phratch
visual programing for grow up

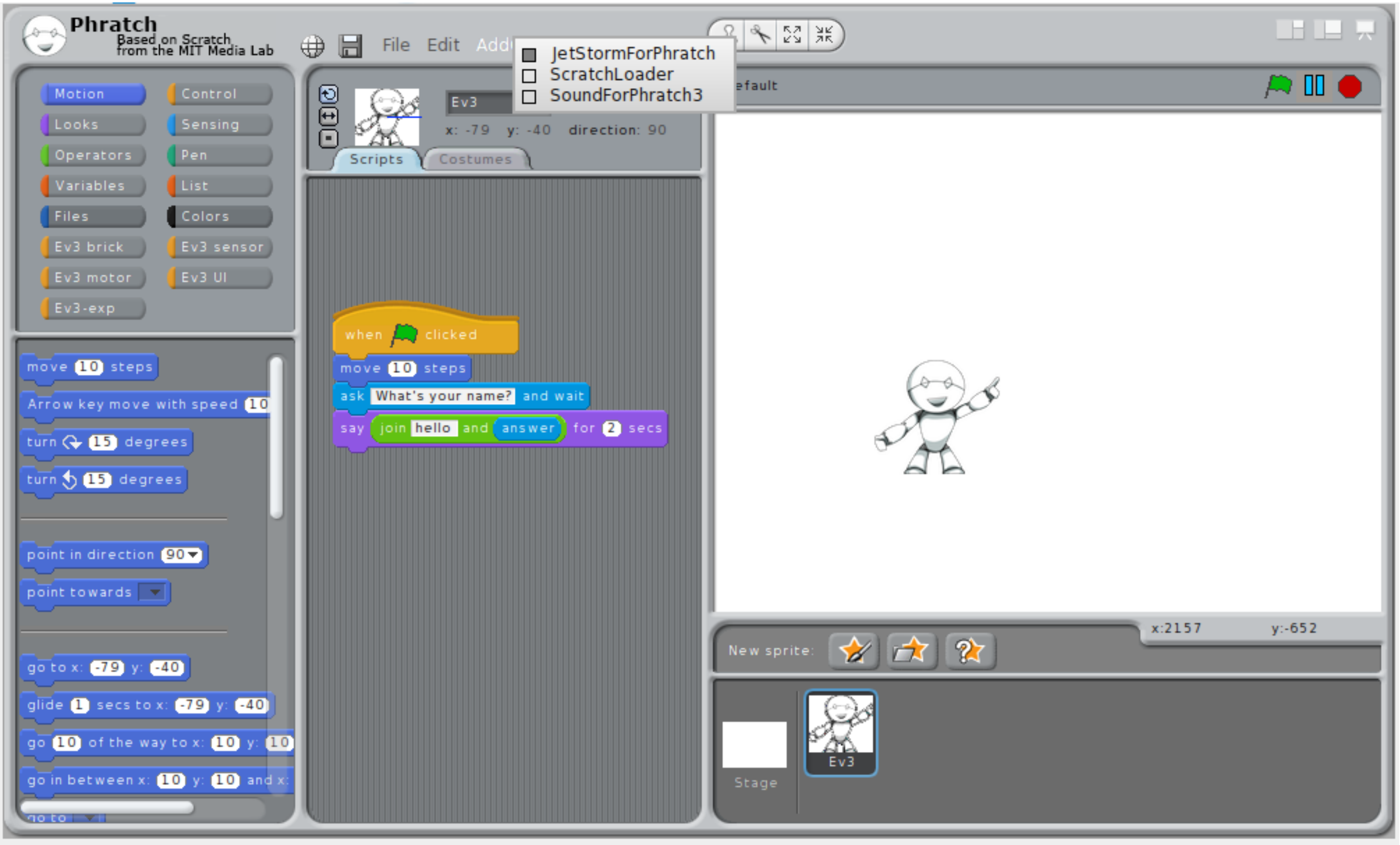
Jannik LAVAL
jannik@phratch.com

http://www.phratch.com

MINES
Douai

ESUG 2014
Cambridge

You know phratch !



■ JetStormForPhratch
□ ScratchLoader
□ SoundForPhratch3

```
when 🏁 clicked
move 10 steps
ask What's your name? and wait
say join hello and answer for 2 secs
```

You know phratch !

on top of Pharo

https://code.google.com/p/scat/

You know phratch !

SCRATCH

on top of Pharo

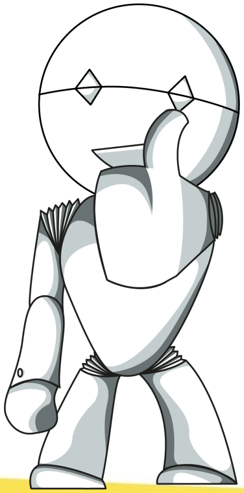
https://code.google.com/p/scat/

You know phratch !

SCRATCH

on top of Pharo

https://code.google.com/p/scat/



ESUG 2013 Annecy

3rd

Innovation Technology Awards

phratch.com sponsored by

EUROPEAN SMALLTALK USER GROUP

esug

www.esug.org

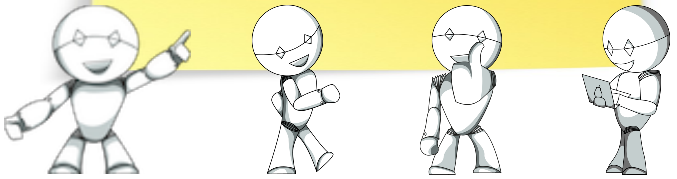
What is new since ESUG 2013?

What is new since ESUG 2013?

What is new since ESUG 2013?

phratch.com

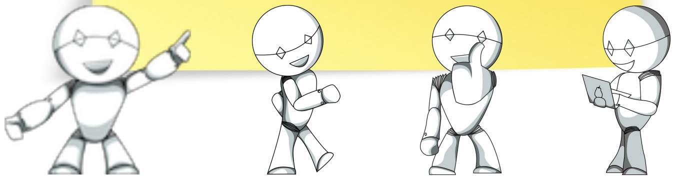
smart icons



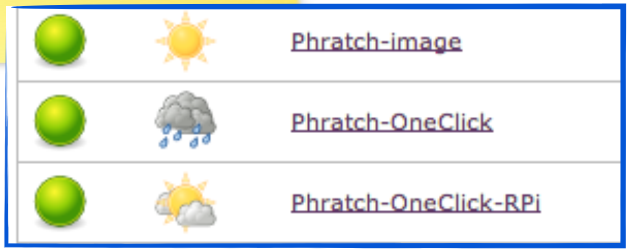
What is new since ESUG 2013?

phratch.com

smart icons



Jenkins

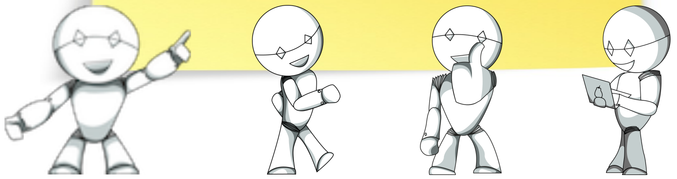


●	☀	Phratch-image
●	🌧	Phratch-OneClick
●	⛅	Phratch-OneClick-RPi

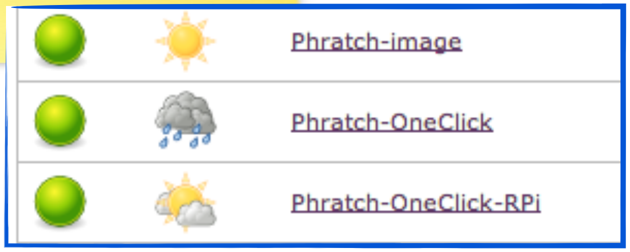
What is new since ESUG 2013?

phratch.com

smart icons

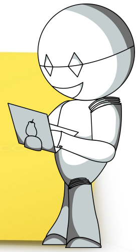


Jenkins



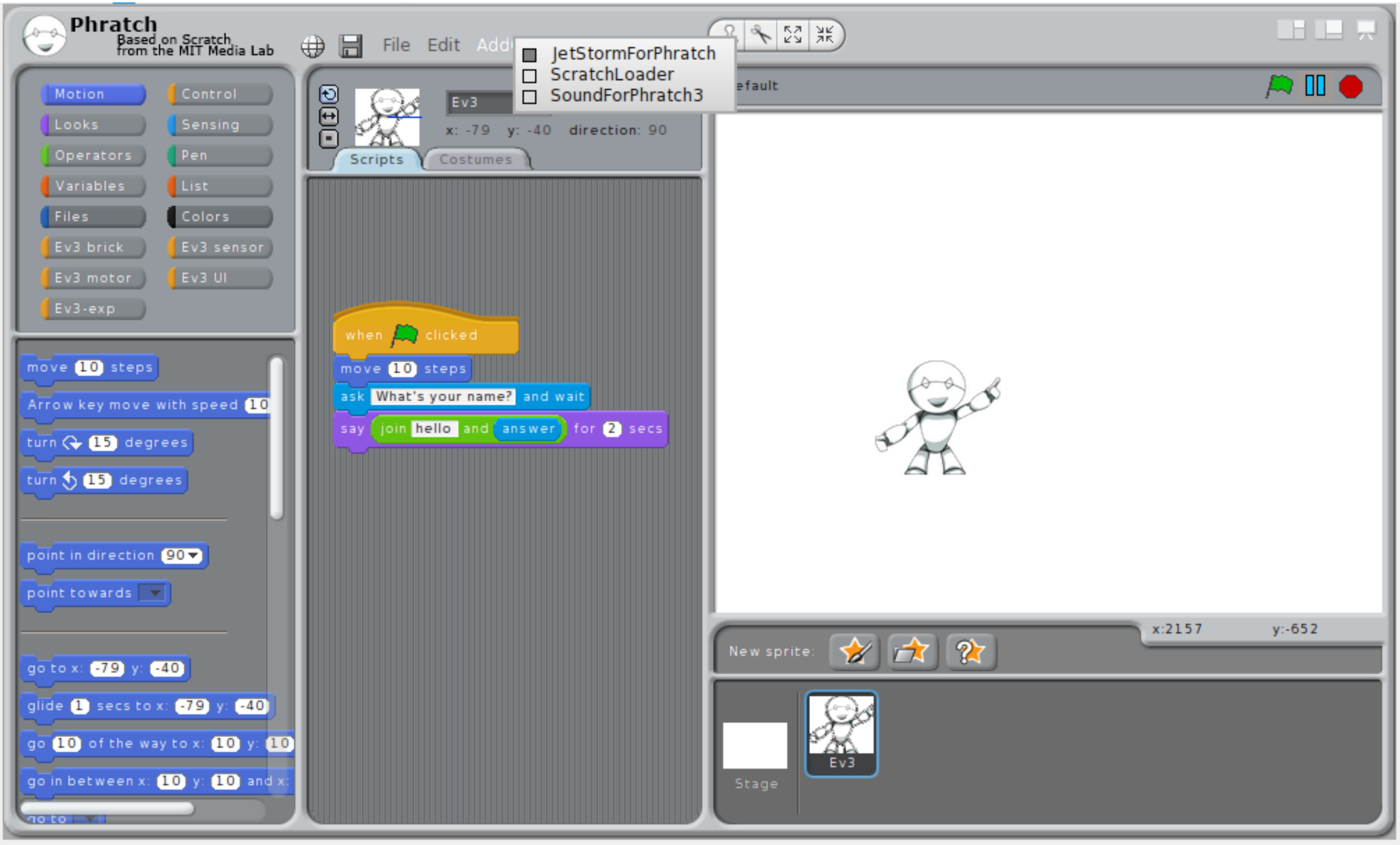
●	☀	Phratch-image
●	🌧	Phratch-OneClick
●	⛅	Phratch-OneClick-RPi

Jetstorm - robotics



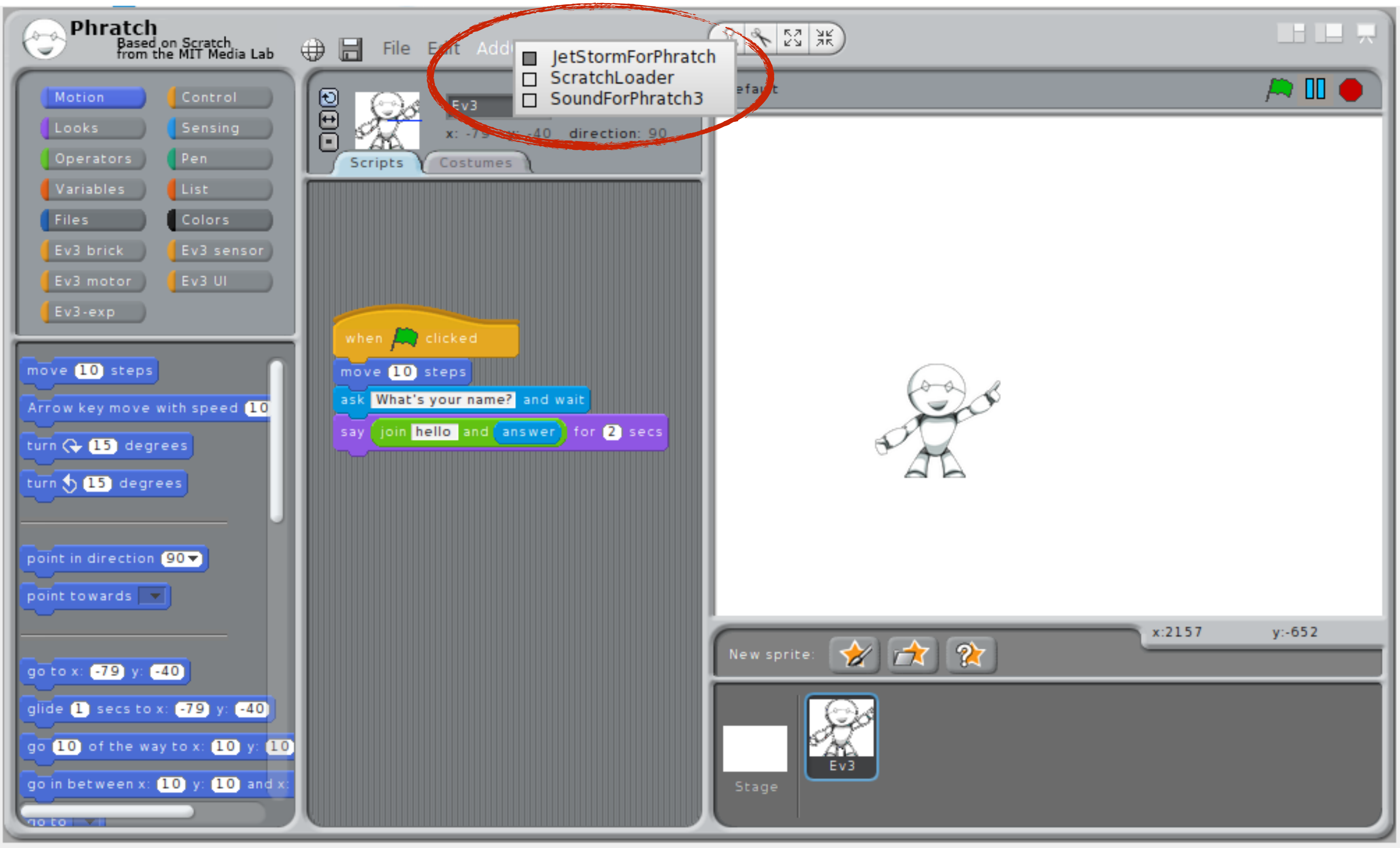
New features: Add-ons

New features: Add-ons

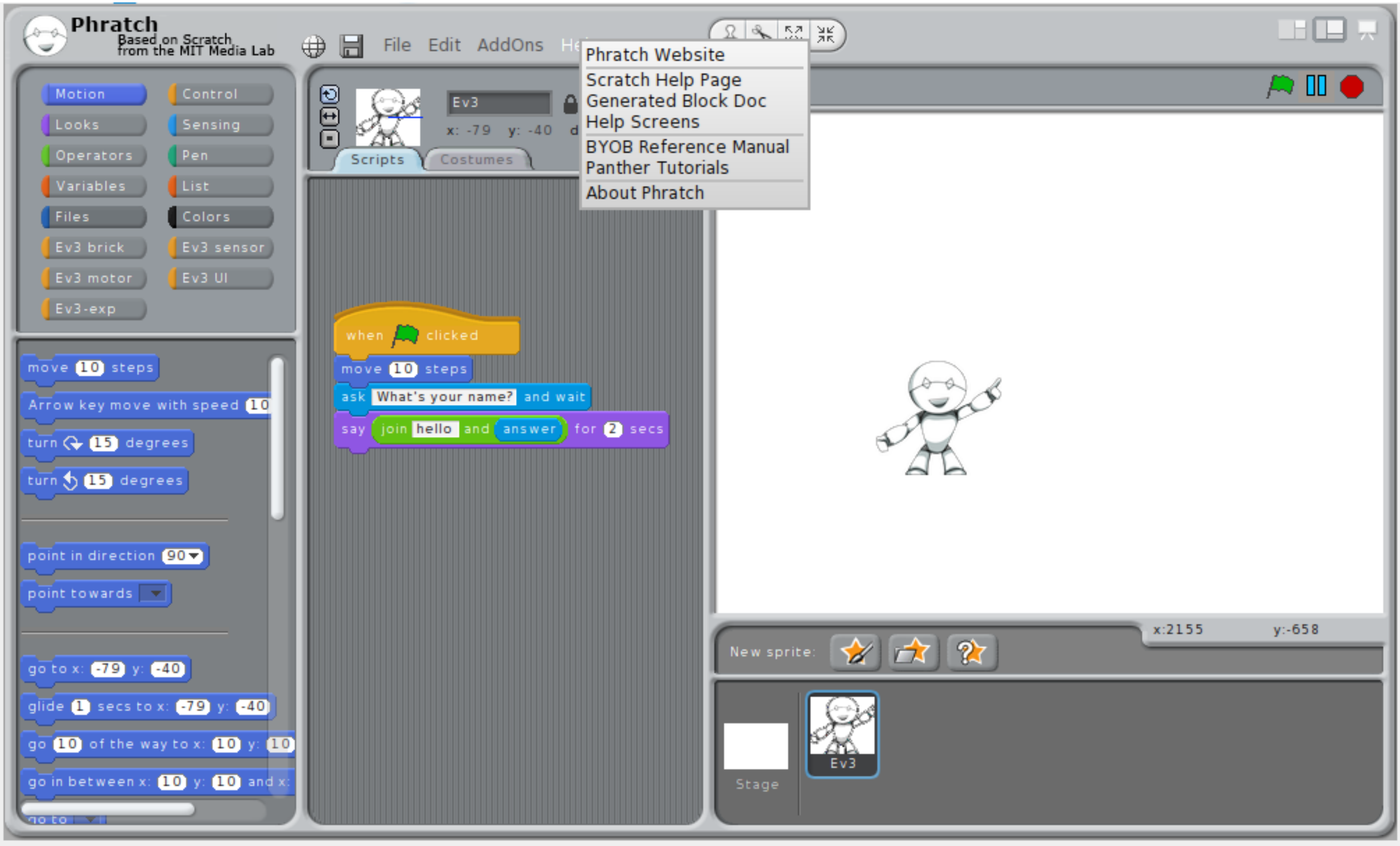


```
when clicked
move 10 steps
ask What's your name? and wait
say join hello and answer for 2 secs
```

