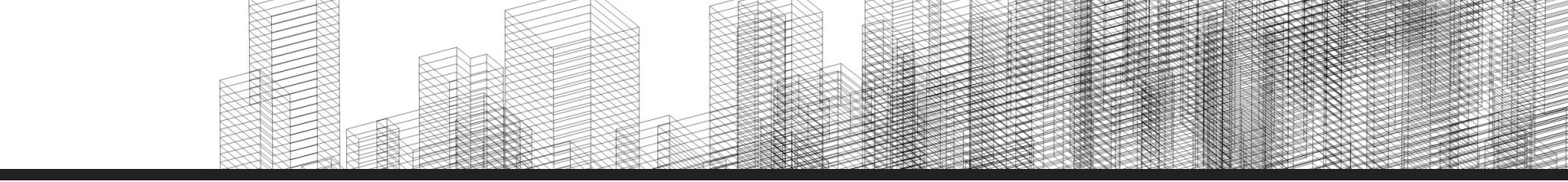


Computer Science in Early Education

Jaydon Smith
Division of Science and Mathematics
University of Minnesota, Morris, Minnesota, USA

October 31, 2020

Computer Science

Was there a lot of opportunities in school before college to study Computer Science?

Computer Science

Was there a lot of opportunities in school before college to study Computer Science?

Have you noticed how people perceive Computer Science to be such an advanced and difficult field?

The Problem

Can younger students comprehend computer science concepts?

The Problem

Can younger students comprehend computer science concepts?

What methods are available to teach these students computer science concepts?

The Problem

Can younger students comprehend computer science concepts?

What methods are available to teach these students computer science concepts?

Are there limitations to being able to teach these students?

Exploring the Problem

Computer science concepts and methods

Two methods:

- Using Scratch and metaphors (Students aged 9 - 12)
- Using ScratchJr (Students aged 5 - 8)

Professional training camps for teachers

Outline

1. Scratch and ScratchJr
2. Computational Thinking
3. Methods and their results
4. Advantages
5. Limitations
6. Conclusion

