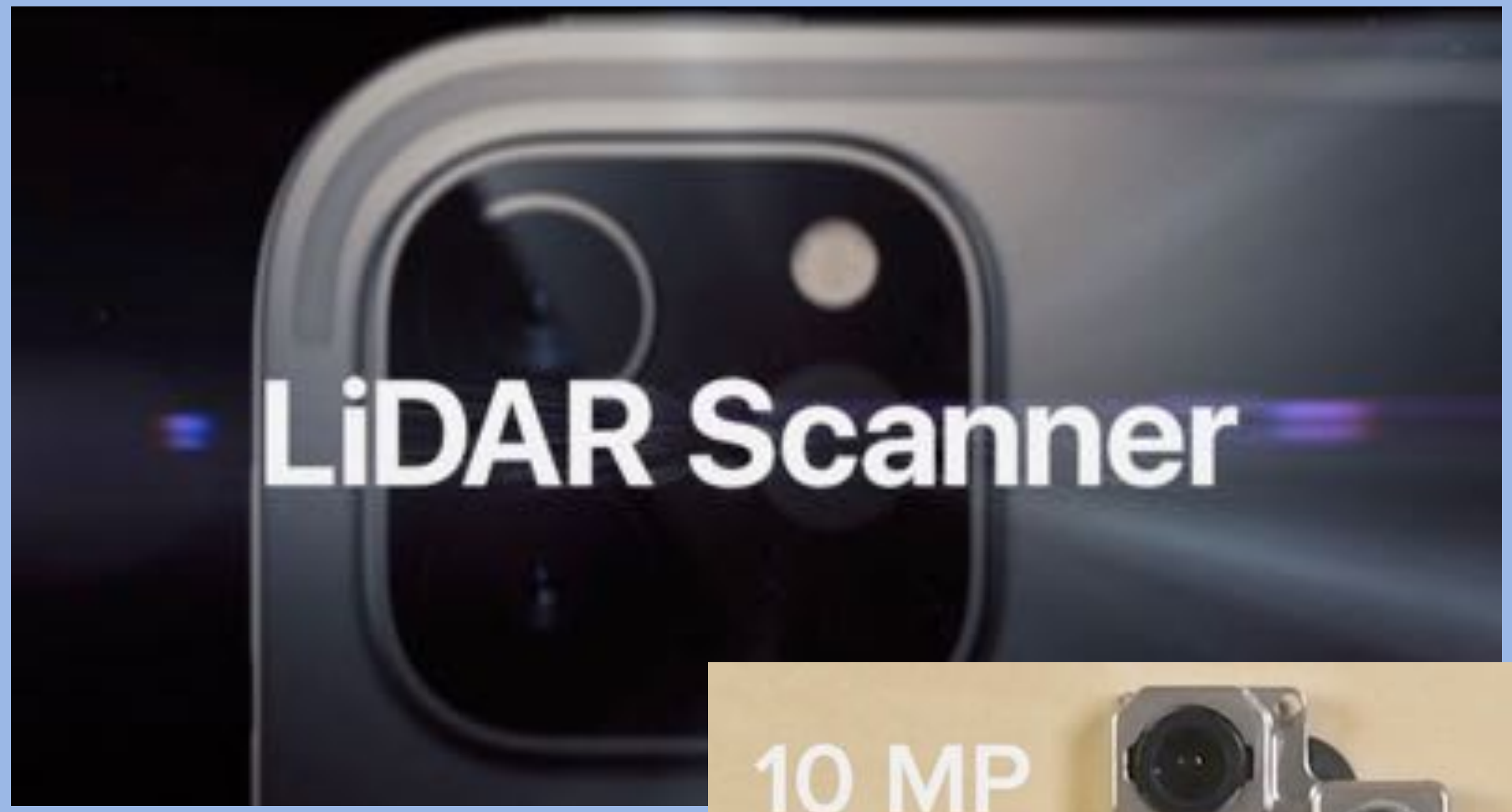
- In the near future
- Eventually:
  - CleanTheHouse
  - PatrolTheBuilding
  - ...
- For now:
  - demonstrations
  - experiments
  - challenge entries (!)

https://www.ros.org/presentations/2009-07_Gerkey_IJCAI-robotics-workshop.pdf

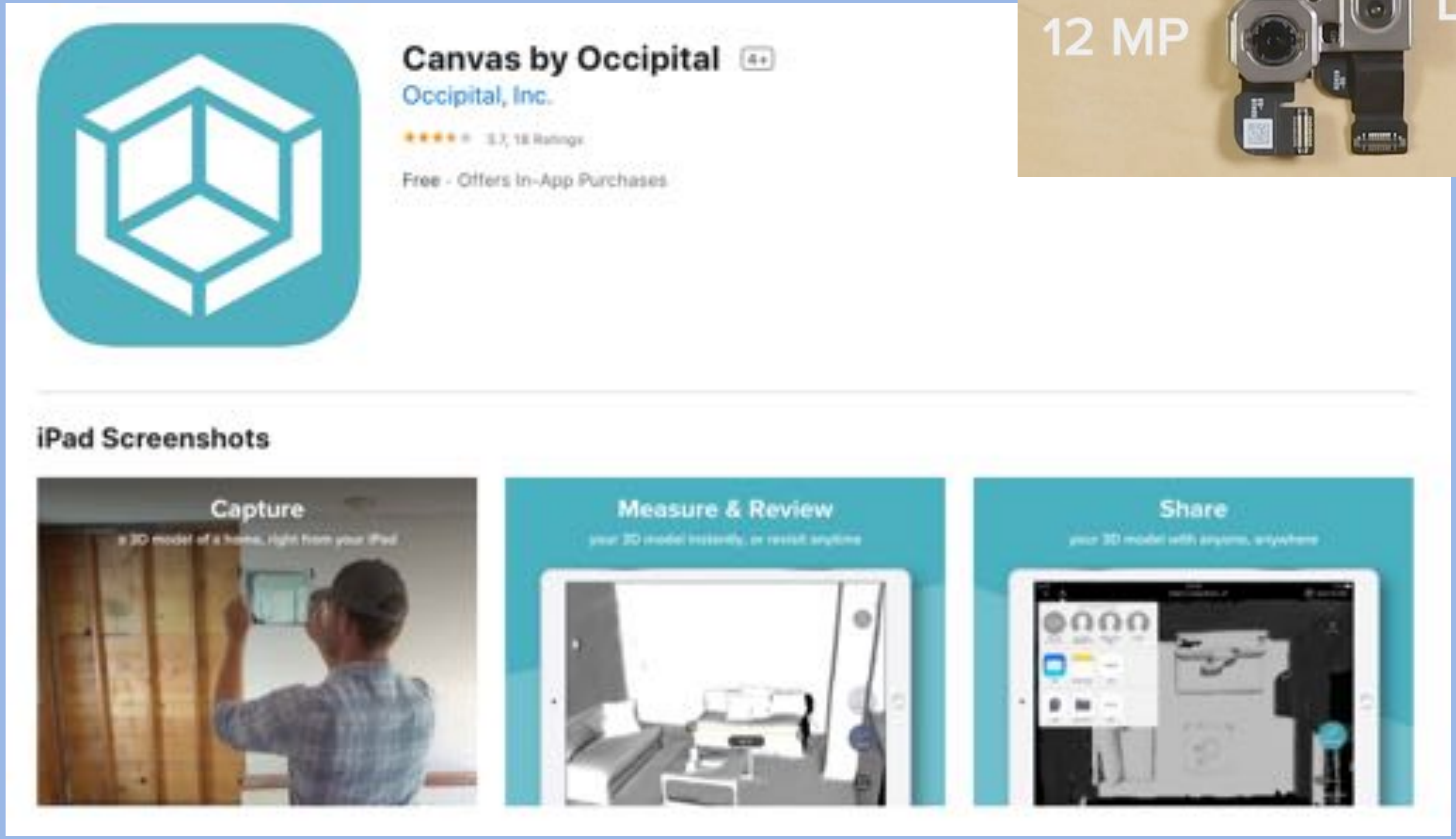
2009


MapIt!
Autonomous exploration and mapping for any indoor environment.
Click to Buy ($49.99)