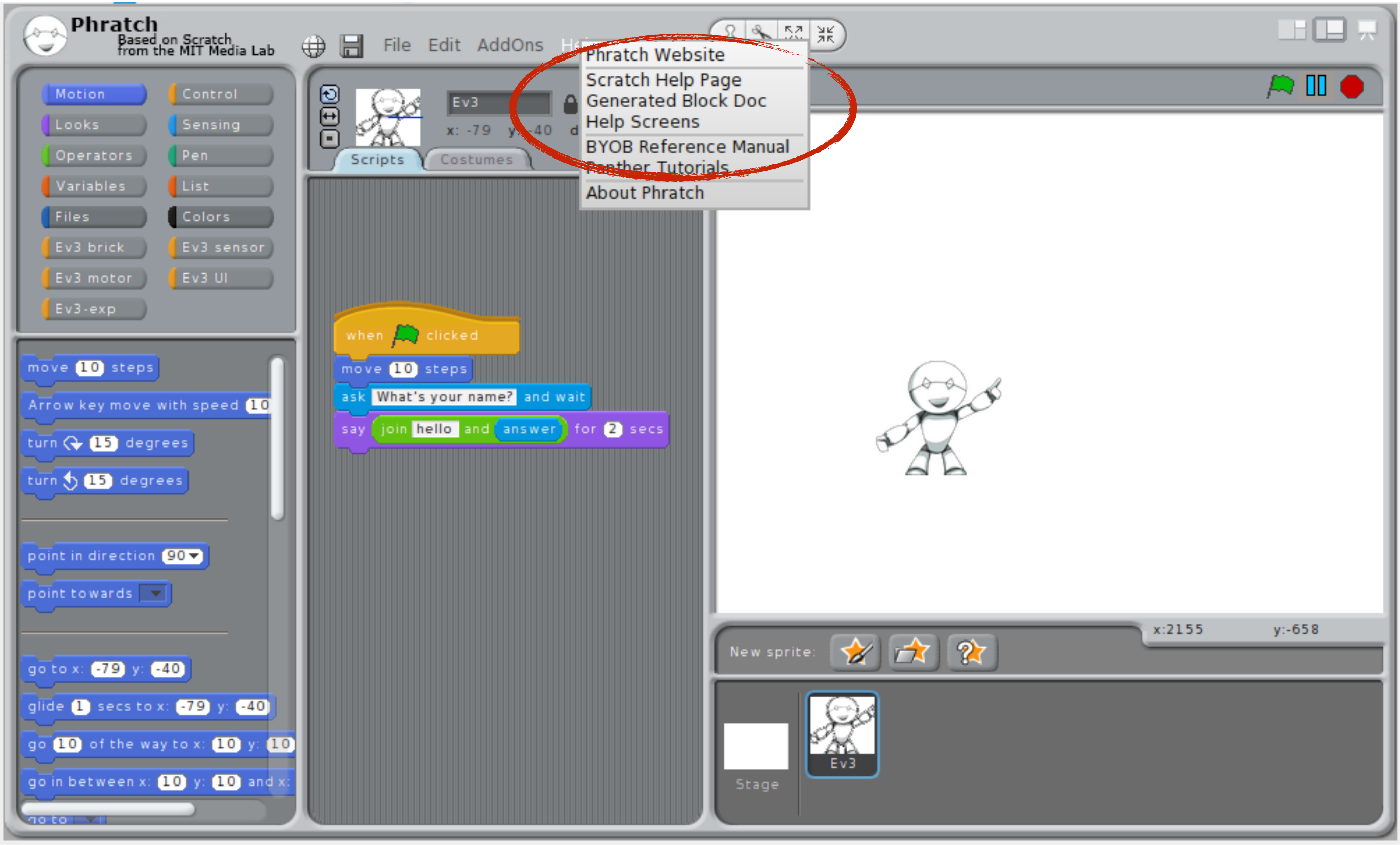
New features: Add-ons



New features: documentation

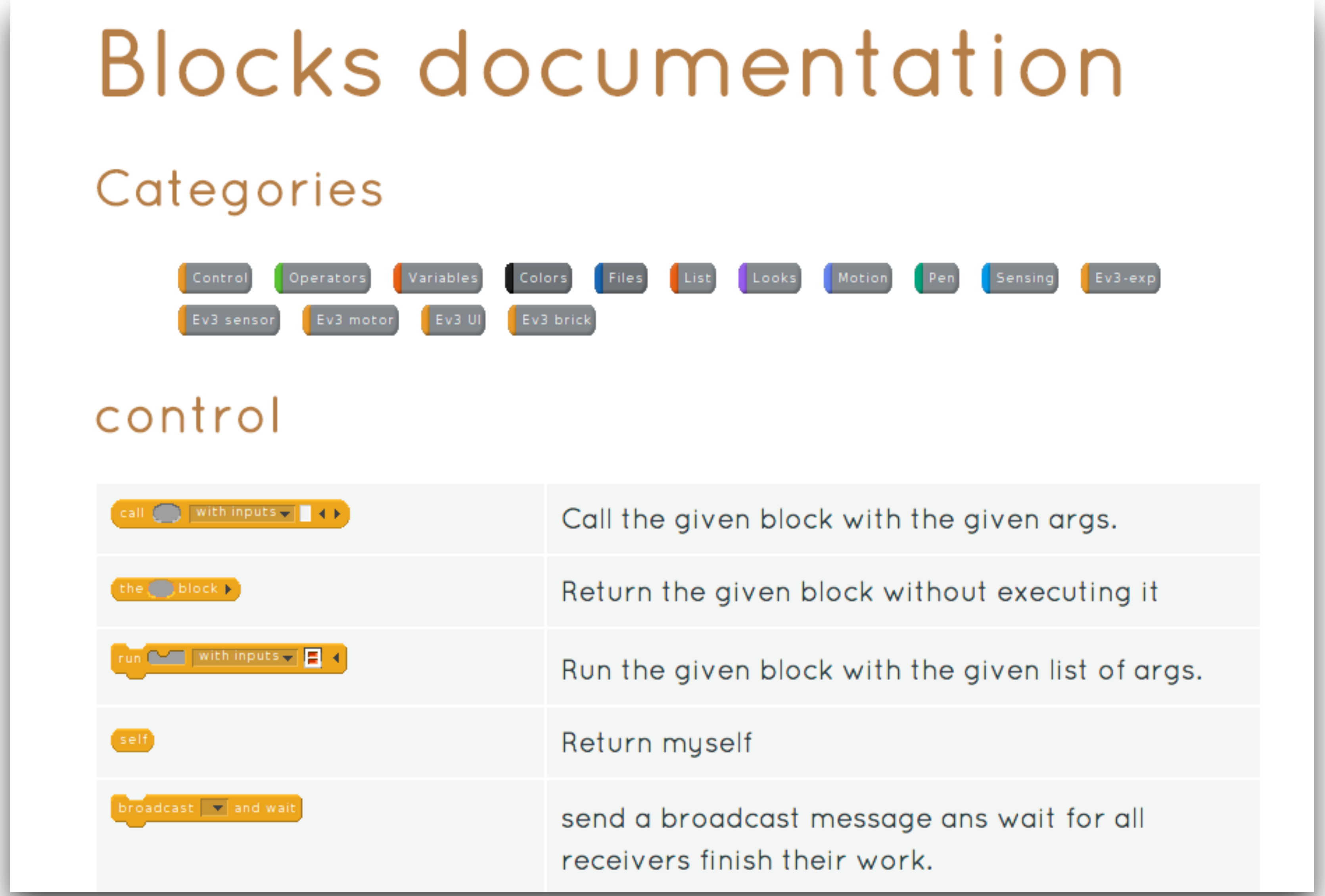


New features: documentation



```
when [flag] clicked
move 10 steps
ask What's your name? and wait
say join hello and answer for 2 secs
```

New features: documentation



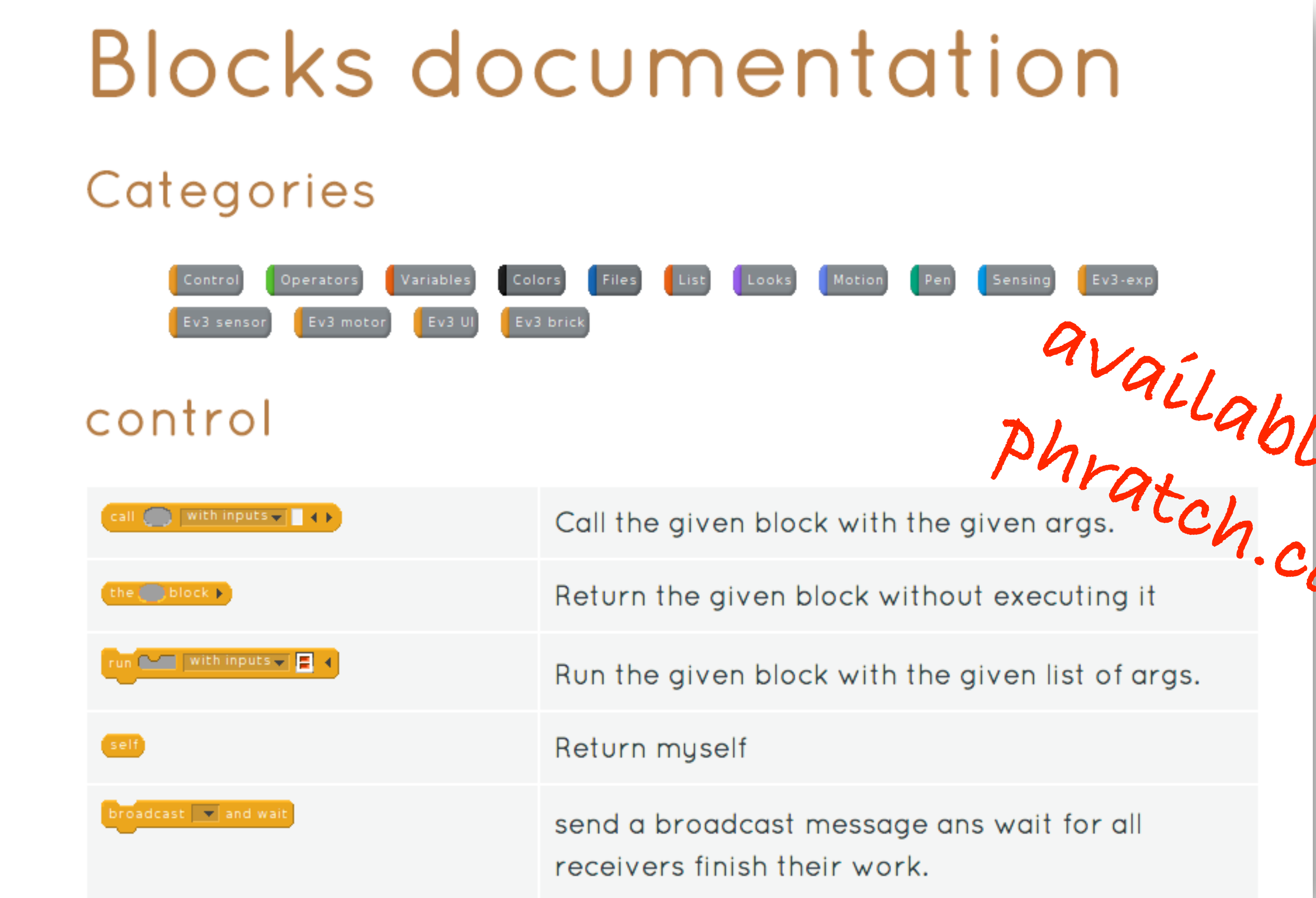
Blocks documentation

Categories

control

call with inputs	Call the given block with the given args.
the block	Return the given block without executing it
run with inputs	Run the given block with the given list of args.
self	Return myself
broadcast and wait	send a broadcast message ans wait for all receivers finish their work.

New features: documentation



Blocks documentation

Categories

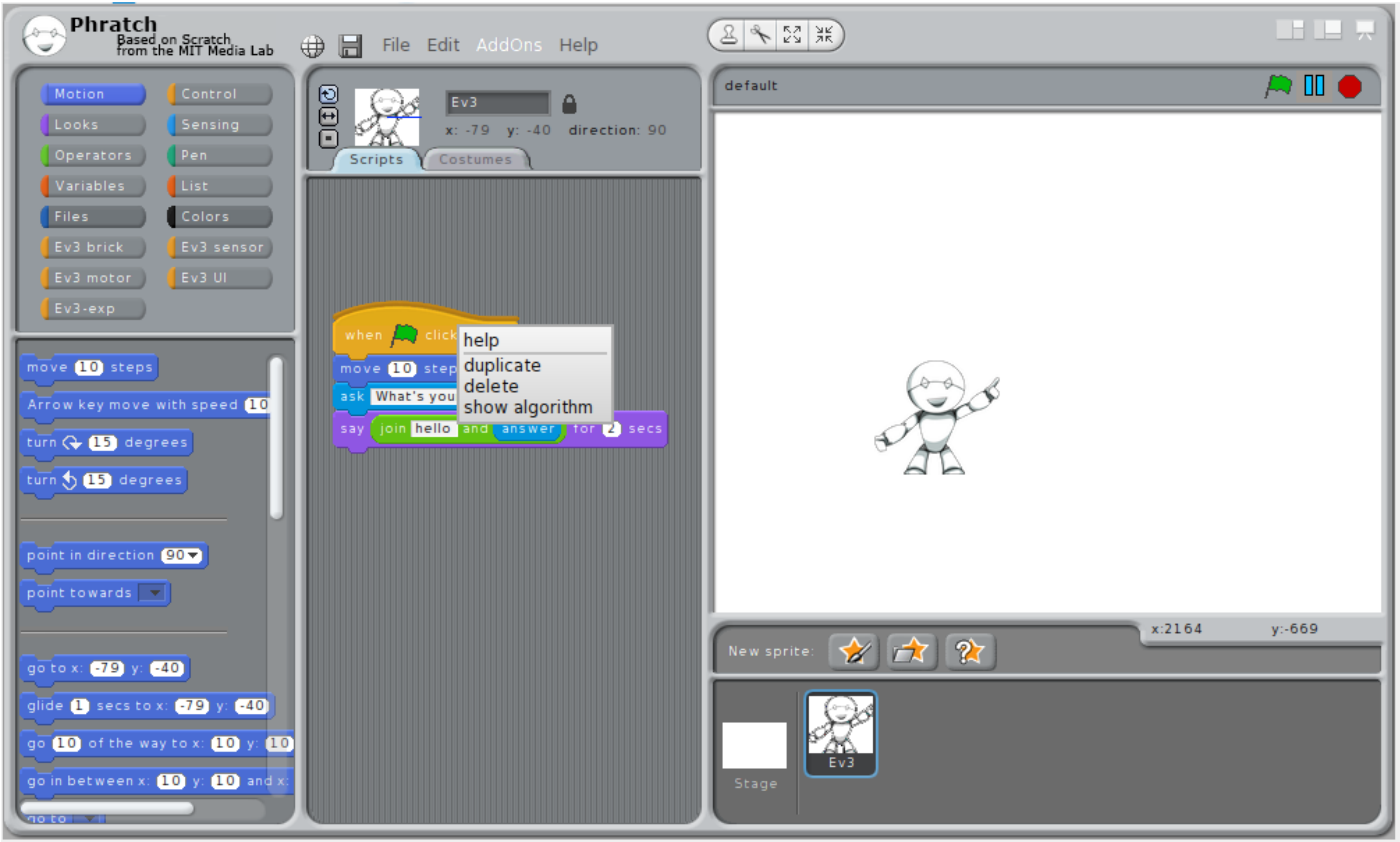
Control Operators Variables Colors Files List Looks Motion Pen Sensing Ev3-exp Ev3 sensor Ev3 motor Ev3 UI Ev3 brick

control

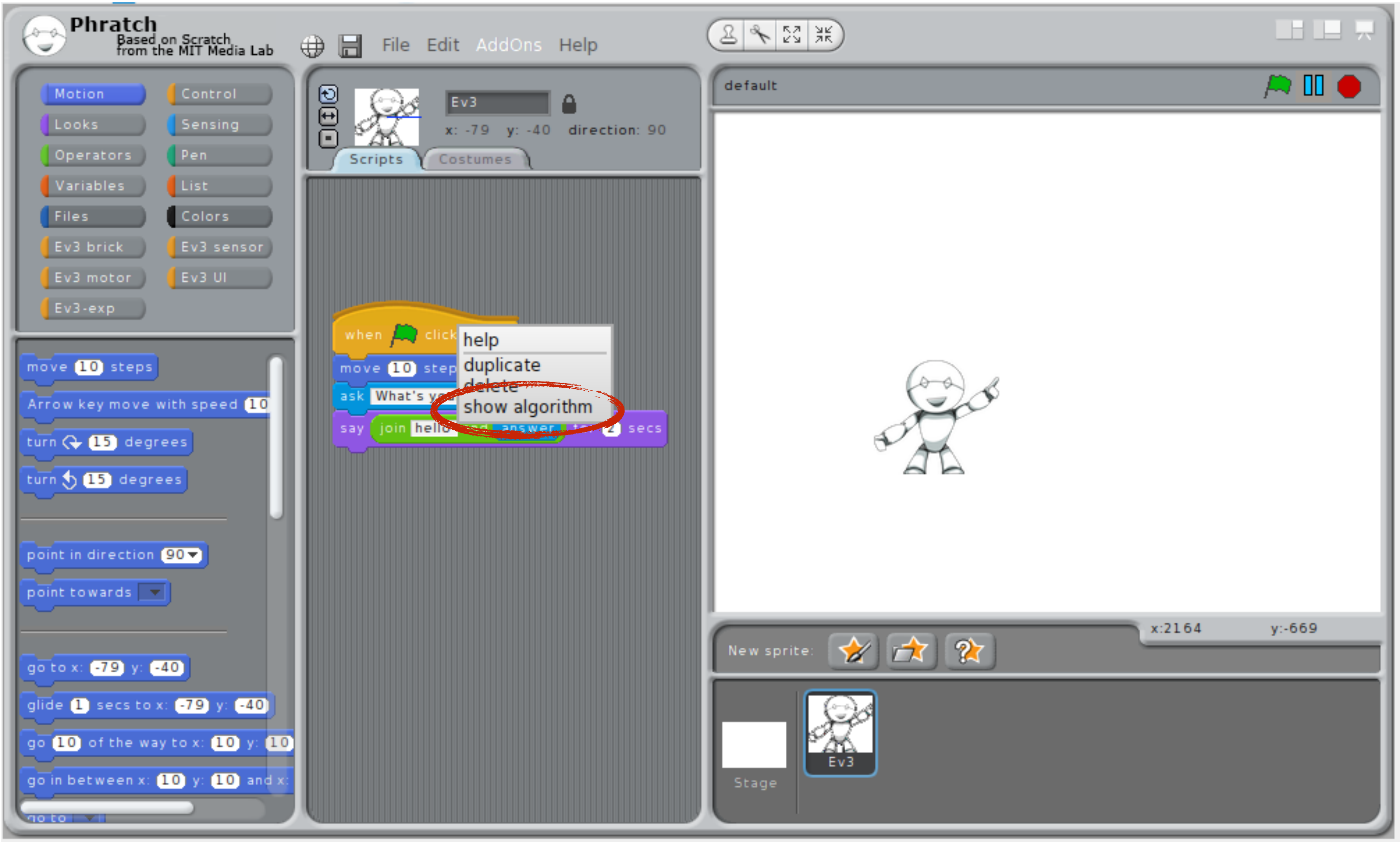
call ( ) with inputs ▾	Call the given block with the given args.
the ( ) block	Return the given block without executing it
run ( ) with inputs ▾	Run the given block with the given list of args.
self	Return myself
broadcast ▾ and wait	send a broadcast message ans wait for all receivers finish their work.