Scratch and ScratchJr

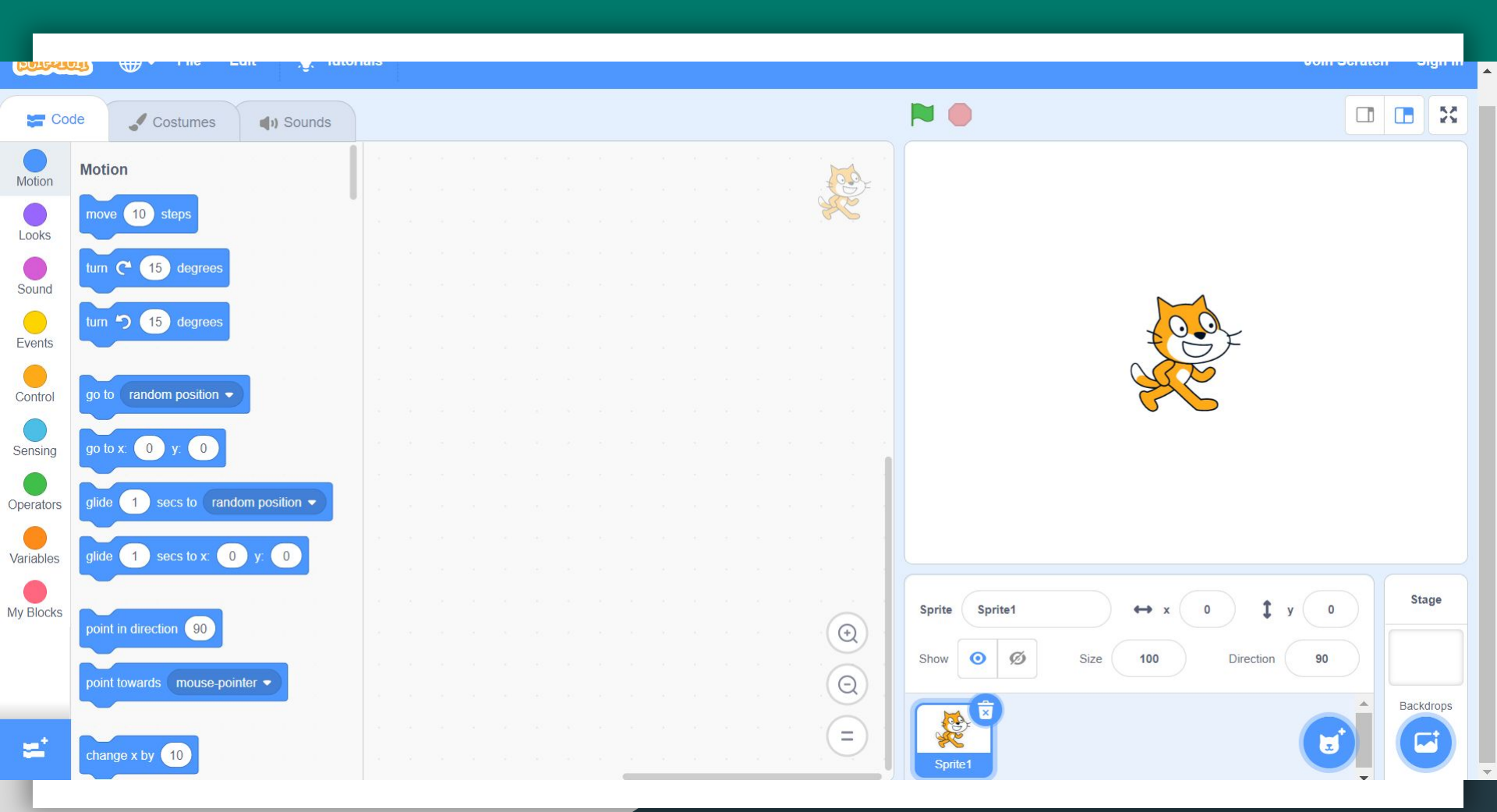
Scratch and ScratchJr

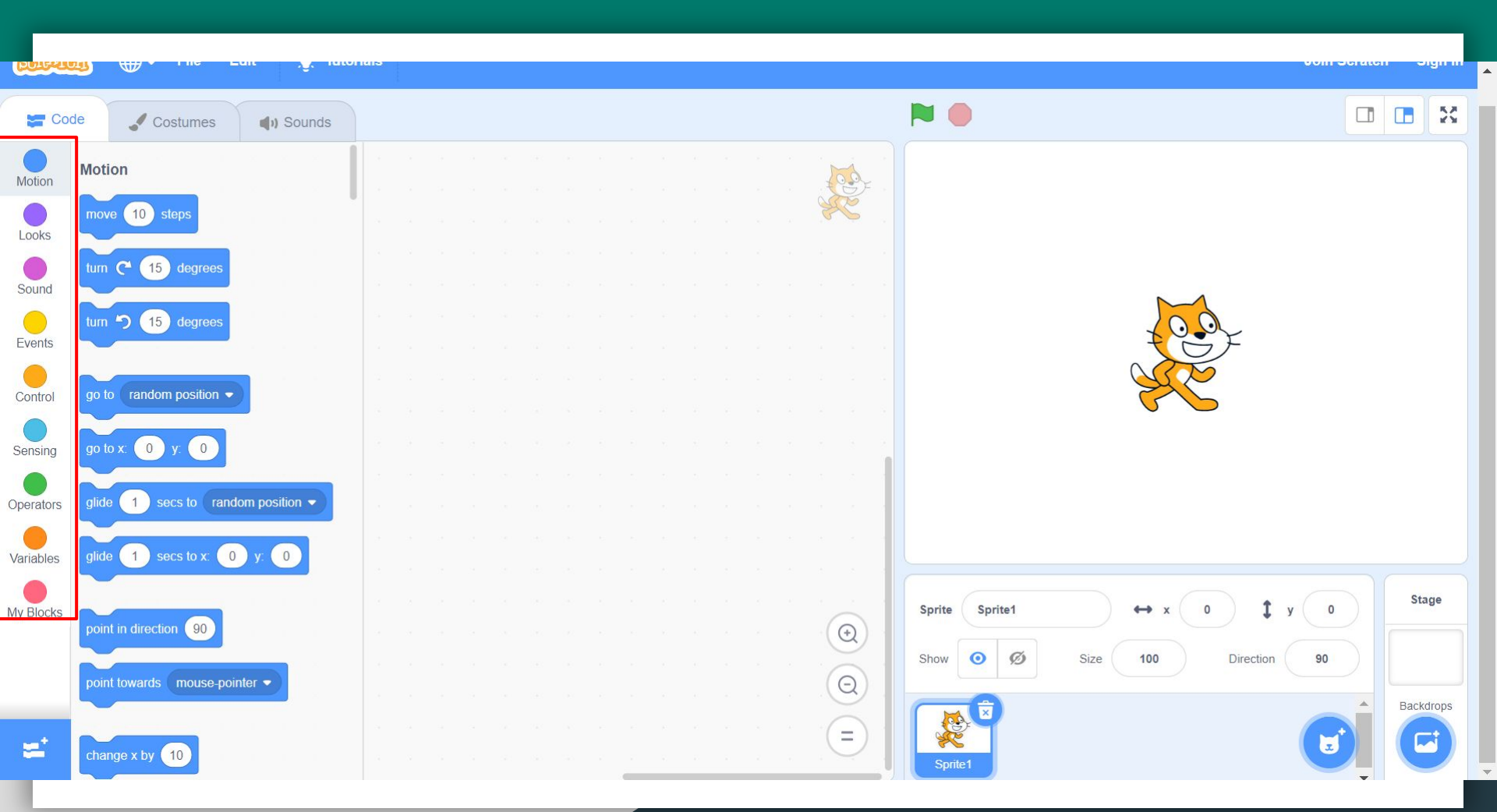
Similarities:

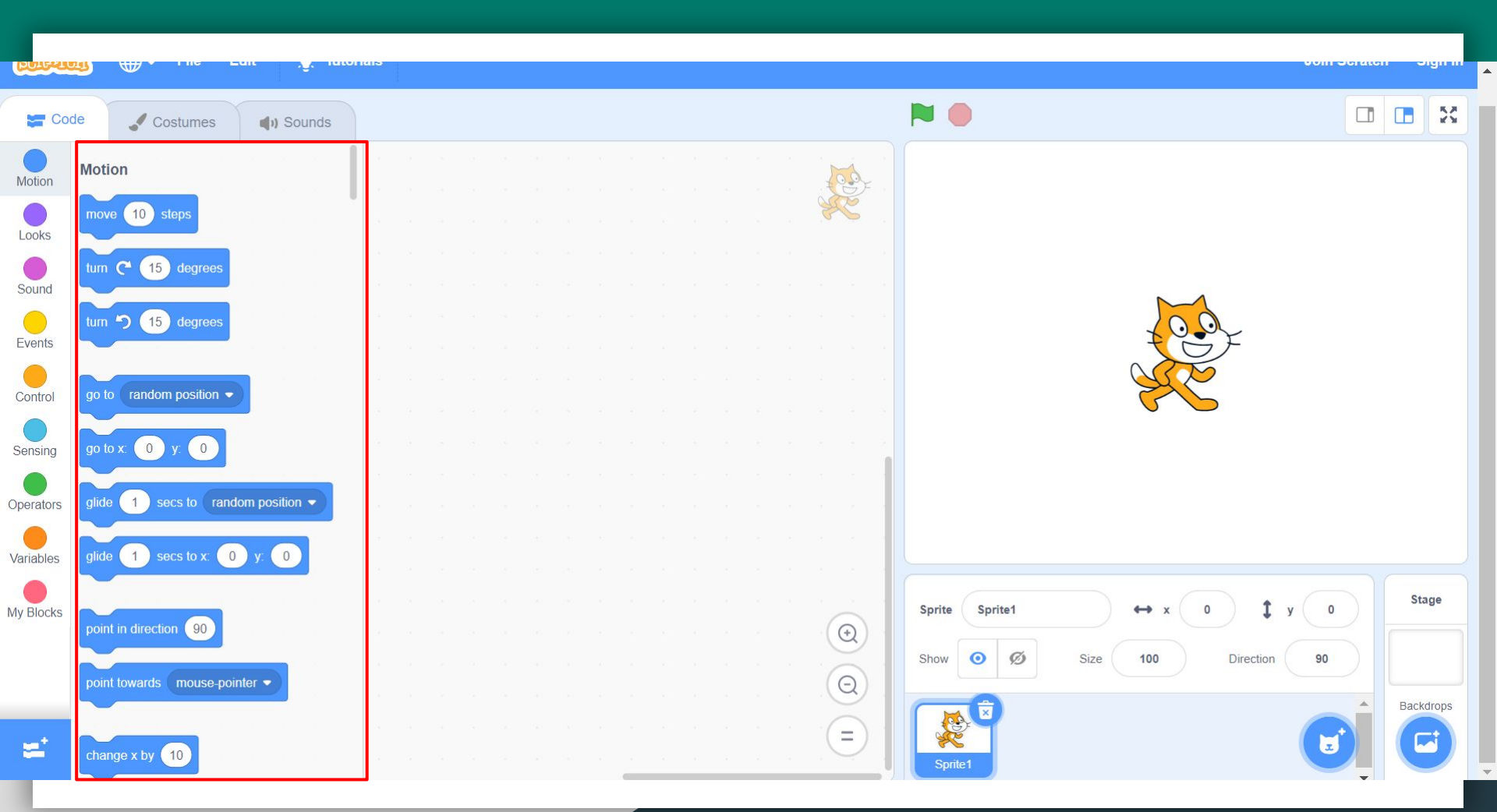
- Simple user interface
- Command blocks instead of code
- Sprites on a “stage”
- Creativity