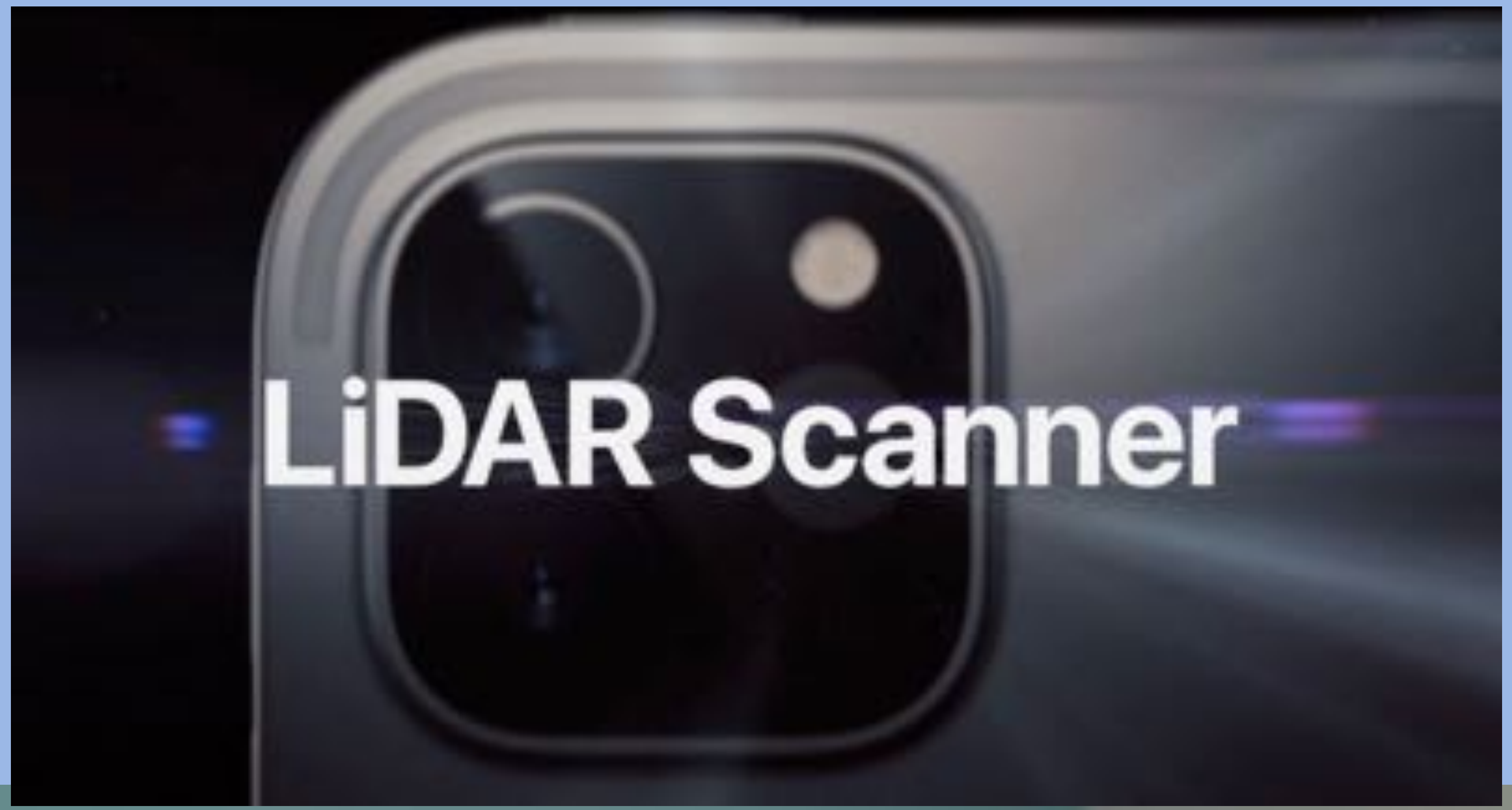
2022

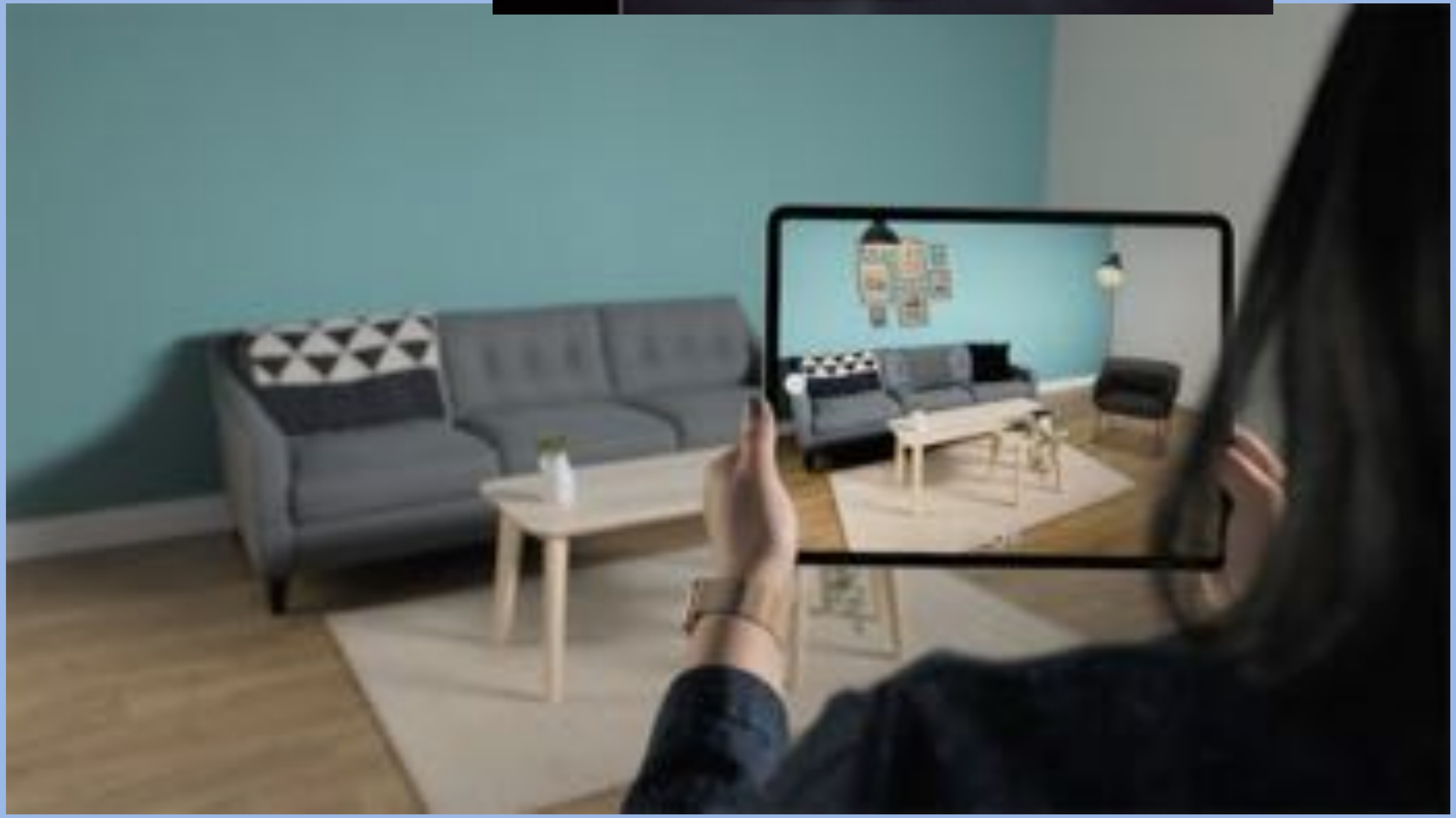

LiDAR Scanner

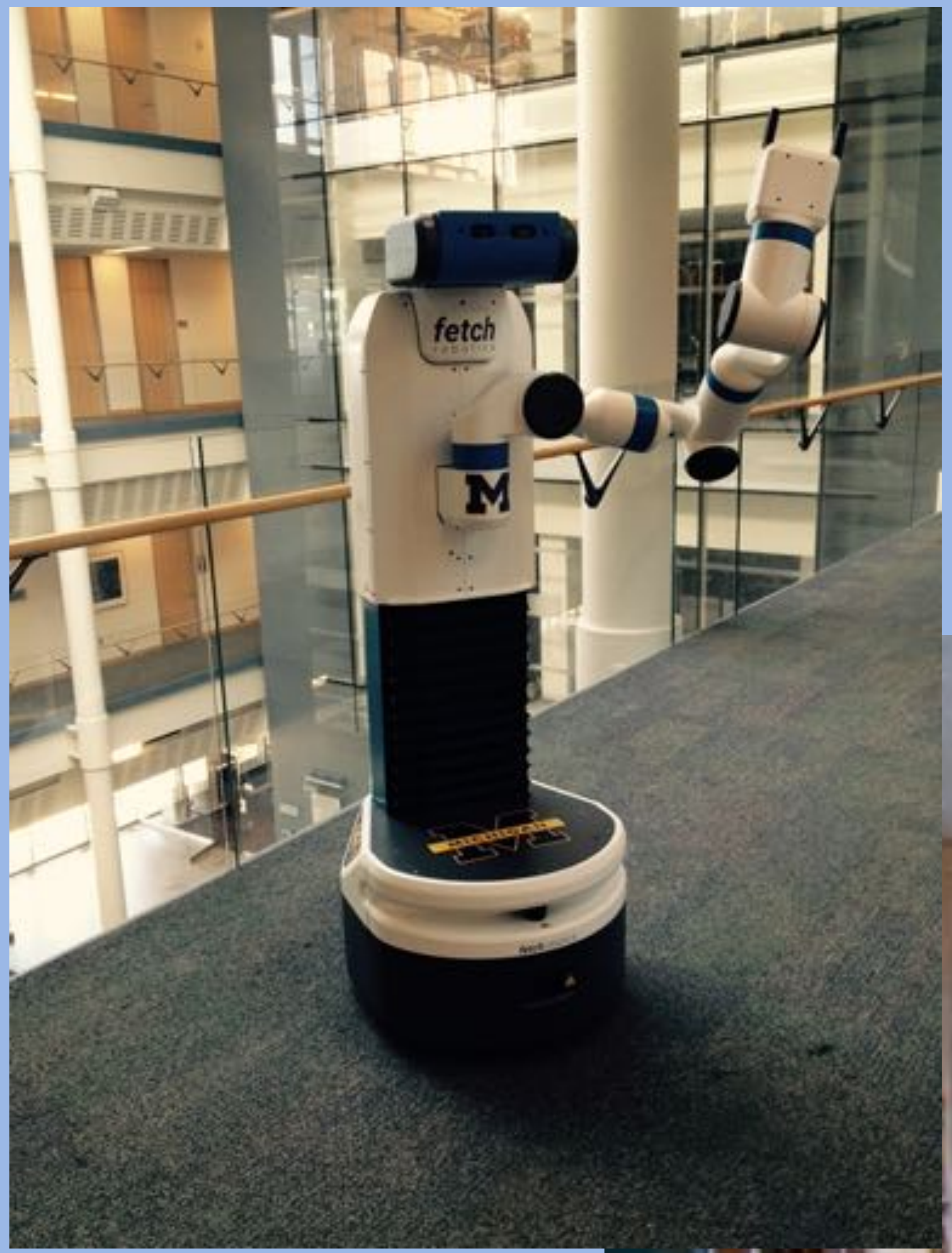

10 MP
12 MP
LiDAR


Canvas by Occipital 4+
Occipital, Inc.
Free · Offers In-App Purchases
iPad Screenshots
Capture
Measure & Review
Share

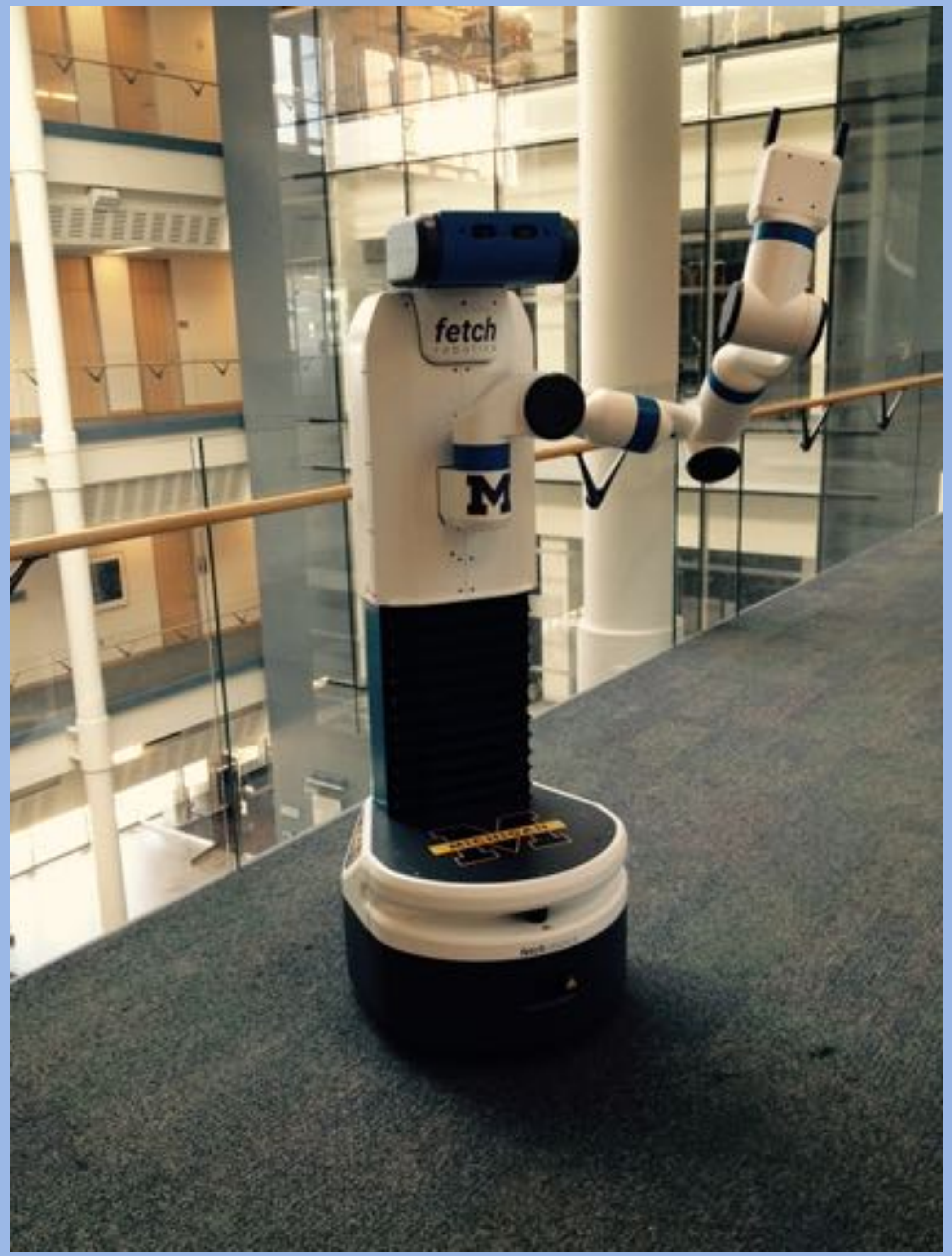

fetch


LiDAR Scanner




fetch

Use any robot *x*



to perform any task *y*

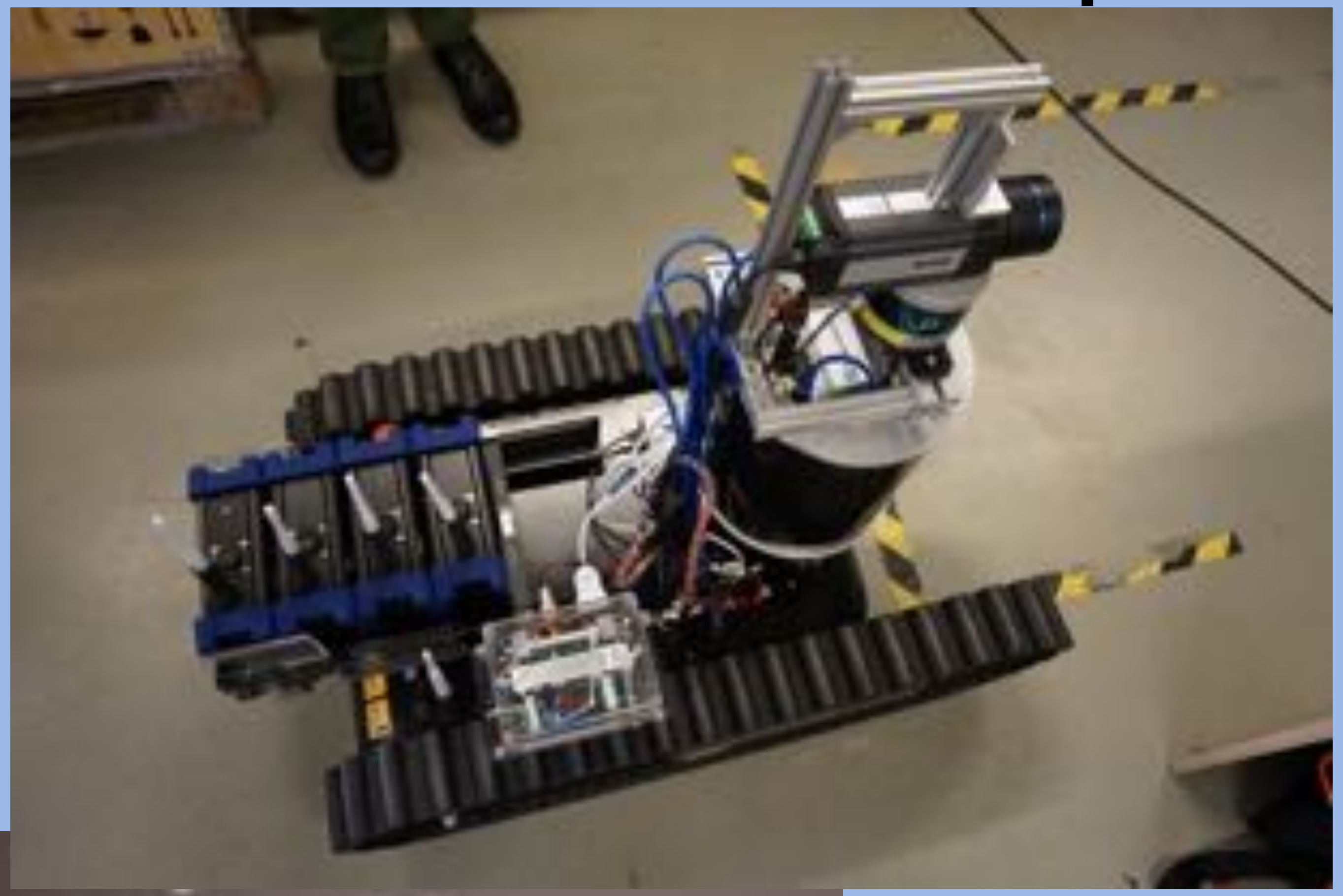
in any environment *z*

The 3Ds: Dirty, Dull, and Dangerous

“Autonomous” Driving

Infrastructure inspection

Nuclear cleanup



https://www.shadowrobot.com/blog/robots-saving-humans-from-dangerous-jobs/

https://techcrunch.com/2018/06/05/remote-control-driverless-car-startup-partners-with-vehicle-manufacturers/