available on phratch.com

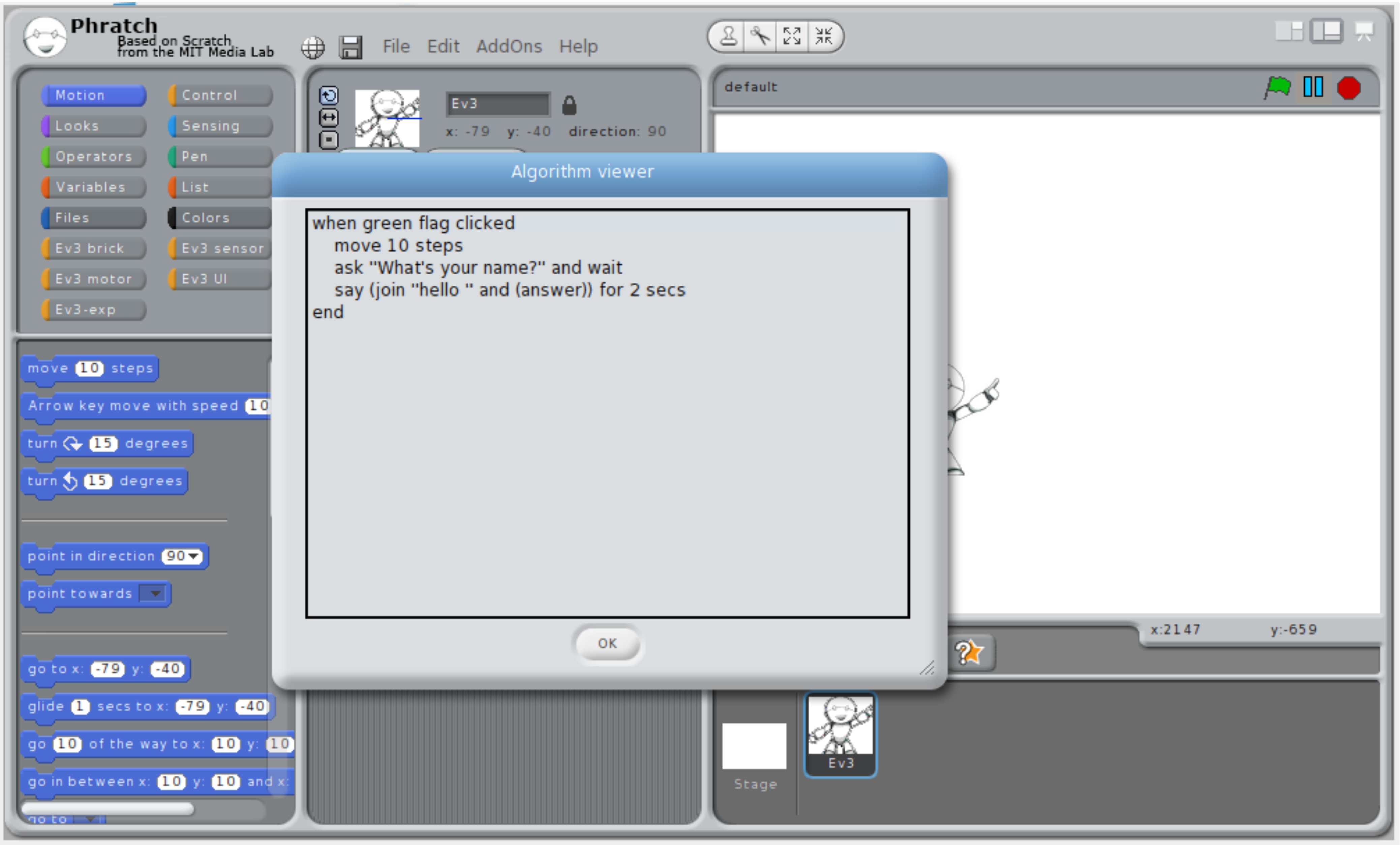
Code accessibility



Code accessibility

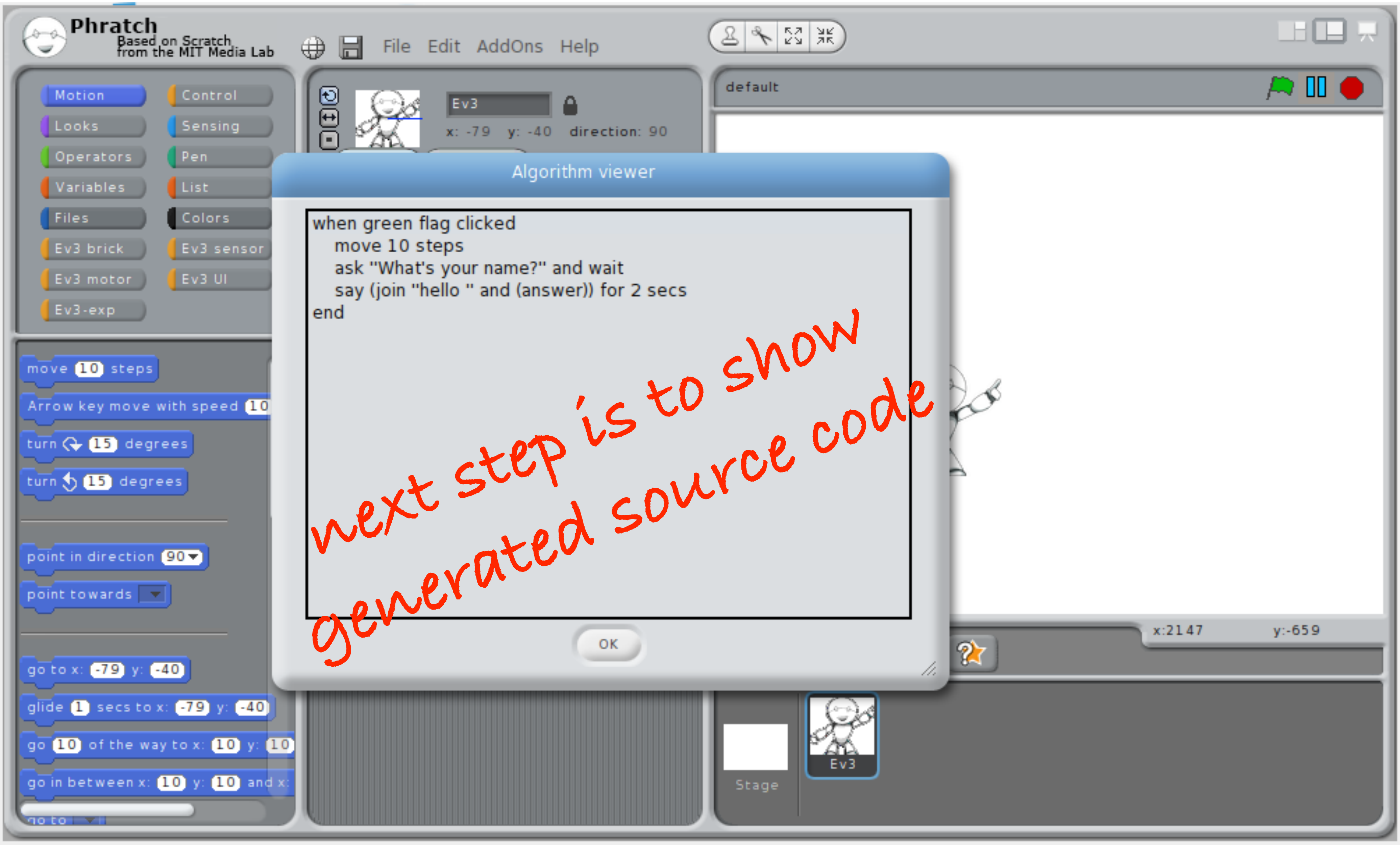


Code accessibility



```
when green flag clicked
    move 10 steps
    ask "What's your name?" and wait
    say (join "hello " and (answer)) for 2 secs
end
```

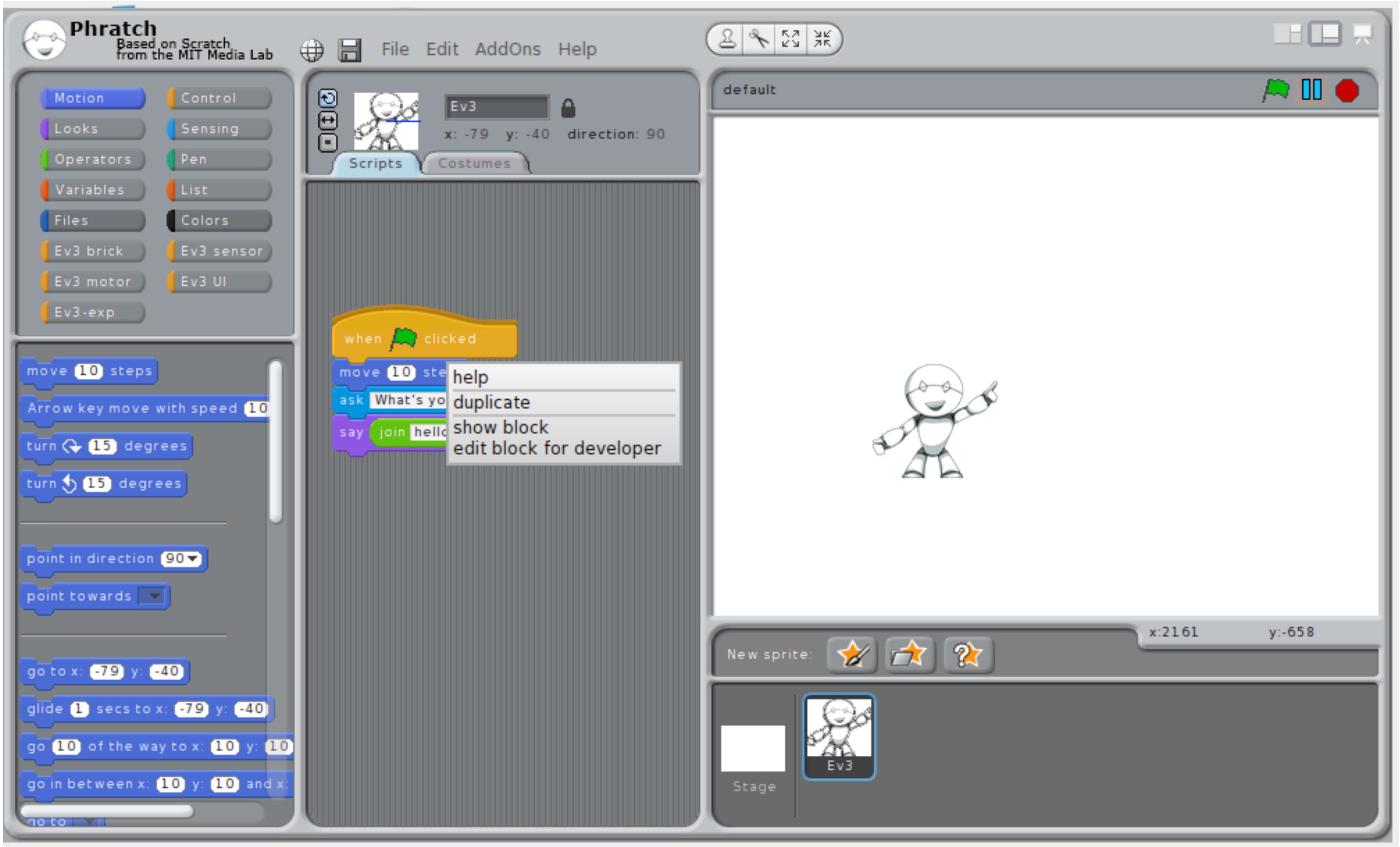
Code accessibility



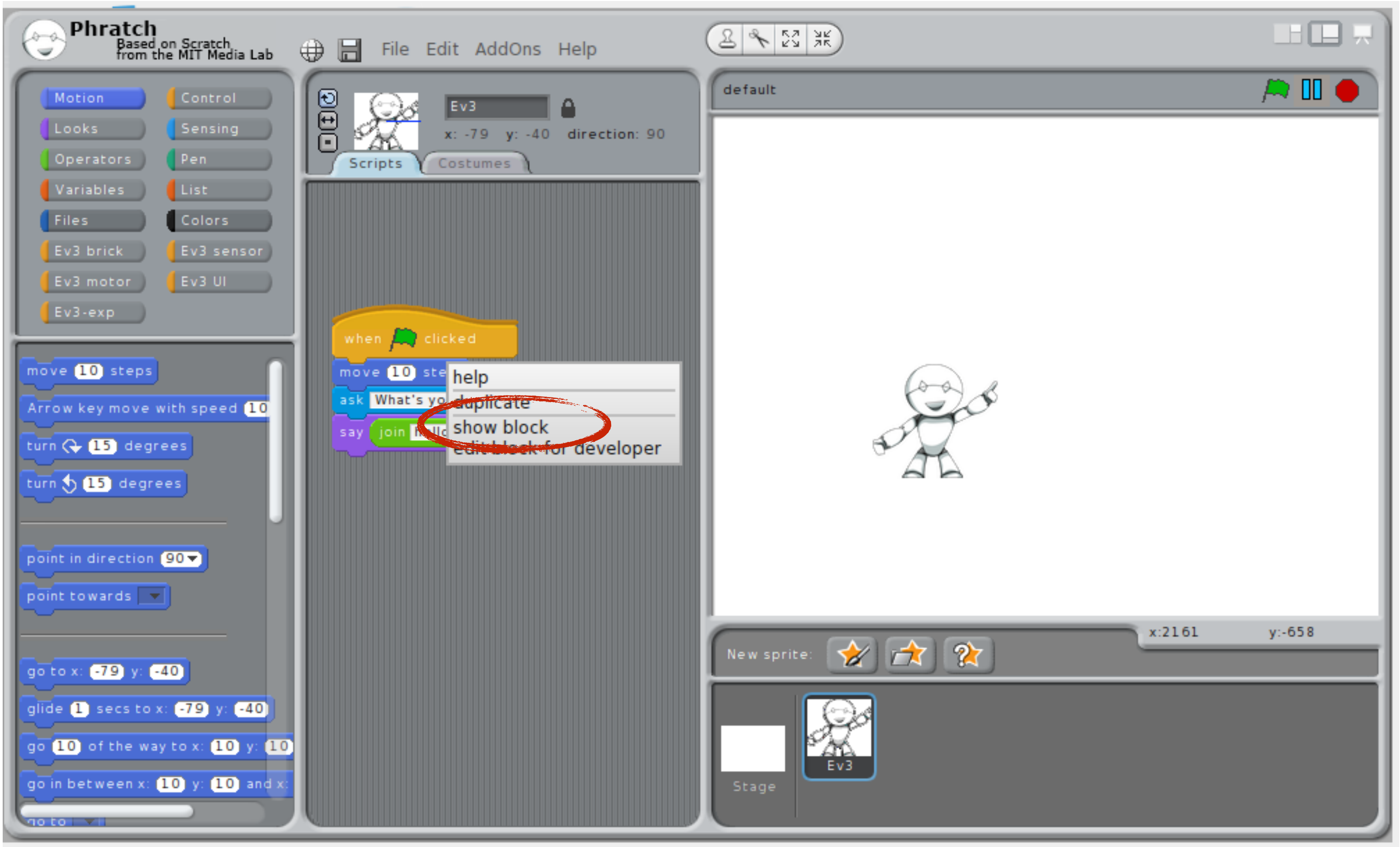
```
when green flag clicked
    move 10 steps
    ask "What's your name?" and wait
    say (join "hello " and (answer)) for 2 secs
end
```