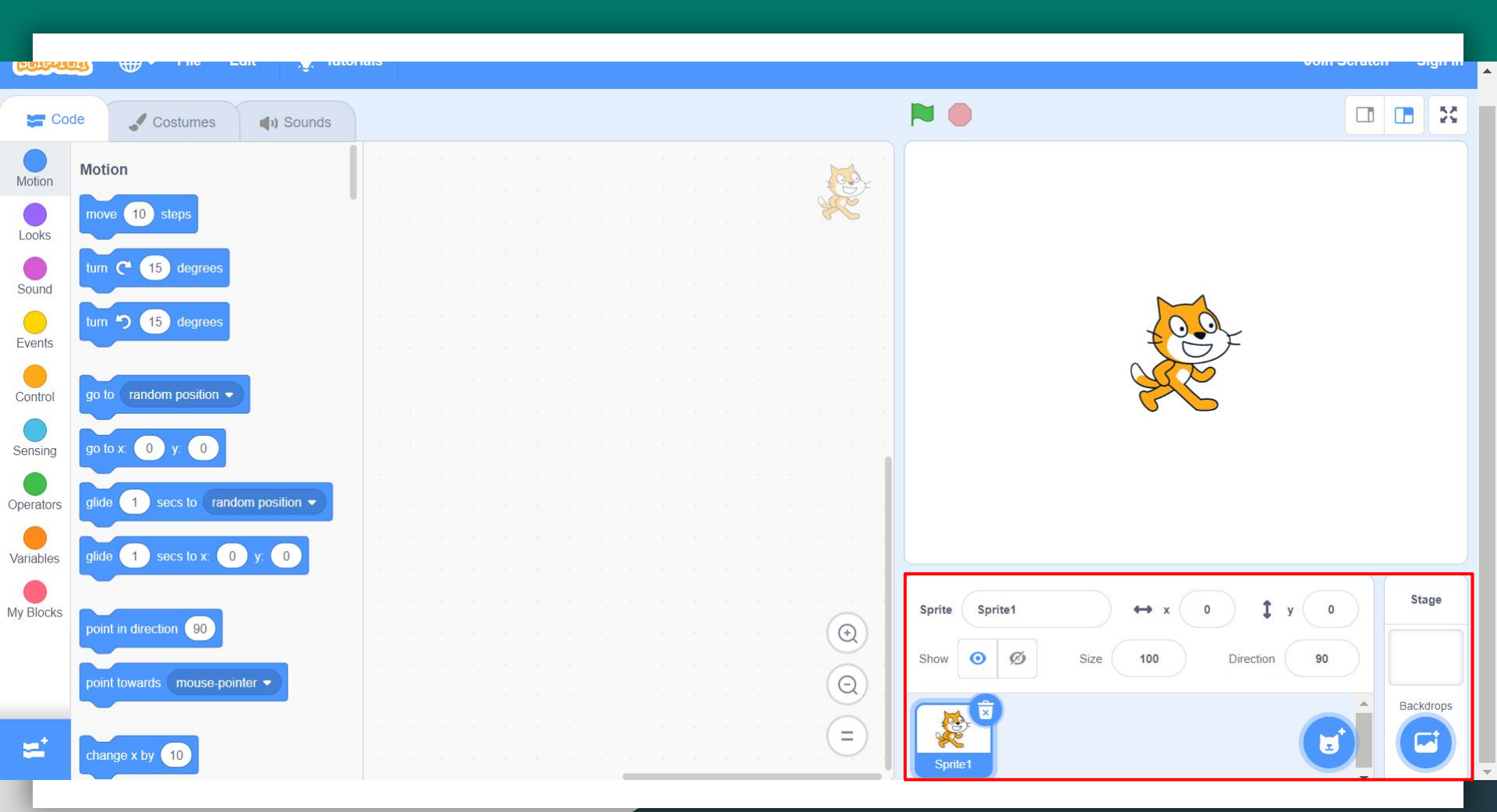
Differences:

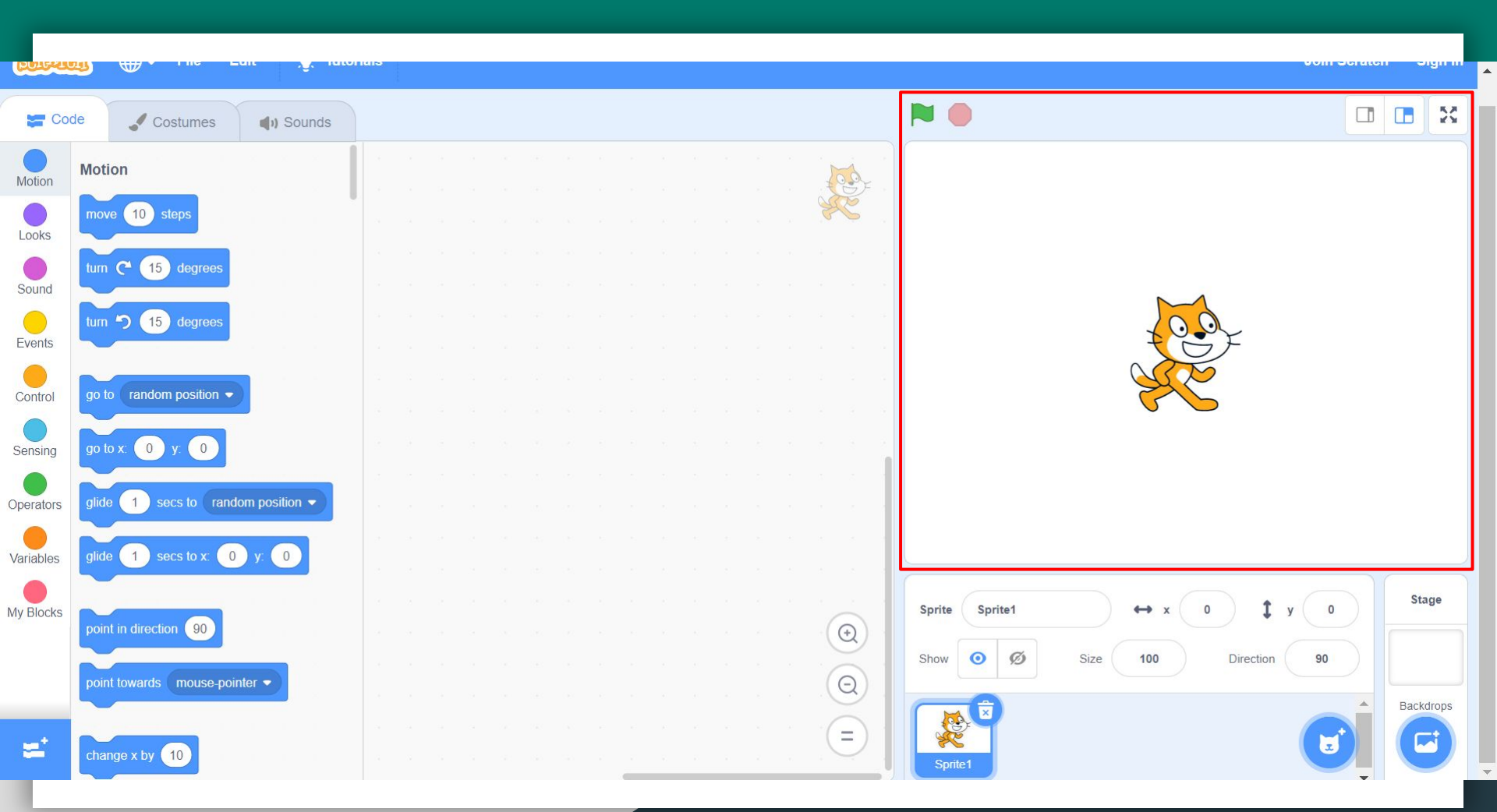
- ScratchJr has an even simpler user interface
- ScratchJr uses image blocks for command blocks
- ScratchJr has less features