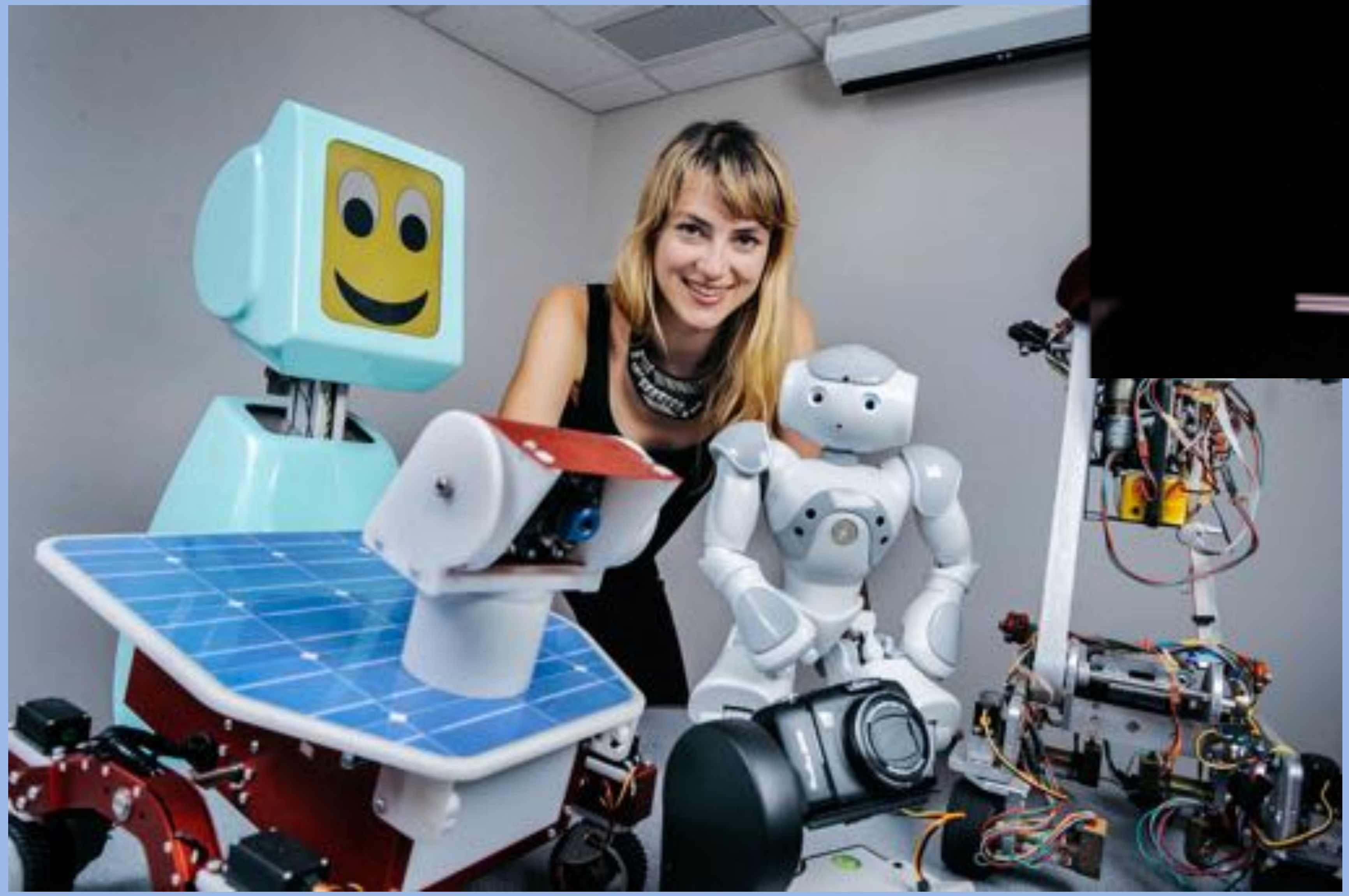
Social Robotics

Autism treatment

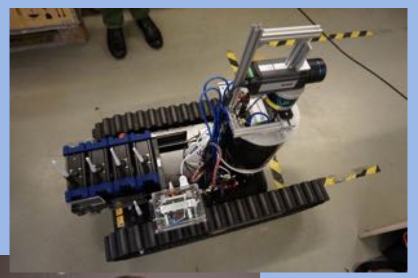
Education

Rehabilitation

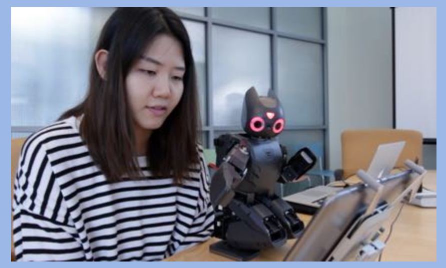
Entertainment

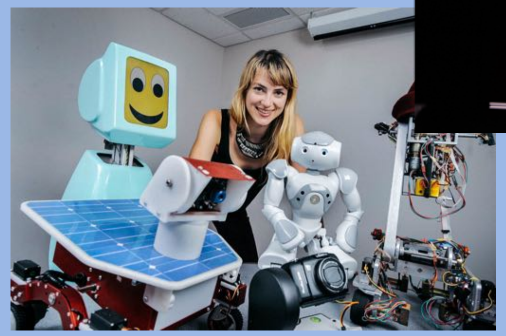


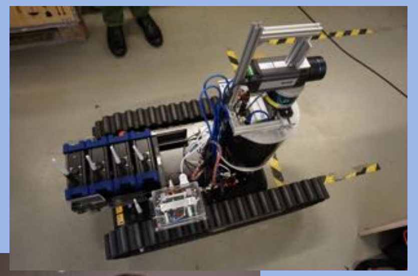
Elder care



Medical Robotics







Agriculture

Exploration

Manufacturing

Security


Menu Search Bloomberg Opinion Sign In Subscribe

Nobody's Ready for the Killer Robot

A Q&A with General Robert Latiff on the ethics of warfare in the autonomous future.

By Tobin Harshaw
December 30, 2017, 8:00 AM EST

Popular in Opinion

History Suggests Post-Pandemic Peace Is Rare

America Needs President Bill Lincoln

Is there a human in the loop? Photographer: Scott Barbour/Getty Images

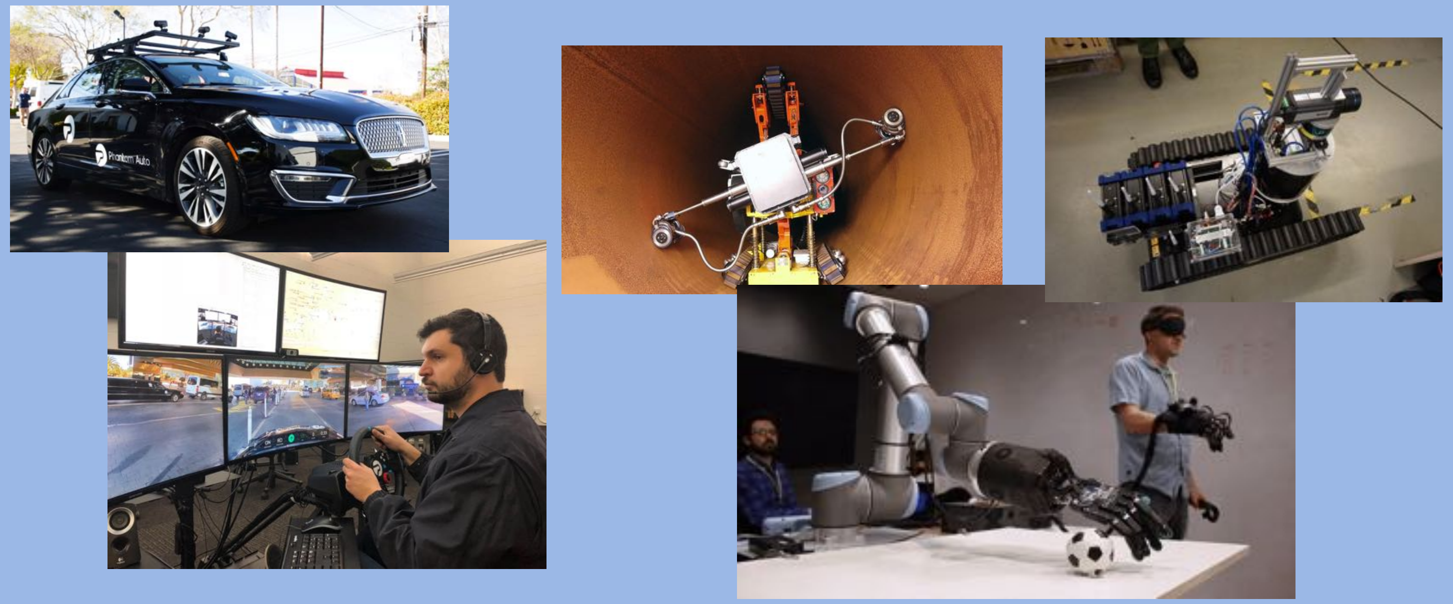
Ethical Use

Lethal Force

Social Robotics

Medicine

Dirty, Dull, Dangerous



Users

Robot Applications

Custom applications,
Taskable autonomy research

Robot Operating System

LCM
ROS
Player
YARP
MOOS

Operating System

Hardware

Users

Robot Applications

Custom applications,
Taskable autonomy research

Robot Operating System

~~LCM~~ ~~ROS~~ ~~Player~~ ~~YARP~~ ~~MOOS~~

Operating System

Hardware

Users

Robot Applications

Custom applications,
Taskable autonomy research

Robot Operating System

Build your own Robot OS

Operating System

Hardware

Robot Operating System
Build your own Robot OS

Localization and Mapping

Path Planning

Feedback Control

Robot Vision

Motion Planning

Dynamical Simulation

Collision Detection

Decision Making Systems

Forward Kinematics

Multi-robot Coordination

Task planning

Inverse Kinematics

Robot Operating System

Build your own Robot OS

Robot Operating System

Covered at breadth in AutoRob

Users

Robot Applications

Robot Operating System

Operating System

Hardware

Work with a real robot once this semester

https://autorob.org

AUTOROB schedule kineval git FAQ assignments: 1

AutoRob

Introduction to Autonomous Robotics
Michigan EECS 367

Robot Operating Systems
Michigan ROB 320

Winter 2022

Flipped Classroom Hybrid COVID-19 Edition

Agenda

- Introduction
- So, where is my robot?
- Course administrative overview
- Action items: what I need from you now
  - Student workflow survey, Join autorob Slack and GitHub Classroom

https://autorob.org

AUTOROB | schedule | kineval | git | FAQ | assignments: 1

AutoRob

Introduction to Autonomous Robotics
Michigan EECS 367

Robot Operating Systems
Michigan ROB 320

Winter 2022

Flipped Classroom Hybrid COVID-19 Edition

Course website
http://autorob.org

AutoRob can be done remotely

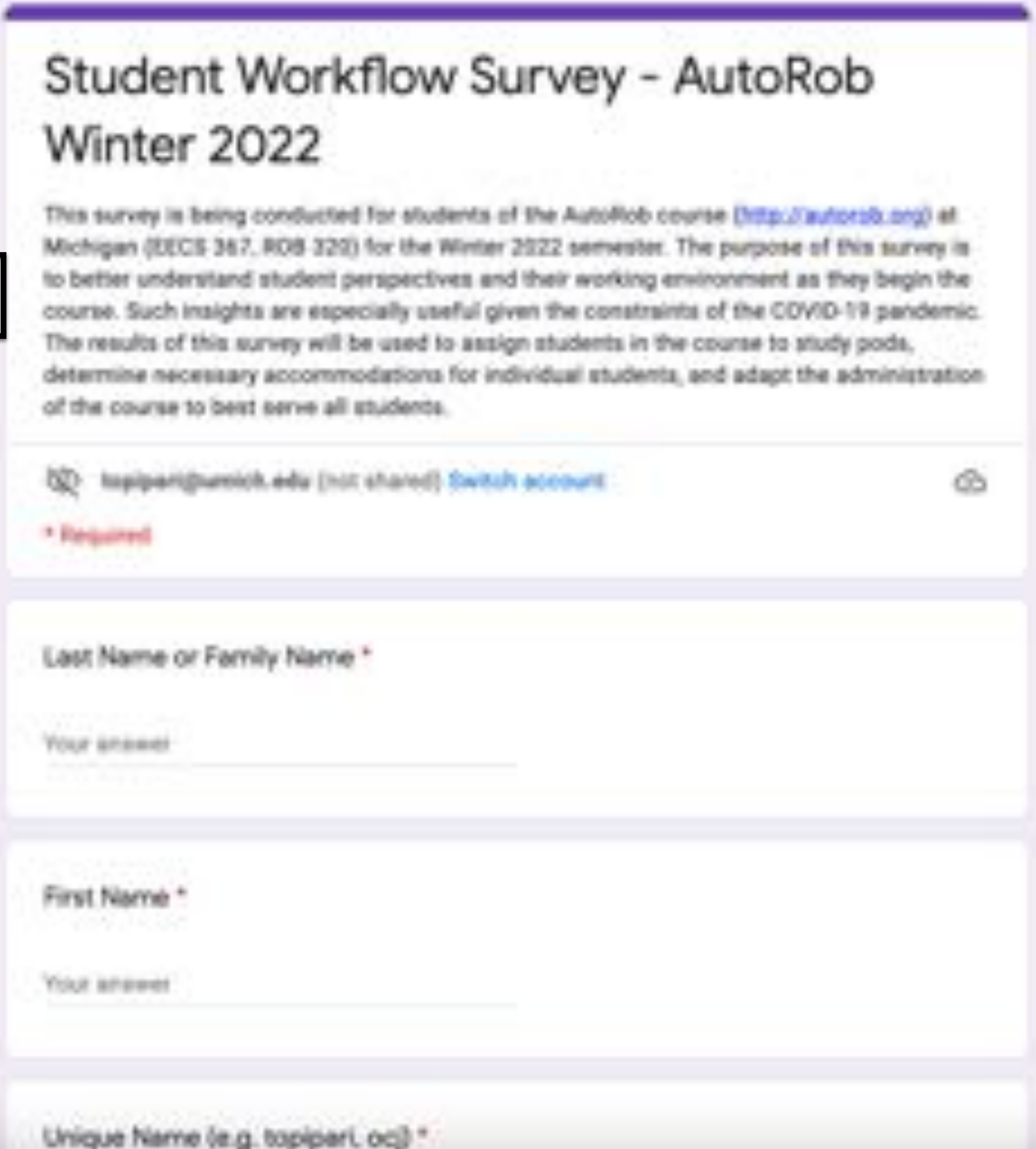
Flipped Classroom Format

- Lectures will be recorded and available online (autorob.org)
- Course Zoom link: https://umich.zoom.us/j/94439912243
- Interactive Session MW 1:30-3:00pm EST
  - General issues, Q&A, Weekly quizzes (starting Jan 24)
- Lab Section: F 2:30-3:20 EST
  - Walkthrough of projects
- Interactive study pods
  - Clusters of 5 students with instructor collaboration

Student Workflow Survey

https://forms.gle/uEfkN8aPWaYZ87K3A

Please complete TODAY!