next step is to show generated source code

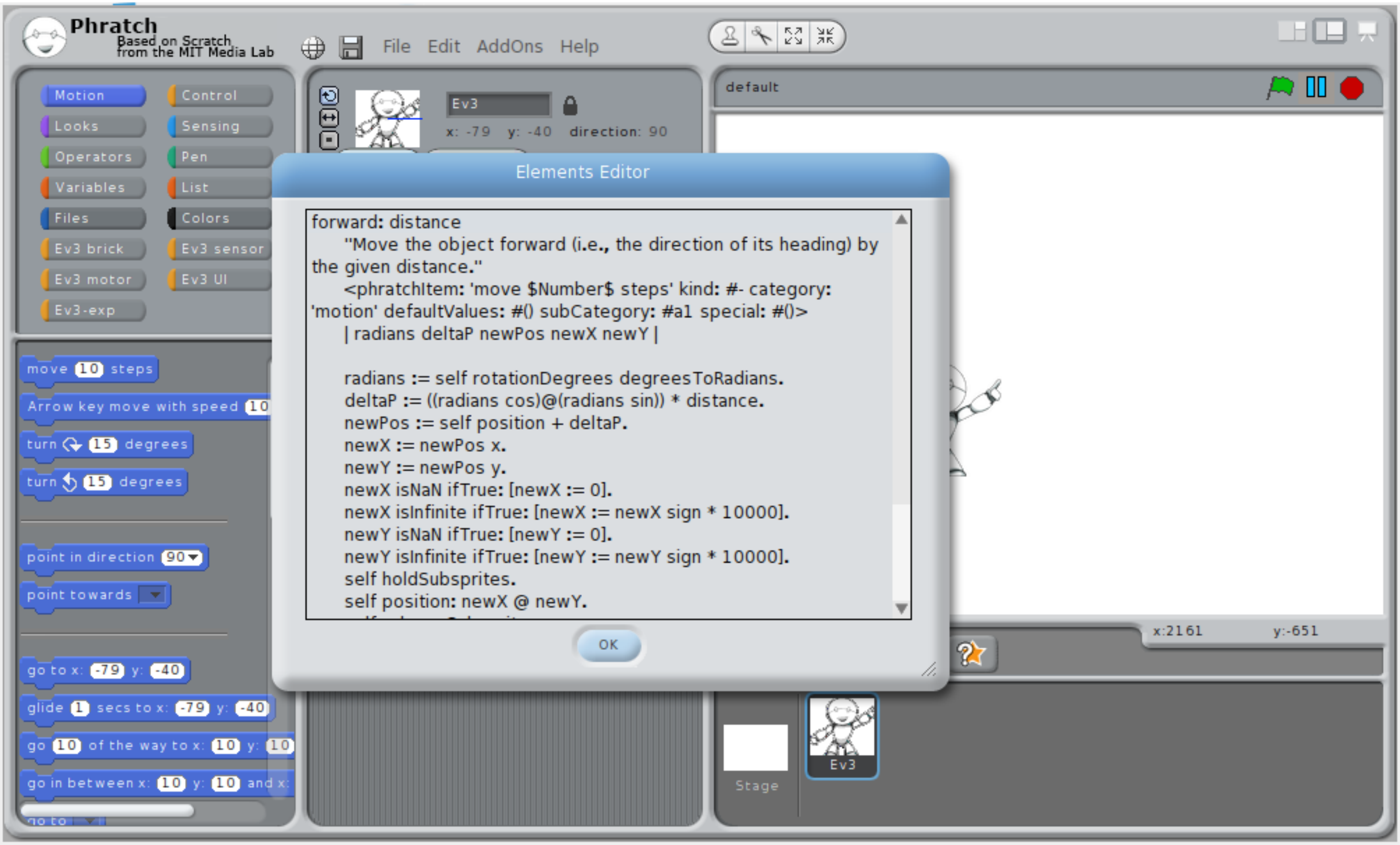
Code accessibility



Code accessibility

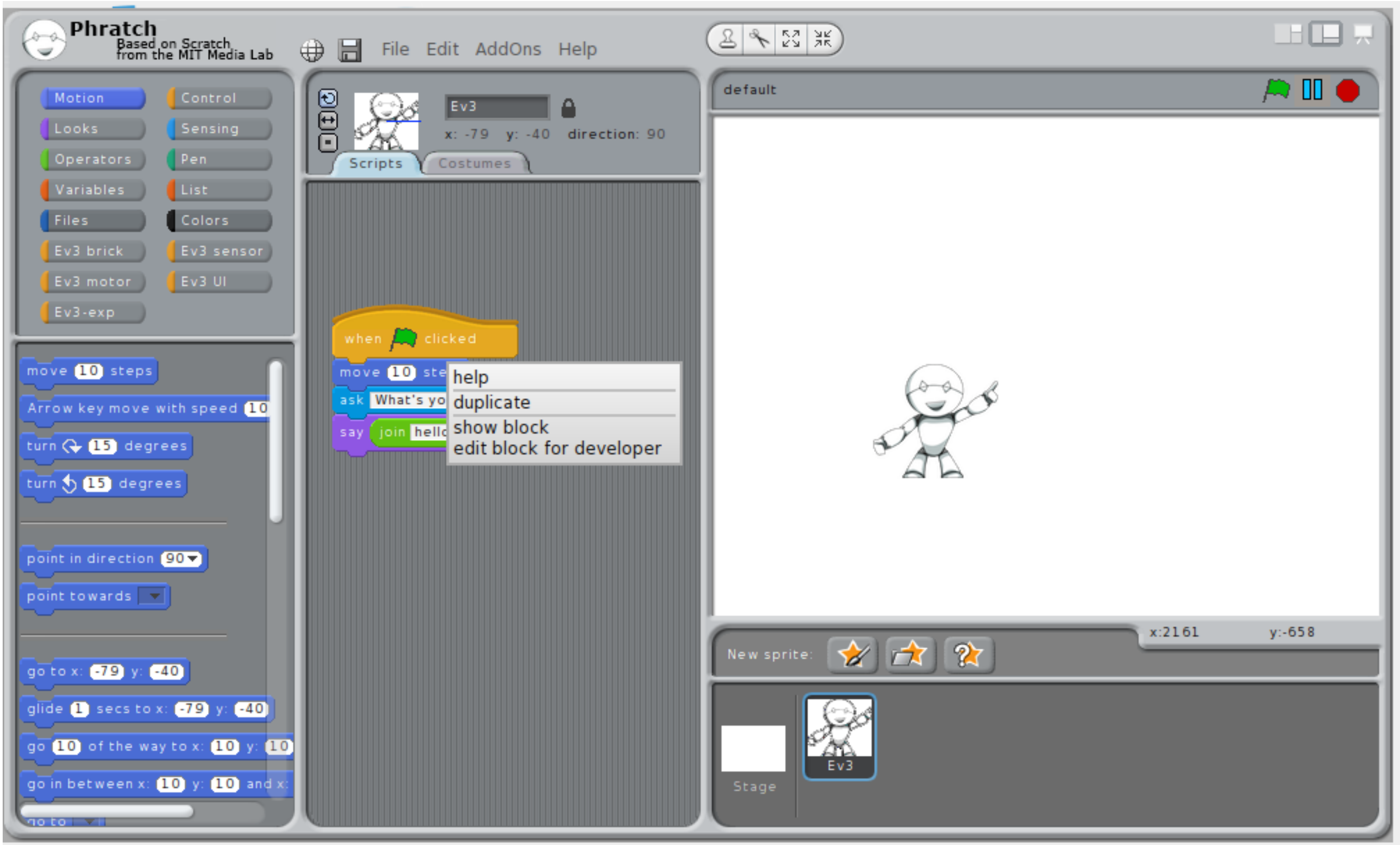


Code accessibility

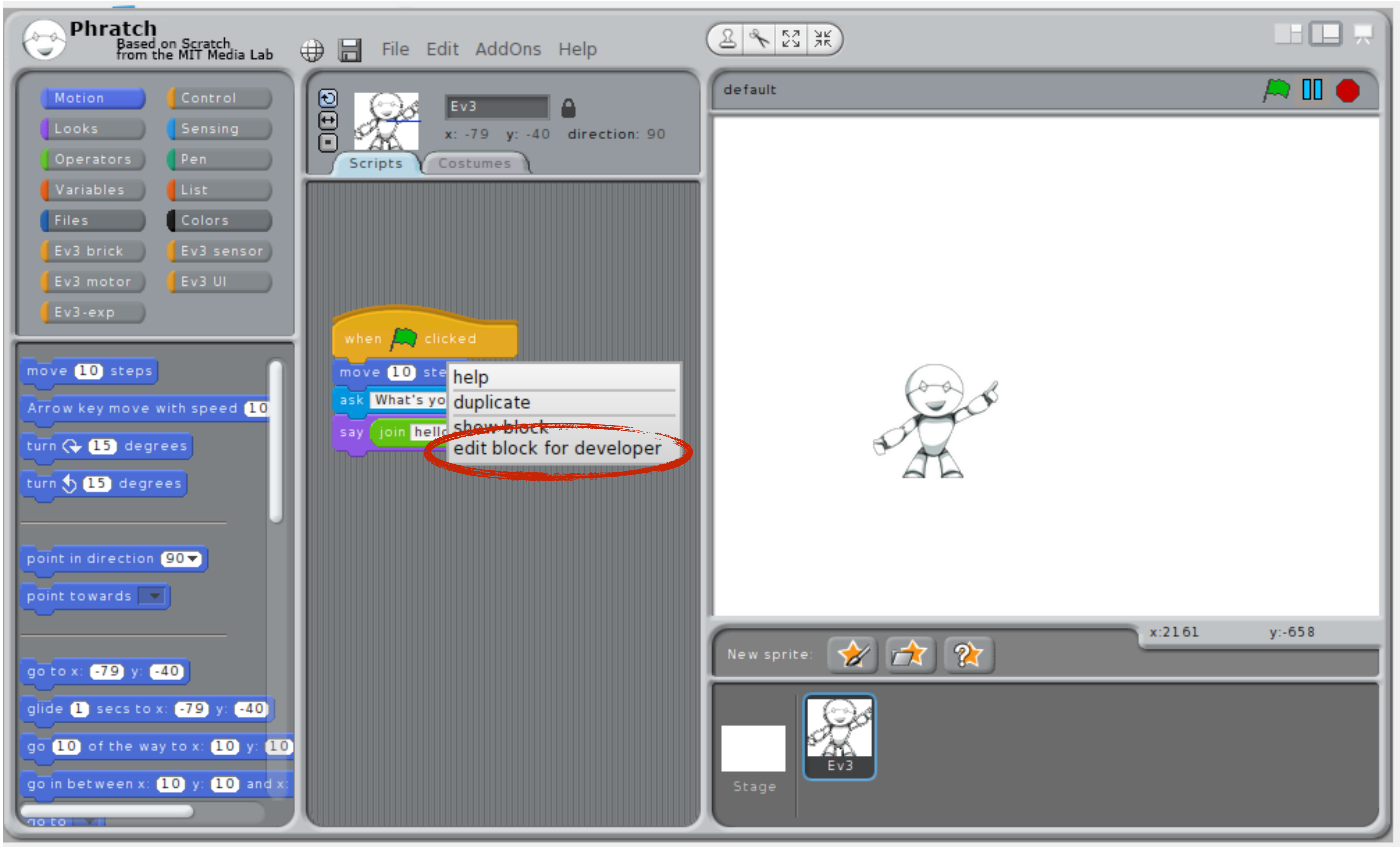


```
forward: distance
	"Move the object forward (i.e., the direction of its heading) by the given distance."
	<phratchItem: 'move $Number$ steps' kind: #- category: 'motion' defaultValues: #() subCategory: #a1 special: #()>
	| radians deltaP newPos newX newY |

	radians := self rotationDegrees degreesToRadians.
	deltaP := ((radians cos)@(radians sin)) * distance.
	newPos := self position + deltaP.
	newX := newPos x.
	newY := newPos y.
	newX isNaN ifTrue: [newX := 0].
	newX isInfinite ifTrue: [newX := newX sign * 10000].
	newY isNaN ifTrue: [newY := 0].
	newY isInfinite ifTrue: [newY := newY sign * 10000].
	self holdSubsprites.
	self position: newX @ newY.
```

