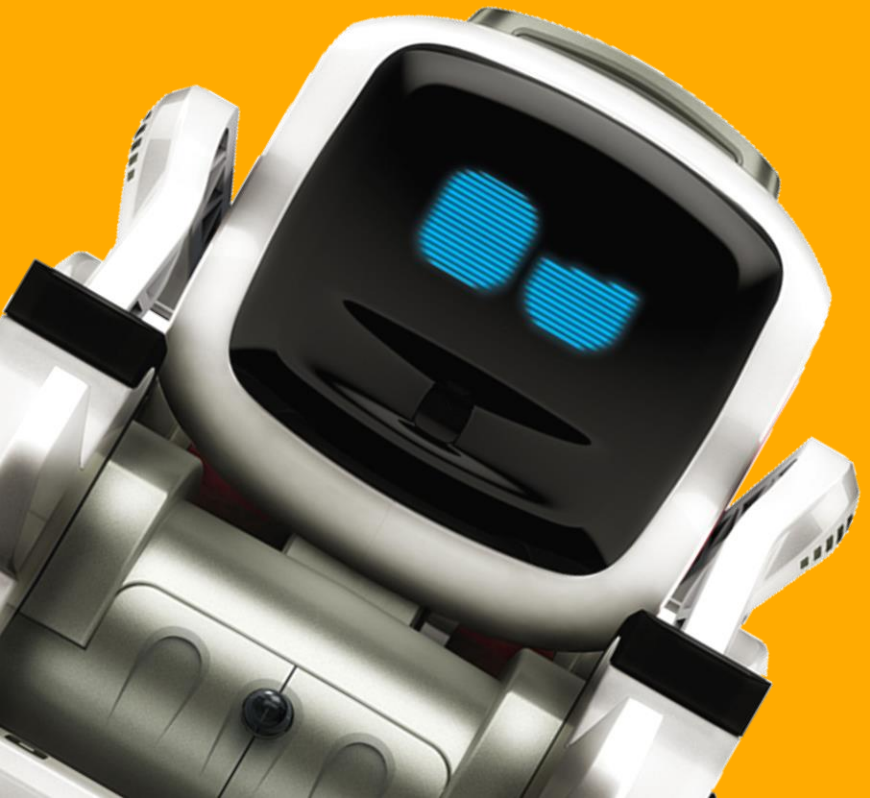
Robotics with *Attitude*



Programming a Real Life Pixar Character

@timcomport

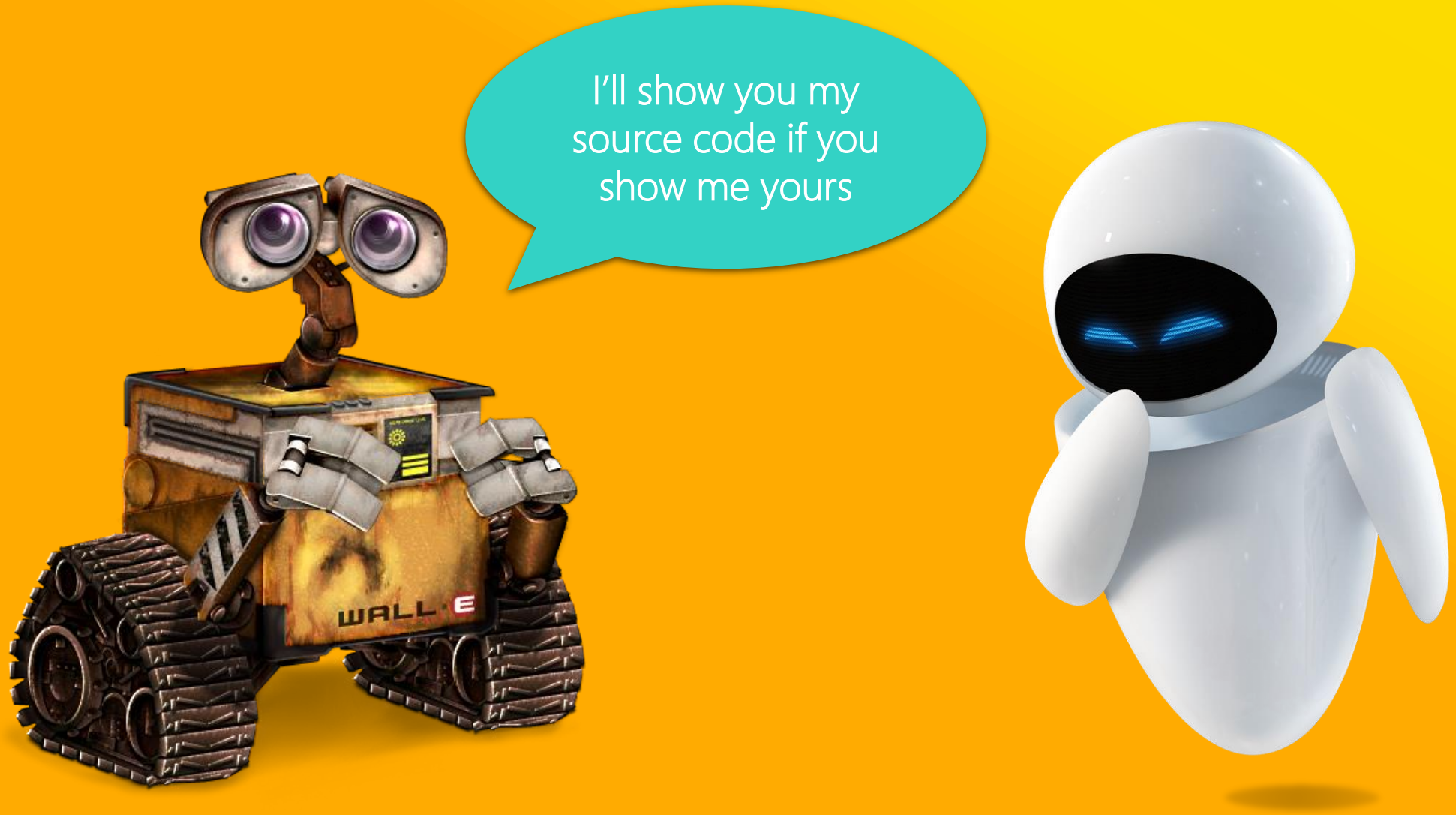
Media courtesy of anki™ and Disney · PIXAR