Student Workflow Survey - AutoRob Winter 2022

This survey is being conducted for students of the AutoRob course (http://autorob.org) at Michigan (EECS 367, ROB 320) for the Winter 2022 semester. The purpose of this survey is to better understand student perspectives and their working environment as they begin the course. Such insights are especially useful given the constraints of the COVID-19 pandemic. The results of this survey will be used to assign students in the course to study pods, determine necessary accommodations for individual students, and adapt the administration of the course to best serve all students.

topipari@umich.edu (not shared) Switch account

* Required

Last Name or Family Name *

Your answer

First Name *

Your answer

Unique Name (e.g. topipari, ocj) *

https://autorob.org

AUTOROB schedule kineval git FAQ assignments: 1

AutoRob

Introduction to Autonomous Robotics
Michigan EECS 367

Robot Operating Systems
Michigan ROB 320

Winter 2022

Flipped Classroom Hybrid COVID-19 Edition

Winter 2022 Course Format

The AutoRob course will have a flipped classroom hybrid format this Winter semester. At their discretion, a student will be able to complete the course remotely in its entirety. Course meetings, quizzes, and office hours will be held both in-person and virtually with consideration of the public health situation. All lectures will be pre-recorded and available online through this course website. The course staff will be available to students through regularly scheduled all-class interactive sessions twice per week, a laboratory section once per week, and small interactive study pods led by a member of the course staff, as well as office hours that are scheduled as needed.

Follow this

IMPORTANT

Students enrolled in EECS 367 or ROB 320 should complete the AutoRob Student Workflow Survey as soon as possible, in preparation for the first week of classes.

Course Meetings

The AutoRob course will have a flipped classroom hybrid format this Winter semester. At their discretion, a student will be able to complete the course remotely in its entirety. Course meetings, quizzes, and office hours will be held in-person only as needed and with consideration of the public health situation. All lectures will be pre-recorded and available online through this course website. The course staff will be available to students through regularly scheduled all-class interactive sessions twice per week, a laboratory section once per week, and office hours with small "study pods" that are scheduled as needed.

Course Lectures will be recorded and available on this site as listed in the course schedule.

Course Interactive Sessions will be dedicated to addressing general questions and comments regarding course concepts in relation to lectures, projects, quiz synchronization, and course administrivia.
Monday and Wednesday 1:30-3:00 PM EST 133 CHRYS
Sessions can be joined remotely at Zoom ID: 944 3991 2243

Laboratory Sections will provide guidance through the workflow of course projects.
Friday 2:30-3:20 PM EST, 1500 EECS
Lab sections can be joined remotely at Zoom ID: 944 3991 2243

Interactive Study Pods

Small pods of students will be assigned to regularly meet together with a member of the course staff once per

Course Schedule (tentative and subject to change)

Preview slides from lectures during the Fall 2020 offering of AutoRob are provided. These preview slides will be replaced with recorded lectures for Winter 2022 as the videos become available.

Date	Topic	Reading	Project | Quiz
Jan 5	Initialization: "So, where is my robot?", "What is a Robot OS?", Course administration and logistics What is a robot?: Brief history and definitions for robotics	Spong Ch.1 Corke Ch.1	Setup git repository Out: Path Planning
Jan 7	Lab Session: Git-ing started with git, JavaScript, and KinEval		
	Week 2		
Jan 10	Path Planning: Navigation as graph search, DFS, BFS, Dijkstra shortest paths, A-star, Greedy best first, Priority queues and binary heaps	Wikipedia	
Jan 12	JavaScript and AutoRob workflow: Project workflow with git, JS/HTML5 tutorial, Document Object Model, Version Control, LaTeX math mode, Licensing, Michigan Honor License	Crockford, HTML Sandbox, hello.html, JavaScript by Example, hello_anim, hello_anim_text	
Jan 14	367 Lab: KinEval Path Planning code overview		
	Week 3		
Jan 17	**No course meeting** - Martin Luther King, Jr. Day UM Martin Luther King Jr. Symposium Help broaden participation in computing and robotics		
	Dynamical Simulation: Simple pendulum, Lagrangian equation(s) of	Spong Ch.7 | Corke Ch.9	

