15-494/694: Cognitive Robotics

Spring 2024

Professor:
Dave Touretzky
dst@cs.cmu.edu

TA:
Joyce Truong
btruong@andrew.cmu.edu

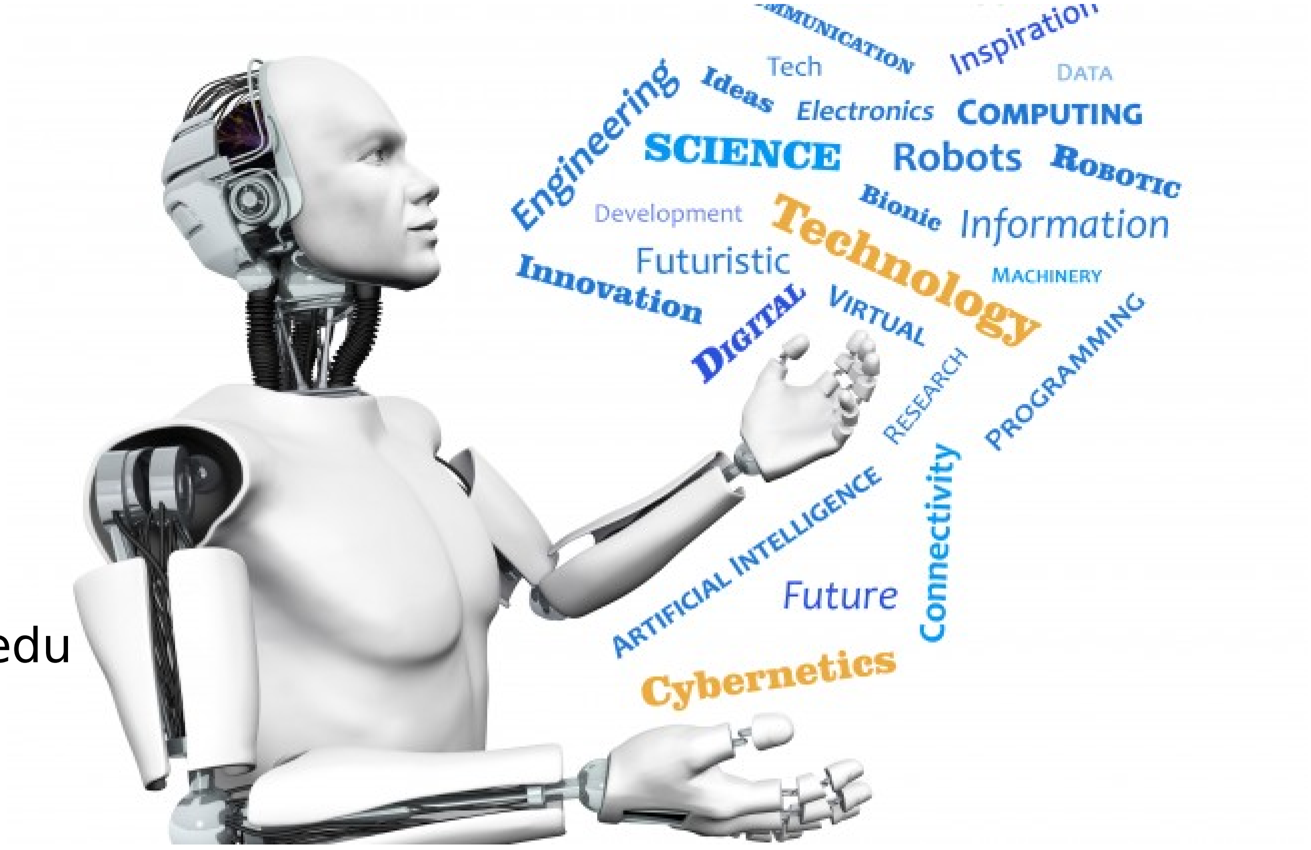


Image from http://www.futuristgerd.com/2015/09/10

What Is This Course About?

- Programming intelligent behavior for the Cozmo robot.
  - Robot vision using OpenCV
  - Machine learning using PyTorch
  - Industrial-strength Python 3
  - Large language models (GPT-4)
  - Real-time intelligence with dedicated NVIDIA GeForce RTX 2070 GPU boards
- Exploring the “Ten Big Ideas in Robotics”.

Administrative Stuff

- Course Times/Locations:
  - Mon / Wed 4:00 to 4:50 in PH A18C
  - Fri 3:30 to 4:50 in NSH 3206 (REL)
    REL = Robotics Education Lab
- Course home page:
  http://www.cs.cmu.edu/afs/cs/academic/class/15494-s24
- Cozmopedia: https://Cozmopedia.org

More Administrative Stuff

- We'll use Piazza for communication. Please ask questions via Piazza instead of emailing the professor or TA directly.

- Assignments will be handed in via Canvas. Assignments are due by 11:59 PM on the day indicated in the schedule.

Grading

- Lab participation (10%)
- In-class quizzes (10%)
- Programming assignments (50%)
- Final project: do something cool (30%)

Difference between 15-494 and 15-694:

- More stringent grading for 694.
- Possibly some extra programming problems.

In-Class Quizzes

- Quizzes test your knowledge of the previous lecture or reading assignment.
- They’re short.
- They’re worth 10% of your grade.
- Be sure to keep up with the reading and review the lecture slides so you’re prepared for the quizzes.

Before Cozmo

Lego Mindstorms and Vex IQ:

- $250 to $350
- No vision: blind robots are boring
- Unreliable components

Sony AIBO ERS-7:

- $2000 in 2003
- Color camera; program it in C++
- Discontinued in 2006
- Came back in 2018!