"Any sufficiently advanced technology is indistinguishable from magic."

Arthur C. Clarke

COZMO®

The Magic

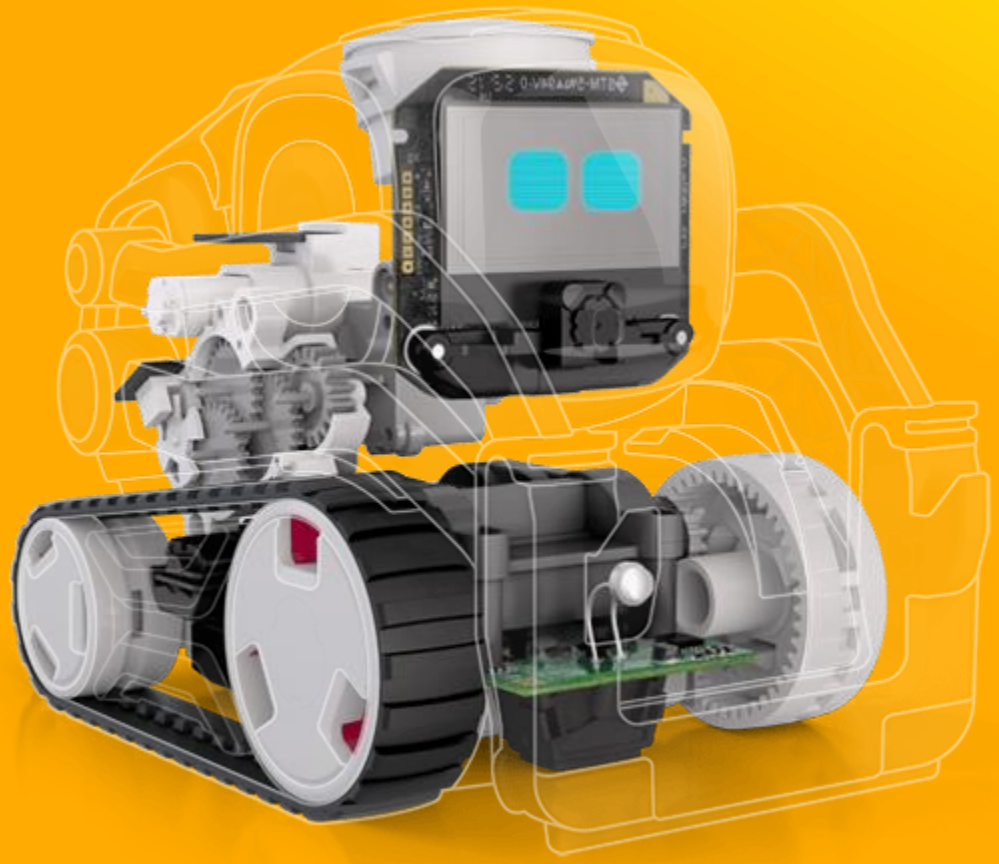

I'll show you my source code if you show me yours
WALL·E

Best Friend