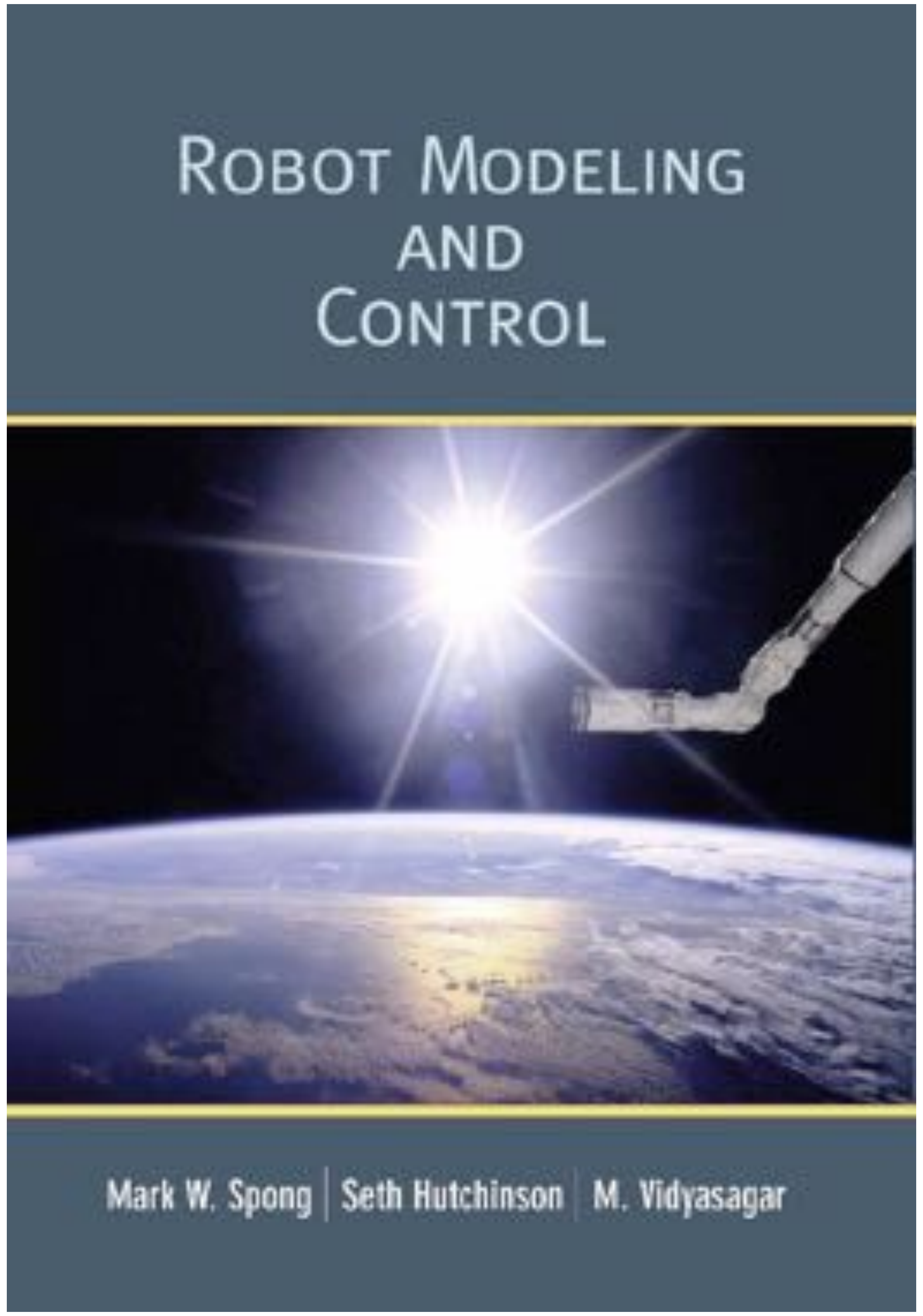
Course Structure

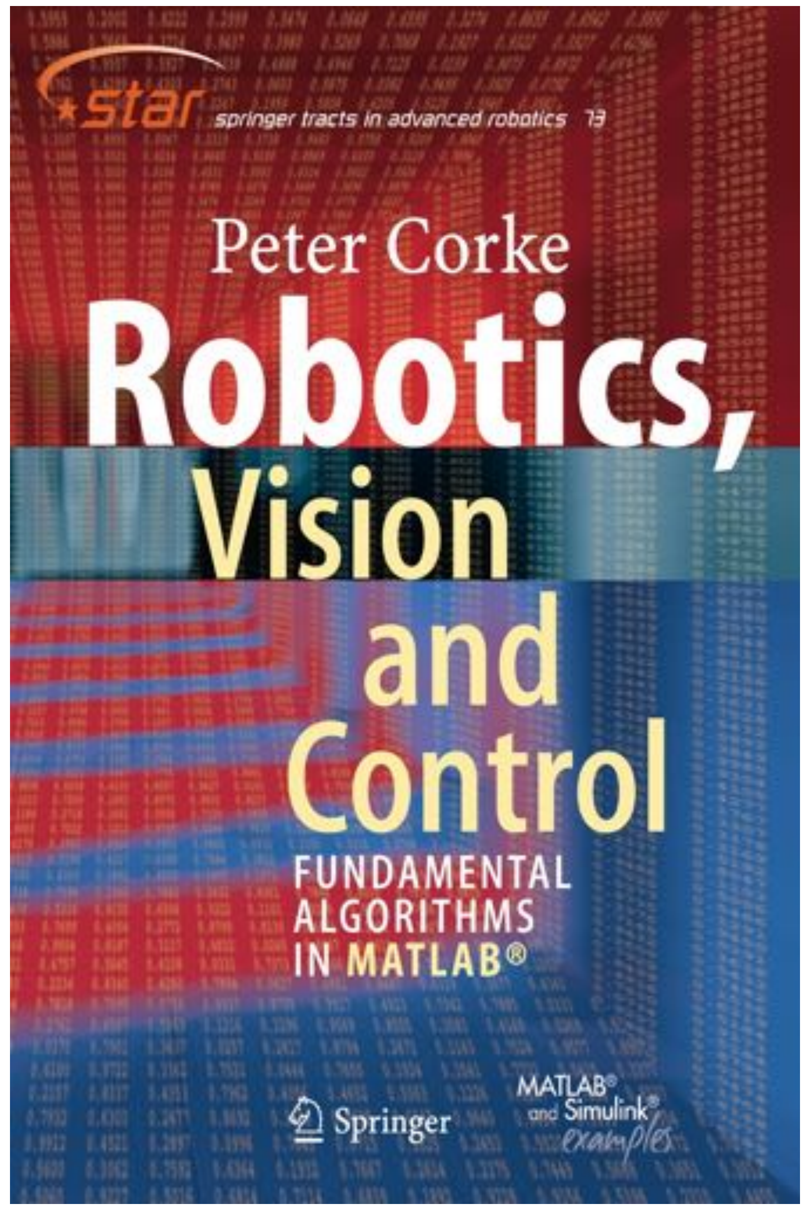
Course Structure

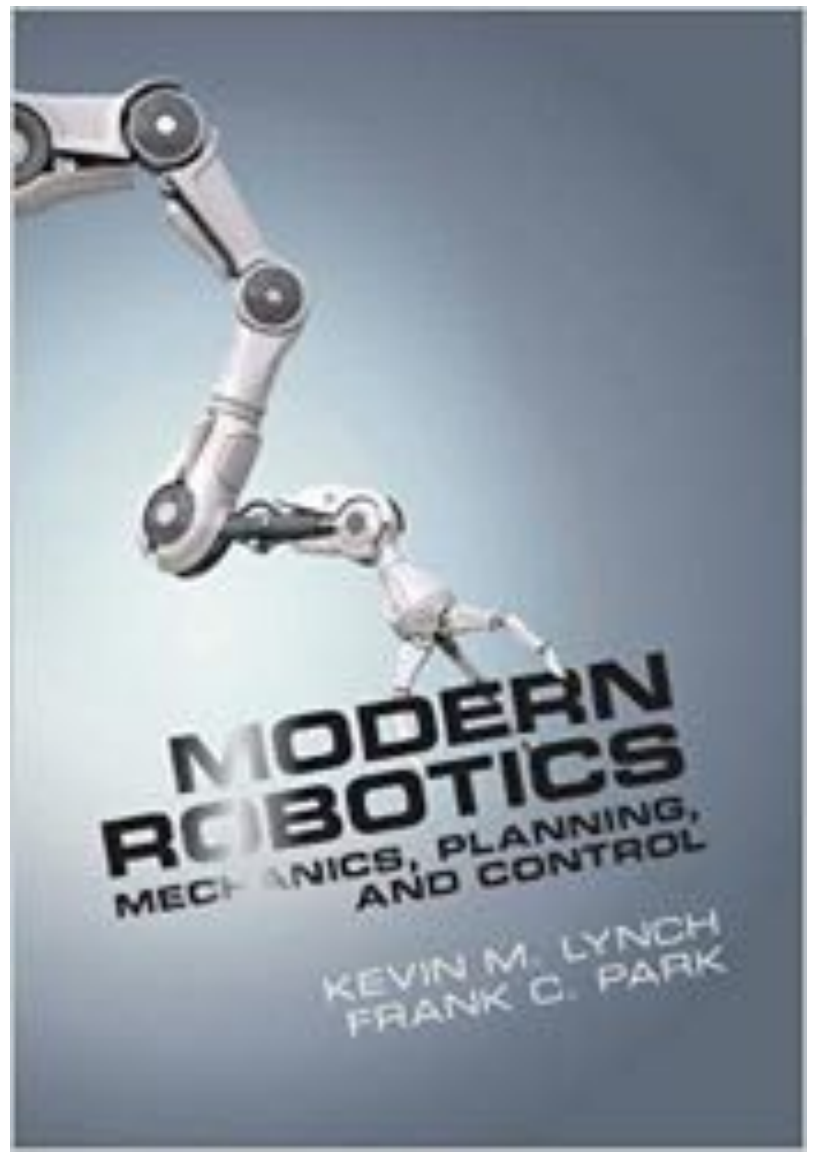
- Autonomous robot modeling and control
  - Objective: Give you the computational skills (and code) to model and control any mobile manipulator
- Project-focused class
  - 7 individual projects: from single joint control up to articulated motion planning with optional robot operating systems tutorials
- 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





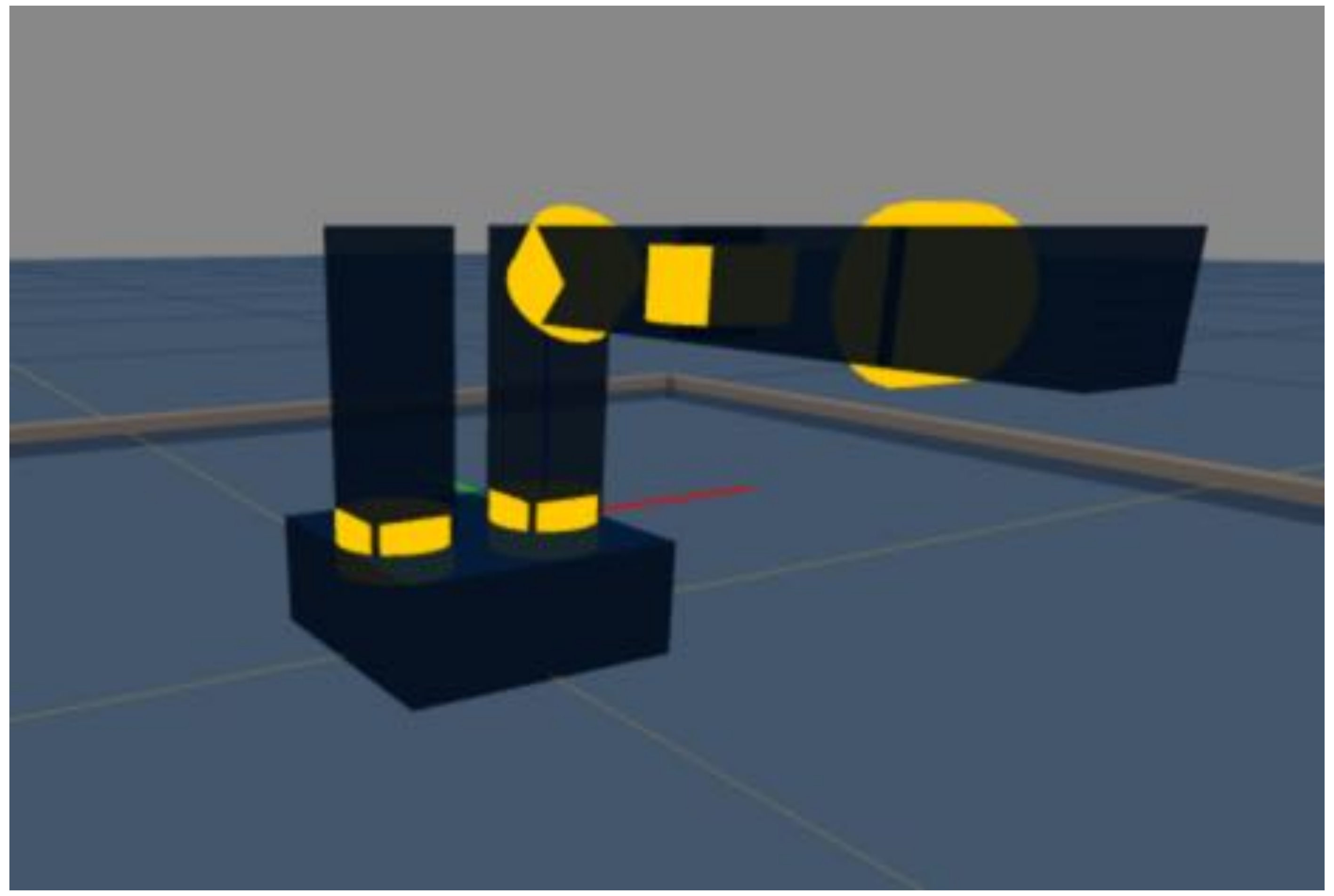


Projects

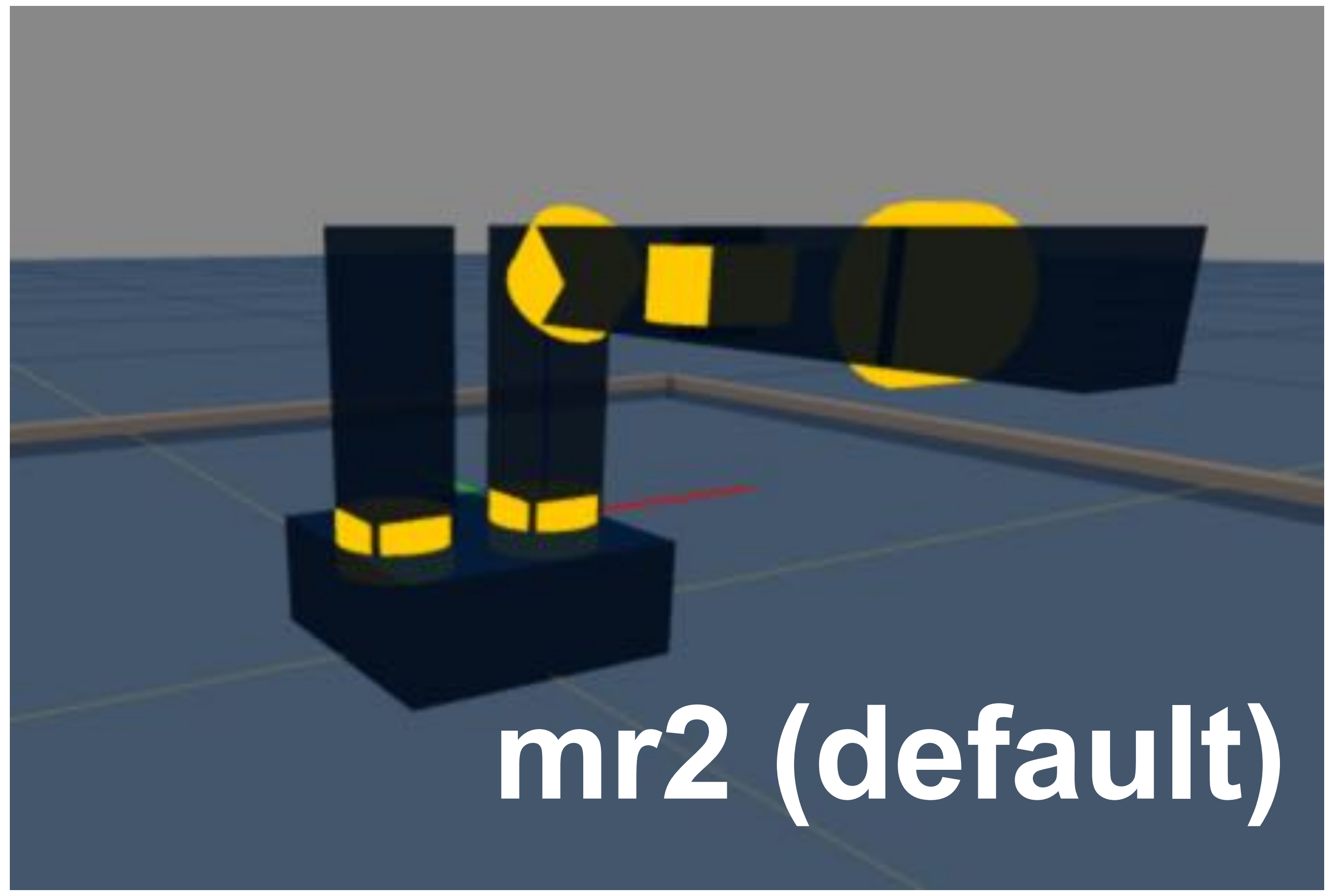
- Projects implemented in JavaScript/HTML5 using KinEval stencil
  - Projects submitted and tracked through github
  - Course staff needs admin access (please complete workflow survey!)
- 7 projects
  - 6 Programming, 1 Written/Oral
- Grading: projects are broken down into features that are “checked”
  - Points are earned through successful implementation of features
  - Continuous Integration grading