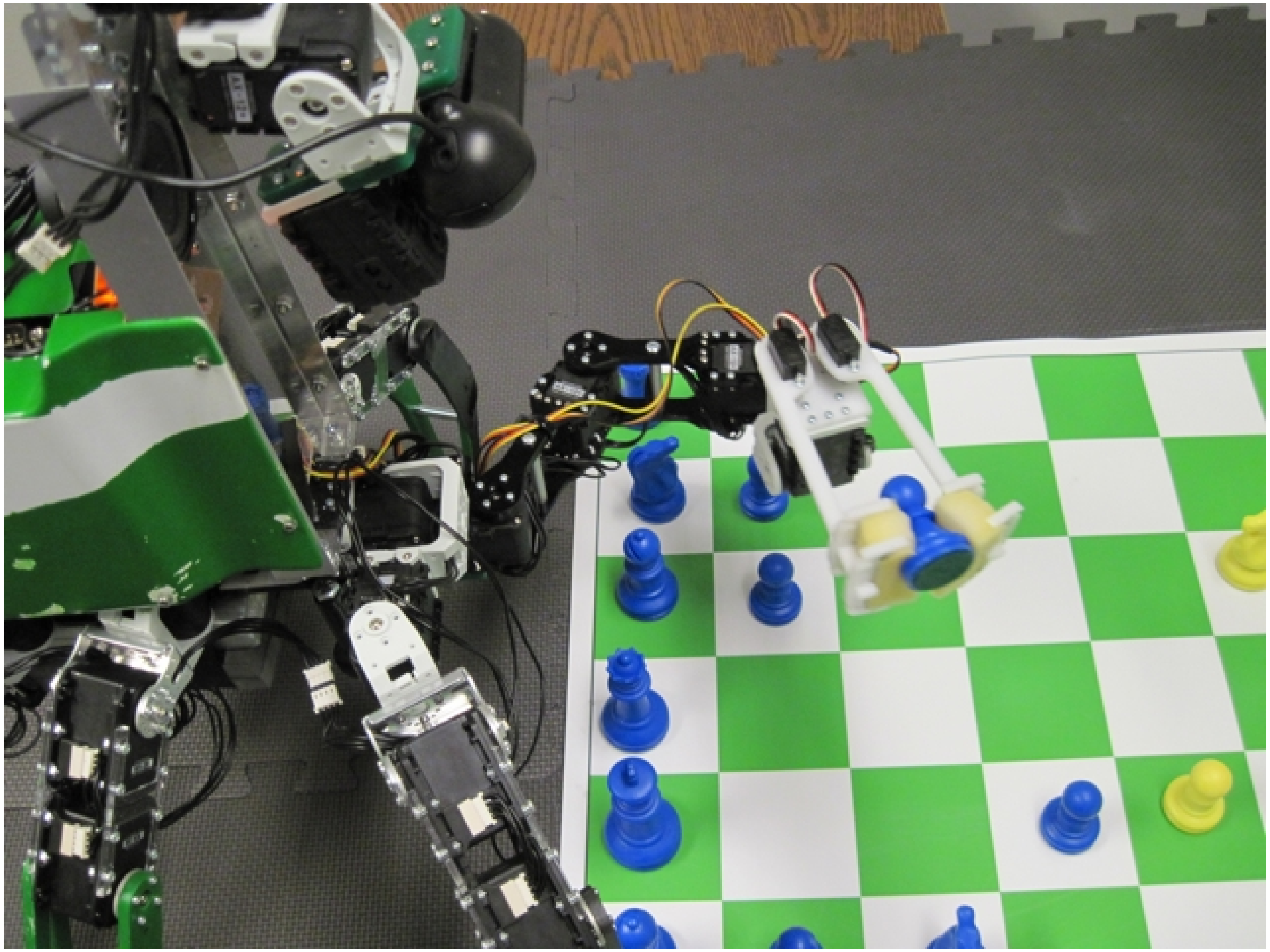
Aldebaran/Softbank's Nao V5 humanoid:

- $9000 in 2017
- Vision, AI algorithms, ROS
- Humanoids are good at two things:
  - Looking cute
  - Falling over

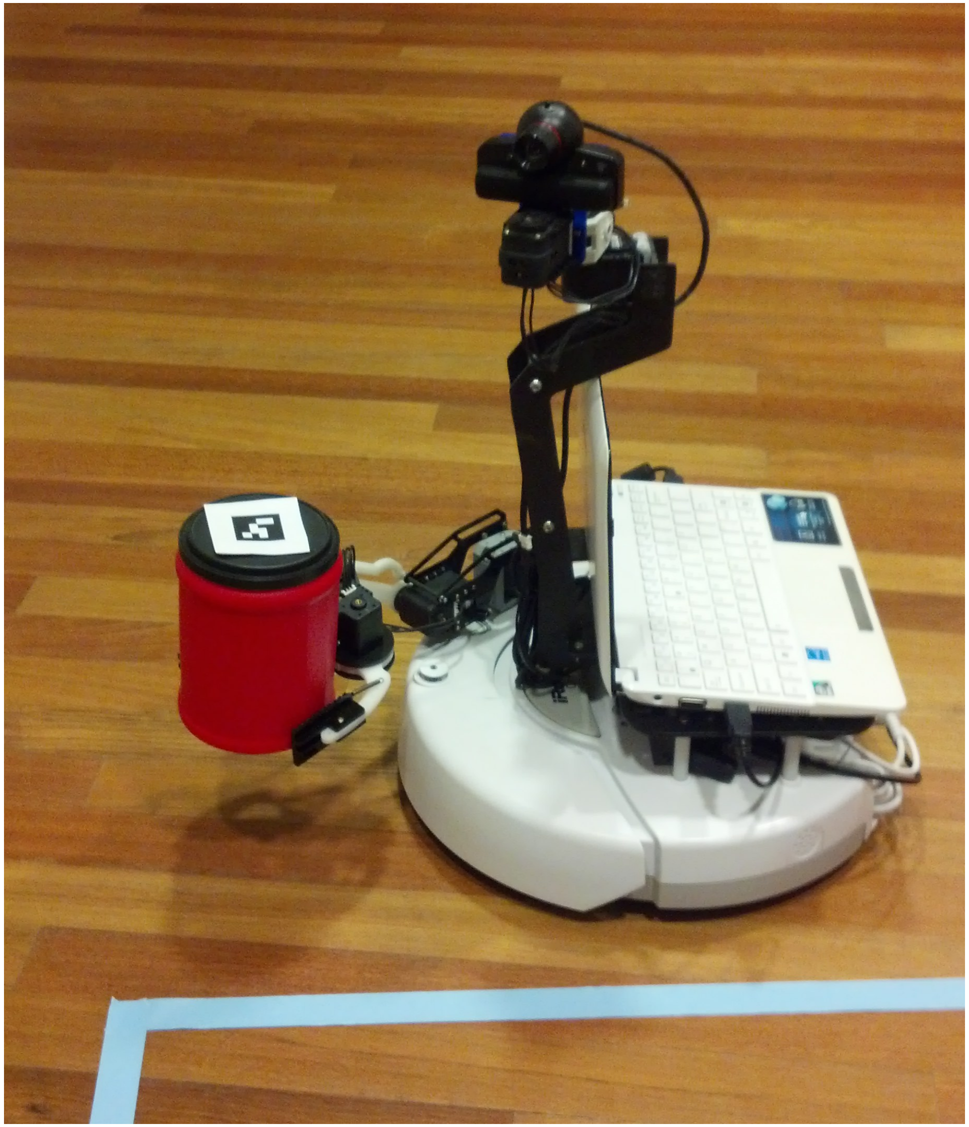
Chiara Playing "*Ode to Joy*"