Game Time

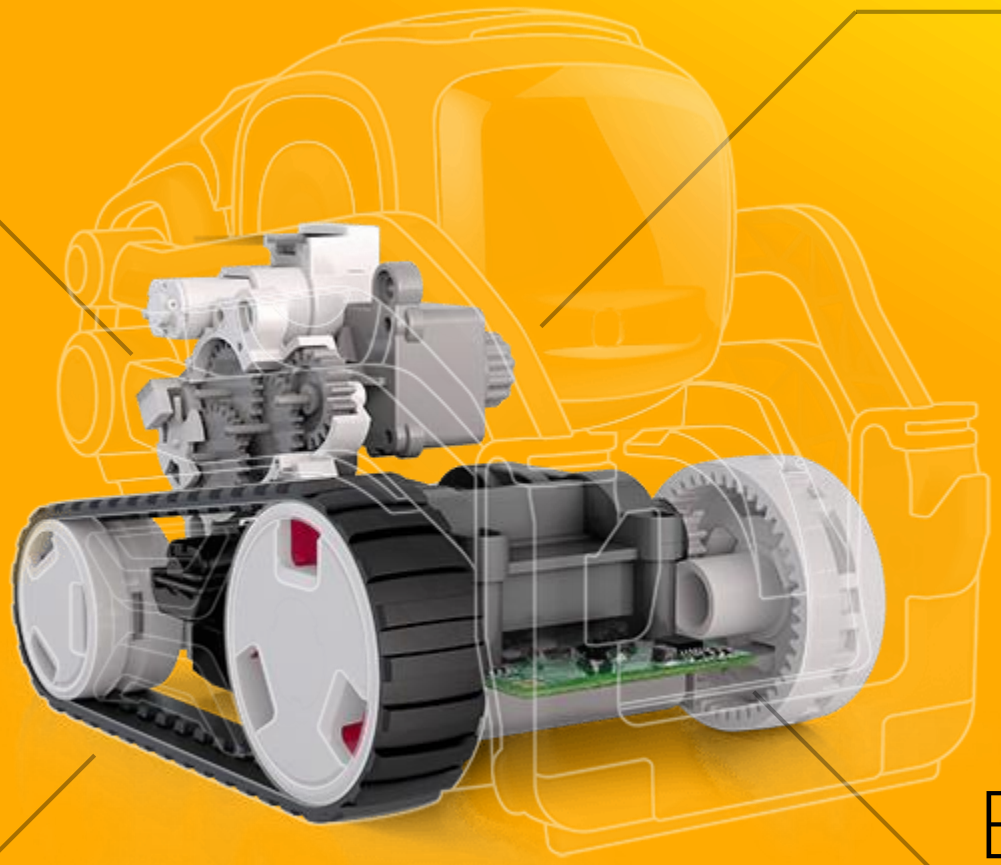
COZMO® | TECH

The Magic Revealed



4 motors

50+ gears

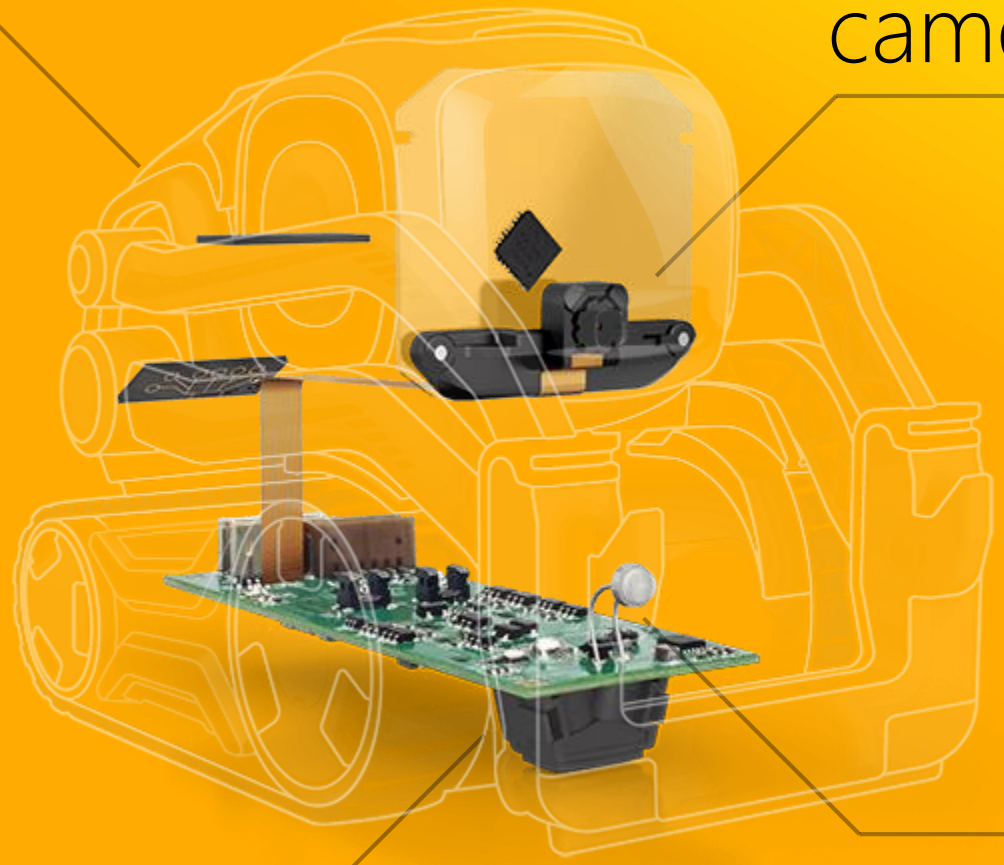


Tank treads

Encoders

RGB status LEDs

30 FPS VGA camera

