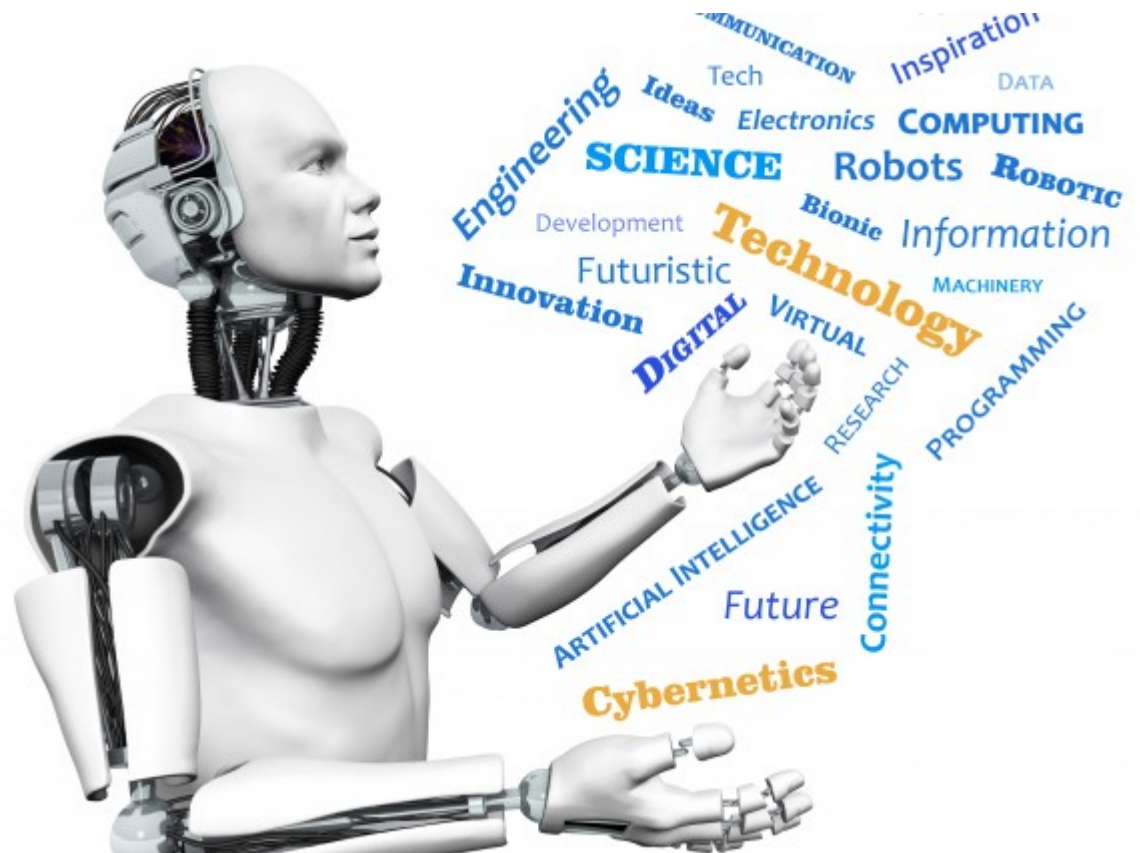


# 15-494/694: Cognitive Robotics

Dave Touretzky

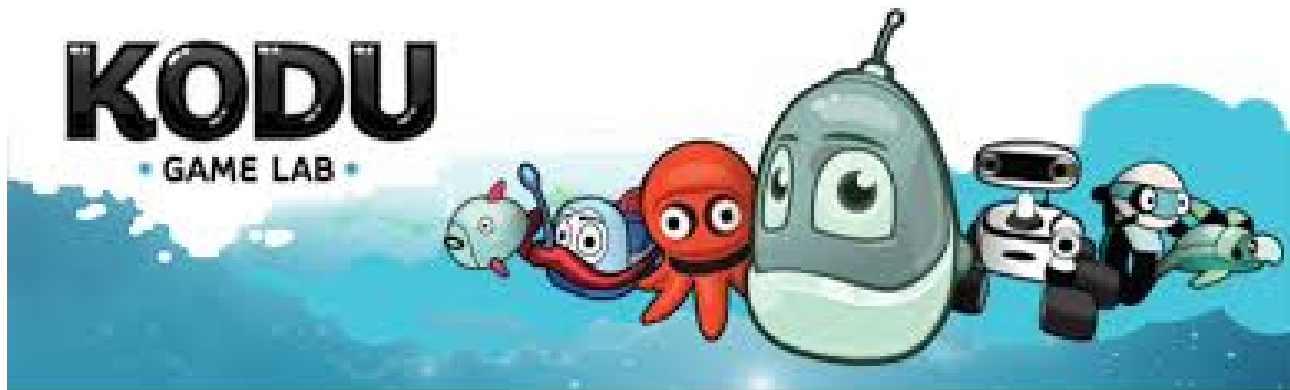
Lecture 15:

Calypso (Kodu for Robots)



# Microsoft's Kodu Game Lab

- Children's programming language: make your own computer games.
- Developed by Microsoft FUSE Labs.
- Released in 2009 for Xbox 360 and Windows.
- Inspired by behavior-based robotics.



# Kodu Worlds

Full 3D, with physics and sound effects.