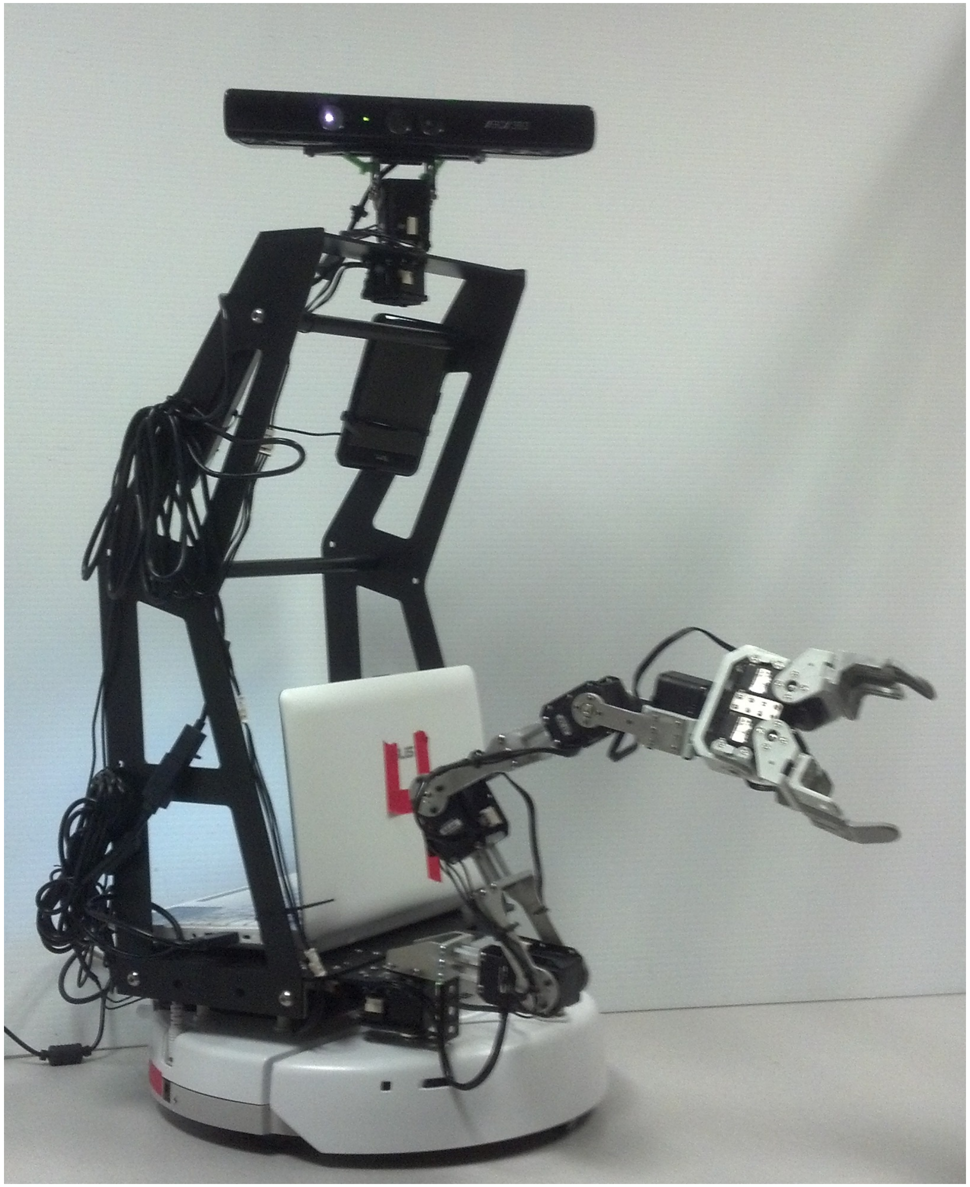
Demo by high school student Ashwin Iyengar, August 2010.

Chiaras Play Chess at AAAI-2010



Calliope Robots





Cozmo Was the Best Platform

- First shipped mid-October 2016.
- Open source Python SDK.
- Anki went bankrupt on May 1, 2019.
- Digital Dream Labs purchased the assets in December 2019, but...
- We need a new platform now.

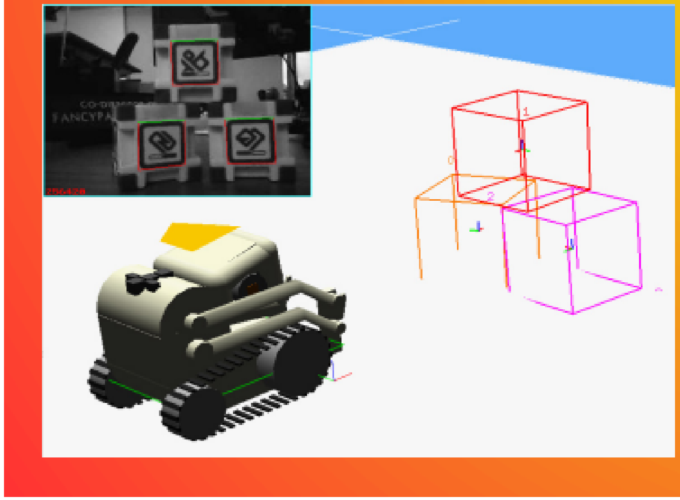
15-494*/694* Cognitive Robotics: The Future of Robot Toys

*Course redesigned for 2017

Cozmo by Anki is a new vision-guided mobile manipulator with built-in artificial intelligence and an open source Python SDK. This course will analyze and program Cozmo.

Topics include:

- Robot software architecture
- Human-robot interaction
- Computer vision
- Navigation
- Path planning
- Manipulation



Spring 2017 – 12 Units
Instructor: Dave Touretzky
Dates/Times: Mon/Wed 3:30 – 4:20
Fri 3:00 – 4:20

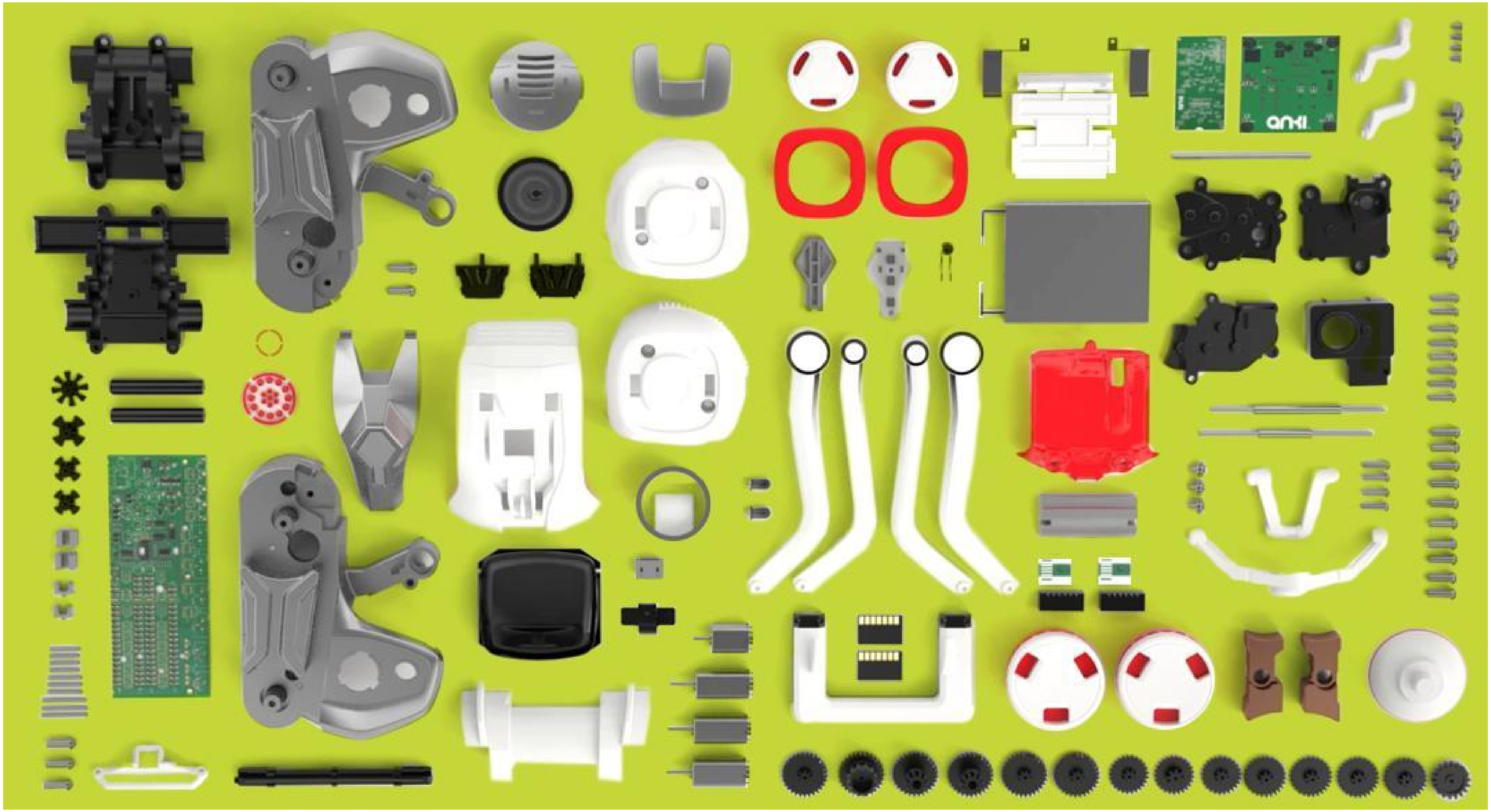
No prior robotics experience required, just strong programming skills.