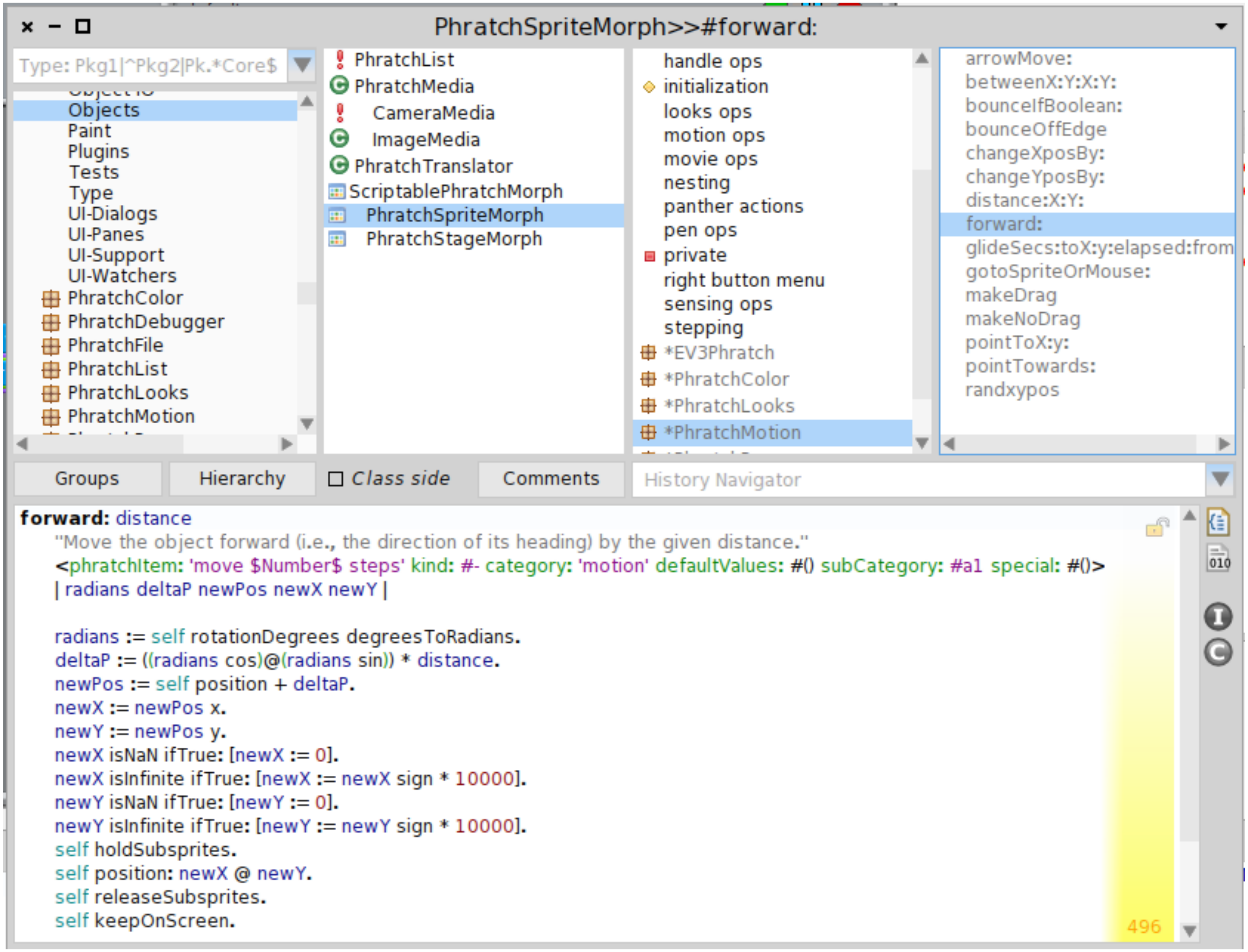
Code accessibility



Code accessibility

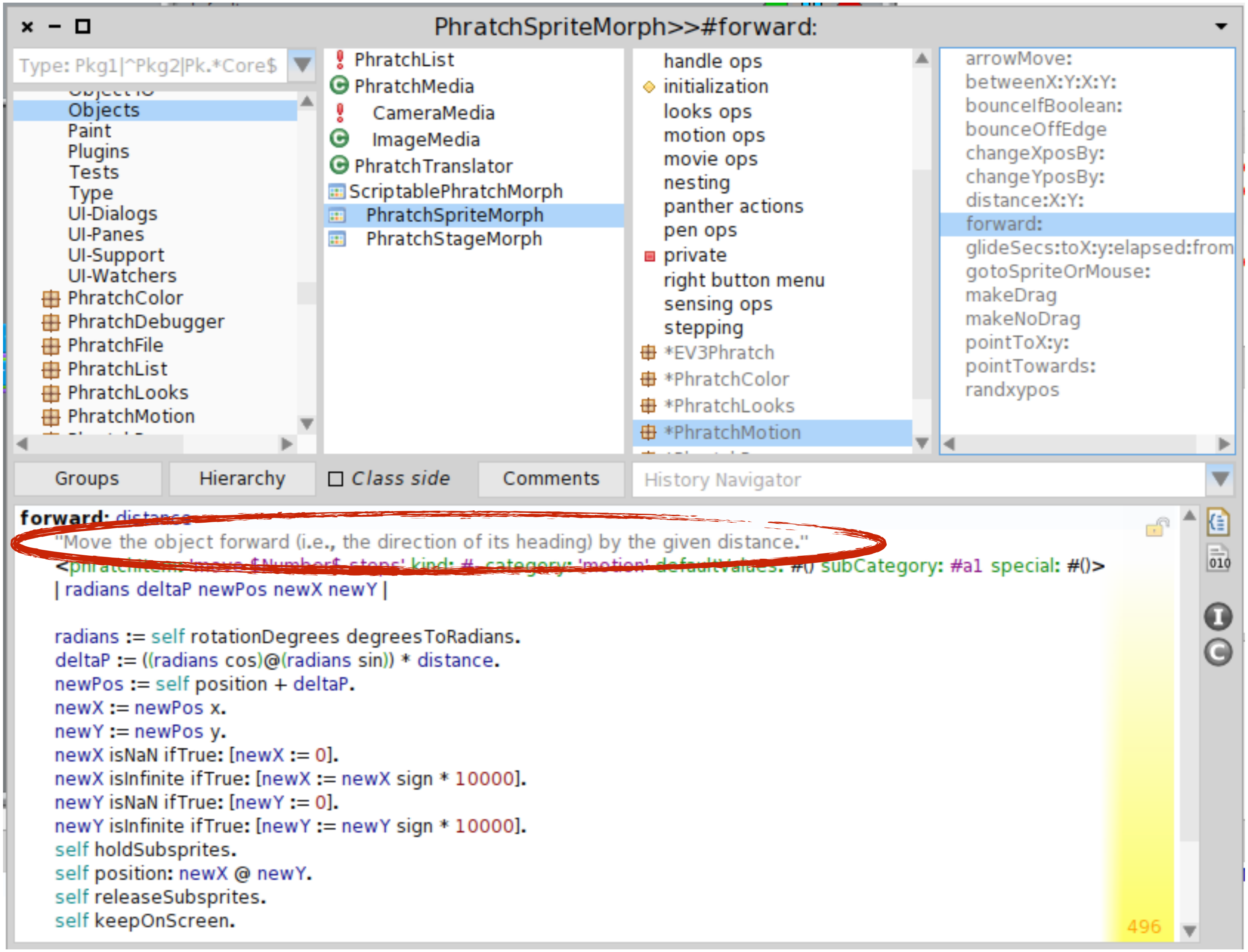


Code accessibility



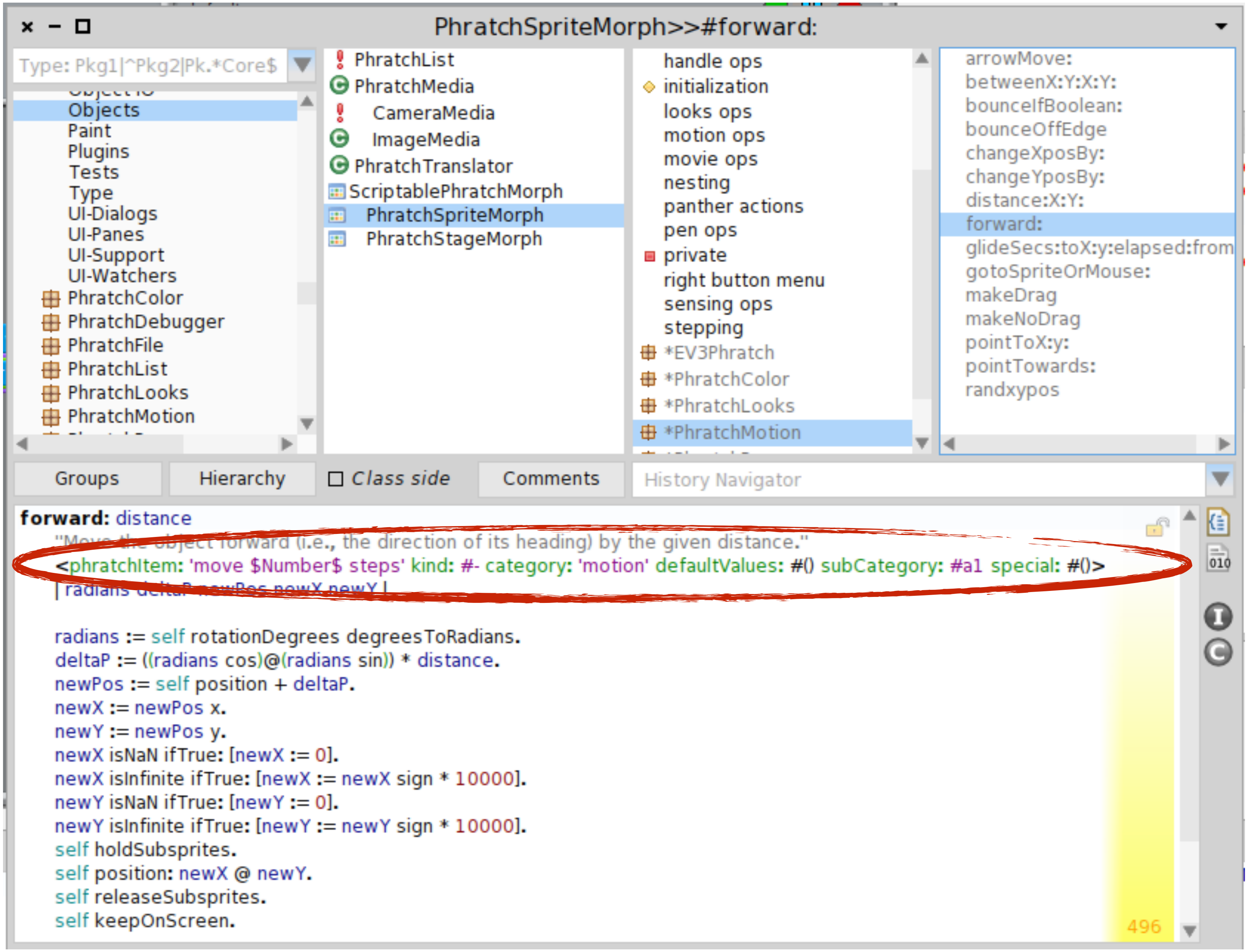
```
forward: distance
	"Move the object forward (i.e., the direction of its heading) by the given distance."
	<phratchItem: 'move $Number$ steps' kind: #- category: 'motion' defaultValues: #() subCategory: #a1 special: #()>
	| radians deltaP newPos newX newY |

	radians := self rotationDegrees degreesToRadians.
	deltaP := ((radians cos)@(radians sin)) * distance.
	newPos := self position + deltaP.
	newX := newPos x.
	newY := newPos y.
	newX isNaN ifTrue: [newX := 0].
	newX isInfinite ifTrue: [newX := newX sign * 10000].
	newY isNaN ifTrue: [newY := 0].
	newY isInfinite ifTrue: [newY := newY sign * 10000].
	self holdSubsprites.
	self position: newX @ newY.
	self releaseSubsprites.
	self keepOnScreen.
```

High extensibility !



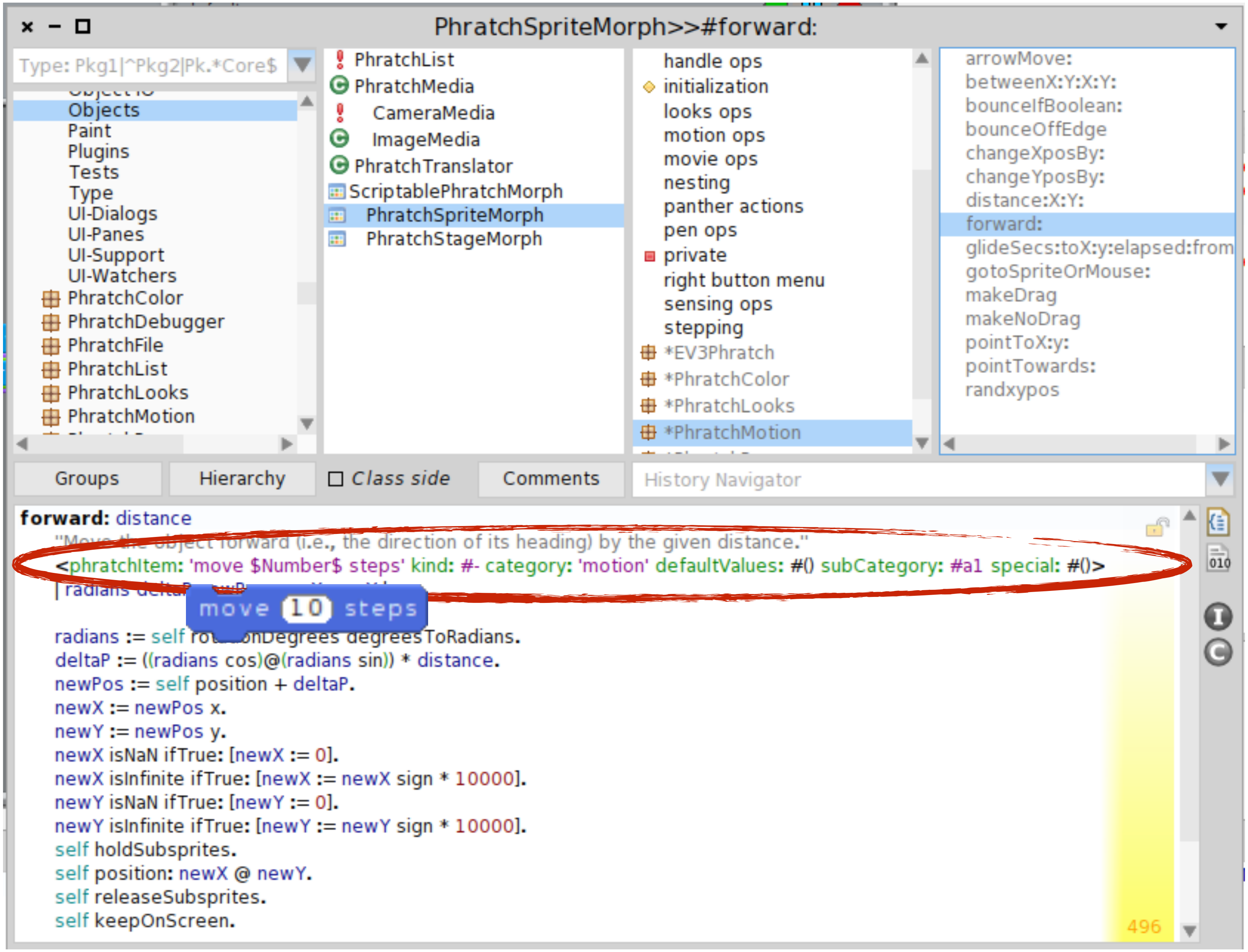
```
forward: distance
    "Move the object forward (i.e., the direction of its heading) by the given distance."
    <phratchItem: 'move $Number$ steps' kind: #- category: 'motion' defaultValues: #() subCategory: #a1 special: #()>
    | radians deltaP newPos newX newY |

    radians := self rotationDegrees degreesToRadians.
    deltaP := ((radians cos)@(radians sin)) * distance.
    newPos := self position + deltaP.
    newX := newPos x.
    newY := newPos y.
    newX isNaN ifTrue: [newX := 0].
    newX isInfinite ifTrue: [newX := newX sign * 10000].
    newY isNaN ifTrue: [newY := 0].
    newY isInfinite ifTrue: [newY := newY sign * 10000].
    self holdSubsprites.
    self position: newX @ newY.
    self releaseSubsprites.
    self keepOnScreen.
```

High extensibility !



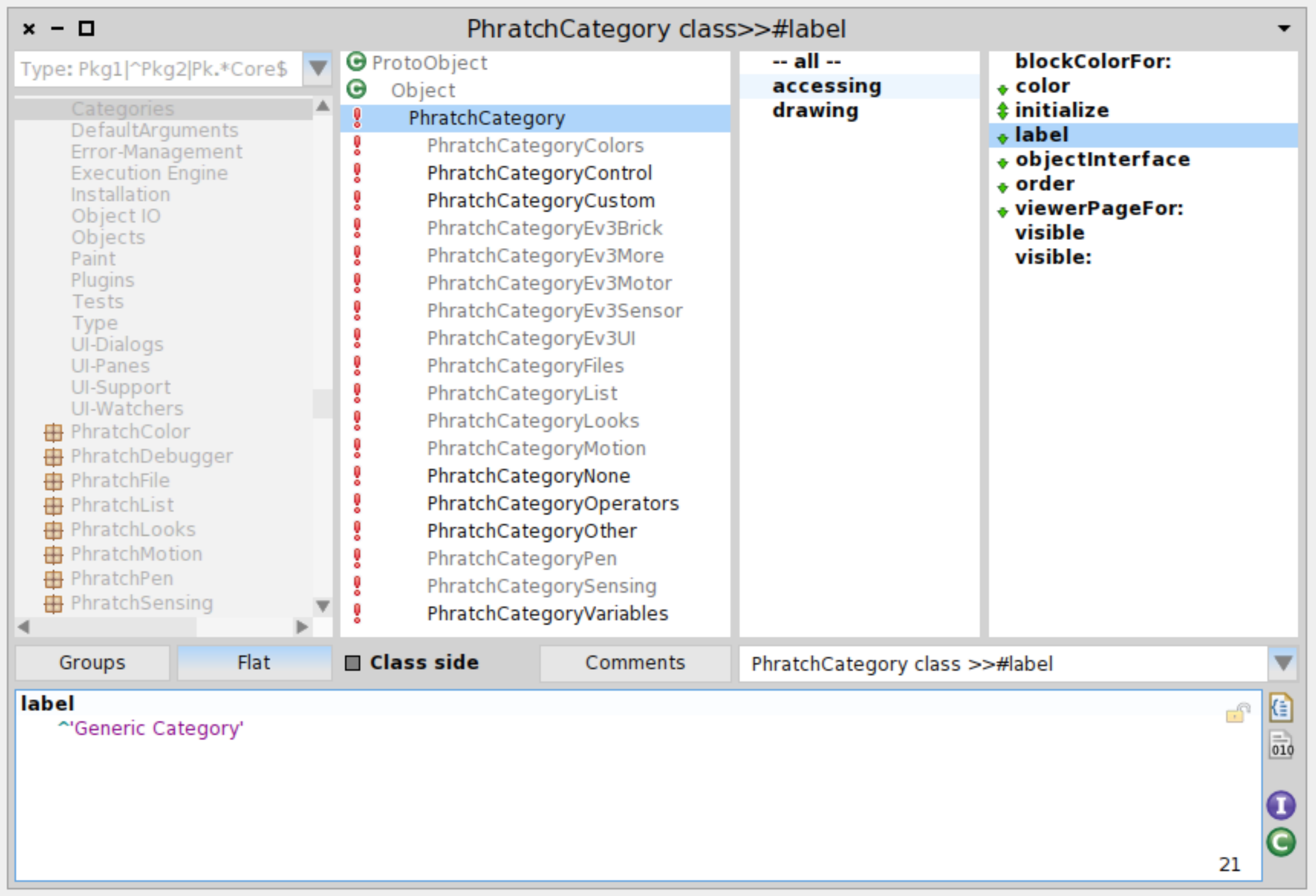
```
forward: distance
    "Move the object forward (i.e., the direction of its heading) by the given distance."
    <phratchItem: 'move $Number$ steps' kind: #- category: 'motion' defaultValues: #() subCategory: #a1 special: #()>
    | radians deltaP newPos newX newY |

    radians := self rotationDegrees degreesToRadians.
    deltaP := ((radians cos)@(radians sin)) * distance.
    newPos := self position + deltaP.
    newX := newPos x.
    newY := newPos y.
    newX isNaN ifTrue: [newX := 0].
    newX isInfinite ifTrue: [newX := newX sign * 10000].
    newY isNaN ifTrue: [newY := 0].
    newY isInfinite ifTrue: [newY := newY sign * 10000].
    self holdSubsprites.
    self position: newX @ newY.
    self releaseSubsprites.
    self keepOnScreen.
```

High extensibility !



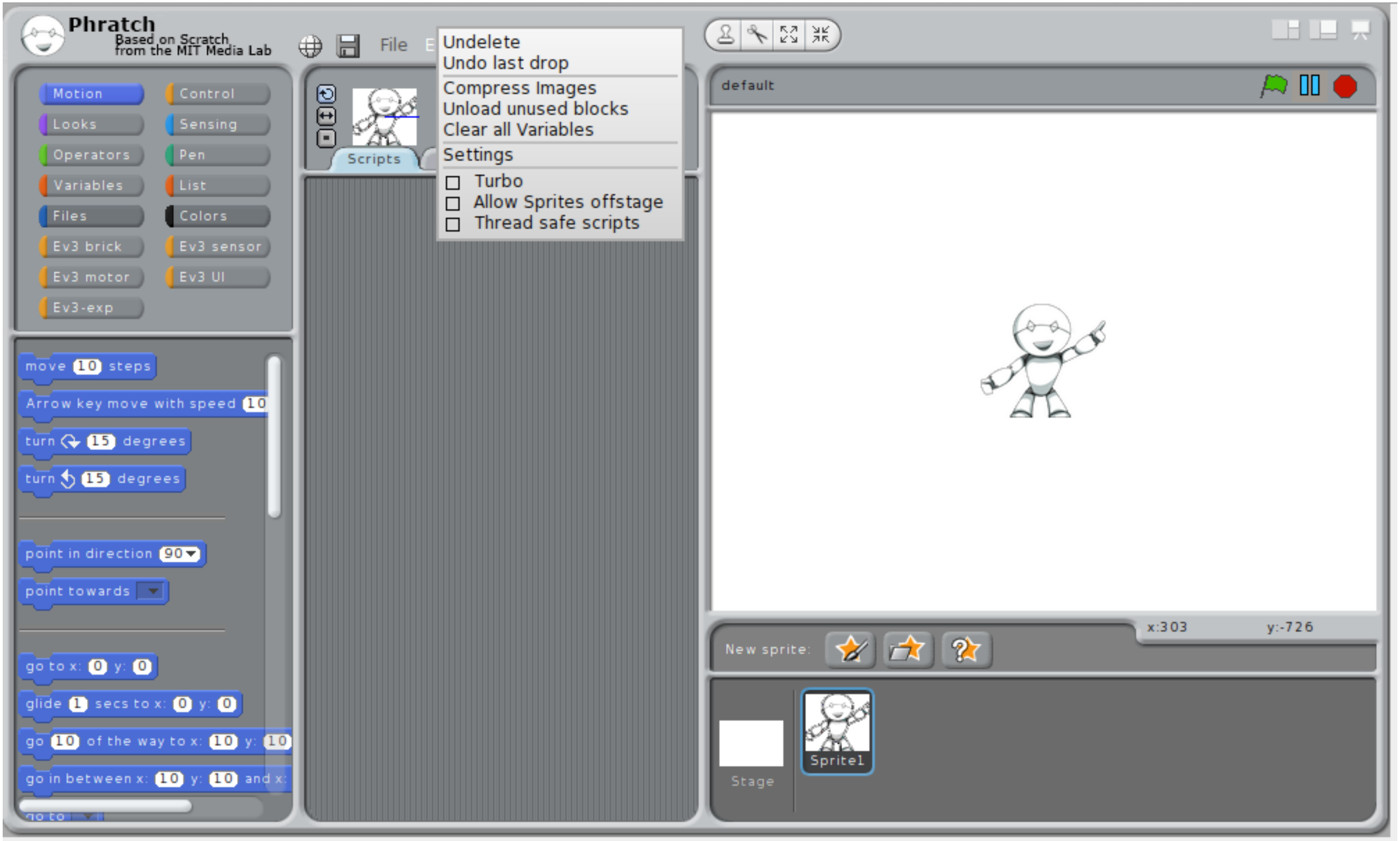
```
forward: distance
    "Move the object forward (i.e., the direction of its heading) by the given distance."
    <phratchItem: 'move $Number$ steps' kind: #- category: 'motion' defaultValues: #() subCategory: #a1 special: #()>
    | radians deltaP newPos newX newY |

    radians := self rotationDegrees degreesToRadians.
    deltaP := ((radians cos)@(radians sin)) * distance.
    newPos := self position + deltaP.
    newX := newPos x.
    newY := newPos y.
    newX isNaN ifTrue: [newX := 0].
    newX isInfinite ifTrue: [newX := newX sign * 10000].
    newY isNaN ifTrue: [newY := 0].
    newY isInfinite ifTrue: [newY := newY sign * 10000].
    self holdSubsprites.
    self position: newX @ newY.
    self releaseSubsprites.
    self keepOnScreen.
```

High extensibility !