# “Parallel” WHEN-DO Rules



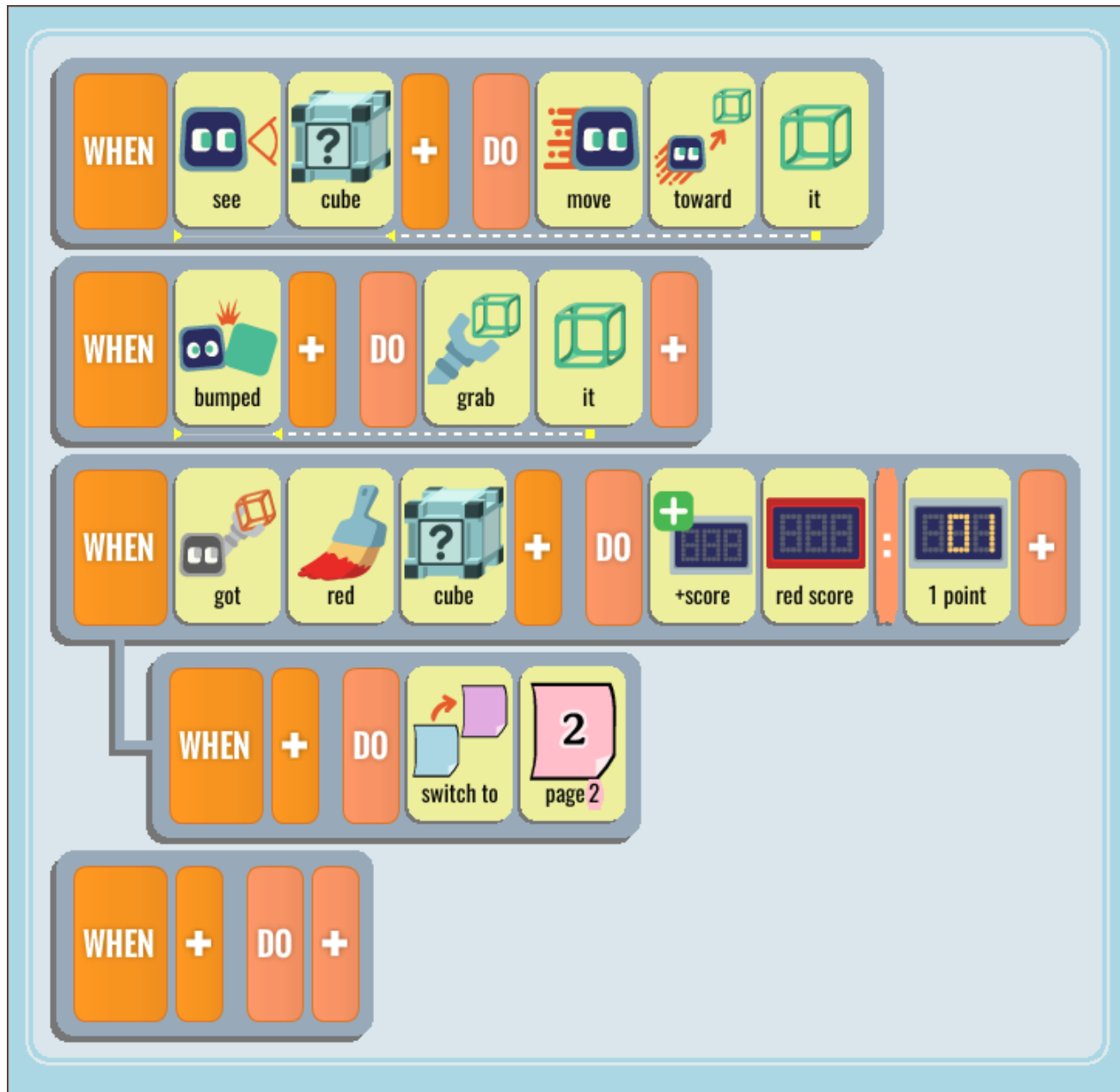
# Menu Selection



# Calypso: Kodu for Robots



# Sample Calypso Program



# Context-Sensitive Petal Menus



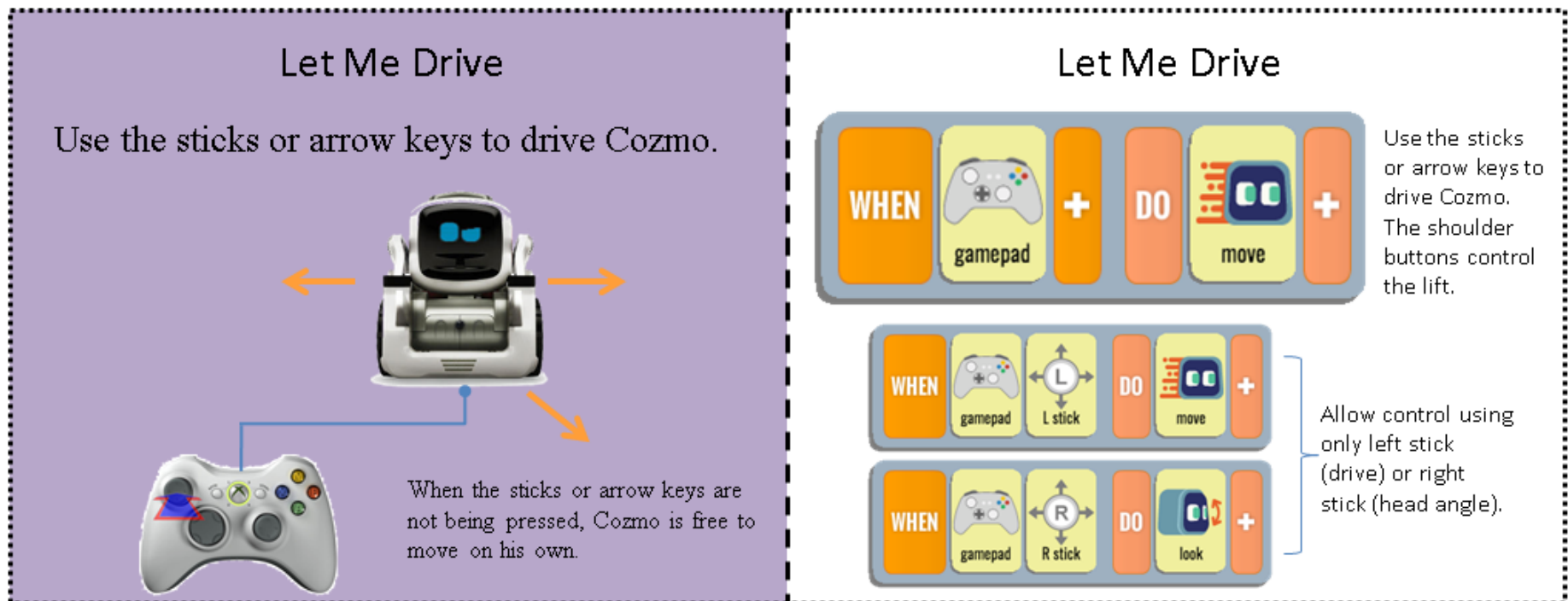


# The Robot's World Map

The screenshot displays the Calypso 0.9.04 web interface. The browser address bar shows the URL `127.0.0.1:43125/Calypso/index.html`. The interface is divided into several sections:

- Control Panel (Left):** Includes buttons for "Stop program", "State machine view" (Ctrl ↑), "Switch characters" (Ctrl ← →), "Map editor view" (Ctrl ↓), and "Scroll up/down" (Esc).
- Scripting Area (Center-Left):** Shows two rows of logic blocks. The top row is active and contains: "WHEN" (orange), "see" (robot icon), "cube" (cube icon), "+", "DO" (orange), "move" (robot icon), "toward" (arrow icon), and "it" (cube icon). The bottom row is inactive and contains: "WHEN" (grey), "bumped" (robot icon), "cube" (cube icon), "+", "DO" (grey), "grab" (robot icon), "it" (cube icon), and "+".