You can get your own Cozmo for $180 at anki.com.

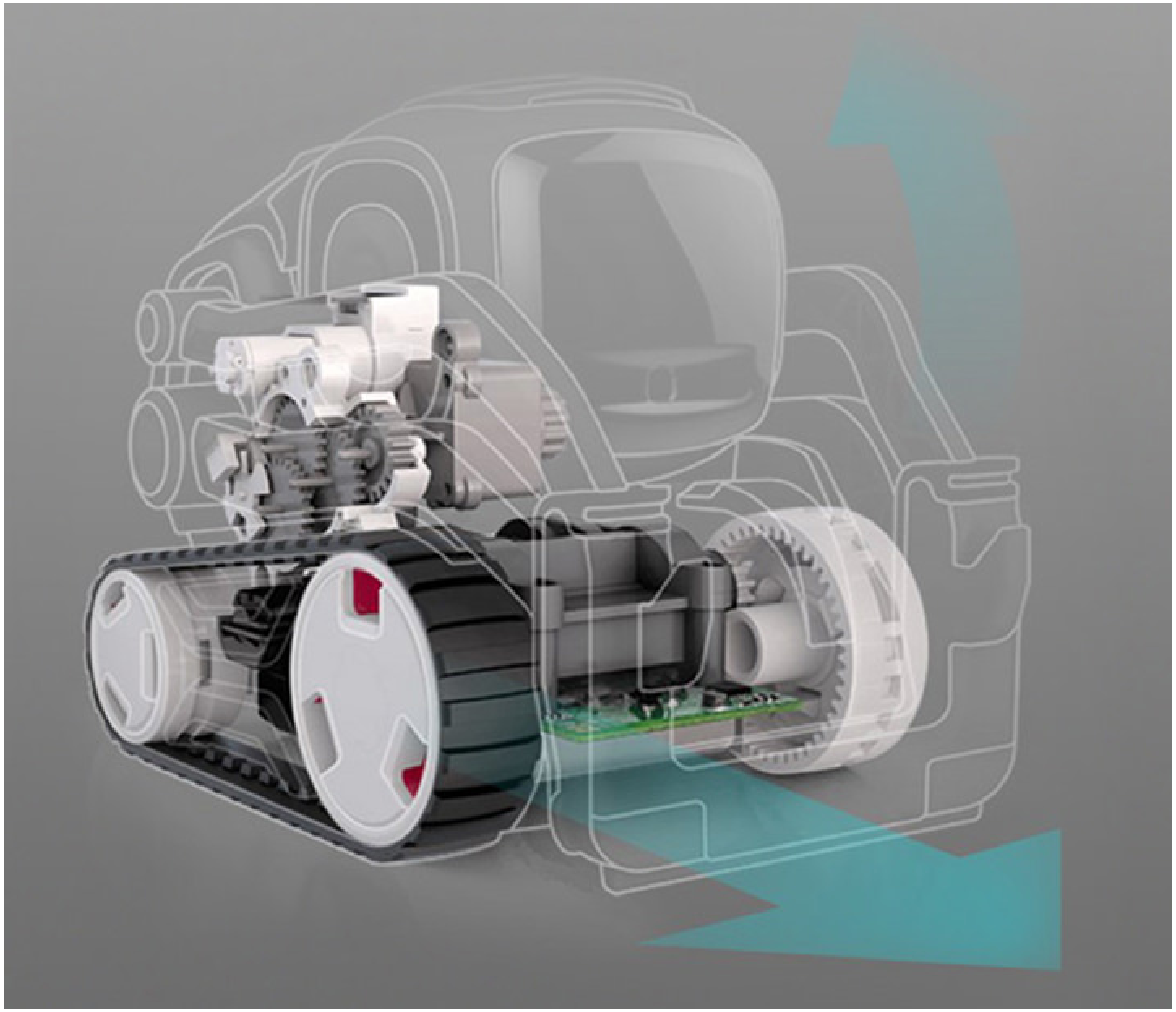
What's Cool About Cozmo?

- Cozmo is a **vision-guided mobile manipulator.**
  - Very few consumer robots have vision.
  - It can see special markers, recognize human faces.
- Open source SDK, plus 2 million lines of proprietary code.
- Cozmo uses AI algorithms internally.
- Cozmo was ridiculously cheap: $179 at Amazon or Best Buy.