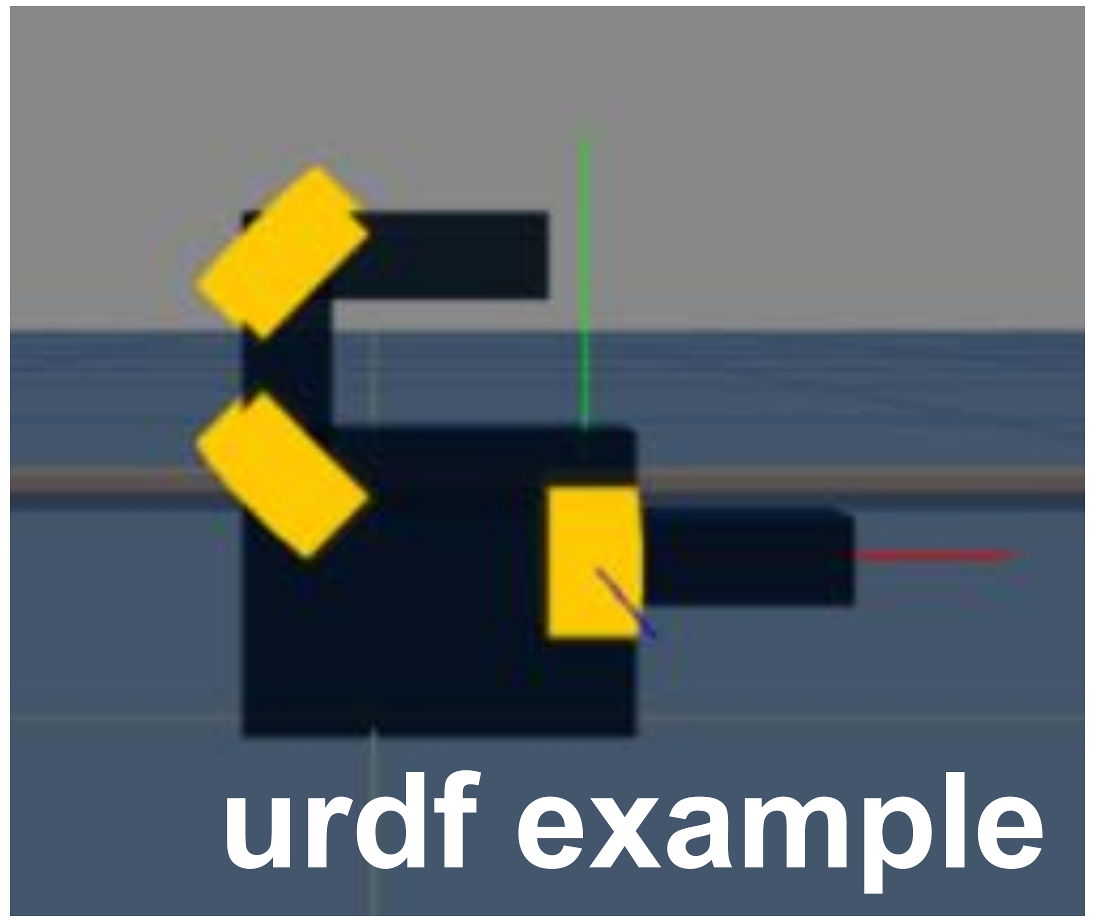
KinEval code stencil

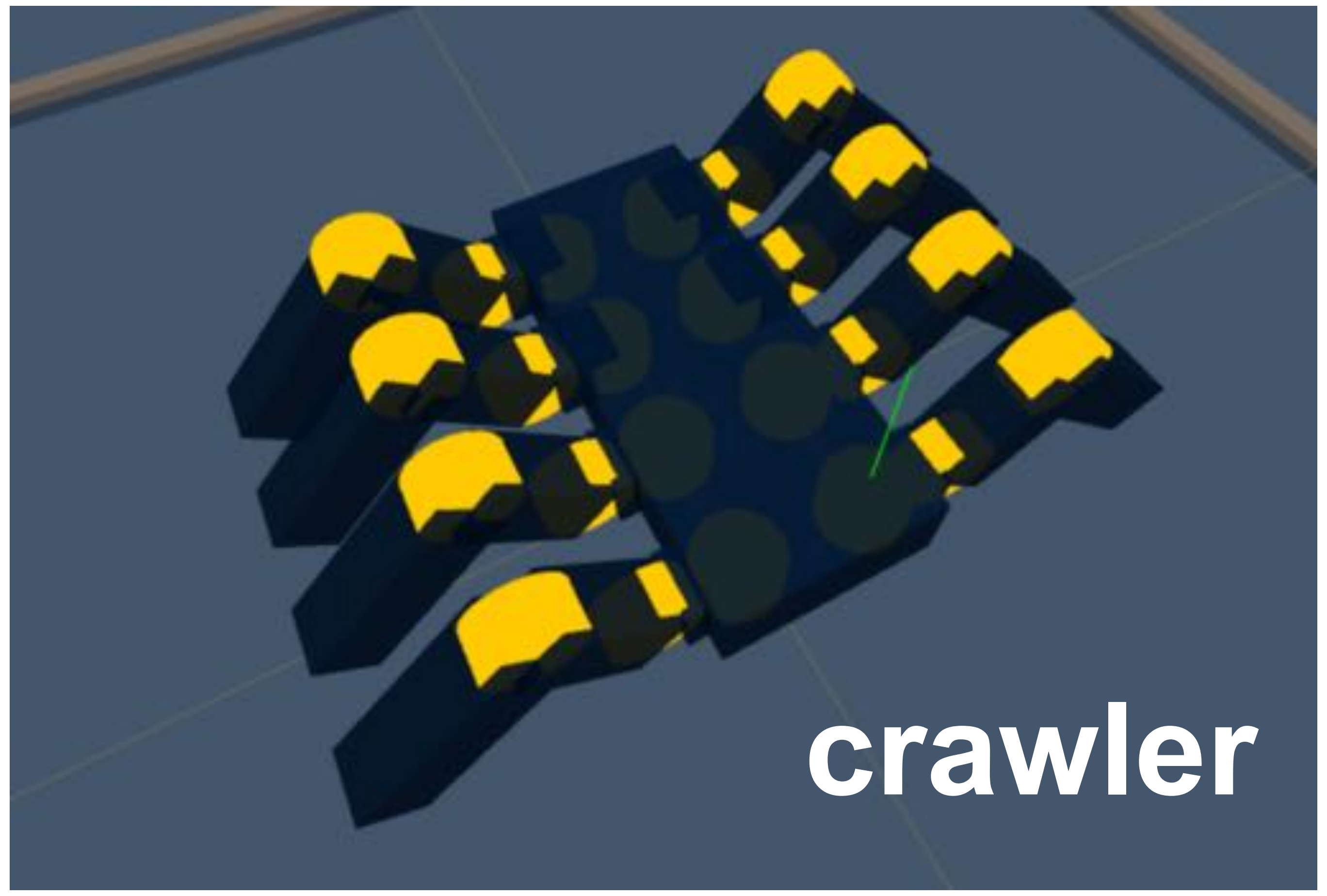
- Code stencil for AutoRob projects in 3D
- Uses threejs 3D rendering library and WebGL
- URDF-like robot description
- Usable, but not perfect, camera and UI controls
- AABB collision detection provided for planning
- Warning: professor-level coding