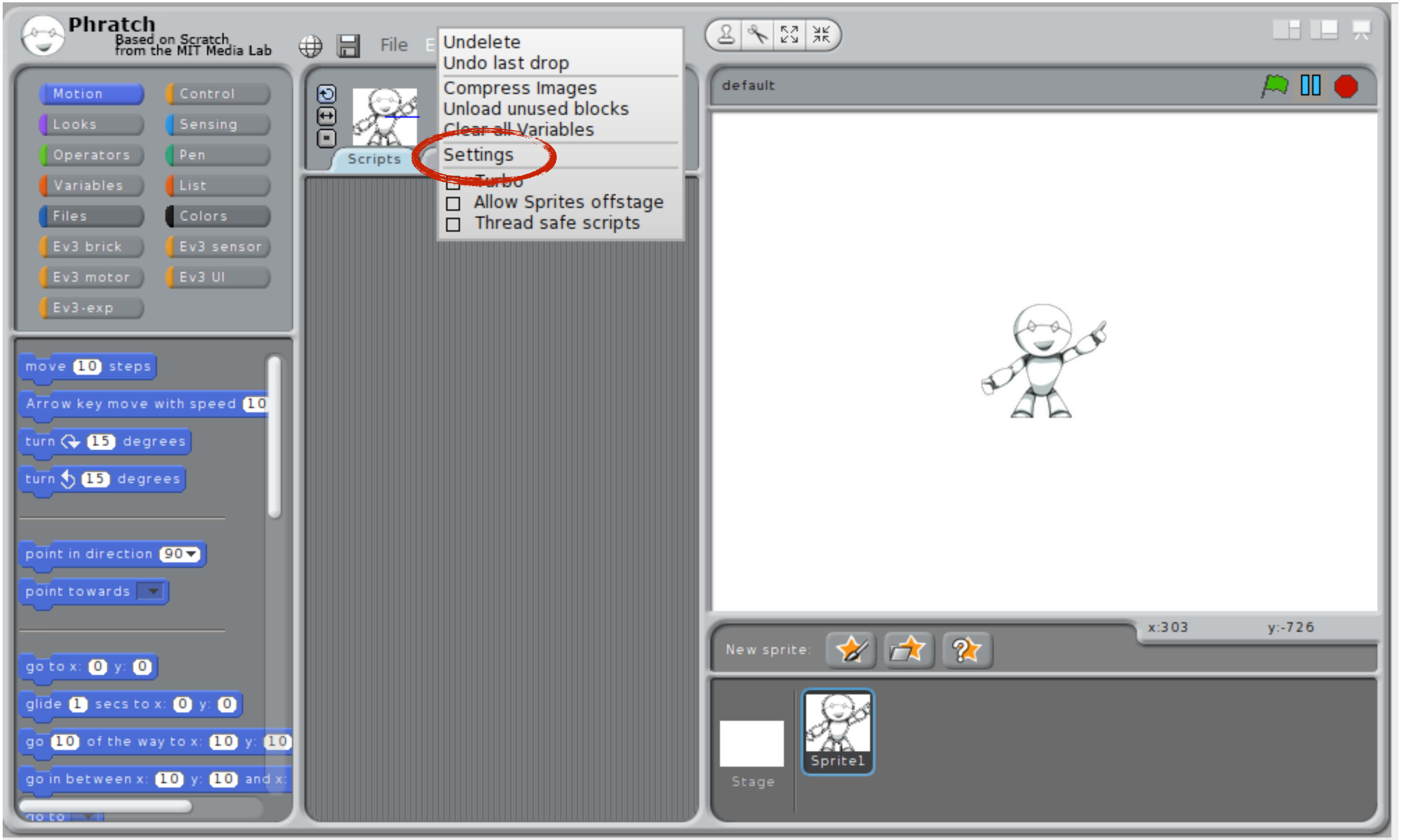


```
label
    ^'Generic Category'
```

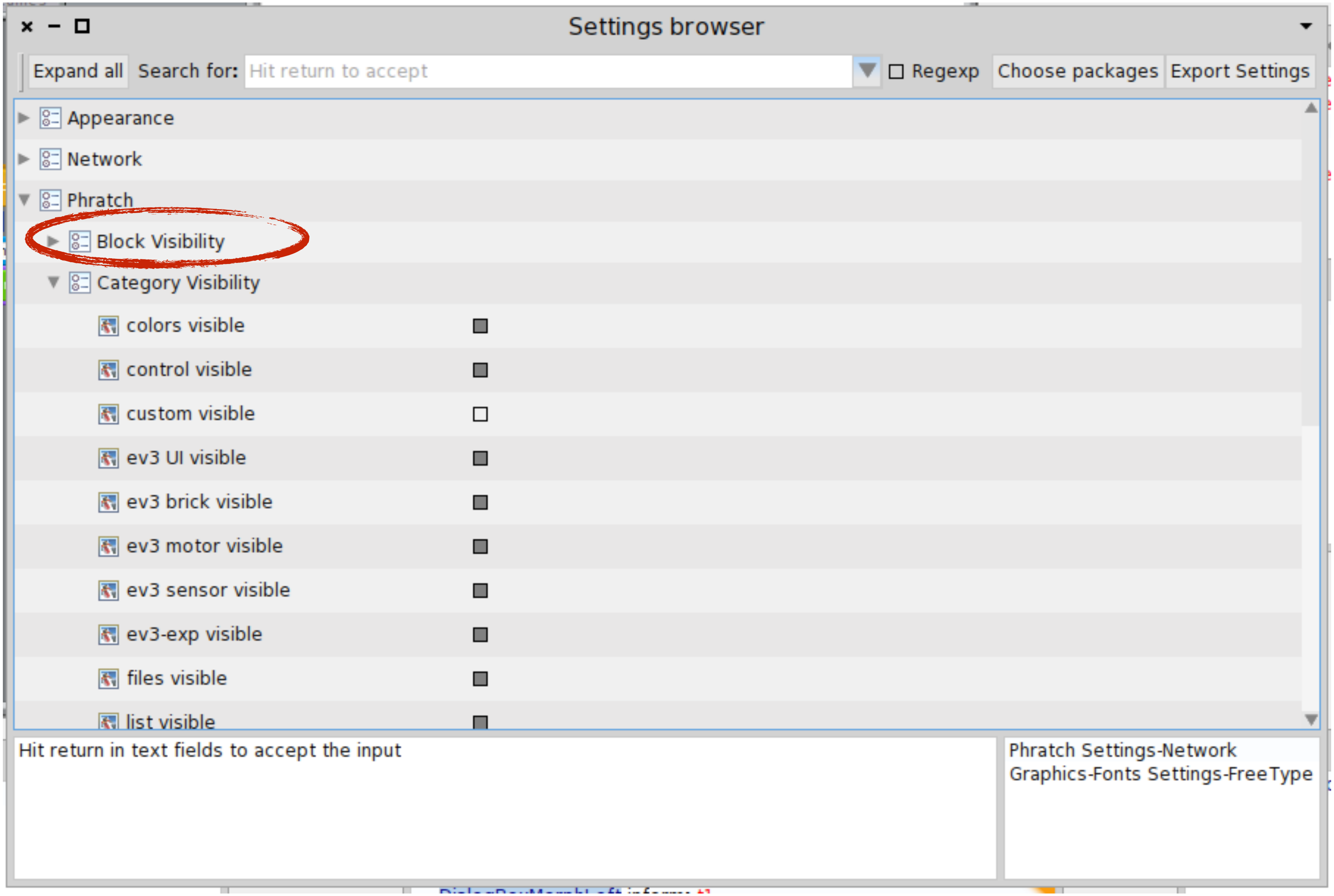
Customization



Customization



Customization

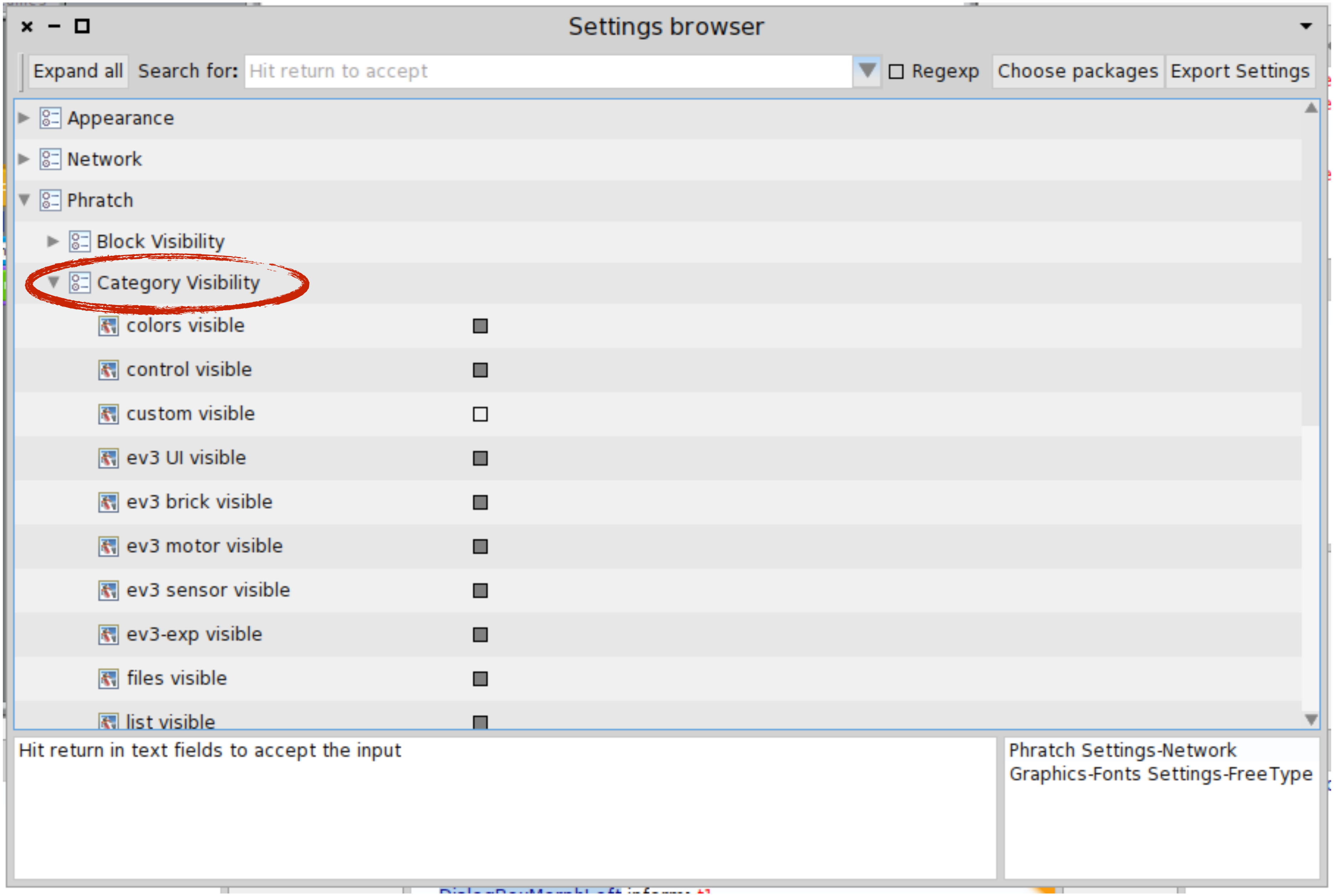


- Appearance
- Network
- Phratch
  - Block Visibility
  - Category Visibility
    - colors visible ■
    - control visible ■
    - custom visible □
    - ev3 UI visible ■
    - ev3 brick visible ■
    - ev3 motor visible ■
    - ev3 sensor visible ■
    - ev3-exp visible ■
    - files visible ■
    - list visible ■

Hit return in text fields to accept the input

Phratch Settings-Network Graphics-Fonts Settings-FreeType

Customization

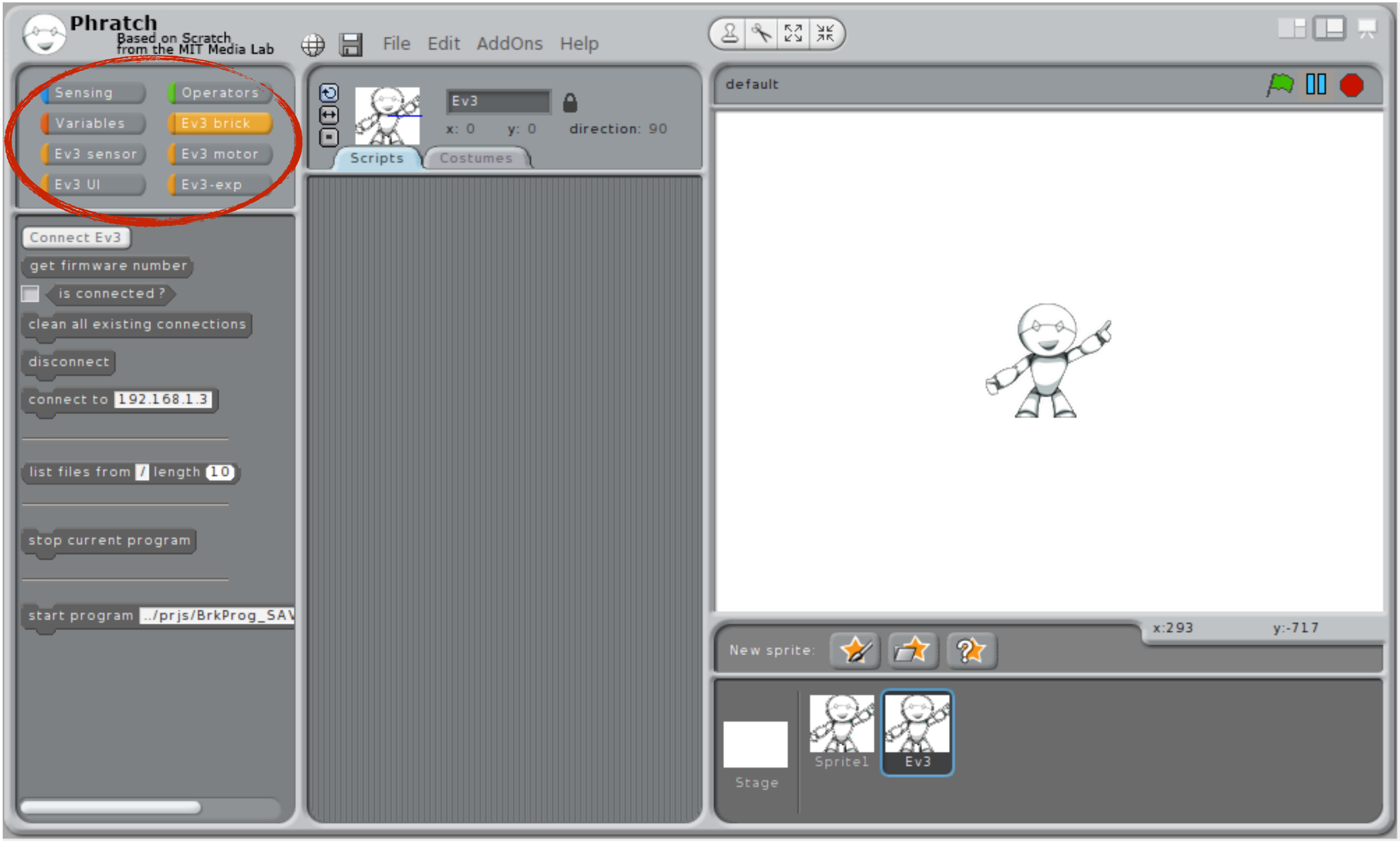


- ▶ Appearance
- ▶ Network
- ▼ Phratch
  - ▶ Block Visibility
  - ▼ Category Visibility
    - colors visible ■
    - control visible ■
    - custom visible □
    - ev3 UI visible ■
    - ev3 brick visible ■
    - ev3 motor visible ■
    - ev3 sensor visible ■
    - ev3-exp visible ■
    - files visible ■
    - list visible ■

Hit return in text fields to accept the input

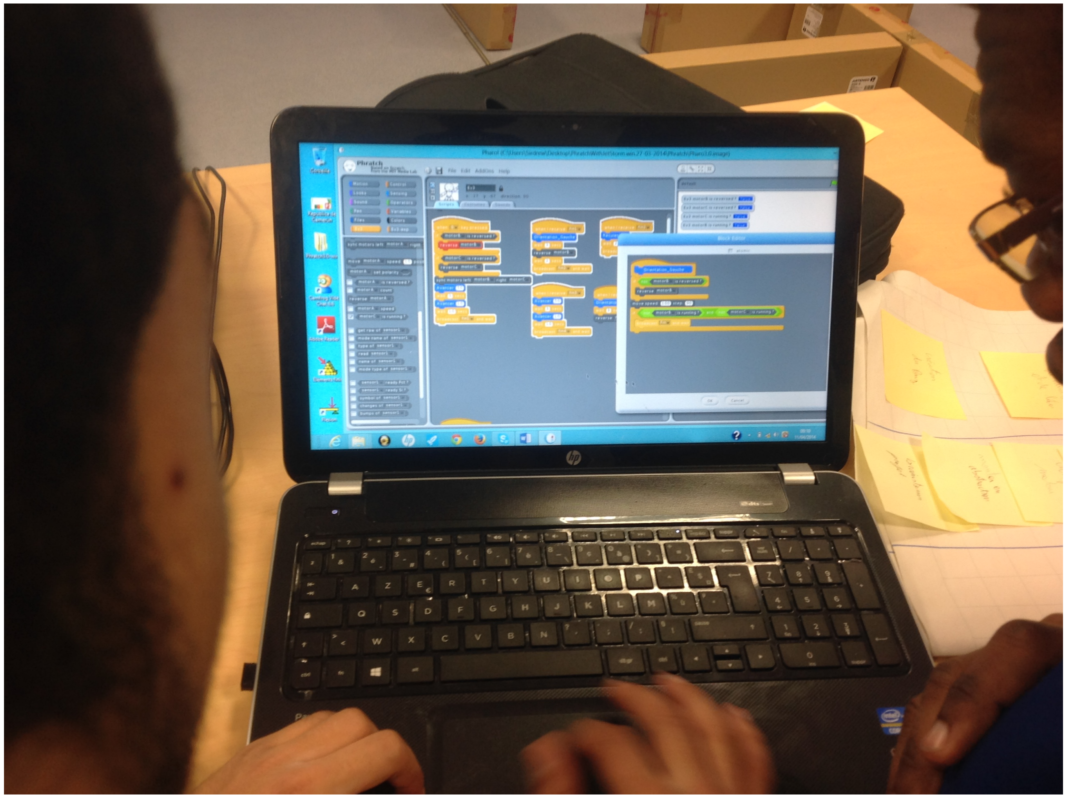
Phratch Settings-Network Graphics-Fonts Settings-FreeType

