

Snap! (by Berkeley)



Andrea Sterbini – sterbini@di.uniroma1.it

Snap! (by Berkeley)

Evolution of Scratch

“Scratch for the Computer Scientist”

Object orientation

Many extensions/libraries

Support for code documentation

Support for debugging

Concurrency

Coroutines

...

Music

Relative motion of sprites

The image shows the Snap! programming environment for a project named "swimmer".

- Left Sidebar:** Contains category buttons for Motion, Looks, Sound, Pen, Control, Sensing, Operators, and Variables. The "Control" category is currently selected.
- Costume Area:** Shows a costume named "right arm" with a "draggable" checkbox.
- Script Area:** Contains the following code blocks:
 - "when I receive initialize" block.
 - "point in direction" block with "direction of rump" and "- 20".
 - "when I receive pull" block.
 - "repeat 5" loop containing a "turn 15 degrees" block.
 - "repeat 10" loop containing a "turn 7 degrees" block.
- Stage Area:** Shows a swimmer character in a blue water background. Below the stage are control buttons (play, stop, camera) and a costume palette with items like "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", and "water".

Snap! improves many language constructs

Scratch

NO complex data

NO functions (only procedures)

NO local variables

NO references to clones

NO call methods

NO libraries

Snap!

Objects, Lists, Lists of Objects

Functions



Local variables (easy recursion)

References to clones



Call methods



Global blocks (library of functions)

Inheritance of clone properties

“Lambda” functions



Relative motion of Sprites/Agents

Relatively easy to build

collective motion of many clones (fireworks, snow, birds, ...)
coordinated motion of agent with many parts (man walking)

Example: Swimmer

Main motion: body trunk and head (bumping to the walls)

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

Attached to thighs and biceps: arms and legs (staying in the same direction as the body)

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



Easy recursion



Standard Libraries/Extensions

Loops and compositions

List operations

Generators (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

Other extensions

SOFTWARE:

Cellular automata (Cellular)

Graphs (Edgy)

NLP (NLTK wrapper)

HARDWARE:

Orbotix Sphero

Lego NXT (but not EV3 yet)

Wiimote

Arduino

Raspberry Pi

Speech synthesis

LEAP

Finch, Hummingbird

Styles of programming

Functional

Lists, filters, map, coroutines

Procedural

Concurrent

Concurrent execution

Message events

Object-oriented/Agent based

Agent properties, Agent methods

Clones: references to created clones, inherited properties