- World Map (Right):** A large white area representing the robot's environment. It shows a robot icon at the bottom, a red line indicating its field of view, and several cube icons scattered across the map.
- Video Feed (Bottom-Left):** A black and white camera view of the robot's environment. Two cubes are highlighted with yellow boxes and labeled "Lightcube 2 id=1" and "Lightcube 2 id=2".
- Status Bar (Bottom):** Displays battery levels: "Cozmo's battery 4 volts", "Cube1 batt 1.28V (56%)", and "Cube3 batt 1.08V (16%)".

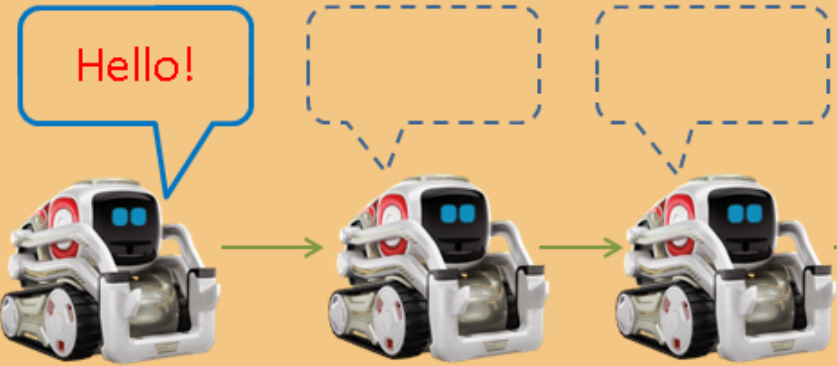
# Calypso Idioms (Design Patterns)



# Calypso Idioms (Design Patterns)

### Once Is Enough

Do something one time instead of repeatedly.

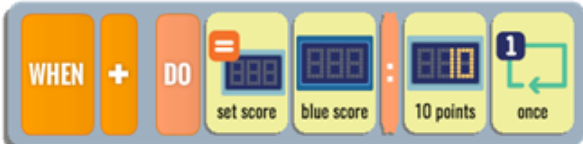


WHEN DO say "Hello!" once


WHEN **condition** DO **action** once

### Once Is Enough


Set the blue score to 10 once; don't try to change it after that:



Act playful when you first see a green cube:



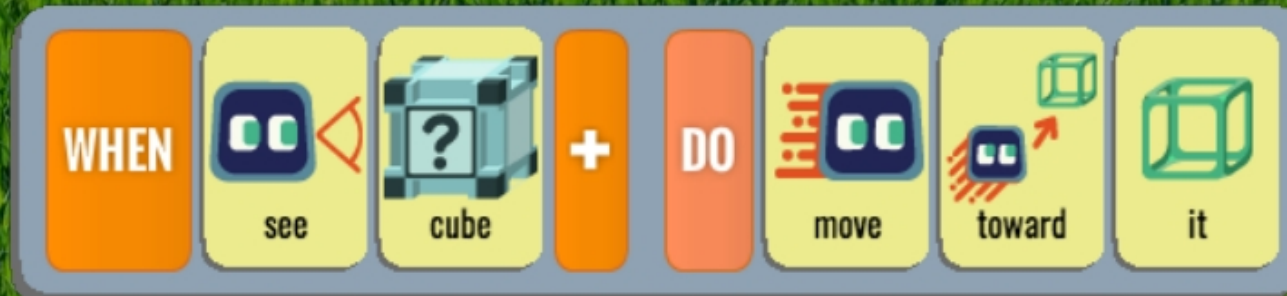
Score one point when you go from "no cube visible" to seeing a cube:



# First Law of Calypso

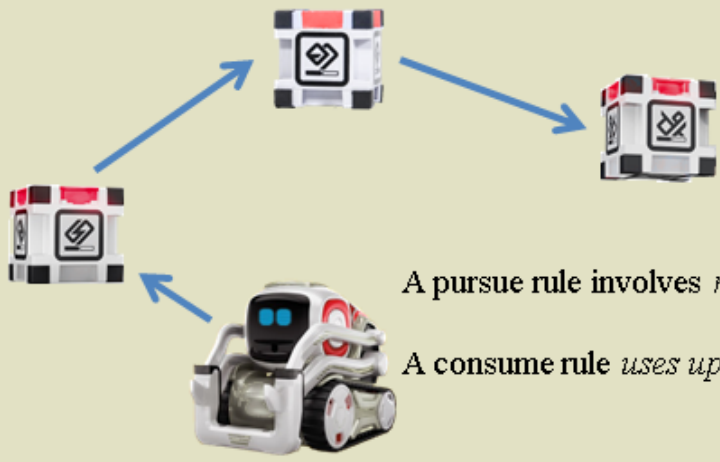
## First Law of *Calypso*

Each rule picks the closest matching object.



# Calypso Idioms (Design Patterns)

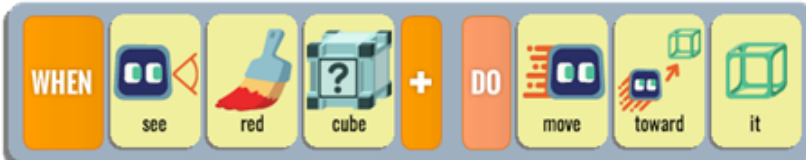
**Pursue and Consume**  
Make Cozmo extinguish all the red cubes.




A pursue rule involves *motion*.  
A consume rule *uses up* the object.

**Pursue and Consume**

**Pursue rule**



**Consume rule**



General Form:  
WHEN see *thing* DO move toward it  
WHEN bumped *thing* DO *consume* it

# Second Law of Calypso

**Second Law of *Calypso***  
Any rule that can run, will run.

The diagram illustrates the Second Law of Calypso with three rule blocks and a robot. The top block is a grey rule: WHEN bumped red cube DO grab it. The middle block is an orange rule: WHEN see red cube DO move toward it. The bottom block is a grey rule: WHEN bumped red cube DO grab it. A red dotted arrow points from the robot to the top rule block, labeled "Seeing + Moving". A red dotted arrow points from the middle rule block to the top rule block, labeled "same behavior as:". The robot is labeled "Not Bumping".

WHEN bumped red cube DO grab it

WHEN see red cube DO move toward it

WHEN bumped red cube DO grab it

Seeing + Moving

same behavior as:

Not Bumping

# Third Law of Calypso

**Third Law of *Calypso***  
When actions conflict, the earliest wins.

The diagram illustrates the Third Law of Calypso with two conflicting action sequences and a robot in a game environment. The top sequence is active, while the bottom sequence is faded.

**Top Sequence (Active):** WHEN see red cube + DO move toward it

**Bottom Sequence (Faded):** WHEN see blue cube + DO move toward it

The robot is shown in the center, with a blue dashed arrow pointing left and a red dashed arrow pointing right, indicating conflicting actions.

# Calypso Idioms (Design Patterns)

## Default Value

When the A button is pressed, glow red.  
Otherwise glow blue.



*situation* → DO **action1** **value**  
*otherwise* → DO **action1** **default-value**

## Default Value

When the A button is pressed, glow red; otherwise glow blue.



General Form:

WHEN *situation* DO **action1** **value**  
WHEN DO **action1** **default-value**

The default case must come *after* the specific case. The action must be the same in both rules; only the value is different. For different actions, use the If-Then-Else idiom.



# Fourth Law of Calypso

## Fourth Law of *Calypso*

An indented rule can run only if its parent's action succeeds.

A Kodu rule block with a yellow background. It starts with a 'WHEN' block (orange) followed by three conditions: 'bumped' (robot bumping a green block), 'green' (a green block), and 'cube' (a white cube with a question mark). This is followed by a 'DO' block (orange) and two actions: 'grab' (a hand holding a cube) and 'it' (a cube). The block ends with a plus sign (+).