Customization

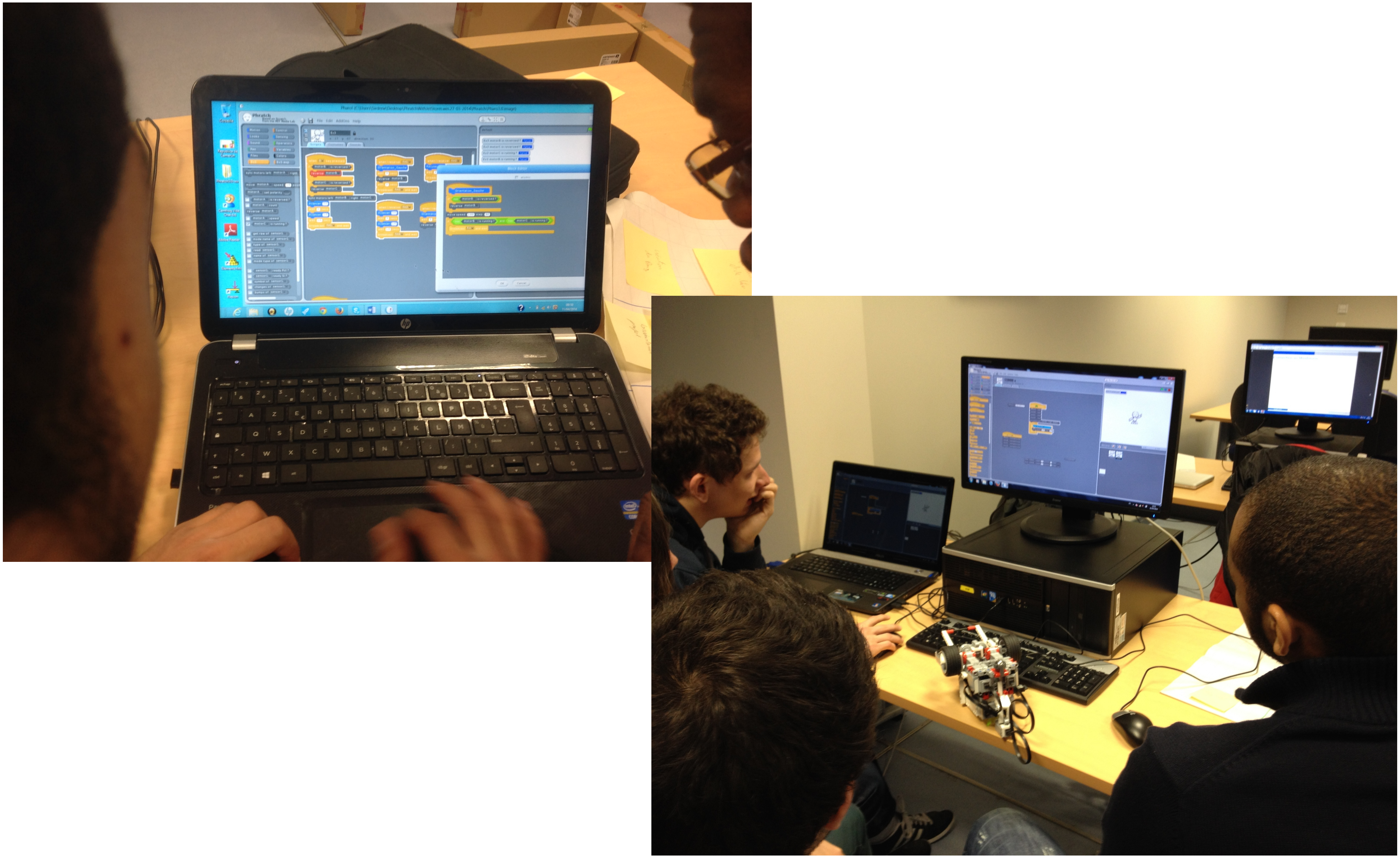


Used for real !

Used for real !

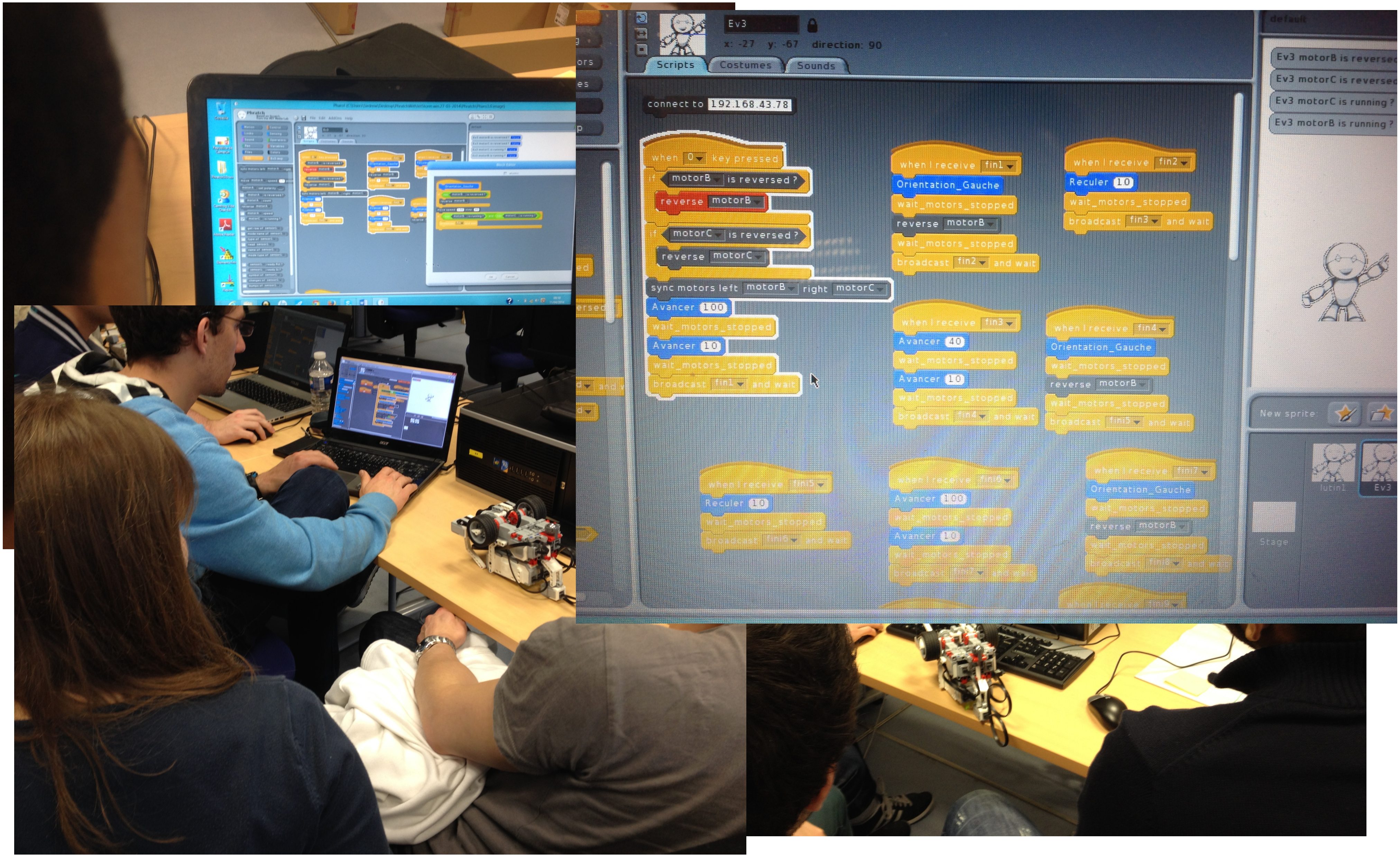


Used for real !



Used for real !

Used for real !



```
connect to 192.168.43.78

when 0 key pressed
if motorB is reversed ?
    reverse motorB
if motorC is reversed ?
    reverse motorC
sync motors left motorB right motorC
Avancer 100
wait_motors_stopped
Avancer 10
wait_motors_stopped
broadcast fin1 and wait
```

```
when I receive fin1
Orientation_Gauche
wait_motors_stopped
reverse motorB
wait_motors_stopped
broadcast fin2 and wait
```

```
when I receive fin2
Reculer 10
wait_motors_stopped
broadcast fin3 and wait
```

```
when I receive fin3
Avancer 40
wait_motors_stopped
Avancer 10
wait_motors_stopped
broadcast fin4 and wait
```

```
when I receive fin4
Orientation_Gauche
wait_motors_stopped
reverse motorB
wait_motors_stopped
broadcast fini5 and wait
```

```
when I receive fini5
Reculer 10
wait_motors_stopped
broadcast fini6 and wait
```

```
when I receive fini6
Avancer 100
wait_motors_stopped
Avancer 10
wait_motors_stopped
broadcast fini7 and wait
```

```
when I receive fini7
Orientation_Gauche
wait_motors_stopped
reverse motorB
wait_motors_stopped
broadcast fini8 and wait
```

```
when I receive fini9
```

Next steps

Raspberry-Pi

Next steps

Raspberry-Pi

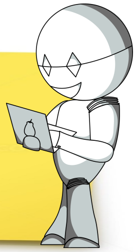
Arduino

Next steps

Raspberry-Pi

Arduino

Android



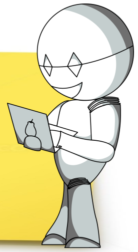
Next steps

Raspberry-Pi

Modularity

Arduino

Android



Next steps

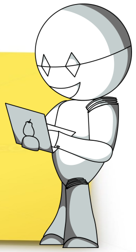
Raspberry-Pi

Modularity

Arduino

Integration with Pharo

Android



Next steps

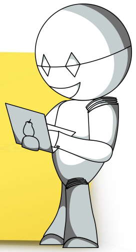
Raspberry-Pi

Modularity

Arduino

Integration with Pharo

Android



Web interface

Next steps

Raspberry-Pi

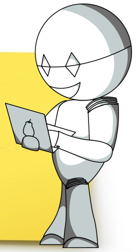
Modularity

Robotics

Arduino

Integration with Pharo

Android



Web interface

Next steps

Raspberry-Pi

Modularity

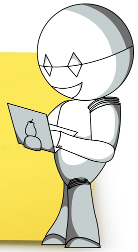
Robotics

Arduino

Integration with Pharo

Community

Android



Web interface

Next steps

Raspberry-Pi

Modularity

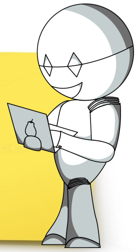
Robotics

Arduino

Integration with Pharo

Community

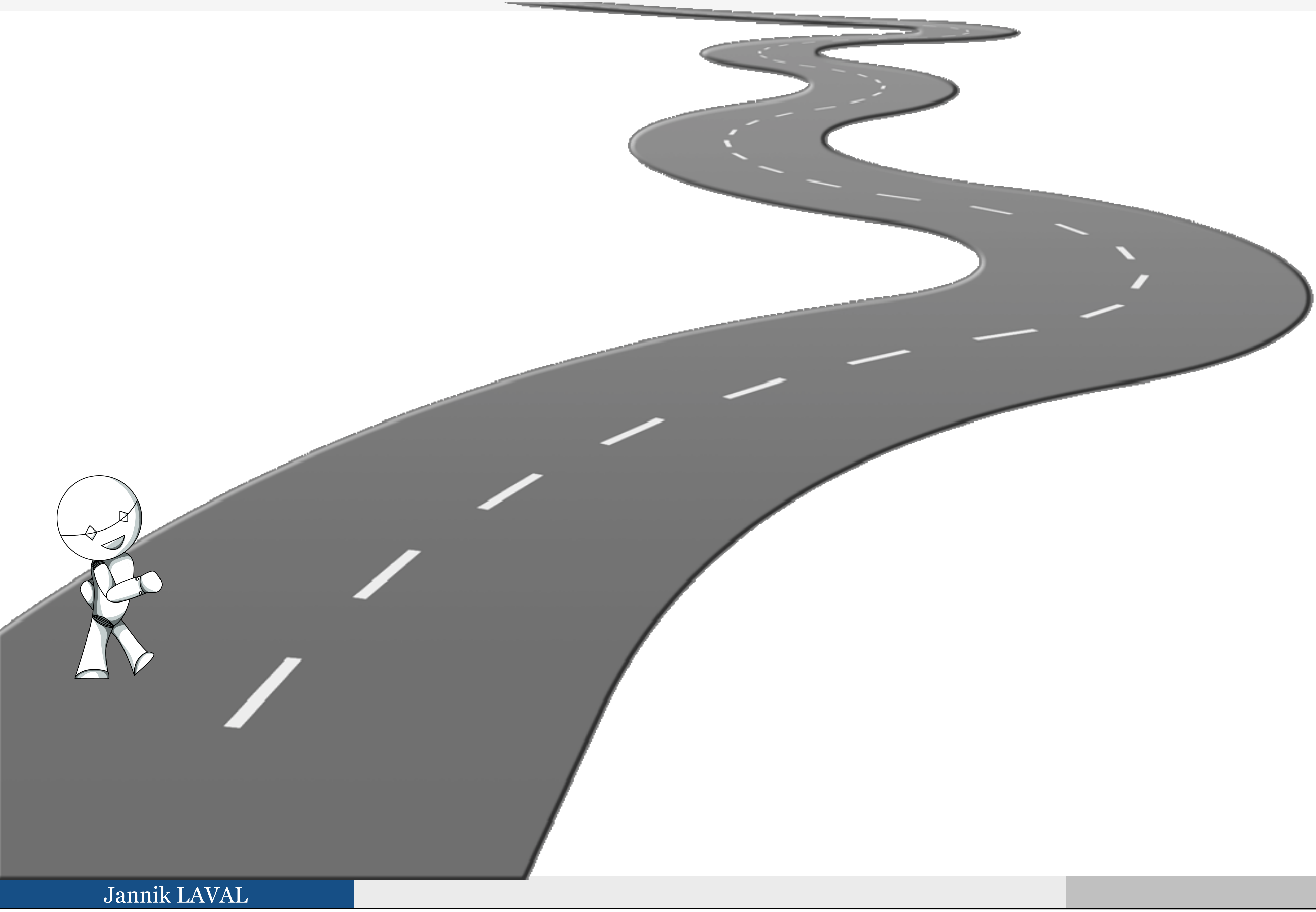
Android



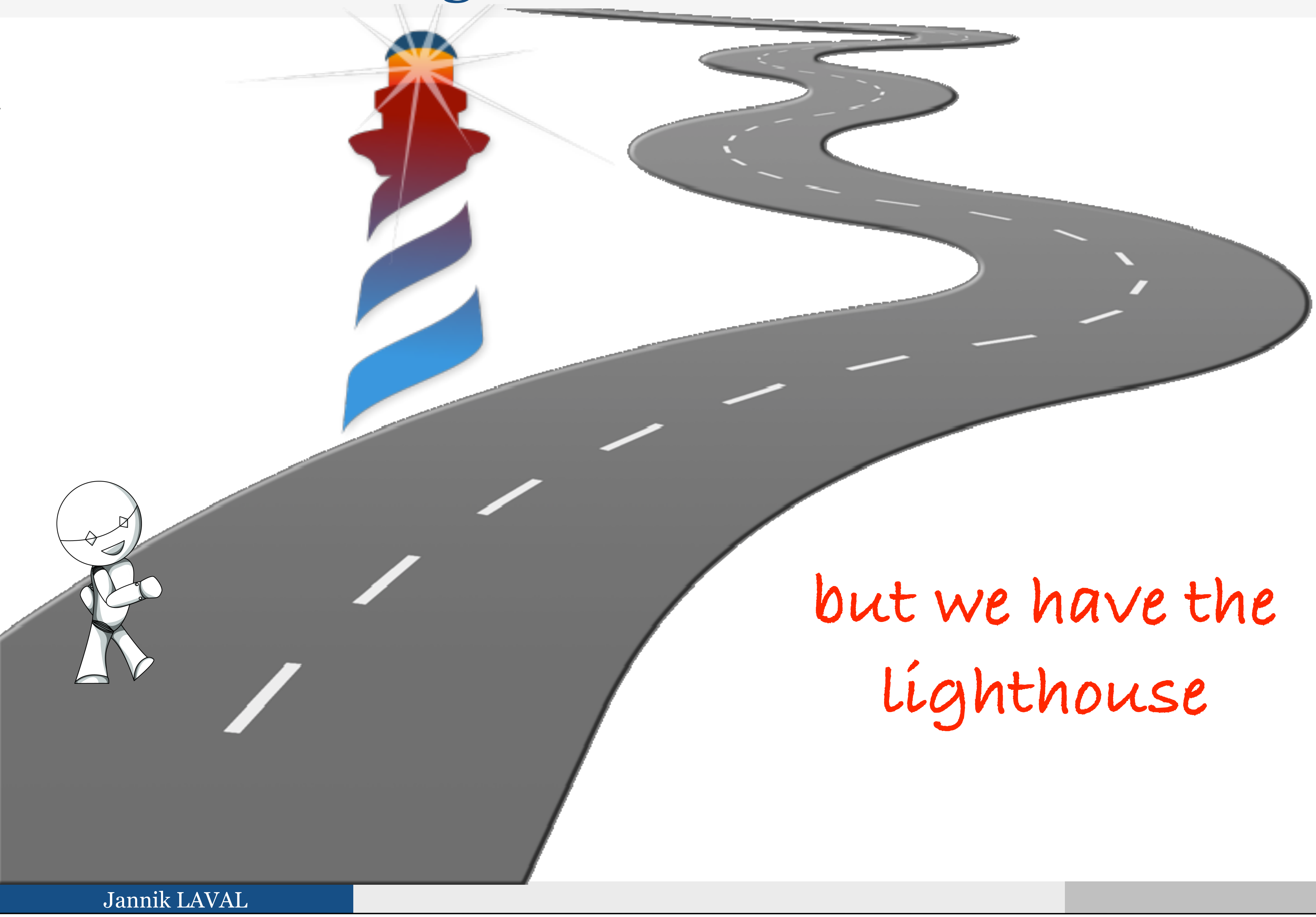
Web interface

...

The road is long...