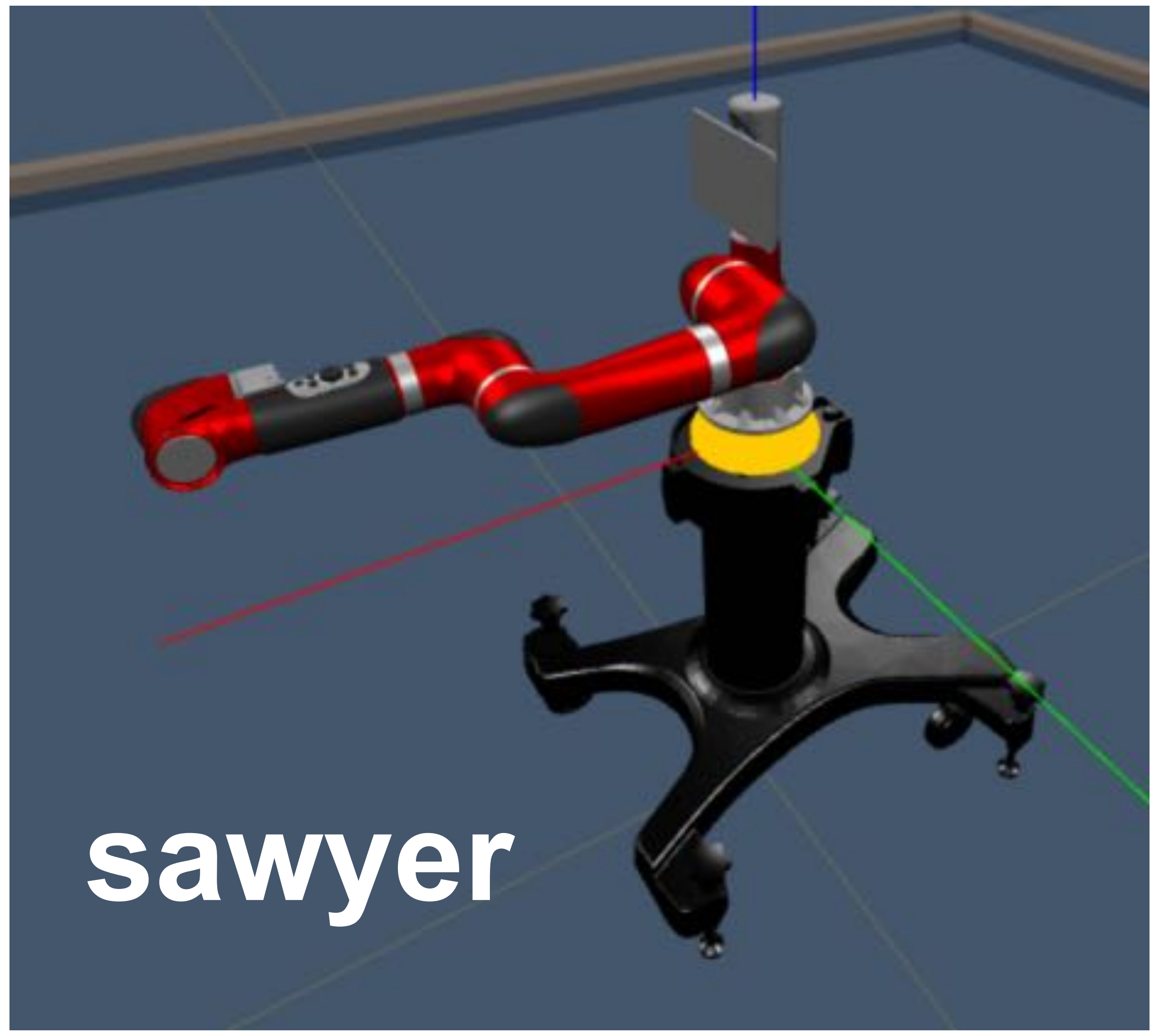


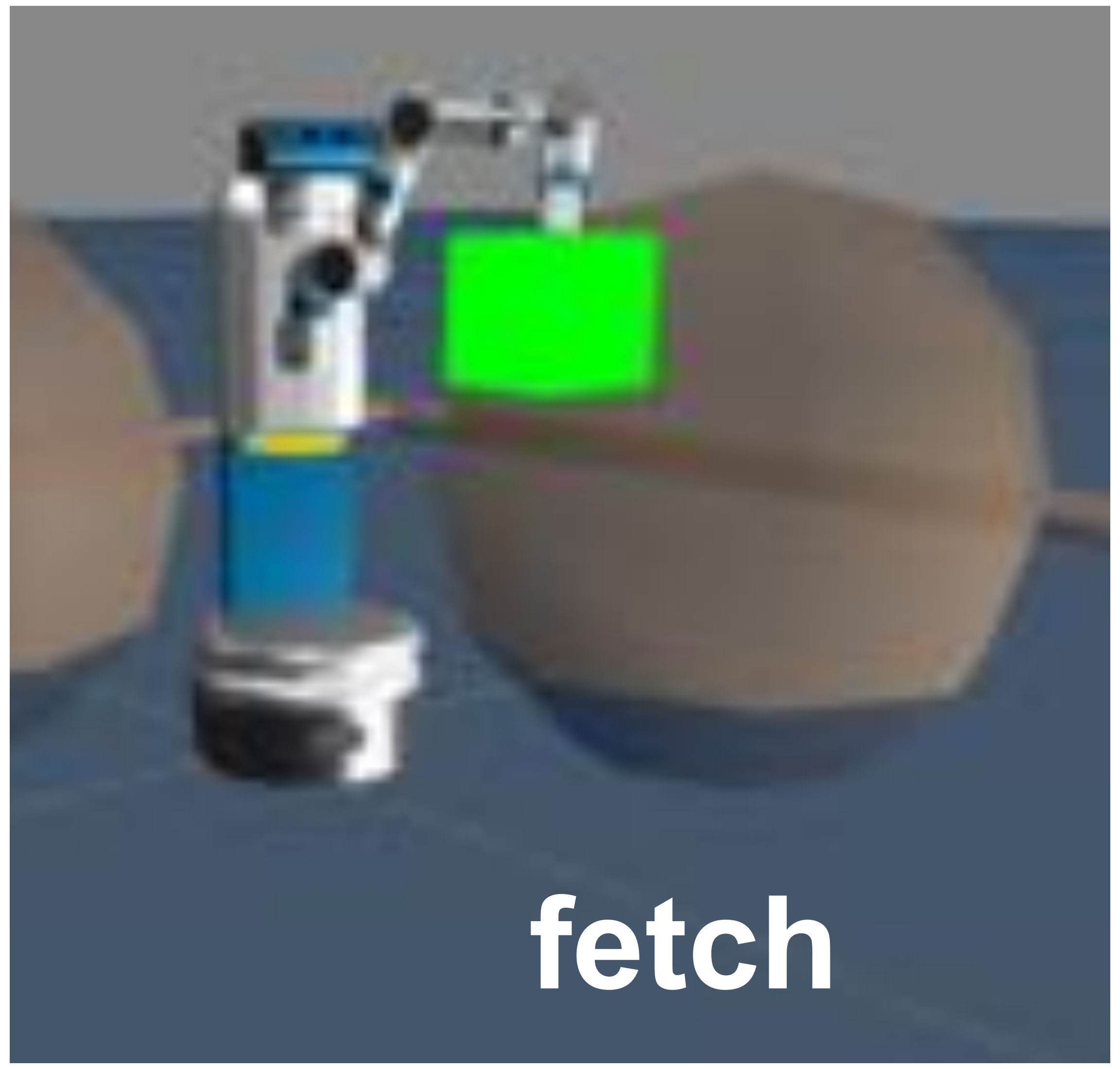
KinEval code stencil

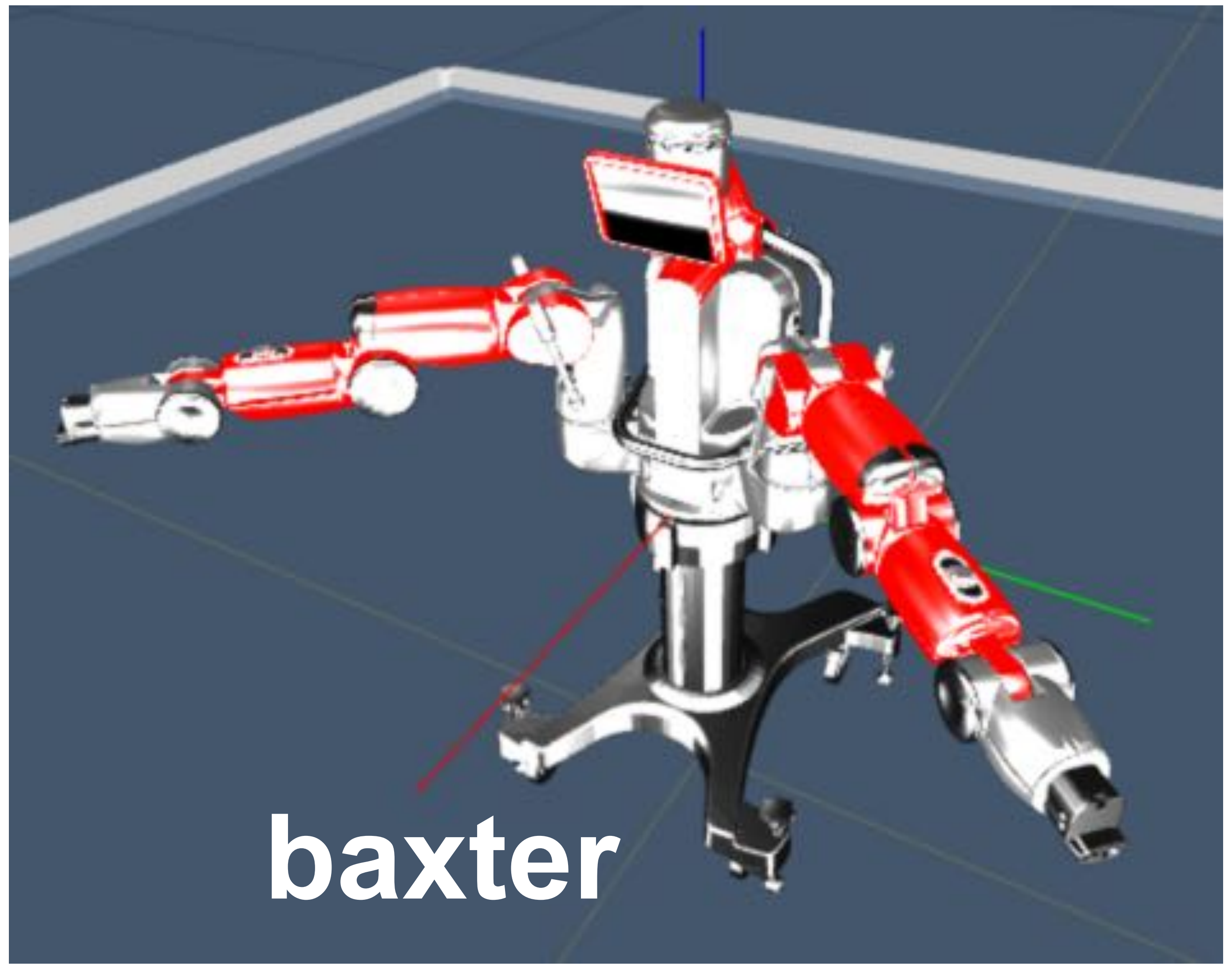








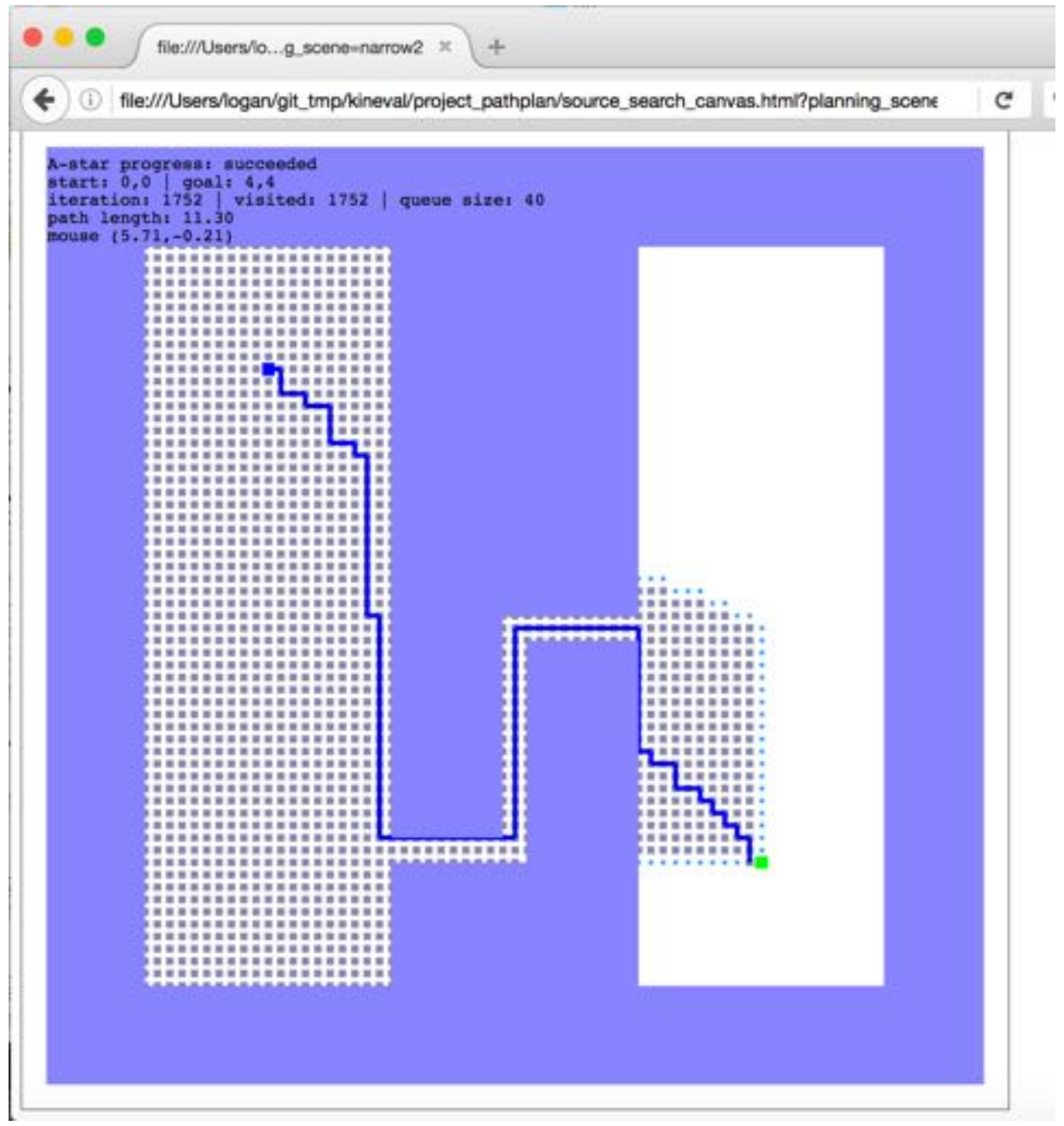


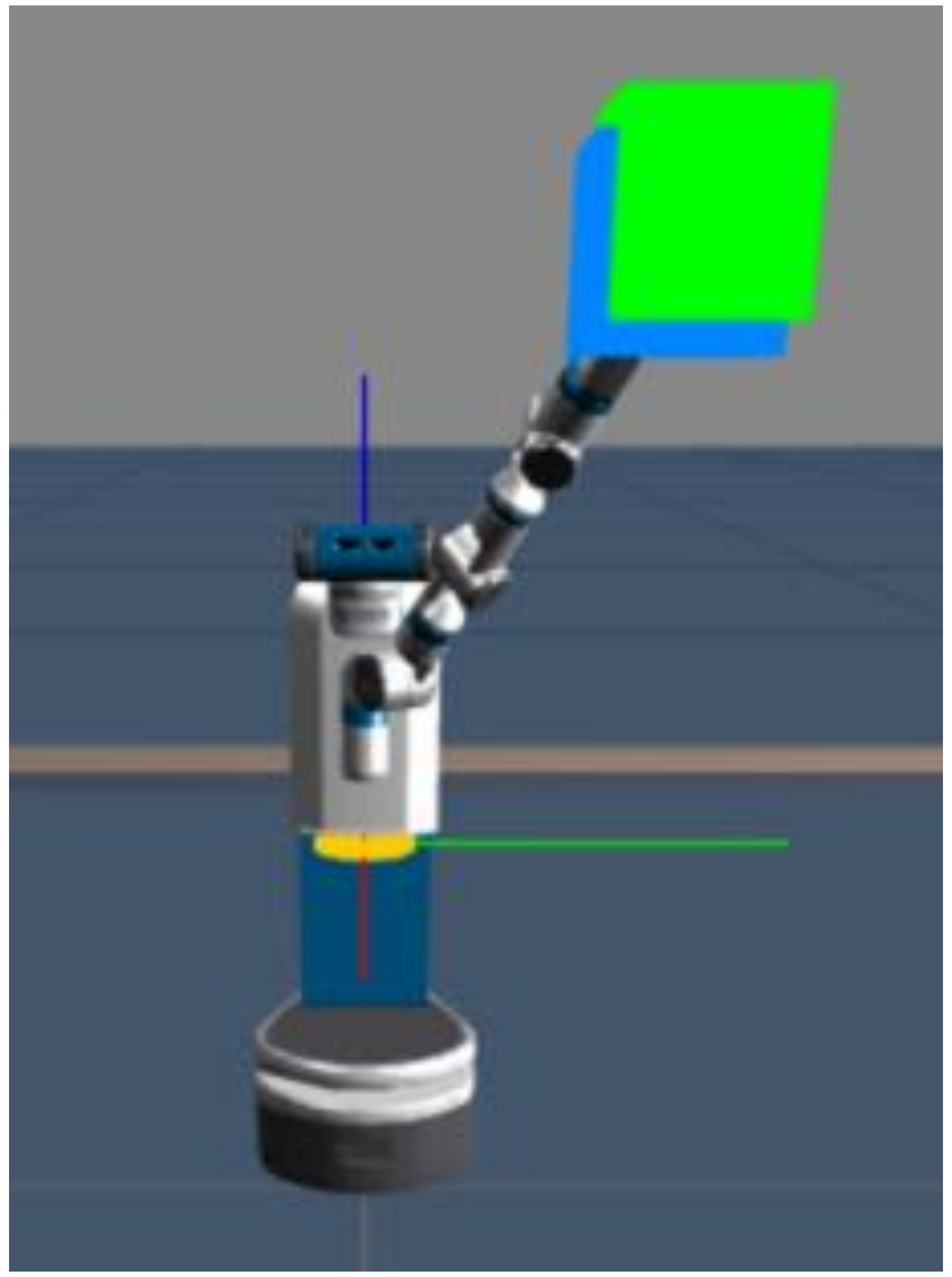


Projects

- **Path Planning** A-star search in 2D world
- **Pendularm** 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?

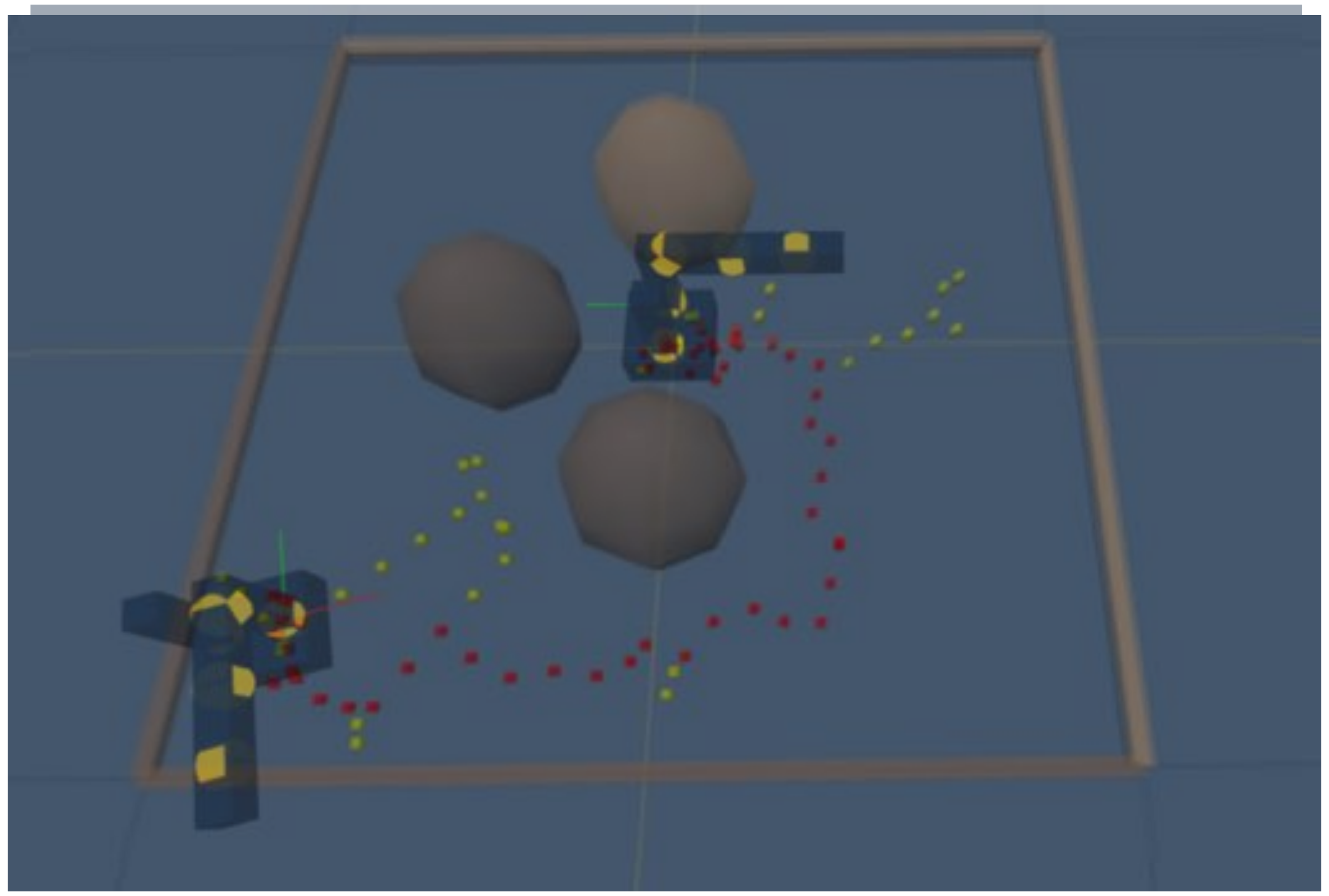
- Project 1: **Path Planning**





- Projects 3-5: **Forward and Inverse Kinematics**

- Project 2: **Pendularm**



- Project 6: **Motion Planning**


https://github.com/autorob-WN22/kineval-stencil- student
Search or jump to...
Pull requests
Issues
Marketplace
Explore
autorob-WN22 / kineval-stencil- student
Private
generated from autorob-WN22/kineval-stencil
Watch 1
Fork 0
Star 0
Code
Issues
Pull requests
Actions
Projects
Security
Insights
master
1 branch
0 tags
Go to file
Add file
Code
github-classroom Initial commit
17 hours ago
1 commit
js
kineval
project_pathplan
project_pendularm
robots
tutorial_heapsort
tutorial_js
worlds
LICENSE
README.md
Initial commit
17 hours ago
About
kineval-stencil- student created by GitHub Classroom
Readme
View license
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages

Will you work with a real robot?