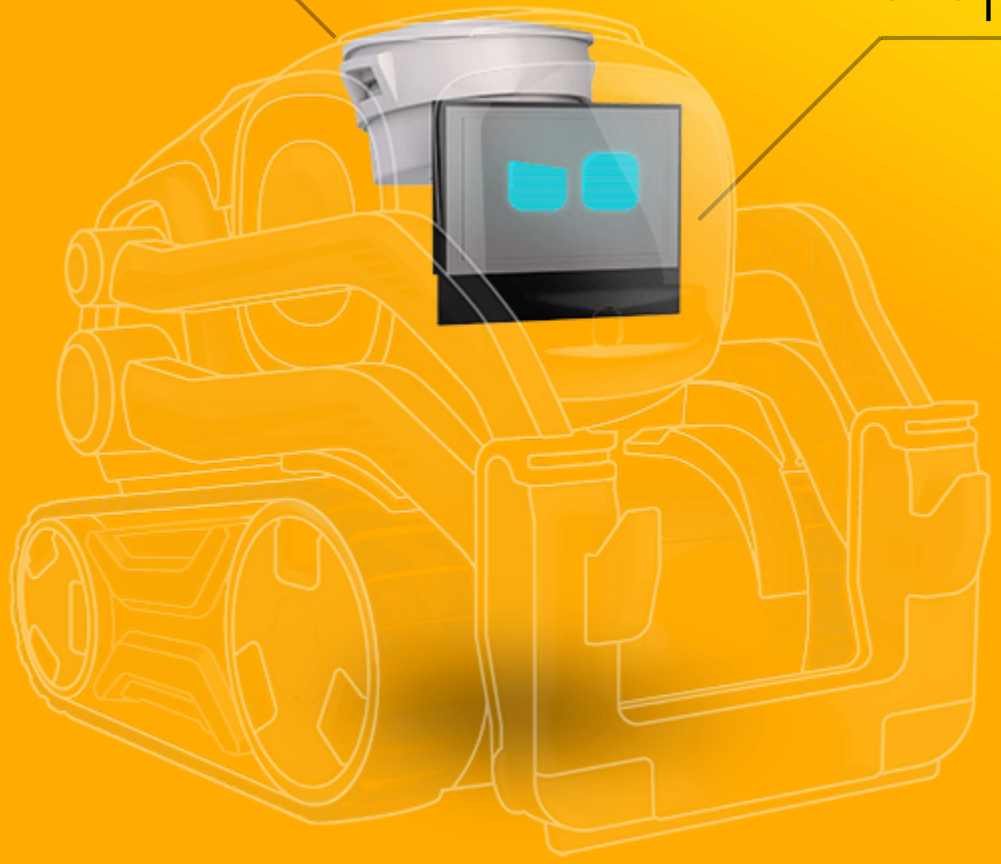


IR drop sensor

IR LED

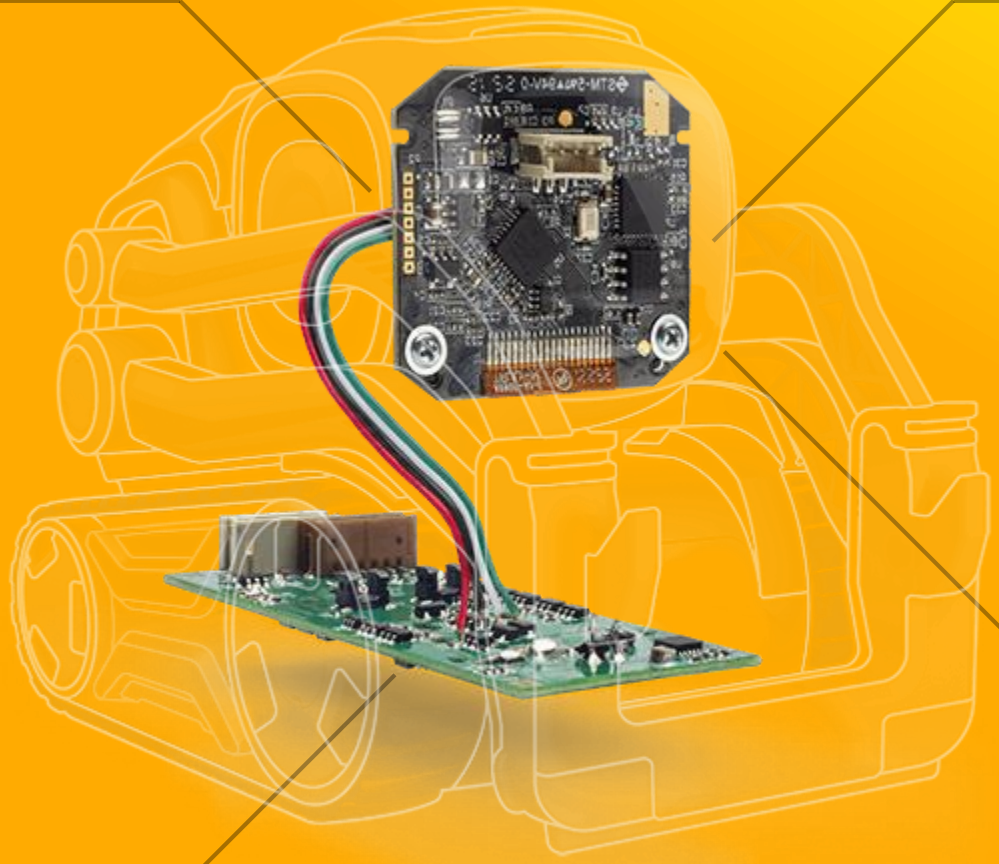
Speaker

128 x 64 OLED display



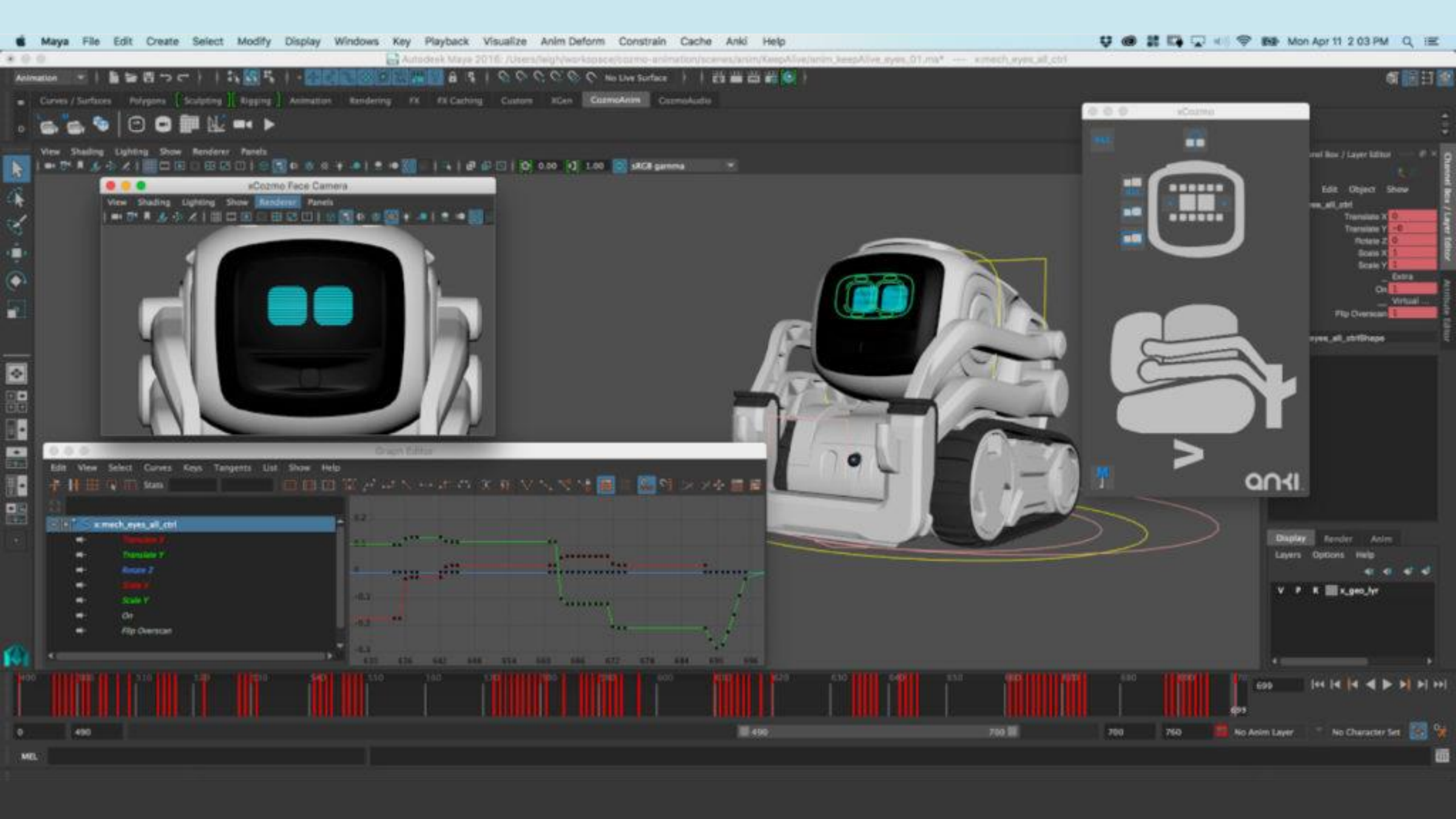
Multiple On-board CPUs

WiFi



