

Snap! (by Berkeley)



Andrea Sterbini – sterbini@di.uniroma1.it

Snap! (by Berkeley univ.)

“Scratch for the Computer Scientist”

Object orientation

Many extensions/libraries

Support for code documentation

Support for debugging

Concurrency

Coroutines

Functional programming (APL)

...

Scalar programming (APL)

Music

Relative motion of sprites

HTML5 web app

Easy local install (just unzip)

See the [Snap! manual](#) for more info

?

PACE

+

☰

✉

🌐

λ

🌟

🔄

Snap! Build Your Own Blocks 4.0 X

+

←

→

↺

🏠

🖨

🔒

🌐

https://snap.berkeley.edu/snapsource/snap.html#present:Us

⋮

🛡

★

TV

CS

📖

📄

🔴

🛡

🟢

🎮

0

☰

λ

Snap!

📁

☁

⚙

swimmer

❗

❗

🖼

🔄

🚩

▶

🔴

Motion

Looks

Sound

Pen

Control

Sensing

Operators

Variables

when clicked

when space key pressed

when I am clicked

when

when I receive

broadcast

broadcast and wait

message

warp

wait 1 secs

wait until

forever

repeat 10

repeat until

right arm

draggable

Scripts

Costumes

Sounds

when I receive initialize

point in direction direction of rump - 20

when I receive pull

repeat 5

turn 15 degrees

repeat 10

turn 7 degrees

Stage

right arm

left bicep

left arm

right bicep

right hand

left hand

left foot

right foot

left thigh

rump

right thigh

left leg

right leg

water

Snap! improves many Scratch language constructs

Scratch

- NO complex data
- NO functions (only procedures)
- NO local variables
- NO references to clones
- NO call methods
- NO libraries
- NO print text on canvas

Snap!

- Objects, Lists, Lists of lists, Lists of Objects
- Functions (return)
- Local variables (easy recursion)
- References to clones
- Call methods
- Global blocks (library of functions)
- Inheritance of clone properties
- Anonymous “Lambda” functions

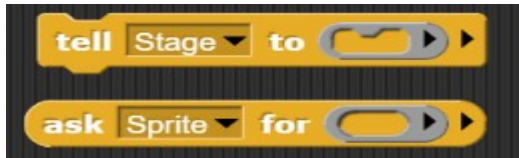


Other functions

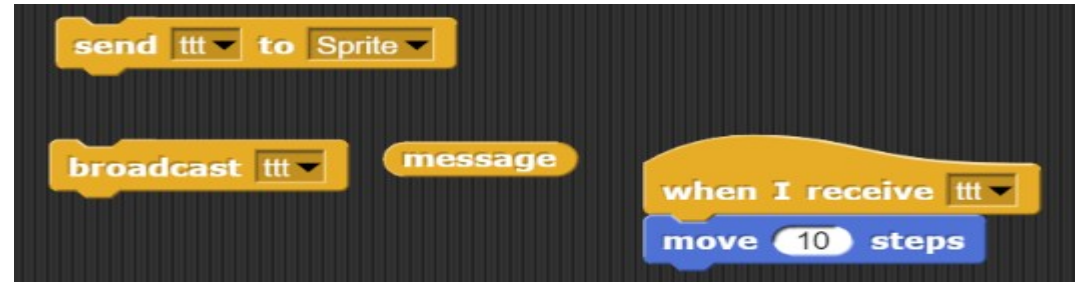
Can create a “costume” by drawing on it



Objects can ask each other to do / report something



Can use individual messages
Or broadcast messages to all



Can define generic events (SLOW!)
(e.g. variable observer)



Relative motion of Sprites/Agents wrt other agents

It makes easy building:

collective motion of many clones (fireworks, snow, birds, ...)

coordinated motion of an agent with many parts
(e.g. man walking, multipropeller drone)