A Kodu rule block with a yellow background. It starts with a 'WHEN' block (orange) followed by three conditions: 'scored' (a digital display showing '222'), 'yellow score' (a digital display showing '000'), and 'greater' (a red greater-than sign). This is followed by a plus sign (+), a 'DO' block (orange), and one action: 'play beeprobo' (a robot head with musical notes). The block ends with a plus sign (+).

A smaller version of the first Kodu rule block, showing the 'WHEN' block and the three conditions.

Score: 5



A smaller version of the second Kodu rule block, showing the 'WHEN' block and the three conditions.

Score: 0



A smaller version of the second Kodu rule block, showing the 'WHEN' block and the three conditions.

Score: 5



Actions don't fail in Kodu, but they do on real robots.


# Calypso Idioms (Design Patterns)

## Do Two Things

Make Cozmo take two actions with one WHEN condition.

WHEN *something* ... DO **this** 

*and also* →

DO **that** 

## Do Two Things

When you feel a cube being tapped, move the lift *and also* play a sound.



General Form:

WHEN *something* DO *action1*

↳ WHEN DO *action2*

Indenting the second rule makes it dependent on the success of the action of the parent rule.

# Calypso Idioms (Design Patterns)

## Count Actions

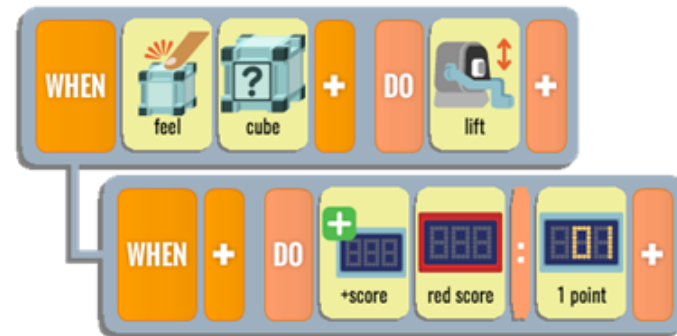
Make Cozmo keep a count of an action he takes.  
This is a special case of Do Two Things.



WHEN *something* DO **action**  
↳ *and also* → score **color** 1 point

## Count Actions

When you move the lift, add one to the red score.



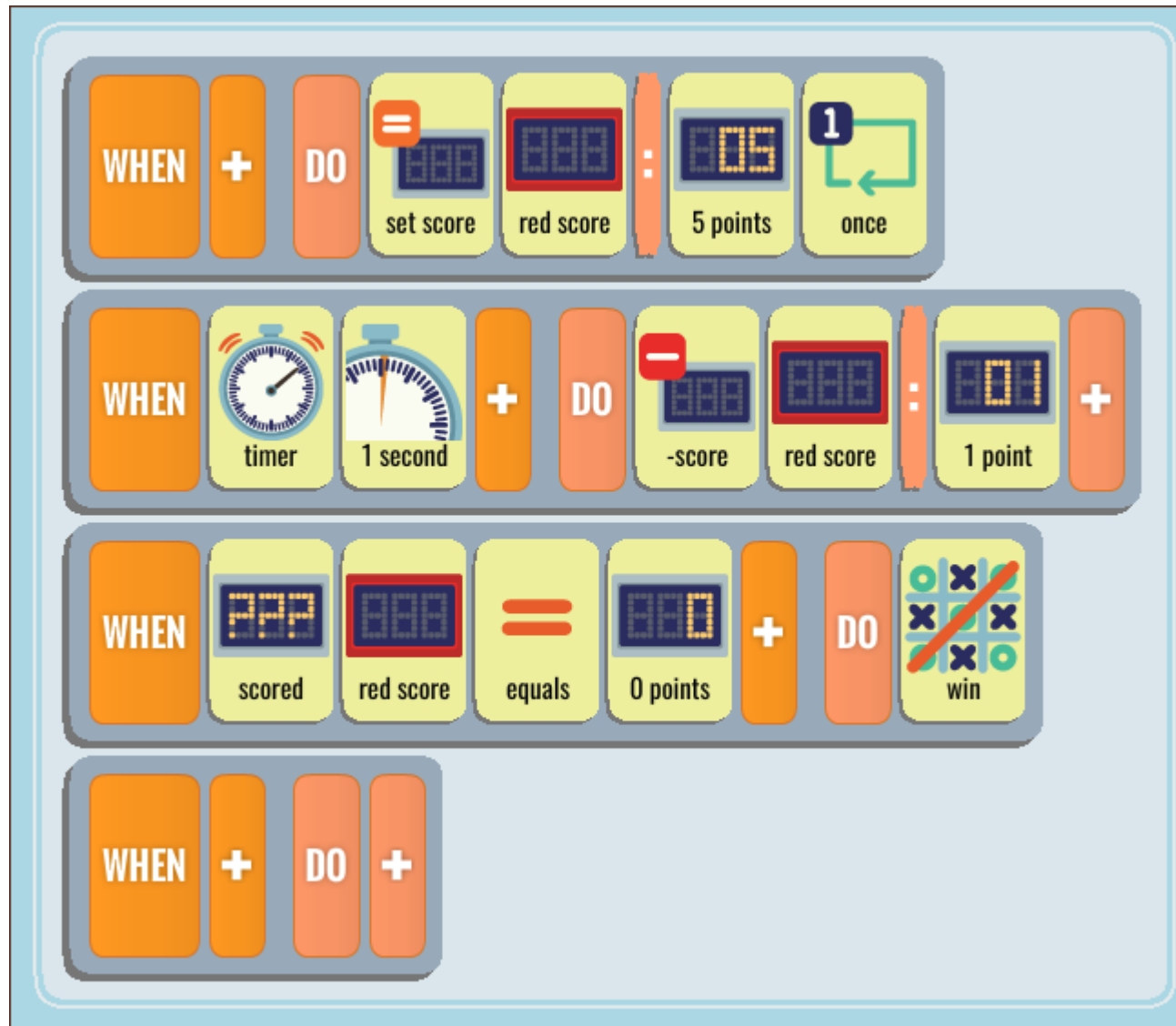
General Form:

WHEN *something* DO **action**

↳ WHEN DO score **color** 1 point

Scores are named by colors and displayed above the world map.