Will you work with
a real robot?

Yes, at least once
using rosbridge/ROS

Course Policies

Grading Policy

EECS 367: Introduction to
Autonomous Robotics

- 7 projects (12 points each)
- 10 quizzes (2 points each)
- Participation (4 points)
- Extra credit (4 points total)

A: 95+ points
B: 83+ points
C: 73+ points

Continuous Integration Grading

- AutoRob in Winter 2022 will use a “CI grader”
- The CI grader will
  - pull code from your repository
  - run tests for all projects that past their due dates
  - return test and grading results to your repository
  - test schedule
- The course staff will review all grades for correctness

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 deadline (Apr 18)

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 deadline (Apr 18)

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 + “CC BY 4.0”
- 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

Discussion and Communication

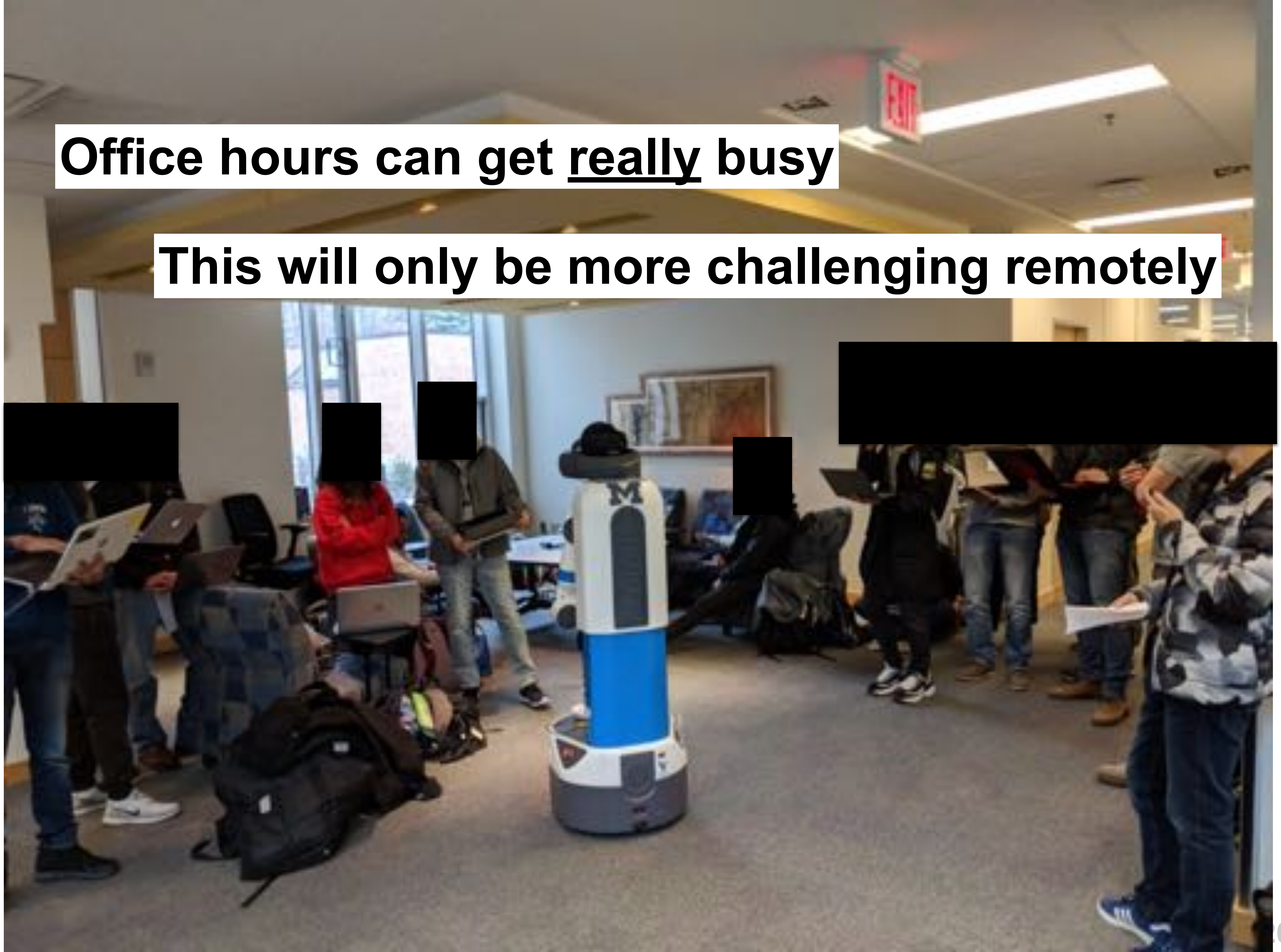
Discussion and Communication

- AutoRob Slack workspace: um-wn22-autorob.slack.com/
  - Formal course discussions allowed
- Course staff office hours
  - Will support hybrid format
  - Virtual for first week
- Instructor email: topipari@umich.edu

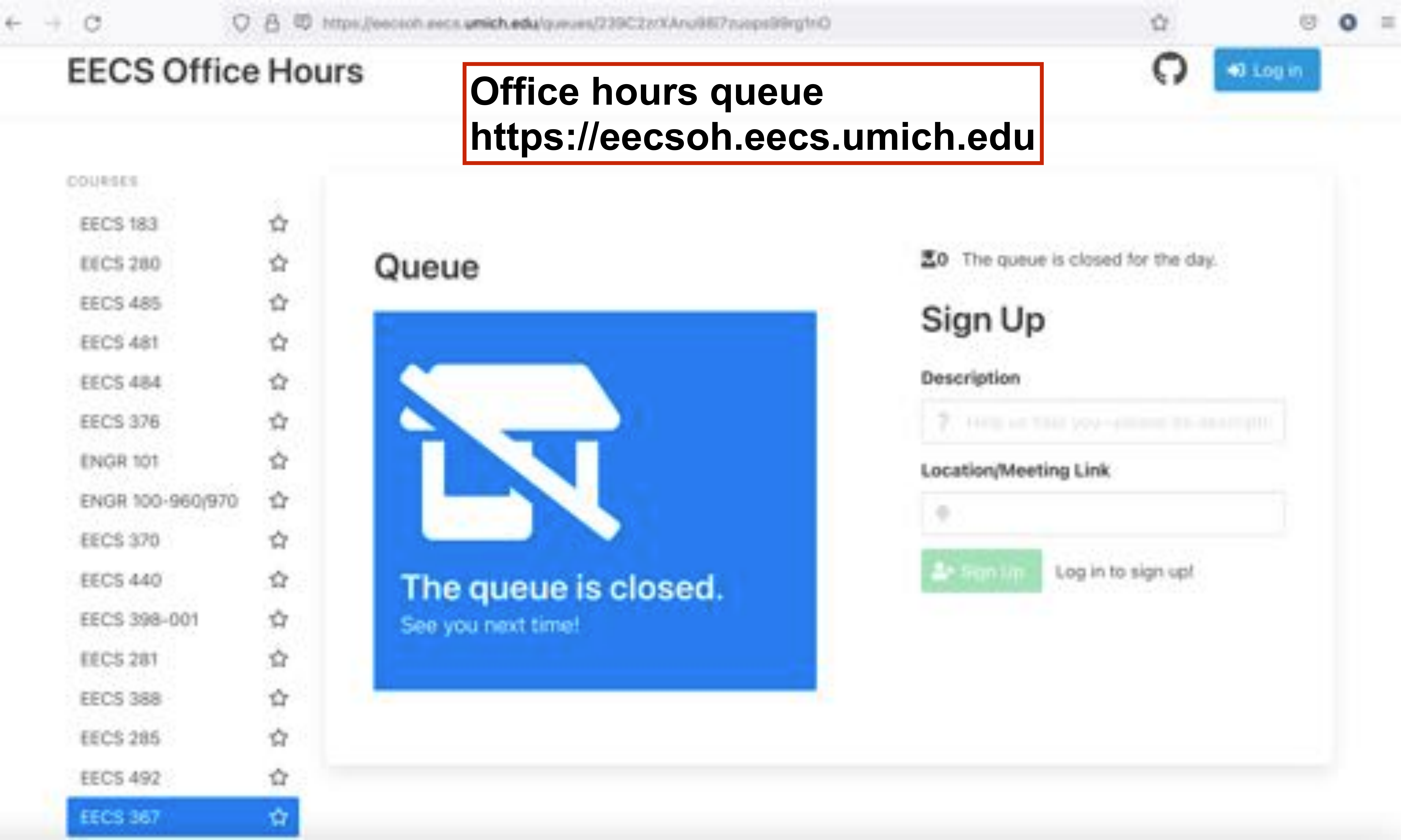

Search University of Michigan
autorob-wn22
general
Company-wide announcements and work-based matters
Add a bookmark
Slack Connect
Browse Slack
Channels
asgn1-pathplan
asgn2-pendularm
asgn3-fk
asgn4-dance-fsm
asgn5-ik
asgn6-rrt
asgn7-best-use
course-staff
general
lecture-questions
optional-extensions
random
Add channels
Direct messages
Slackbot
#general
@itsslack created this channel yesterday. This is the very beginning of the #general channel. Description: This channel is for workspace-wide communication and announcements. All members are in this channel. (edit)
Add people
Send emails to channel
Yesterday
ITS Slack
joined #general along with Anthony Opipari.
See video clips in action
Message #general

Office hours can get <u>really</u> busy

This will only be more challenging remotely



Office hours queue
https://eecsoh.eecs.umich.edu



Agenda

- Introduction
- So, where is my robot?
- Course administrative overview
- Action items: what I need from you now
  - Student workflow survey, Join autorob Slack and GitHub Classroom

What I need from you **now**

What I need from you **now**

- Complete student workflow survey - ASAP!
- Ensure you have access to the "autorob-wn22" UM slack workspace
- Install git and setup your working environment
    - Join the "autorob-WN22" GitHub classroom
    - Ensure you can clone, commit, and push files to your repository
- View recorded course lectures
    - To be posted in advance of interactive sessions

https://forms.gle/uEfkN8aPWaYZ87K3A

Student Workflow Survey - AutoRob Winter 2022

This survey is being conducted for students of the AutoRob course (http://autorob.org) at Michigan (EECS 367, ROB 320) for the Winter 2022 semester. The purpose of this survey is to better understand student perspectives and their working environment as they begin the course. Such insights are especially useful given the constraints of the COVID-19 pandemic. The results of this survey will be used to assign students in the course to study pods, determine necessary accommodations for individual students, and adapt the administration of the course to best serve all students.

topipari@umich.edu (not shared) Switch account

* Required

Last Name or Family Name *

Your answer

First Name *

Your answer

Unique Name (e.g. topipari, ocj) *

What I need from you **now**

- Complete student workflow survey - ASAP!
- Ensure you have access to the "autorob-wn22" UM slack workspace
- Install git and setup your working environment
    - Join the "autorob-WN22" GitHub classroom
    - Ensure you can clone, commit, and push files to your repository
- View recorded course lectures
    - To be posted in advance of interactive sessions

You should have access through UM login
https://um-wn22-autorob.slack.com/

What I need from you **now**

- Complete student workflow survey - ASAP!
- Ensure you have access to the “autorob-wn22” UM slack workspace
- Install git and setup your working environment
    - Join the “autorob-WN22” GitHub classroom
    - Ensure you can clone, commit, and push files to your repository
- View recorded course lectures
    - To be posted in advance of interactive sessions

Join our GitHub Classroom:
https://classroom.github.com/a/j_-yiHwt


GitHub Classroom
GitHub Education
You're ready to go!
You accepted the assignment, Kinematic Evaluator.
Your assignment repository has been created:
https://github.com/autorob-WN22/kineval-stencil-student
We've configured the repository associated with this assignment (update).
Join the GitHub Student Developer Pack
Verified students receive free GitHub Pro plus thousands of dollars worth of the best real-world tools and training from GitHub Education partners — for free. Learn more
Apply

https://github.com/autorob-WN22/kineval-stencil-student

Search or jump to...

Pull requests Issues Marketplace Explore

autorob-WN22 / kineval-stencil-student Private

generated from autorob-WN22/kineval-stencil

Watch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights

master 1 branch 0 tags

Go to file Add file Code

github-classroom Initial commit 17 hours ago 1 commit

js	Initial commit	17 hours ago
kineval	Initial commit	17 hours ago
project_pathplan	Initial commit	17 hours ago
project_pendularm	Initial commit	17 hours ago
robots	Initial commit	17 hours ago
tutorial_heapsort	Initial commit	17 hours ago
tutorial_js	Initial commit	17 hours ago
worlds	Initial commit	17 hours ago
LICENSE	Initial commit	17 hours ago
README.md	Initial commit	17 hours ago

About

kineval-stencil-student created by GitHub Classroom

Readme

View license

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Wrap up

- This week's lab
  - Walkthrough of git and KinEval
- Next week's lectures
  - Path planning
  - JavaScript
- Assignment 1 (Path Planning) released, due January 21, 11:59pm