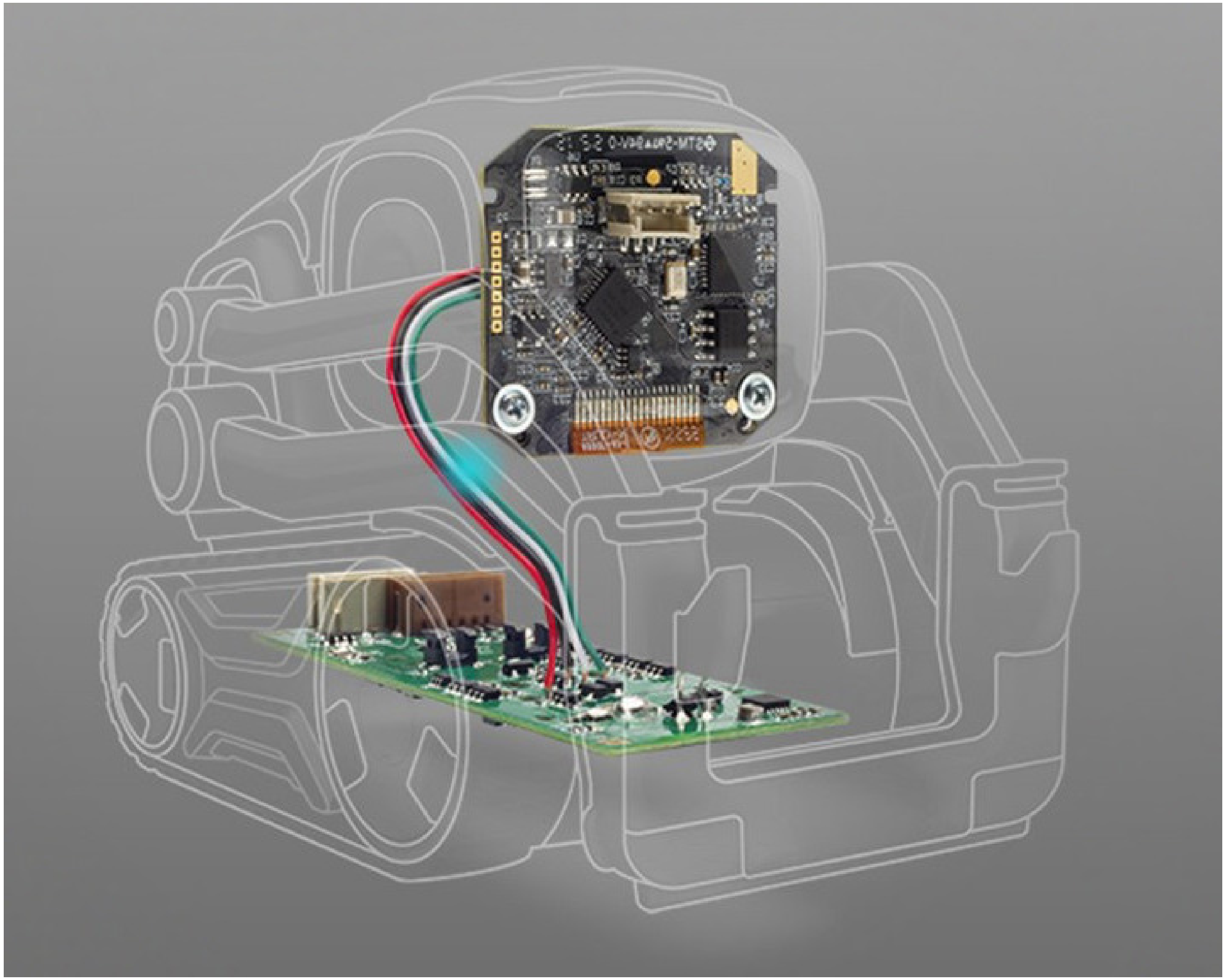
Over 300 Parts



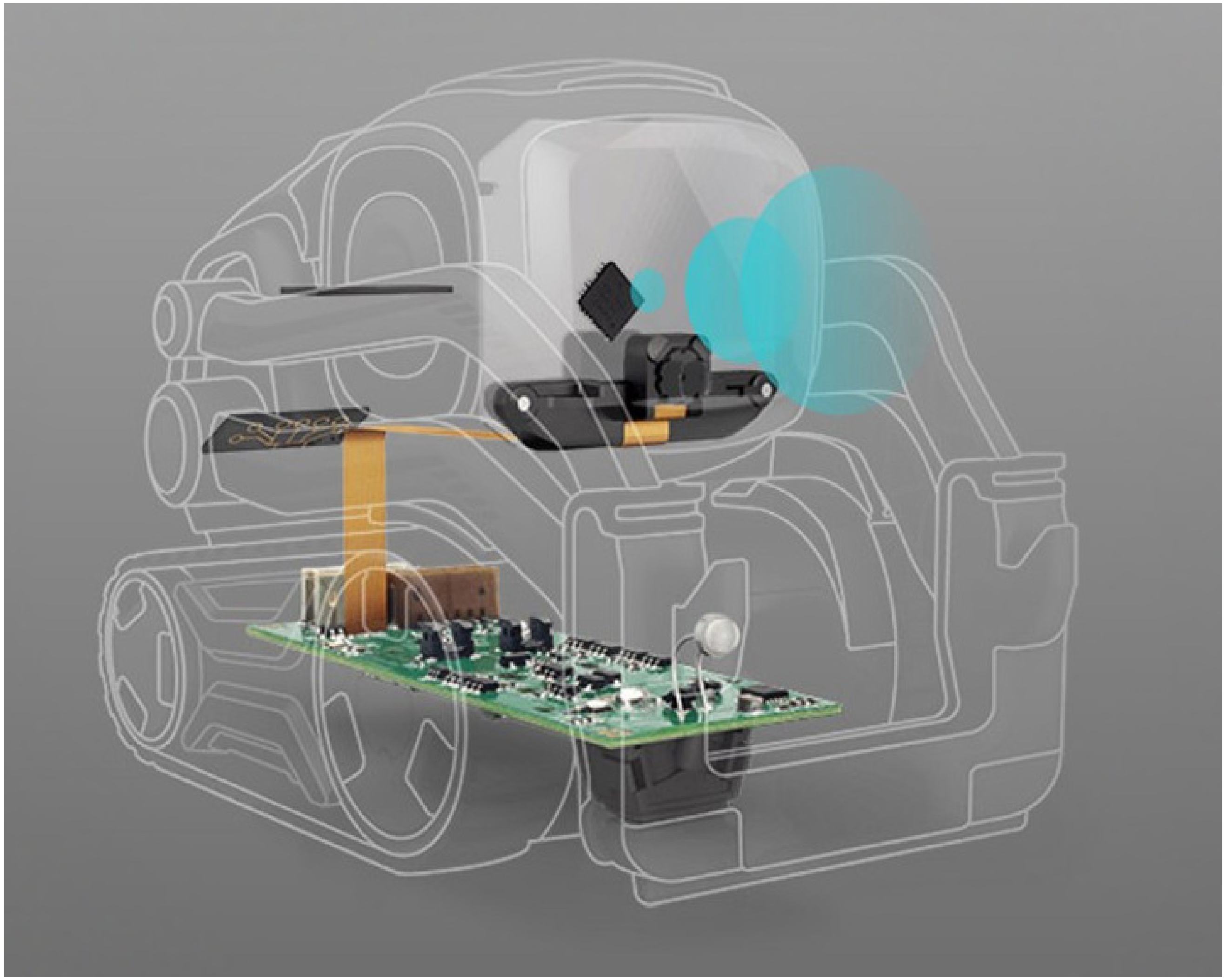
Four Motors: Wheels, Head, Lift



Three Small Circuit Boards