Accelerometer

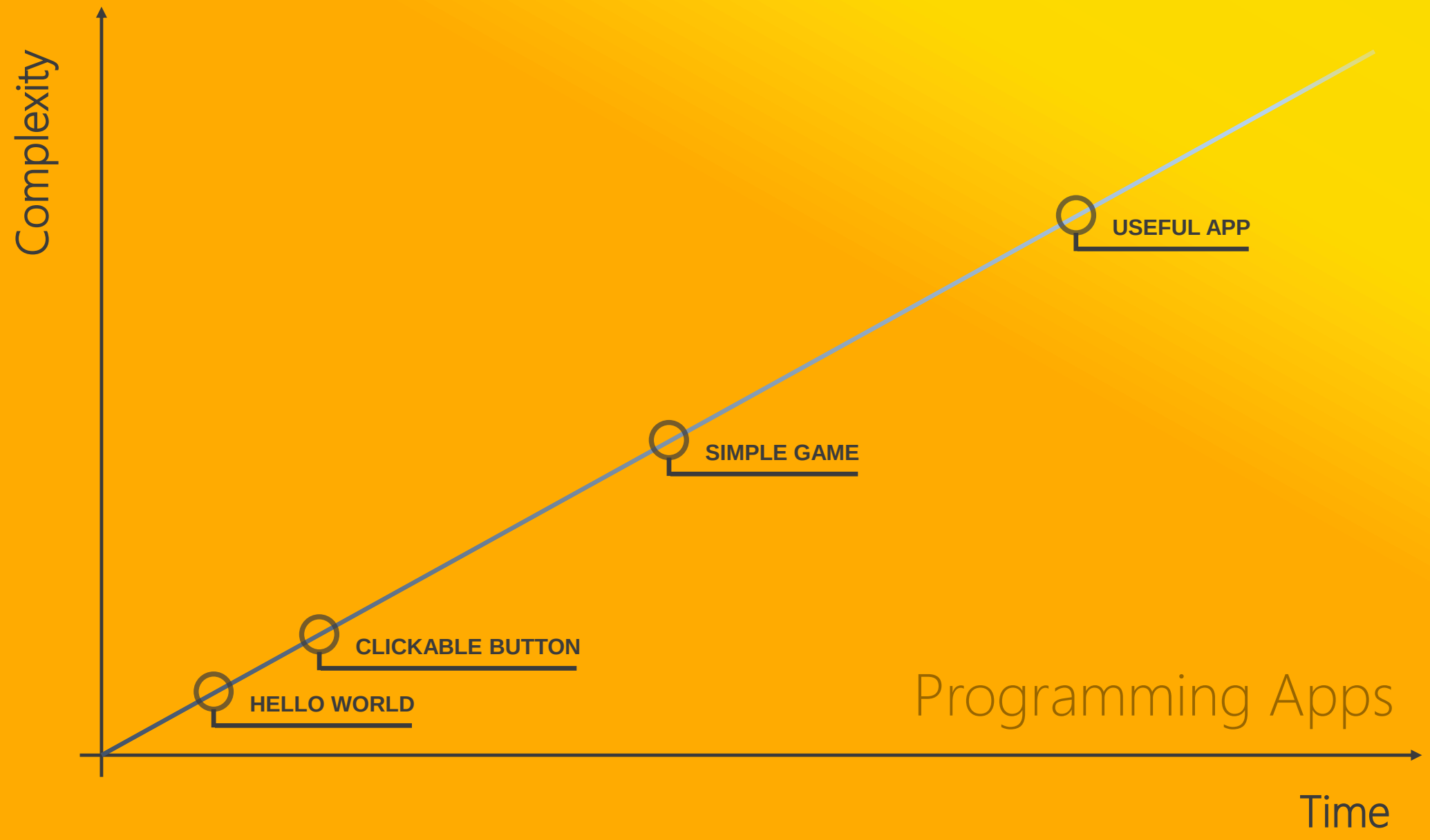
Gyroscope

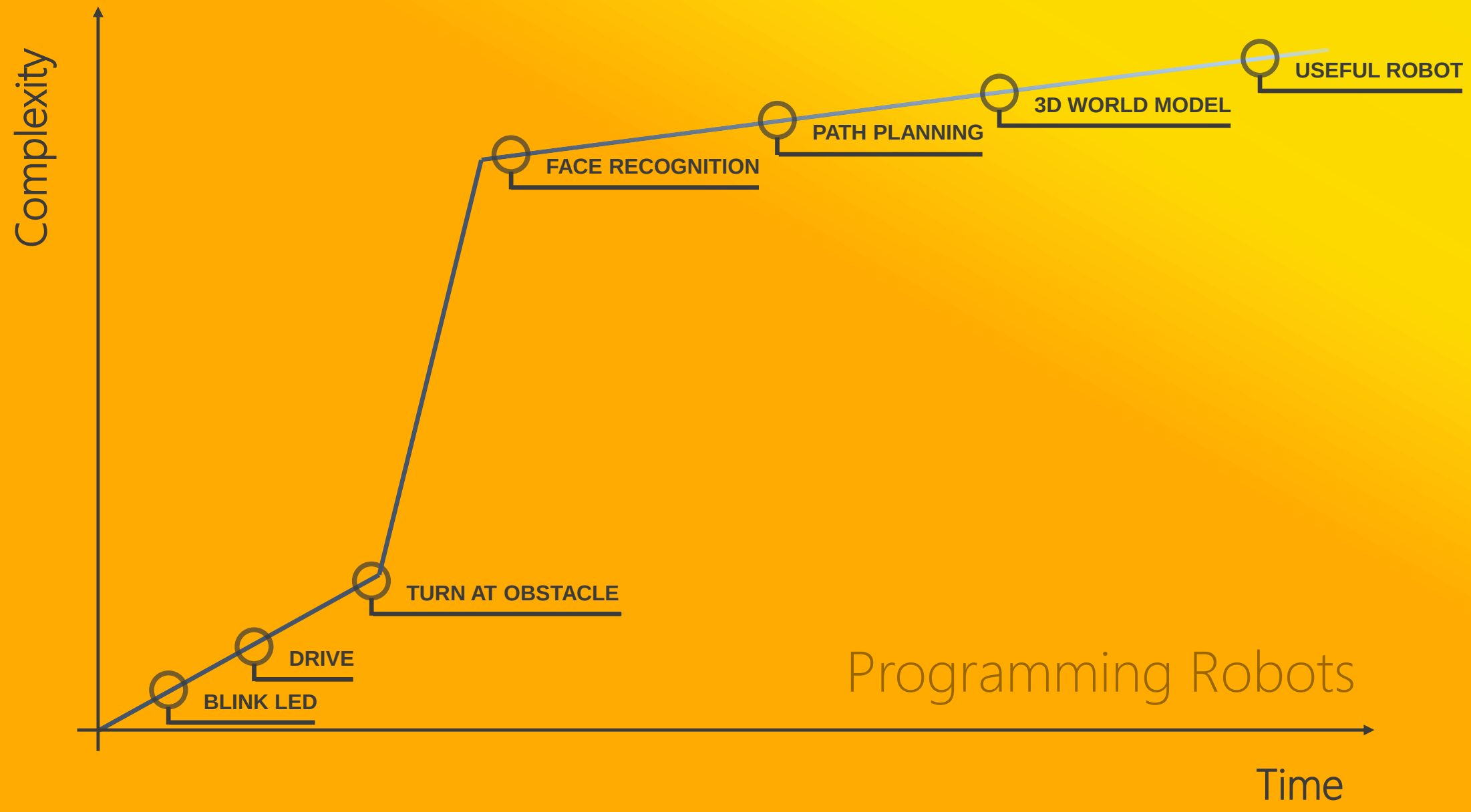

Maya File Edit Create Select Modify Display Windows Key Playback Visualize Anim Deform Constrain Cache Anki Help
Mon Apr 11 2:03 PM
Curves / Surfaces Polygons Sculpting Rigging Animation Rendering FX FX Caching Custom XGen CozmoAnim CozmoAudio
View Shading Lighting Show Renderer Panels
xCozmo Face Camera
xCozmo
anki
Graph Editor
Edit View Select Curves Keys Tangents List Show Help
x:mech_eyes_all_ctrl
Translate Y
Translate Y
Rotate Z
Scale Y
On
Flip Overscan
Display Render Anim
Layers Options Help
x_geo_lyr
No Anim Layer
No Character Set
MEL