# Parallel WHEN Evaluation?



In Kodu this would exit immediately.

# Fifth Law of Calypso

**Fifth Law of *Calypso***  
On every cycle, earlier actions affect later rules.

WHEN bumped cube DO glow it blue

WHEN see blue DO grab it

WHEN got blue DO switch to page 2

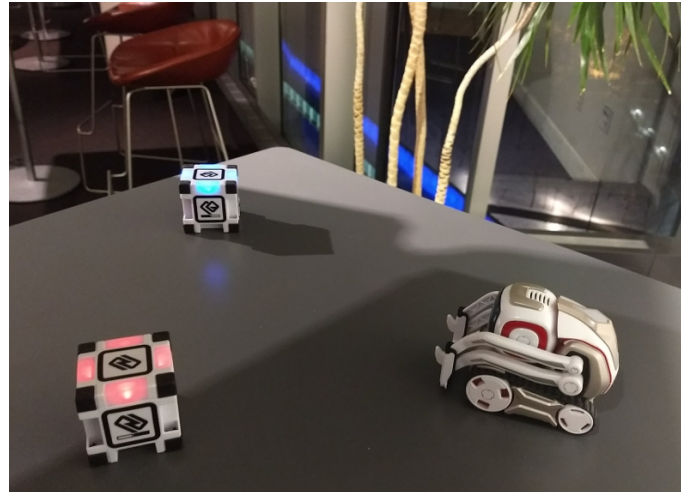
1

1

2

Differs from Kodu, where all WHEN parts are evaluated simultaneously.

# Visiting Cubes in Sequence



PAGE 1:

WHEN	see	red	cube	+	DO	move	toward	it
WHEN	+	DO	turn	wander				
WHEN	bumped	red	cube	DO	switch to	page 2		

PAGE 2:

WHEN	see	blue	cube	+	DO	move	toward	it
WHEN	+	DO	turn	wander				
WHEN	bumped	blue	cube	DO	win			

# State Machine View



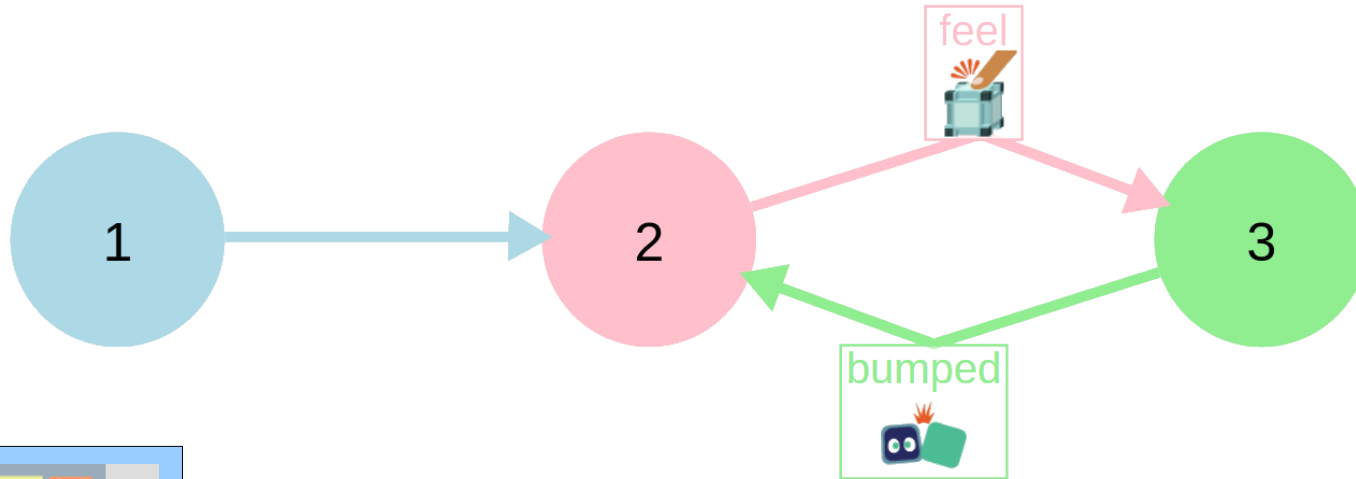
PAGE 1:

WHEN				+	DO			
WHEN	+	DO						
WHEN				DO				

PAGE 2:

WHEN				+	DO			
WHEN	+	DO						
WHEN				DO				

# Loopy State Machine



PAGE 1:

WHEN	+	DO	say	+
WHEN	+	DO	switch to	page 2

PAGE 2:

WHEN	feel	cube	+	DO	glow	it	red	+
WHEN	+	DO	switch to	page 3				

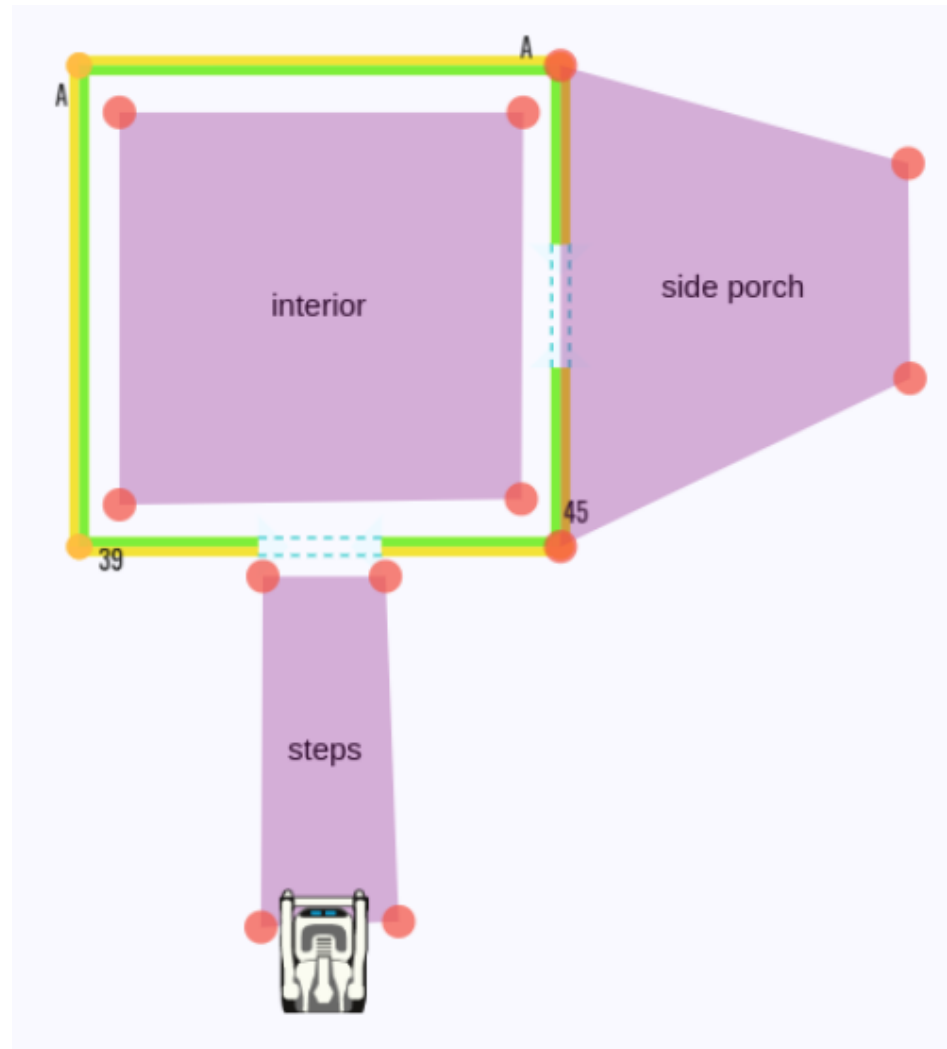
PAGE 3:

WHEN	see	red	cube	+	DO	move	toward	it
WHEN	bumped	red	cube	DO	glow	it	none	+
WHEN	+	DO	say	+				
WHEN	+	DO	switch to	page 2				