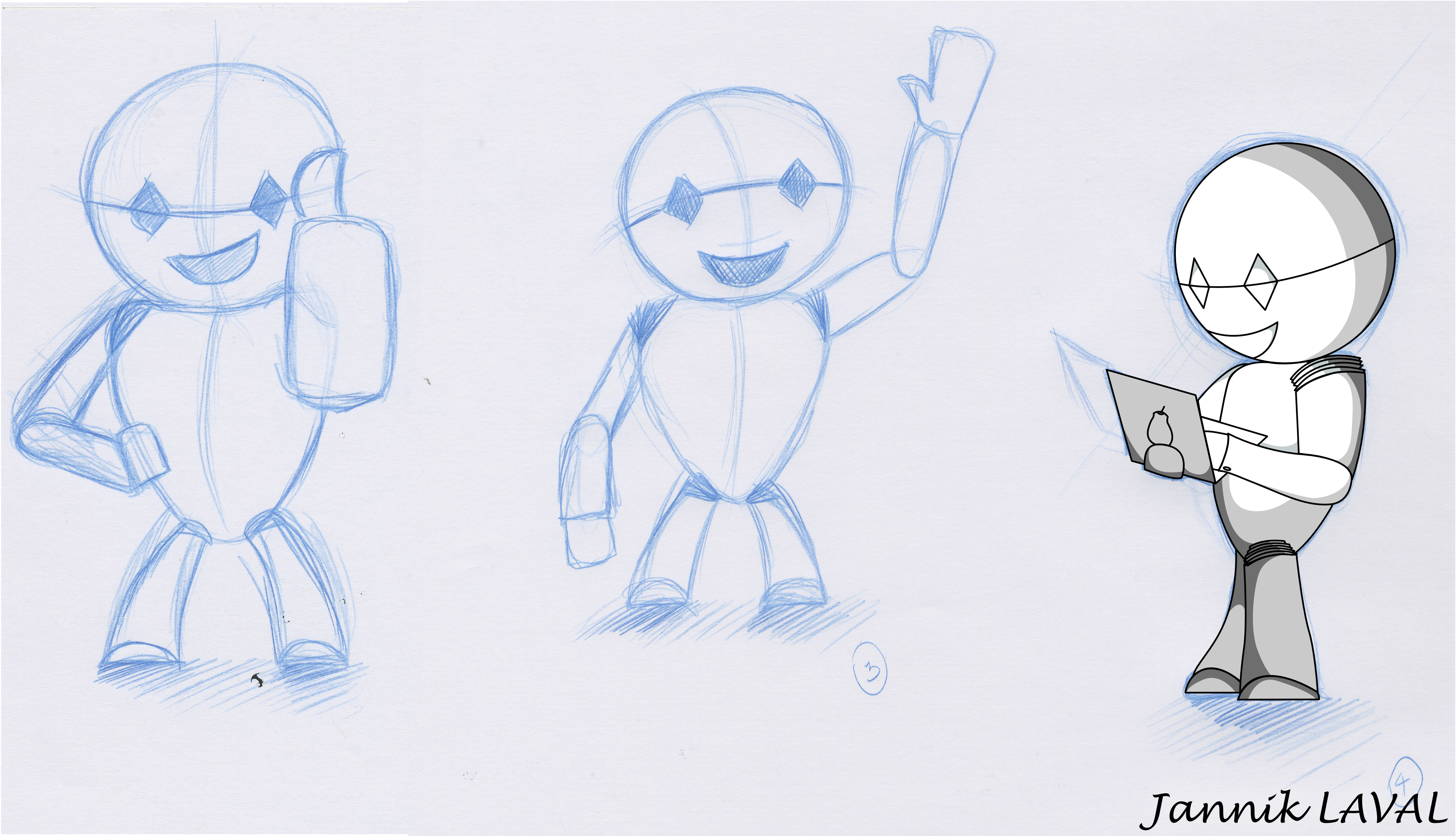


The road is long...



but we have the lighthouse

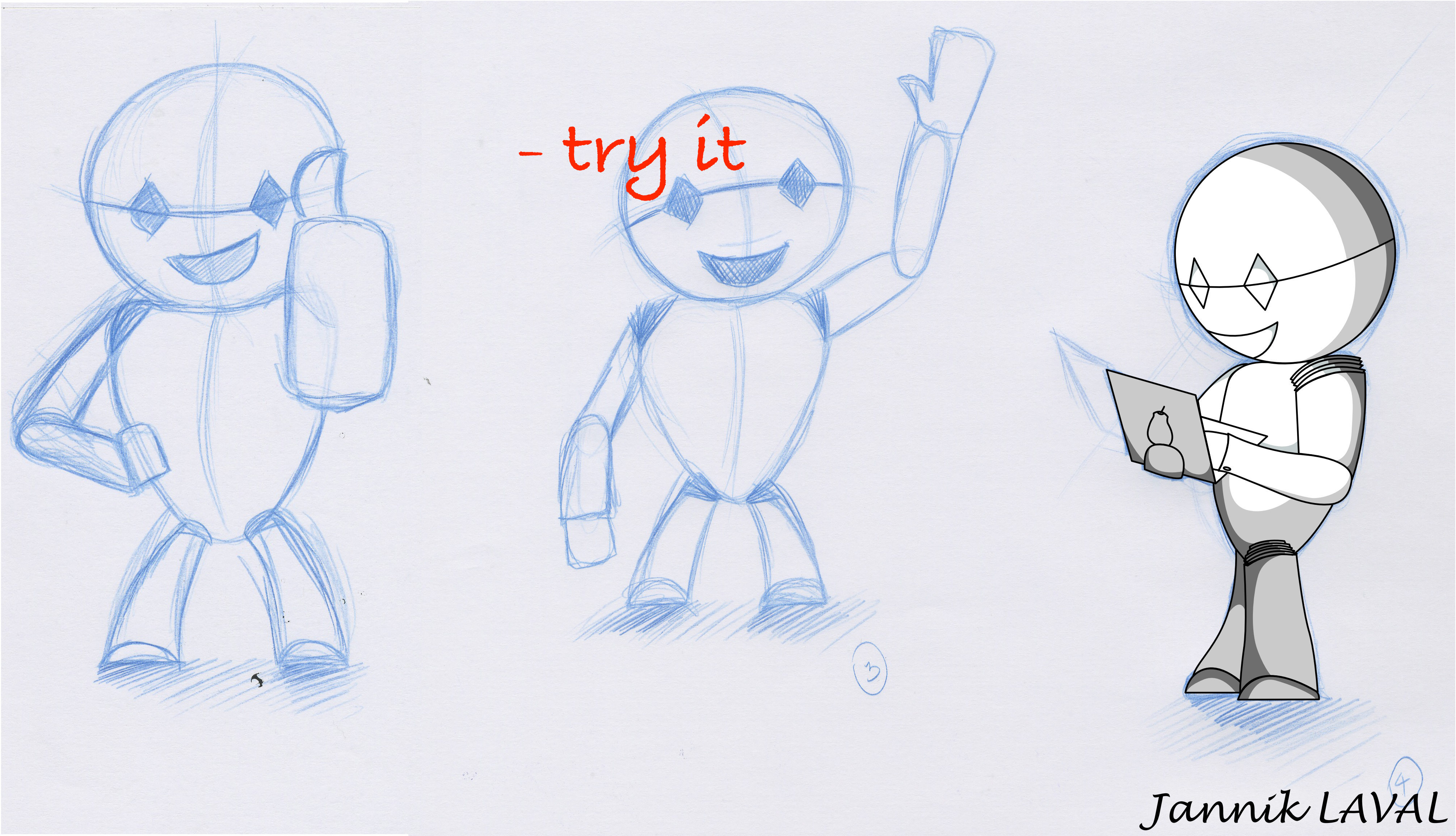
We need you



Jannik LAVAL

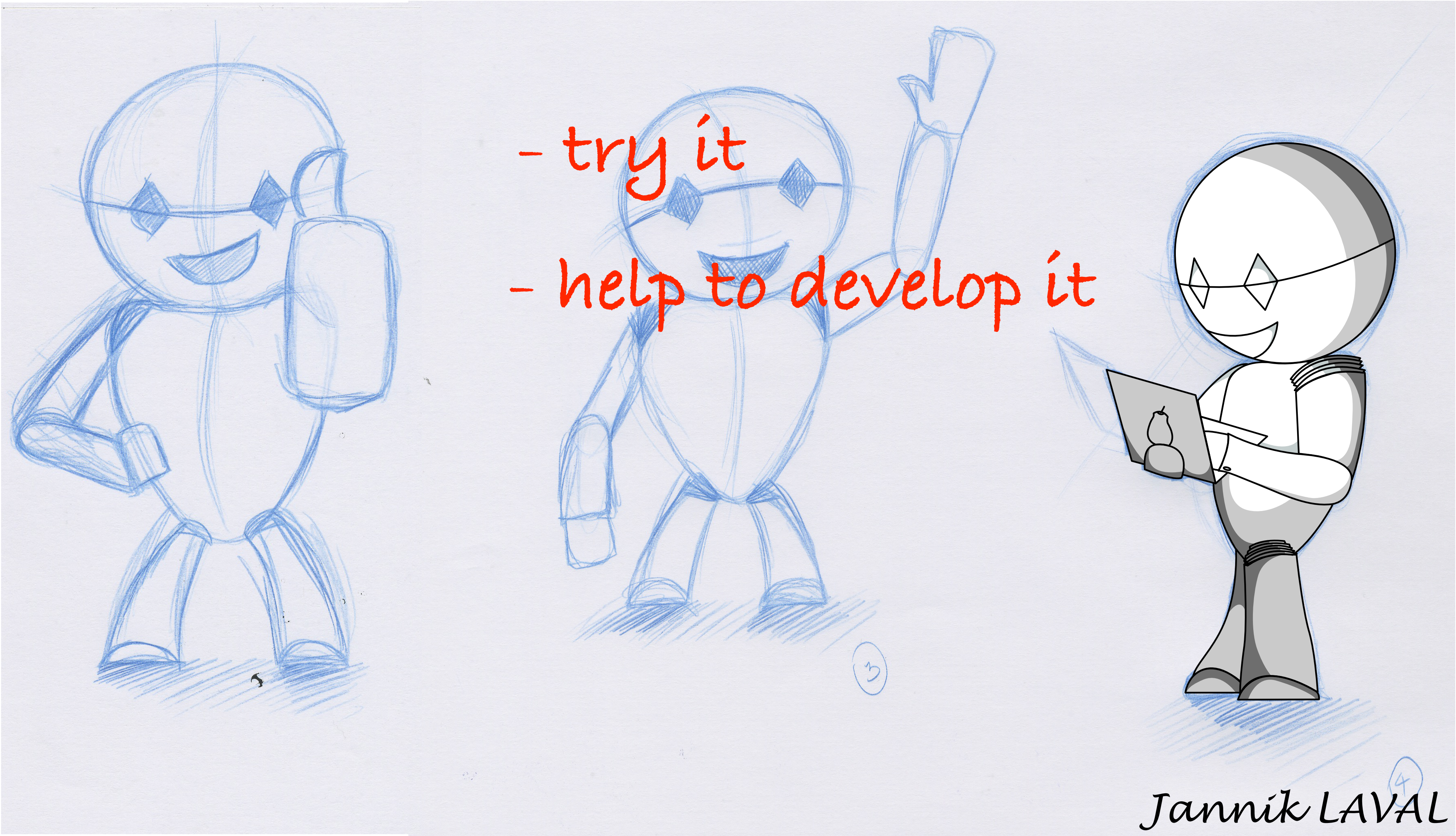
http://www.phratch.com

We need you



http://www.phratch.com

We need you

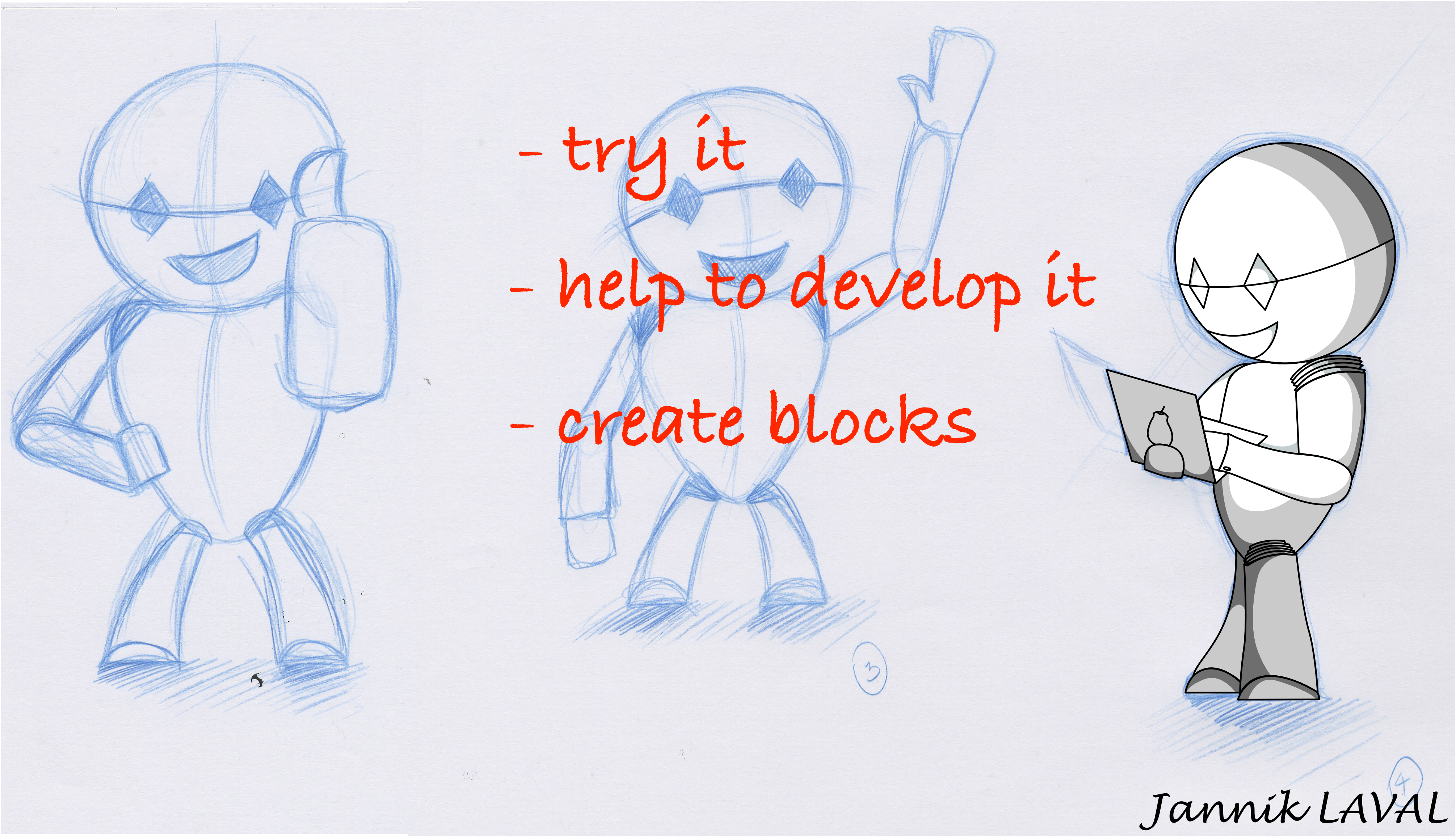


- try it
- help to develop it

Jannik LAVAL

http://www.phratch.com

We need you

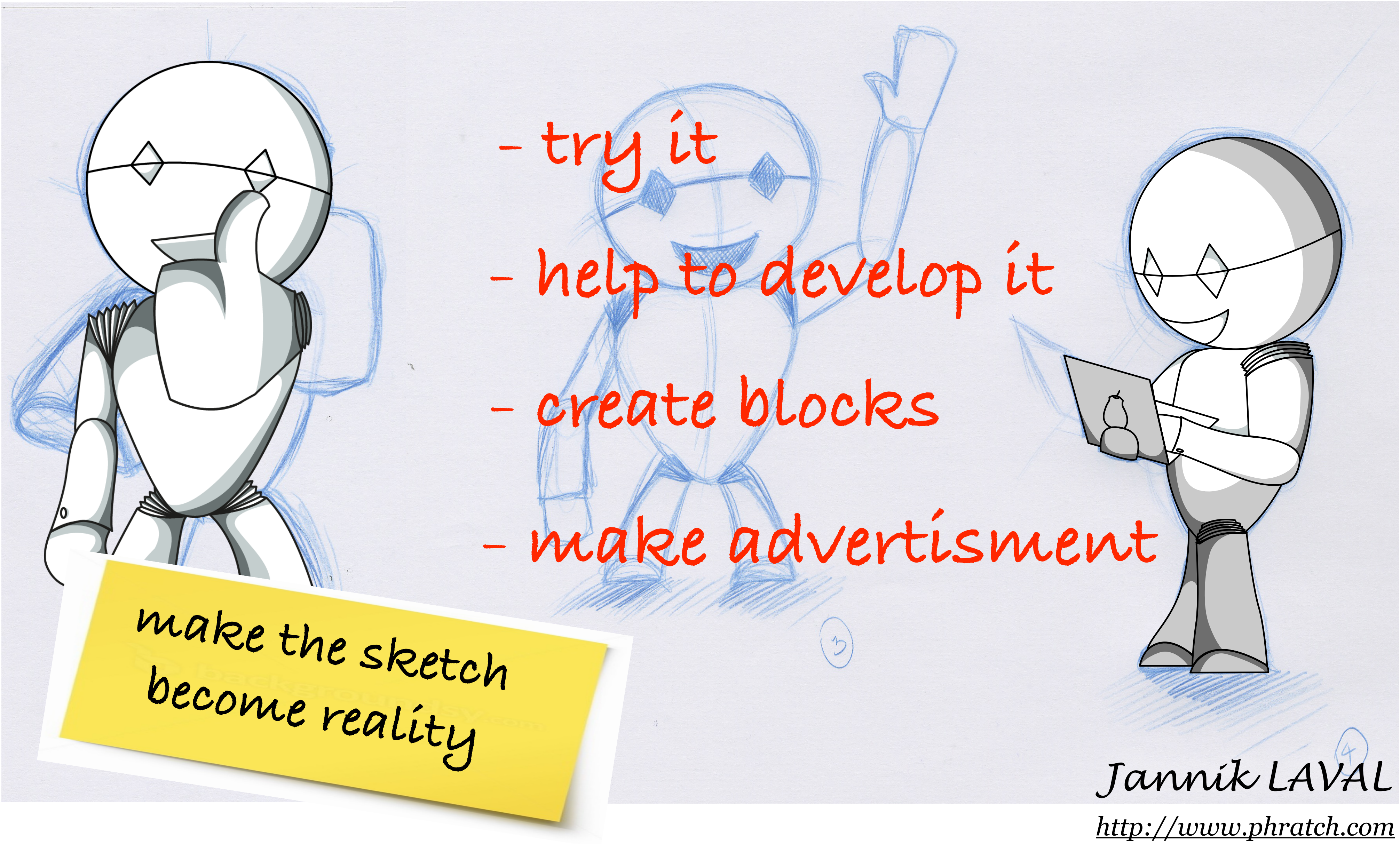


- try it
- help to develop it
- create blocks

Jannik LAVAL

http://www.phratch.com

We need you



- try it
- help to develop it
- create blocks
- make advertisment

make the sketch become reality

Jannik LAVAL

http://www.phratch.com