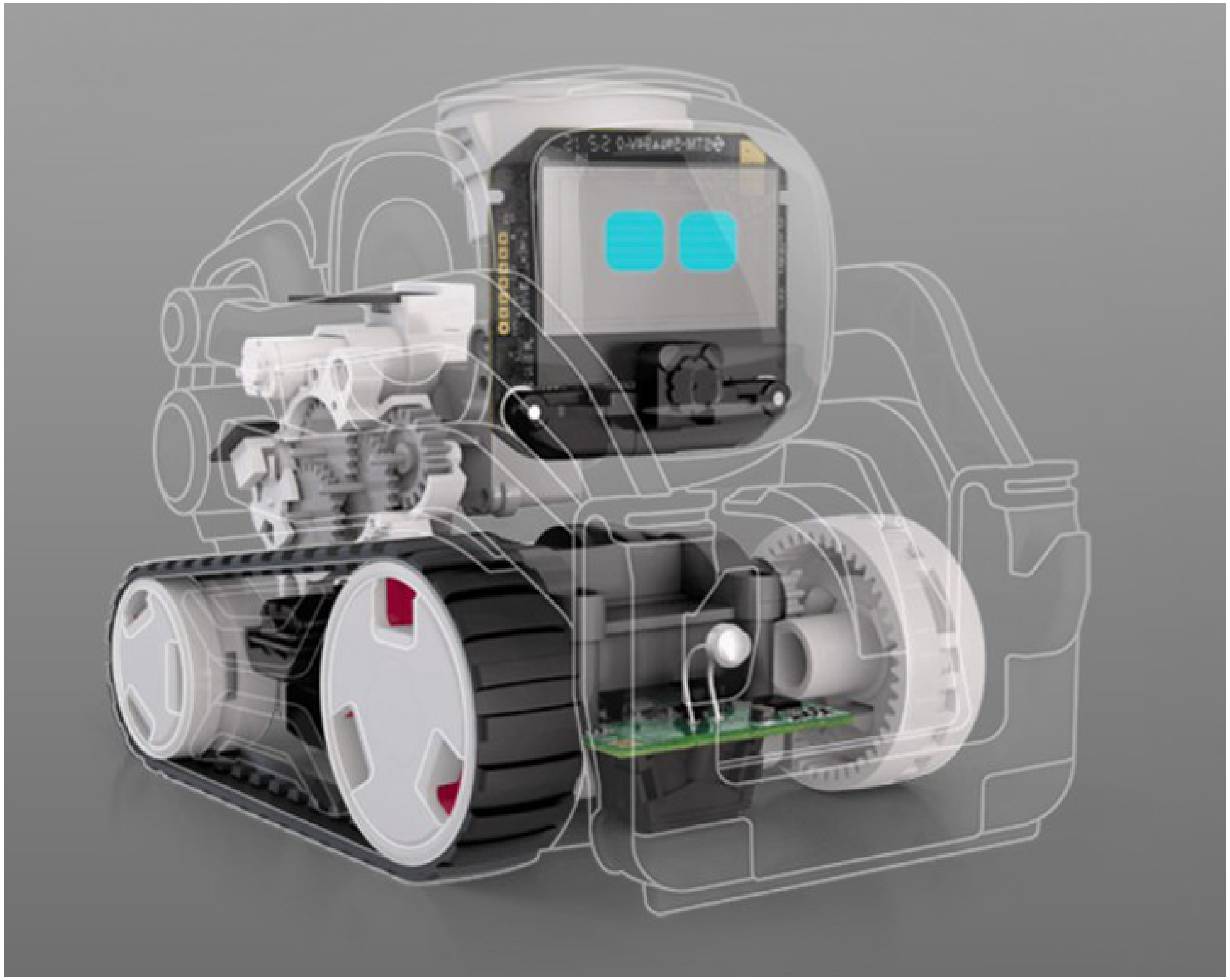


On-Board Camera

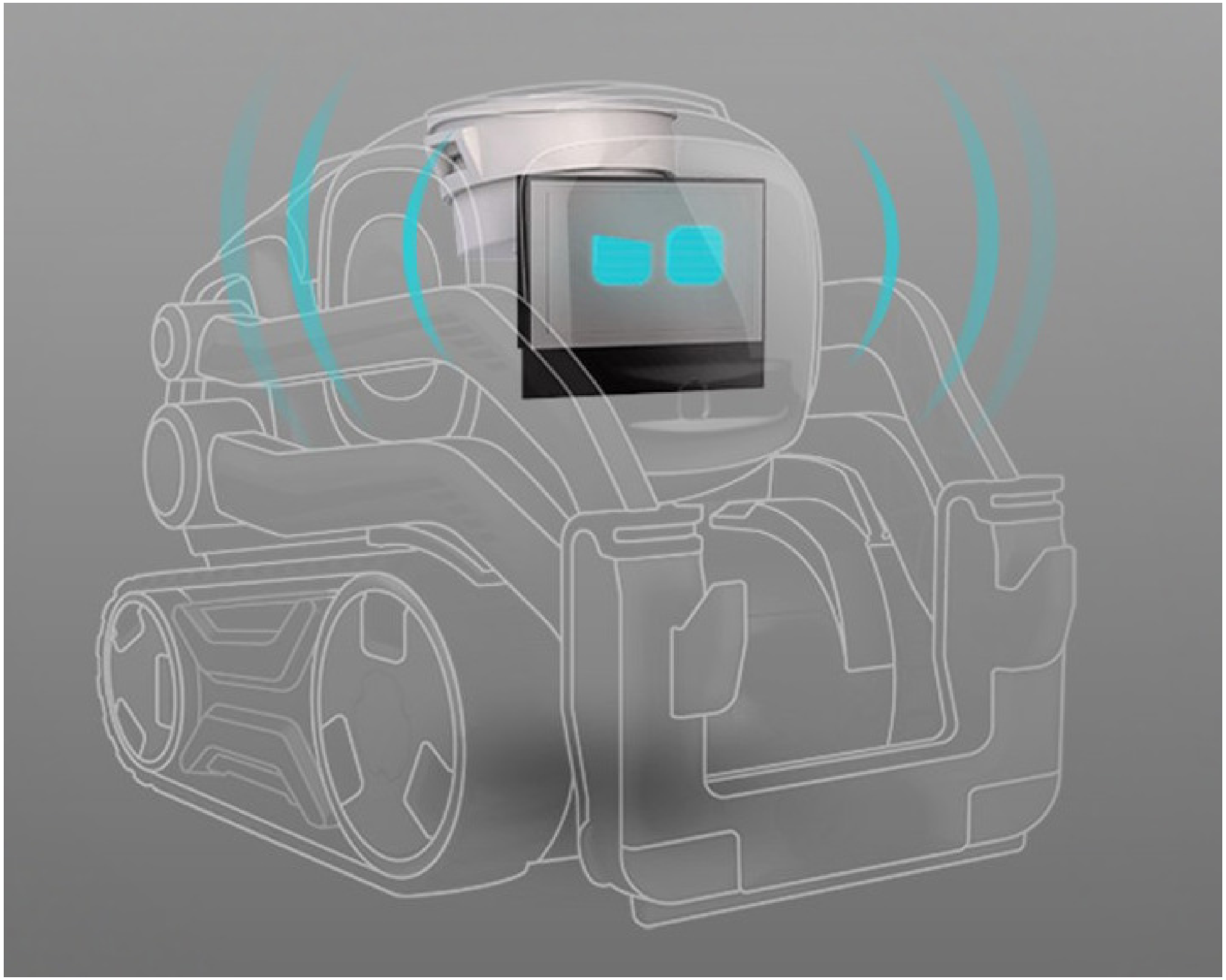


Images are 320x240 grayscale, or 160x240 color.

