Lego SPIKE

Andrea Sterbini – sterbini@di.uniroma1.it

Lego SPIKE HUBs

2 kits with Bluetooth, BLE and USB connectivity and with an internal 6 axis gyro+accel sensor

- “Prime” large HUB has SIX I/O ports,
3 motors, colour / force / distance sensors, 4 buttons, 5x5 led matrix, speaker

- “Essential” small HUB has only TWO I/O ports and a single led/button

Ports are bi-directional Low-power at 115 Khz

MicroPython OS on 100MHz Arm Cortex-M4 CPU with 320KB RAM 1M FLASH, leaving 32M RAM available for programs/data

Sensors and devices

Distance sensor

3x3 light (Essential)

Force sensor

Colour sensor

Motors (sensor)

(Essentials)

Internal Gyro/Accel sensor

IDE apps both for pre-scholar and scholar

Programming IDE for iPad, MacOS, Win, Android

Scratch-inspired Word Blocks or Icon Blocks languages

Transforms visual programs to Python

Deploys the program on the robot

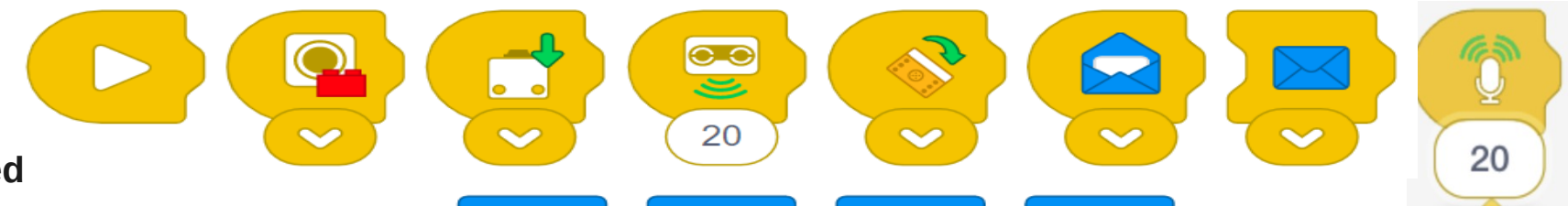
Runs the program on the robot and shows the robot display

4 combinations of IDE configuration:
- 2 kit-related (Prime vs Essentials)
- 2 visual block shapes (pre-scholars IconBlocks vs text-based WordBlocks)
- plus a Python mode

Icon Blocks
pre-scholar language ScratchJr inspired

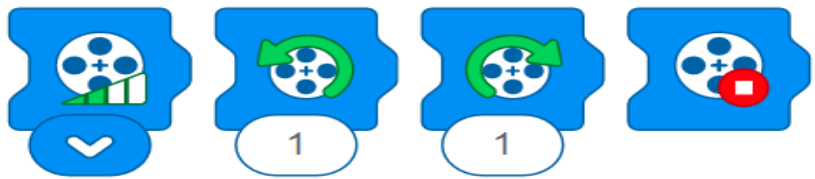
EVENTS

- sensor-based
- message-based



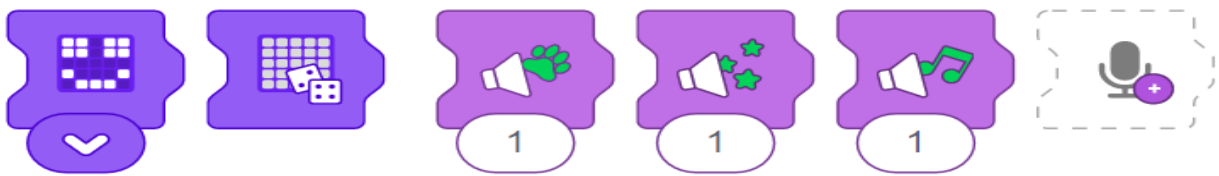
SINGLE MOTOR MOTION

- set speed / turn left / turn right / stop



DISPLAY and SOUNDS

- show picture / random pixels
- 8 animal / 8 effects / 8 music / record a new sound



CONTROL:

- wait / repeat / forever / end

Extensions

DISPLAY (shown in the APP):

show text/show image/show fullscreen

BAR-CHARTS (shown in the APP):