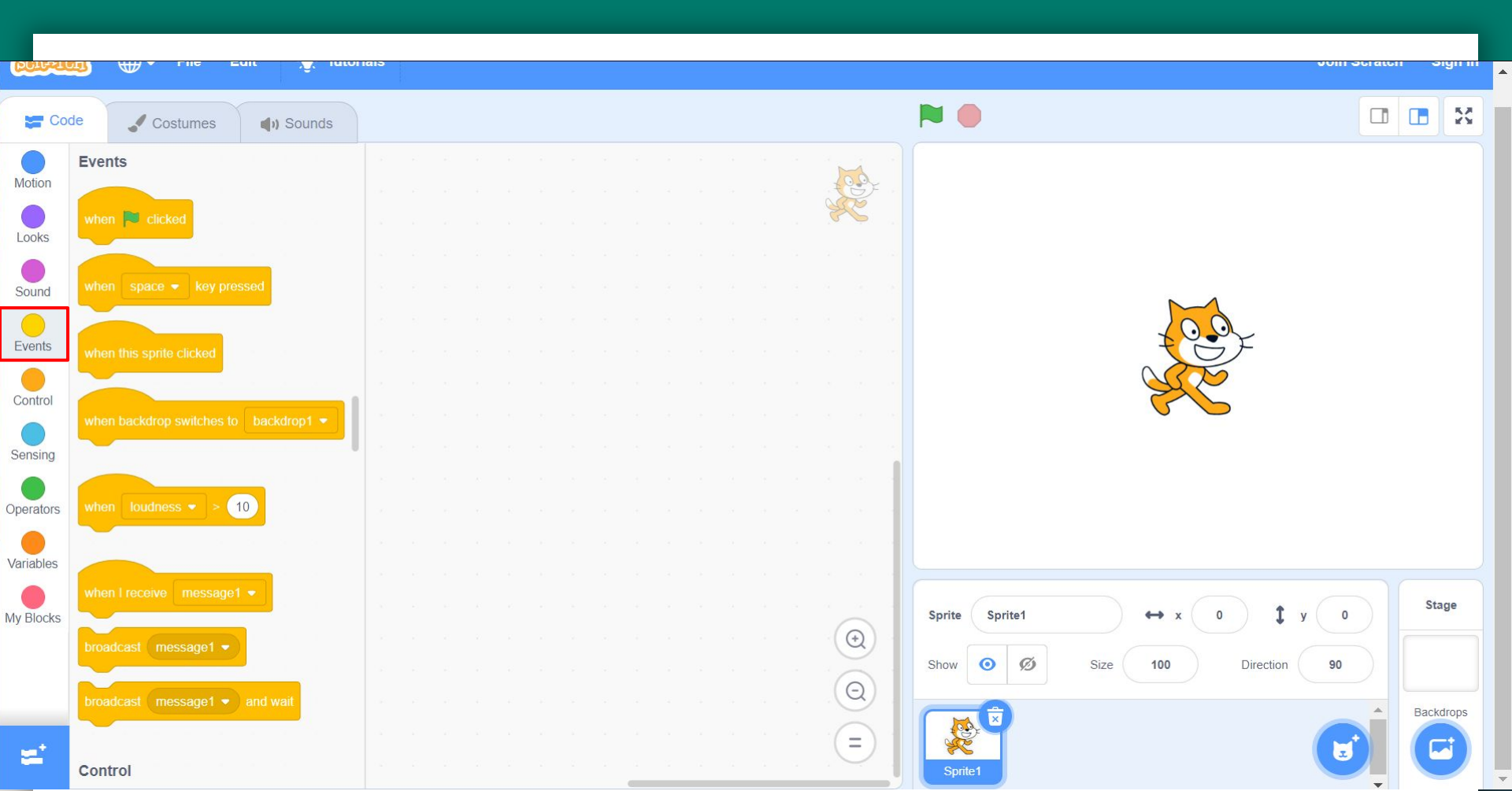

File
Edit
Tutorials
Join Scratch
Sign In
Code
Costumes
Sounds
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
Motion
move 10 steps
turn 15 degrees
turn 15 degrees
go to random position
go to x: 0 y: 0
glide 1 secs to random position
glide 1 secs to x: 0 y: 0
point in direction 90
point towards mouse-pointer
change x by 10
Sprite
Sprite1
x 0
y 0
Show
Size 100
Direction 90
Stage
Backdrops
Sprite1