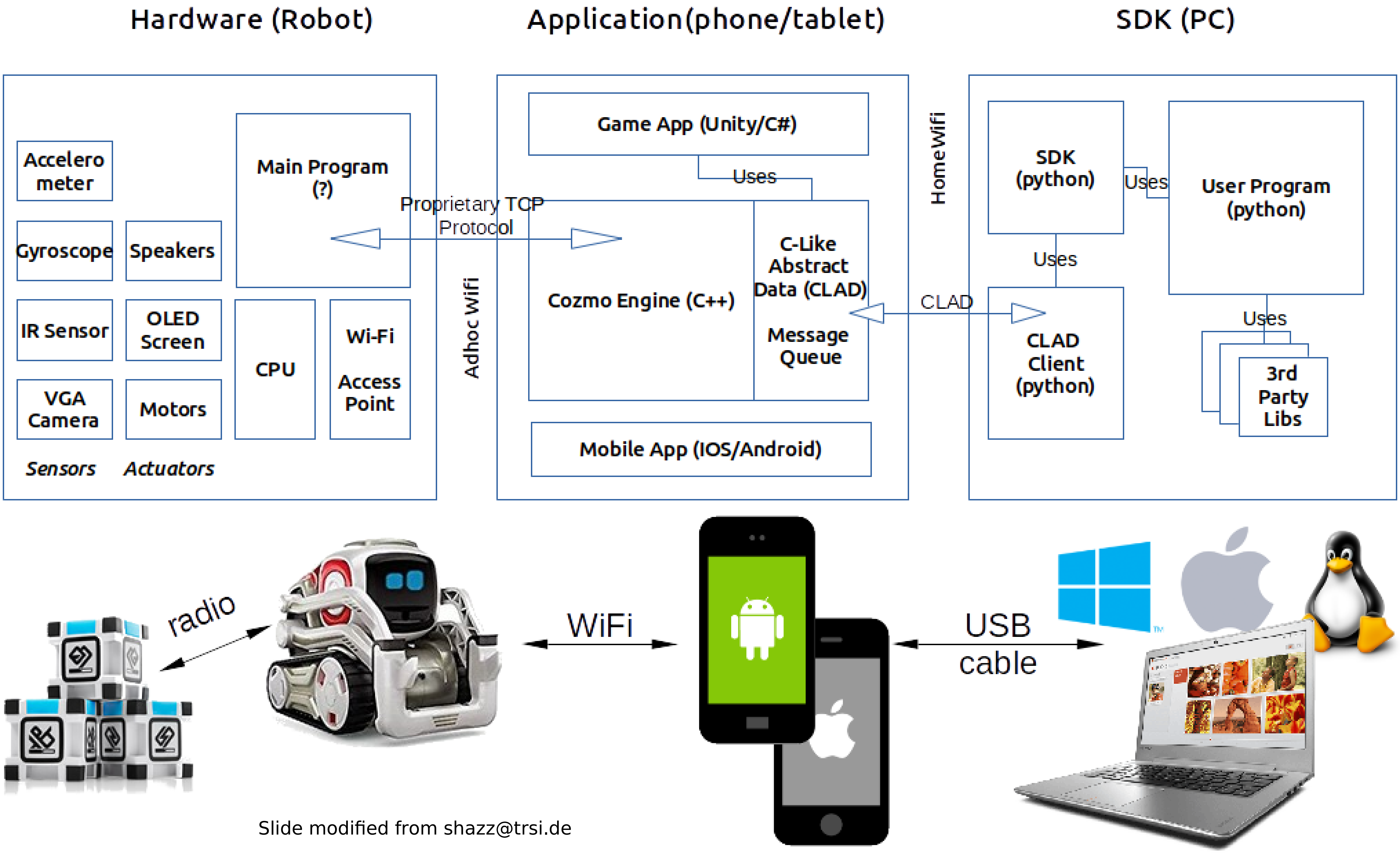
LED “Face” and Speaker



Pixar Designer “Animations”



Cozmo Architecture



Slide modified from shazz@trsi.de

Cozmo Resources

- Cozmo SDK documentation web site:
  See link from course home page.

- Cozmo Python SDK:
  https://github.com/anki/cozmo-python-sdk

- Cozmopedia:
  http://Cozmopedia.org

Vector

Vector Specs

- Qualcomm APQ8009 processor on board
- 1080p color camera
- Array of four microphones
- “Time of flight” laser rangefinder
- Four cliff detectors
- Accelerometer and gyroscope
- Touch-sensitive head panel
- Color “face” display

15-494/694 Cognitive Robotics

Now with GPUs!

Help us celebrate the new AI major by giving big new real-time brains to the Cozmo and Vector robots.

Mon/Wed 3:30-4:20
Fri labs 3:00-4:20

Prior robotics experience not required.

Deep
Neural
Networks

NVIDIA GeForce
RTX 2080 Ti
544 Tensor Cores
250 Watts TDP

Our future robot overlords will be pleased.

Machine Learning for Robotics

What can we do with ML?

- Apply existing deep neural net models for problems like object recognition.
- Train specialized vision-based controllers for tasks such as lane-keeping (self-driving cars) or skilled manipulation.
- Train a robot to recognize another robot.
- Whatever you can think of!

GPU-Enabled Robotics

GeForce RTX 2070

- 256 bit bus width
- 288 Tensor cores
- 10.8 billion transistors
- 8 GB of RAM
- FP16 performance: 12,994 GFLOPS
- 175 watts TDP (thermal design power)

Ten “Big Ideas” in Robotics

- “Big ideas” are the key concepts people should learn when approaching a new field (Wiggins and McTighe).
- “Essential questions” are a way of leading people to the big ideas.
- Read the “Ten Big Ideas” paper linked from the class schedule for today.

1. How Do Robots Know What To Do?

Big idea:

- Autonomous robot behaviors are **mechanisms** constructed from carefully designed algorithms and representations.

Underlying technologies:

- State machines; event-based architectures.

2. How Do Robots See the World?

Big idea:

- Robots use sophisticated but imperfect **computer vision algorithms** to deduce real world object representations from arrays of pixels.

Underlying technologies:

- Hough transforms; face detection algorithms; ArUco markers; much more…

3. How Do Robots Know Where They Are?

Big idea:

- Robots estimate their position in the world using a combination of **odometry**, **visual landmarks**, and other types of sensor information.

Underlying technologies:

- Particle filters; SLAM (Simultaneous Localization and Mapping) algorithms.

4. How Do Robots Know Where To Go?

Big idea:

- Robots navigate through the world using a **path planner** to search for routes around obstacles to reach their goal.

Underlying technology:

- Path planning algorithms such as wavefront algorithms or RRTs (Rapidly-exploring Random Trees).

5. How Do Robots Control Their Bodies?

Big idea:

- Robots describe their bodies as **kinematic trees** and use **kinematics solvers** to translate between joint angles and body coordinates.

Underlying technologies:

- Kinematic description files; Denavit-Hartenberg conventions; forward and inverse kinematics solvers.

6. What Can We Do When A Robot Becomes Too Complex for One Person to Fully Understand It?

Big idea:

- Robots are complex software systems that employ standard **abstraction** and **software engineering** techniques to **manage complexity.**

Underlying technologies:

- Modular design; coding standards; class libraries; documentation generators.

7. How Do We Calculate the Quantities Needed to Make A Robot Function?

Big idea:

- Geometry, trigonometry, and linear algebra are the **mathematical underpinnings** of much of robotics.

Underlying technologies:

- Software libraries for linear algebra, angular arithmetic, quaternions, etc.

8. How Can Robots Solve Complex Problems?

Big idea:

- Robots use **task planning** to search a space of **world states** to find a path to a goal state.

Underlying technologies:

- Task planners; domain description languages; plan execution and monitoring architectures.

9. How Should Robots Behave Around People?

Big idea:

- Successful **human-robot interaction** requires awareness of humans in the environment and adherence to social conventions such as not following too closely.

Underlying technologies:

- Human-tracking tools (e.g., OpenPose); face, gesture, and speech recognition; natural language dialog systems.

10. How Can Robots Work Together?

Big idea:

- **Inter-robot communication** and **multi-robot coordination** algorithms allow robots to collaborate.

Underlying technologies:

- Communications primitives; shared world maps; multi-robot planners.