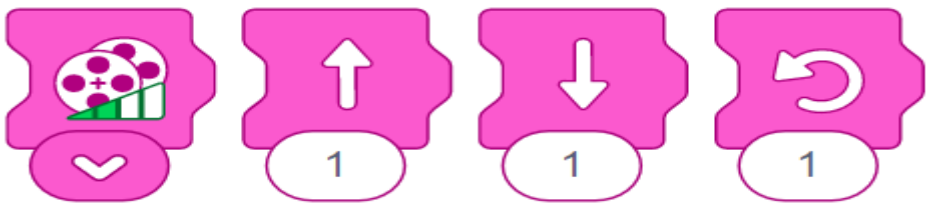
add a value to the bar-chart (6 colors)

clear bar-chart

show fullscreen

DIFFERENTIAL MOVEMENT:

set speed / forward / back / turn



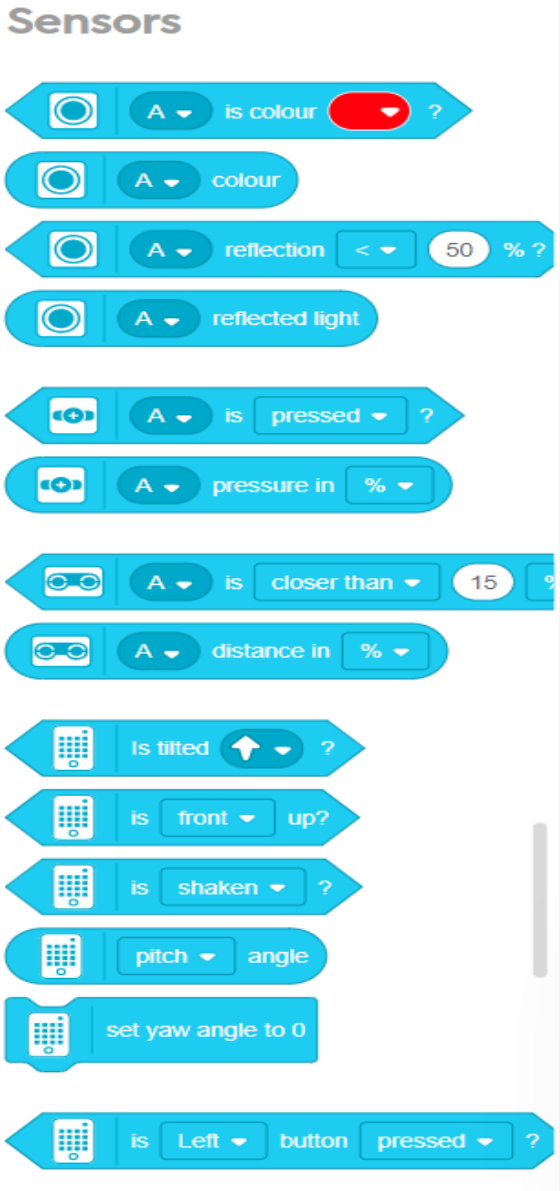
Icon Block language:
ScratchJr – Agents + Robotics

New Functions?	NO	cannot define new blocks
Recursion?	NO	
Return?	NO	
Loops?	PARTIAL	only counted loop or forever
Conditionals?	NO	
Messages?	YES	send 6 colors or a random color
Events?	YES	sensor or message-based
Concurrency?	YES	multiple event definitions
Agents?	NO!!!	