Example: Swimmer

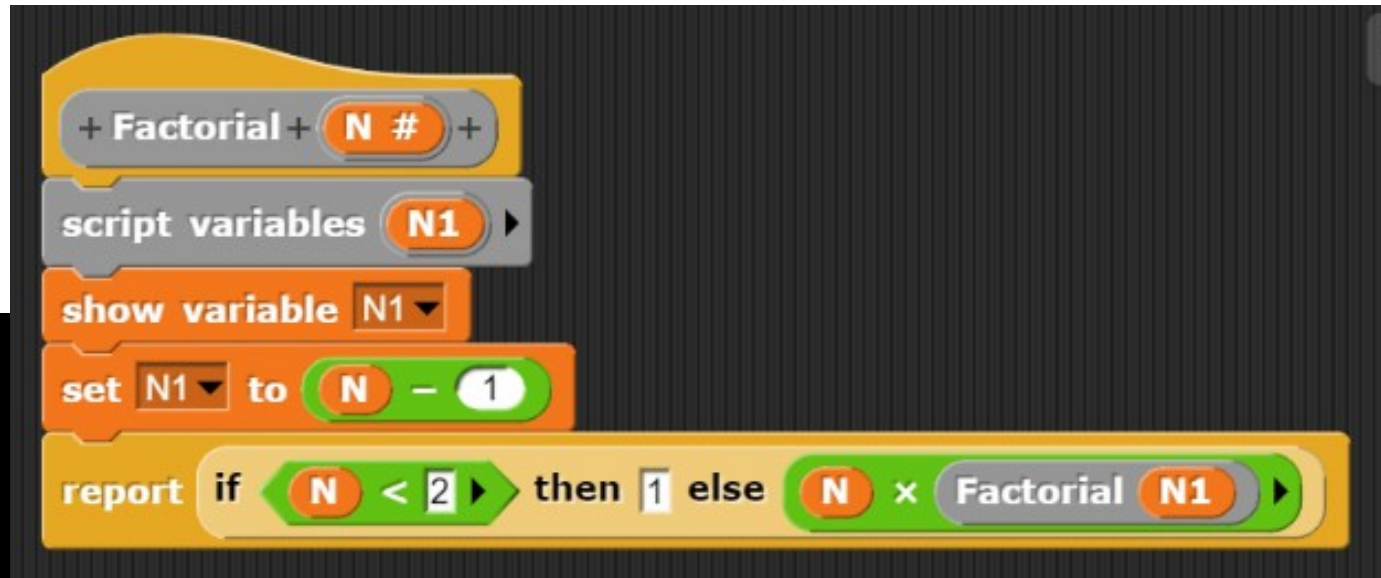
Main motion: body trunk and head (straight motion **bumping** onto the walls)

Attached to body: thighs and biceps (that are **rotating** w.r.t. the body)

Attached to thighs and biceps: arms and legs (that are just kept in the body direction)

Attached to arms and legs: hands and feet (that **rotate** w.r.t. the arm and leg)

Easy recursion (with local variables)



Many Standard Libraries/Extensions

Loops and compositions

List operations

Streams (lazy lists)

Multiple args operators

Web access

Words manipulation

Switch/case

RGB/HSV colors

Handle big lists

Frequency distribution analysis

Try/catch

Multiline input

GUI settings

Bignum, rational, complex

Text to speech

Animations

Image manipulation

Audio generation

Json

Parallelization

and more ...

Other extensions

SOFTWARE:

3D extrusion (BeetleBlocks)

Embroidery (TurtleStitch)

Cellular automata (Cellular)

Graphs (Edgy)

Network programming (NetsBlox)

Music (Tones-Snap)

Minecraft (ProgKids in russian)

HARDWARE:

Orbotix Sphero

Lego NXT (but not EV3 yet)

Wiimote

Arduino

Raspberry Pi

Speech synthesis

LEAP

Finch, Hummingbird

Many programming styles!

Functional

Lists, filters, map, coroutines, continuations, generators

Procedural

Concurrent

Concurrent execution

Message events (with data)

Object-oriented/Agent based

Agent properties, Agent methods

Clones: references to created clones, inherited properties

Snap! for C.T. applied to other Subjects

Pro:

Rich language with all CS constructs and more!

Rich data structures (including objects, Json and CSV tables)

Easy animation of multi-agent groups with relative motion

Many extension libraries

Con?:

Sophisticated constructs for more experienced programmers

Good for older students and more complex projects