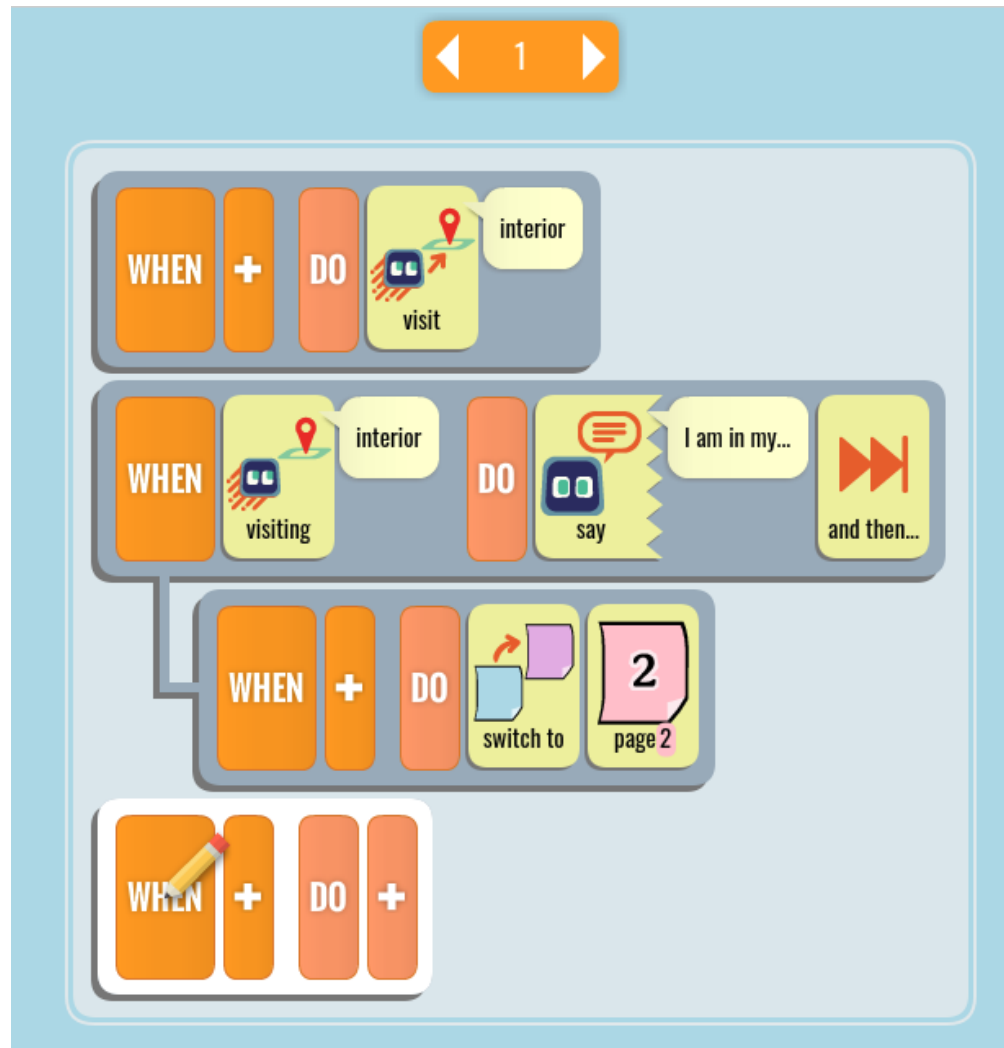
tap on another one



# Walls and Rooms



# Visit Action and Visiting Predicate



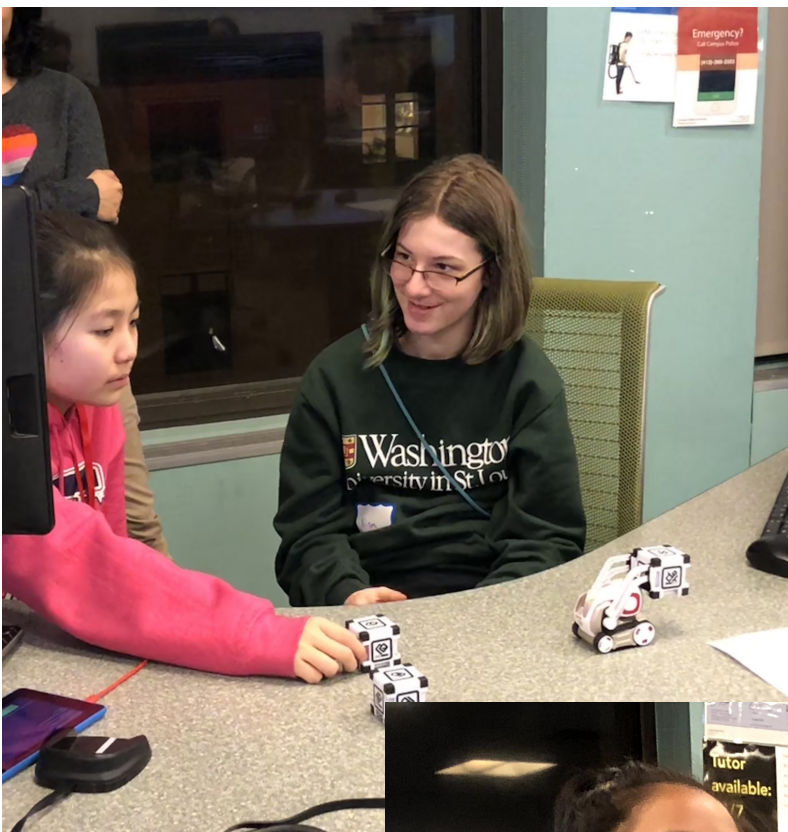
# Suspending the Rule Interpreter

- Some actions require full control of the robot and take time to complete. They must suspend the rule interpreter until they succeed or fail:
  - Grab, Roll, Drop, Express
- Some actions complete immediately and never fail:
  - Glow, score manipulation, switch to page

# Suspending (cont.)

- A third class of actions take time to complete but can run in parallel with other actions, so they don't suspend:
  - Say, Play, Look, Lift
- If we want to suspend execution until these actions complete, we add an “and then...” tile.





Testing  
With  
Real  
Kids

# Code Lab vs. Calypso (1/2)

Feature	Code Lab	Calypso for Cozmo
Free	✓	✗
Familiar to anyone who knows...	Scratch	Kodu Game Lab
Built in to the Cozmo app	✓	✗
Large display; runs on laptop or desktop	✗	✓
Camera viewer shows you what Cozmo is seeing	✗	✓
User-visible world map	✗	✓
Interpreter highlights rules that are running	✗	✓
Xbox game controller, mouse, or keyboard input	✗	✓

# Code Lab vs. Calypso (2/2)

Feature	Code Lab	Calypso for Cozmo
Voice commands	✗	✓
Simulator mode	✗	✓
Support for state machines	✗	✓
Detects failed actions	✗	✓
Free online curriculum	✗	✓

# Calypso Development Plans

- New primitives:
  - Visual search
  - Line following
  - Trainable object recognition
- New object types:
  - Chips
  - Qubes
  - Containers
- Multi-robot support