Word Block language: Scratch + Robotics

Single motor motion

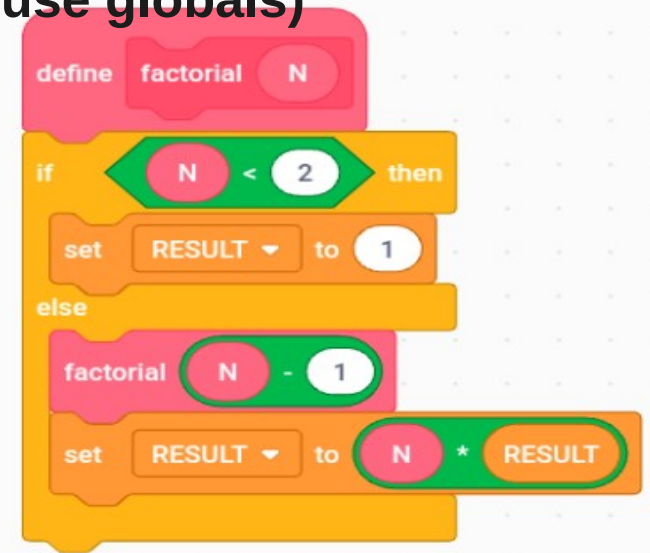
Differential drive motion



Methods in Computer S

Word Block language:
Scratch - Agents + Robotic blocks

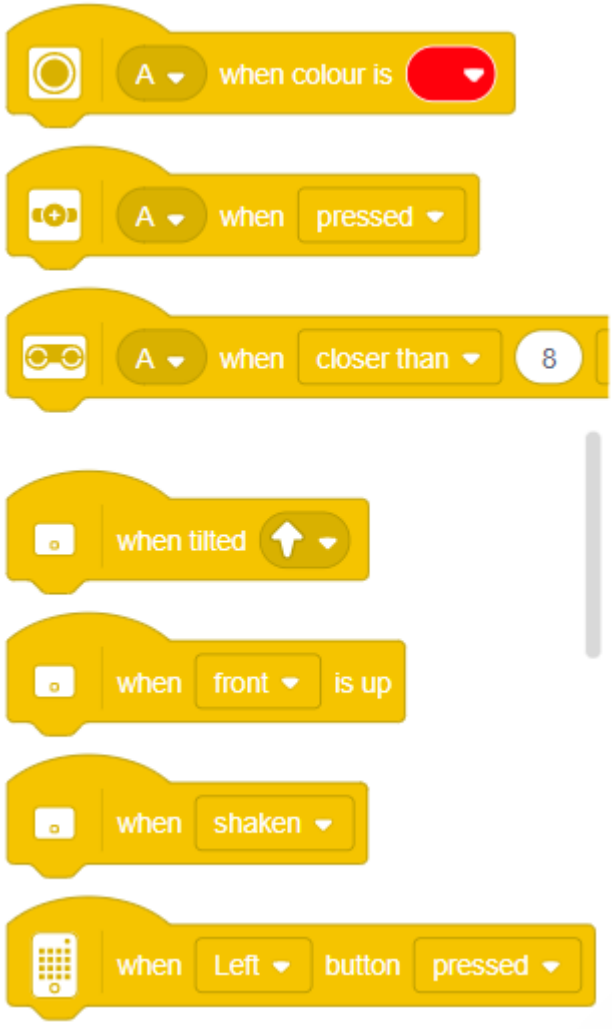
New Functions?	ONLY PROCEDURES
Recursion?	YES
Return?	NO (but you can use globals)
Local variables?	NO! (only globals)
Messages?	YES (named, no args but you can use globals)
Concurrency?	YES (multiple event defs)
Agents?	NO!!!
Clones?	NO!!!
Events?	YES



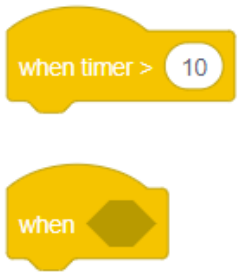
Events and messages

Sensor based events (when sensor CHANGES):

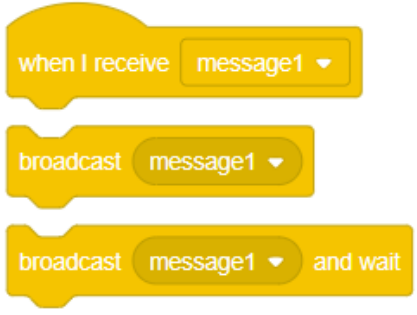
- When colour is
- When pressure is
- When distance is closer than
- When tilted / positioned / shaken
- When button is pressed / released



Timer and polling events



Message-based events and synchronization



(Micro)Python mode in SPIKE App

Single thread of execution (?)

Each sensor, Hub and App is represented by a class

Events are simulated by waits (wait__until__XXX methods)

Simple types: int, float, str, list

```
from spike import PrimeHub, LightMatrix, Button, StatusLight,
from spike import ForceSensor, MotionSensor, Speaker, ColorSensor
from spike import App, DistanceSensor, Motor, MotorPair
from spike.control import wait_for_seconds, wait_until, Timer
from math import *

hub = PrimeHub()

hub.light_matrix.show_image('HAPPY')
```

CALLBACKS → how to manage async events
(see Lego HUB docs or MicroPython docs)

hub.button.Button.callback()	**when the button is pressed**
hub.bluetooth.lwp_monitor()	**when a BT connection is ready**
hub.display.callback()	**when a display operation is completed/interrupted**
hub.motion.orientation()	**when HUB changes orientation**
hub.motion.gesture()	**when HUB is shaken/tapped/falls**
hub.sound.callback()	**when a sound is completed/interrupted**
hub.supervision.callback()	**when an over-current event happens**
hub.BT_VCP.callback()	**when BT connection changes**
hub.Port.callback()	**when a device is plugged/unplugged**
hub.Motor.callback()	**when a command is completed/interrupted/stalled**
hub.MotorPair.callback()	**when a command is completed/interrupted/stalled**

More sensors!!!

LPF2 connection specs allow using other sensors, e.g. from MindSensors.com

- Temperature (contact or infrared)

- Infrared Motor controller

- RFID tag reader

- dual zone infrared distance

- plus an adapter for old EV3 sensors (compass, vision, camera, etc…)

More ways to program Lego !!!

Other Python-based firmware is available

See https://pybricks.com compatible with:

- Lego Technic / Inventor / Boost / Spike

- Lego Mindstorms EV3

- With block coding also!