Est. Pose: (x,y)=(-68.4452, 137.7813) at deg=23.1
True Pose: (x,y)=(-74.7573, 138.9384) at deg=22.7
LookAtFaceVerified
LookAtFaceVerified
RollSucceeded
0.98: Charged
0.39: Social
0.12: Happy
0.12: Confident
0.06: Excited
-0.13: Winning
1.0
-1.0
1.78: StackCubes
0.98: InteractWithFace
0.68: visitPossibleHa
0.48: FindCubes
0.38: Pounce
0.27: FindFaces
0.04: Wander
0.00: DriveOffCharger
0.00: PutDownCube
0.00: RequestSpeedTap
0.00: RollCube
Pose: -67.2, 135.0, ang: 13.3
Head: -1.3 deg, Lift: 32.3 mm
Pitch: -0.1 deg (IMUHead: -1.3 deg)
Speed L: 0 R: 0 mm/s
Batt: 5.0 V AnimTracksLocked: LHB
Video: 15 Hz Proc: 15 Hz
AnimBytesFree[AF]: 16115[4]
Status: PAP
ErrSig x:114.0 y:-3.2 z:22.6 a:-0.01
Action: [DriveToObjectAction+TurnTowardsLastFaceP
LocalizedTo: Block_LIGHTCUBE1_0
WorldOrigin[1]: Robot1_PoseOrigin0
Vision: Tracking
A StackCubes-PickingUpBlock
Anim: (none)
15Hz GoalEvaluatorA StackCubes-PickingUpBloc

COZMO® | SDK

Make Your Own Magic


Complexity
HELLO WORLD
CLICKABLE BUTTON
SIMPLE GAME
USEFUL APP
Programming Apps
Time


Complexity
BLINK LED
DRIVE
TURN AT OBSTACLE
FACE RECOGNITION
PATH PLANNING
3D WORLD MODEL
USEFUL ROBOT
Programming Robots
Time

Vision System

- Face Recognition
- Edge Detection
- “Interesting Objects”
- Marker Detection
- Motion Detection
- Colour Detection
- Camera Streaming

Face Tracking

- High Speed Tracking
- Position / Orientation
- 44 Face Markers
- Face Recognition
- Age / Gender
- Eye-Openness
- Facial Expressions

Planning / A.I.

- Realtime Path Planner
- Realtime SLAM
- World Exploration
- Emotion Engine
- Behaviour Engine

Manipulation

- 3D Position Tracking
- Push
- Pickup
- Stack
- Roll

Animations

- 128 x 64 OLED Face
- Maya Animations
- JSON Animation format
- Sound Streaming
- 4 DOF

All the low level access to Robot, Sensors, Cubes, etc.

Solution1 - hello_world.py

hello_world.py

run

```
import sys
import cozmo

def run(sdk_conn):
    '''The run method runs once Cozmo is connected.'''
    robot = sdk_conn.wait_for_robot()
    robot.say_text("Hello World").wait_for_completed()

if __name__ == '__main__':
    cozmo.setup_basic_logging()

    try:
        cozmo.connect(run)
    except cozmo.ConnectionError as e:
        sys.exit("A connection error occurred: %s" % e)
```

160 %

Solution1 - cube_stack.py

cube_stack.py

run

```
def run(sdk_conn):
    '''The run method runs once Cozmo is connected.'''
    robot = sdk_conn.wait_for_robot()

    lookaround = robot.start_behavior(cozmo.behavior.BehaviorTypes.LookAroundInPlace)
    cubes = robot.world.wait_until_observe_num_objects(num=2, object_type=cozmo.objects.LightCube)
    lookaround.stop()

    if len(cubes) < 2:
        print("Error: need 2 Cubes but only found", len(cubes), "Cube(s)")
    else:
        current_action = robot.pickup_object(cubes[0])
        current_action.wait_for_completed()
        if current_action.has_failed:
            code, reason = current_action.failure_reason
            print("Pickup Cube failed: code=%s reason=%s" % (code, reason))

        current_action = robot.place_on_object(cubes[1])
        current_action.wait_for_completed()
        if current_action.has_failed:
            code, reason = current_action.failure_reason
            print("Place On Cube failed: code=%s reason=%s" % (code, reason))
```

160 %