Three Programming Frameworks for Cozmo

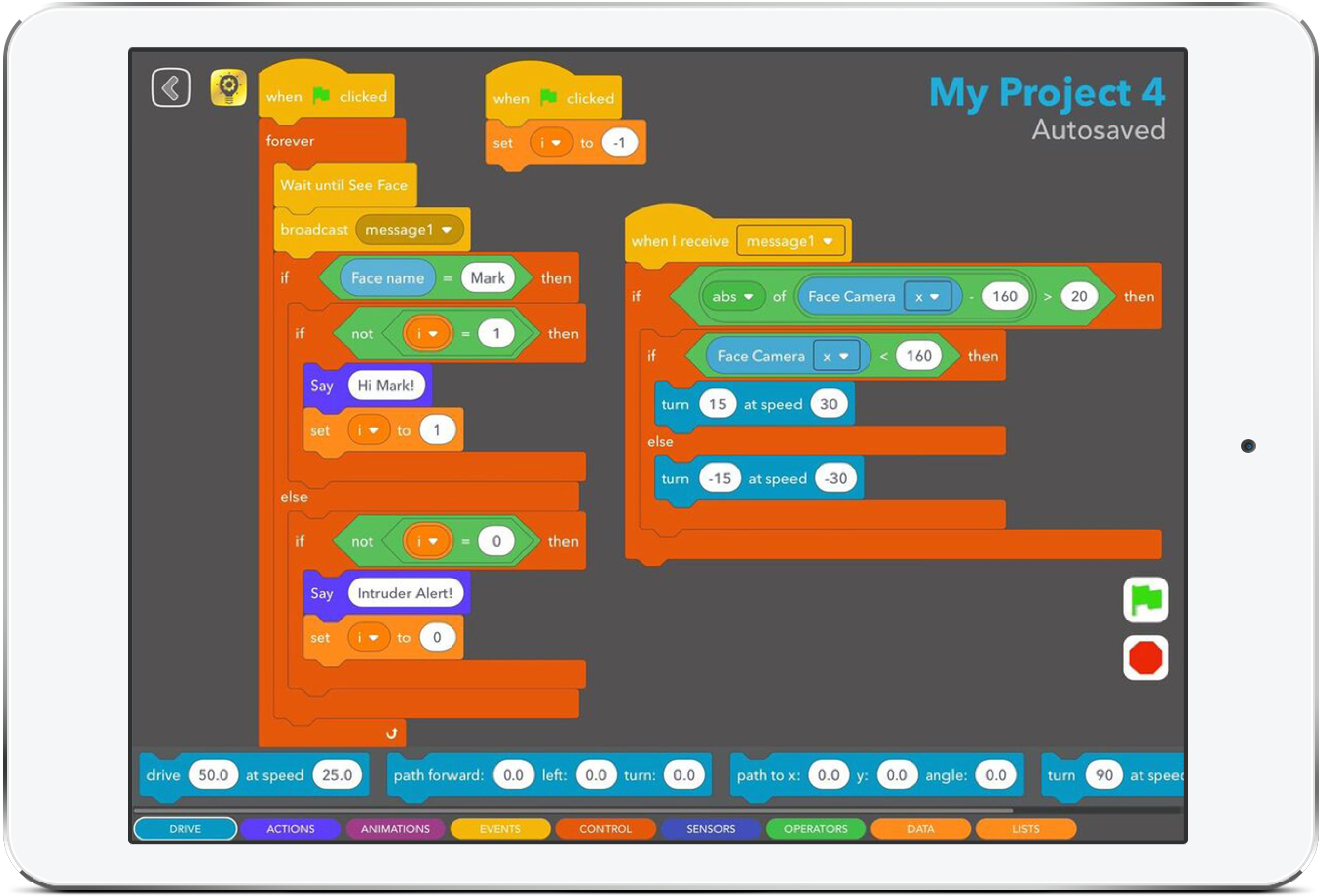
- Cozmo Python SDK
  - cozmo-tools is built on top of this

- Code Lab – built in to the Cozmo app

- Calypso – runs on a laptop

How well does each framework facilitate exploring the ten big ideas?

Code Lab is based on Scratch 3.0



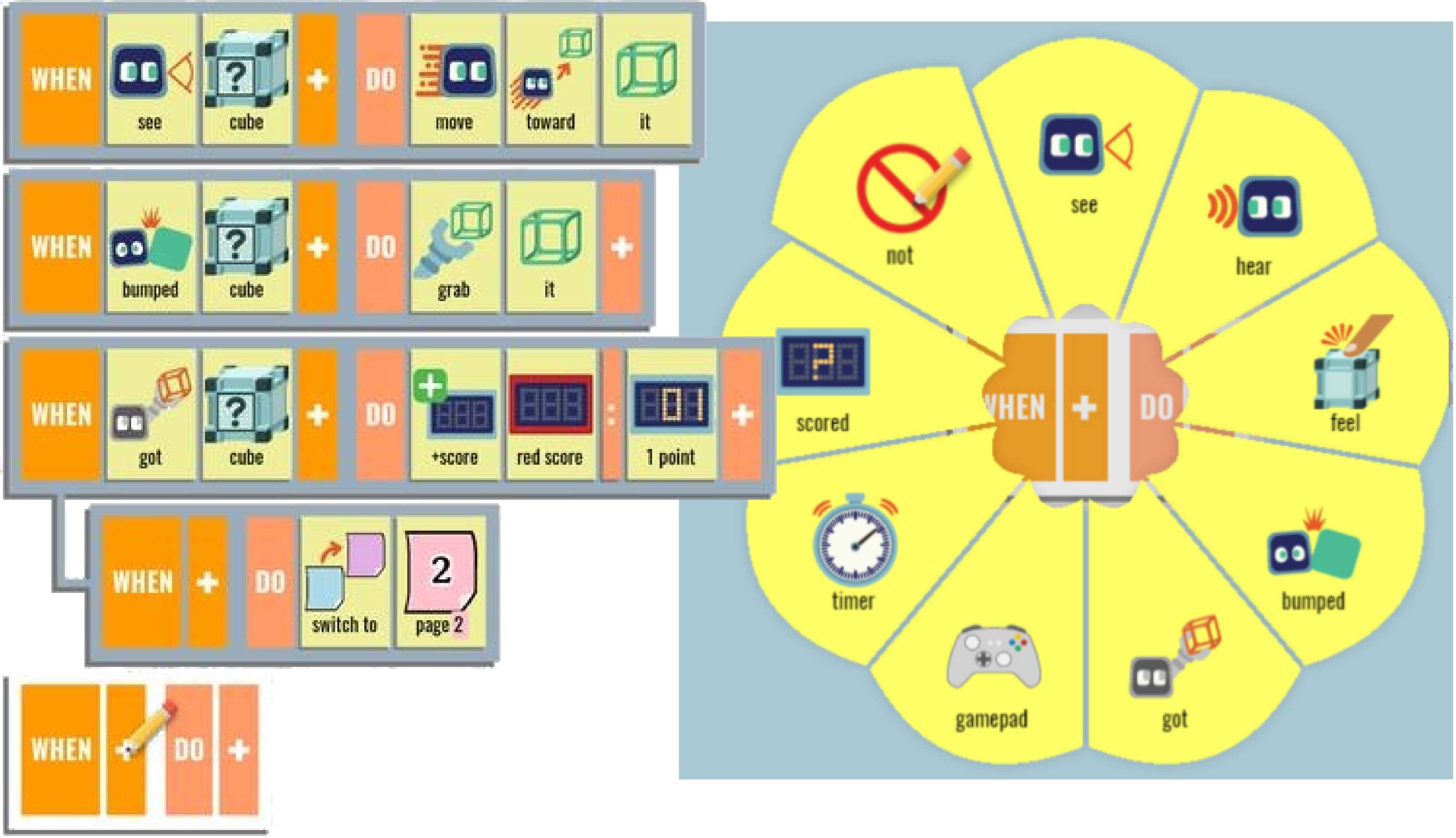
```
when ⚑ clicked
forever
    Wait until See Face
    broadcast message1
    if Face name = Mark then
        if not i = 1 then
            Say Hi Mark!
            set i to 1
    else
        if not i = 0 then
            Say Intruder Alert!
            set i to 0
↺
```

```
when ⚑ clicked
set i to -1
```

```
when I receive message1
if abs of Face Camera x - 160 > 20 then
    if Face Camera x < 160 then
        turn 15 at speed 30
    else
        turn -15 at speed -30
```

```
drive 50.0 at speed 25.0
path forward: 0.0 left: 0.0 turn: 0.0
path to x: 0.0 y: 0.0 angle: 0.0
turn 90 at spee
```

Calypso is “Kodu for Robots”



What You Should Do Now

1. Bookmark the course home page.
2. Read the “Big Ideas” paper.
3. Brush up on your Python 3: classes, modules, numpy.
4. If you want to install the Cozmo SDK on your personal machine, visit the Cozmo SDK documentation page.
5. Note that Friday labs start at 3:30 PM, not 4:00. Lab attendance is required.