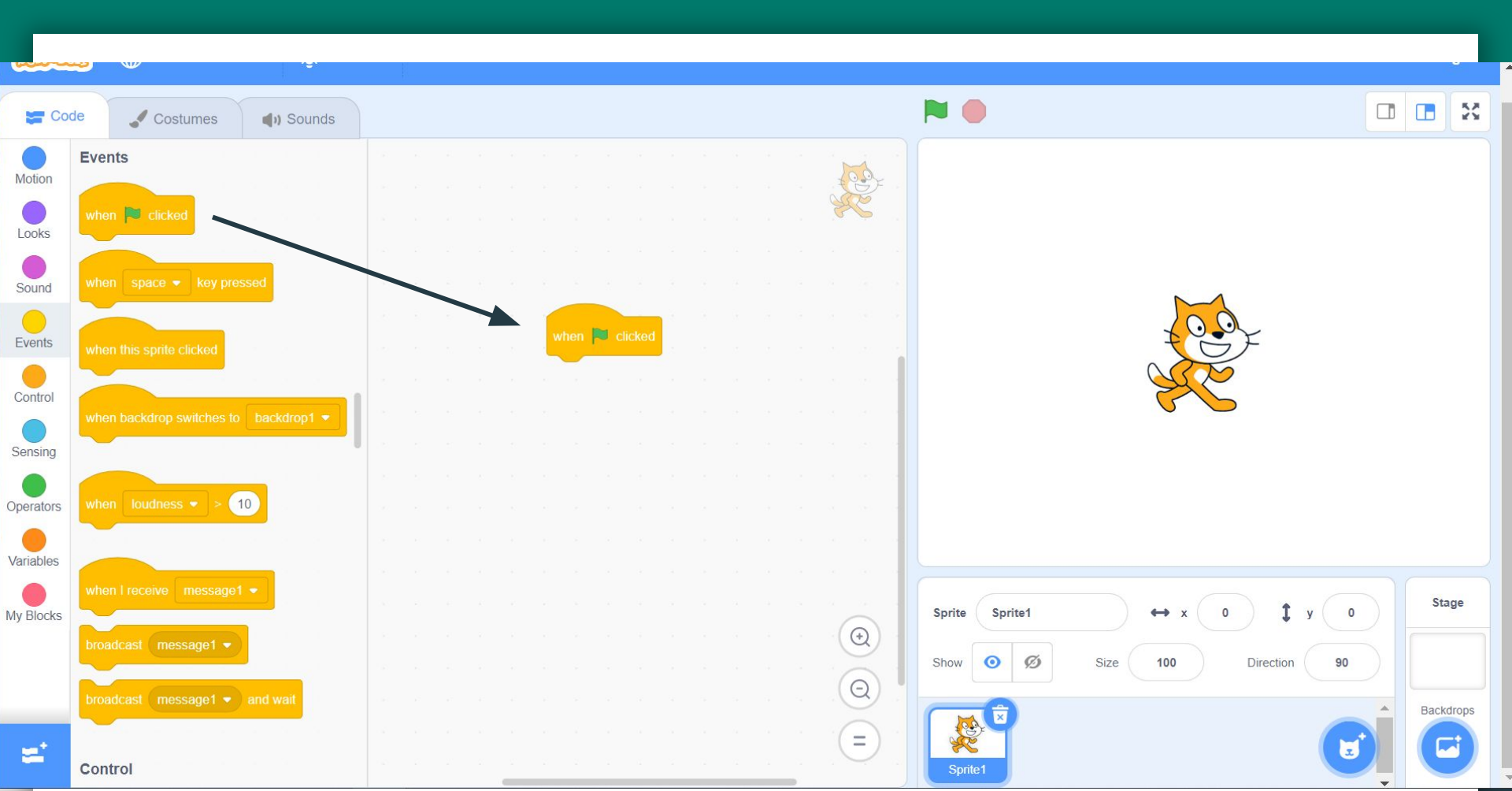

File
Edit
Tutorials
Join Scratch
Sign In
Code
Costumes
Sounds
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
Motion
move 10 steps
turn 15 degrees
turn 15 degrees
go to random position
go to x: 0 y: 0
glide 1 secs to random position
glide 1 secs to x: 0 y: 0
point in direction 90
point towards mouse-pointer
change x by 10
Sprite
Sprite1
x 0
y 0
Show
Size 100
Direction 90
Stage
Backdrops
Sprite1

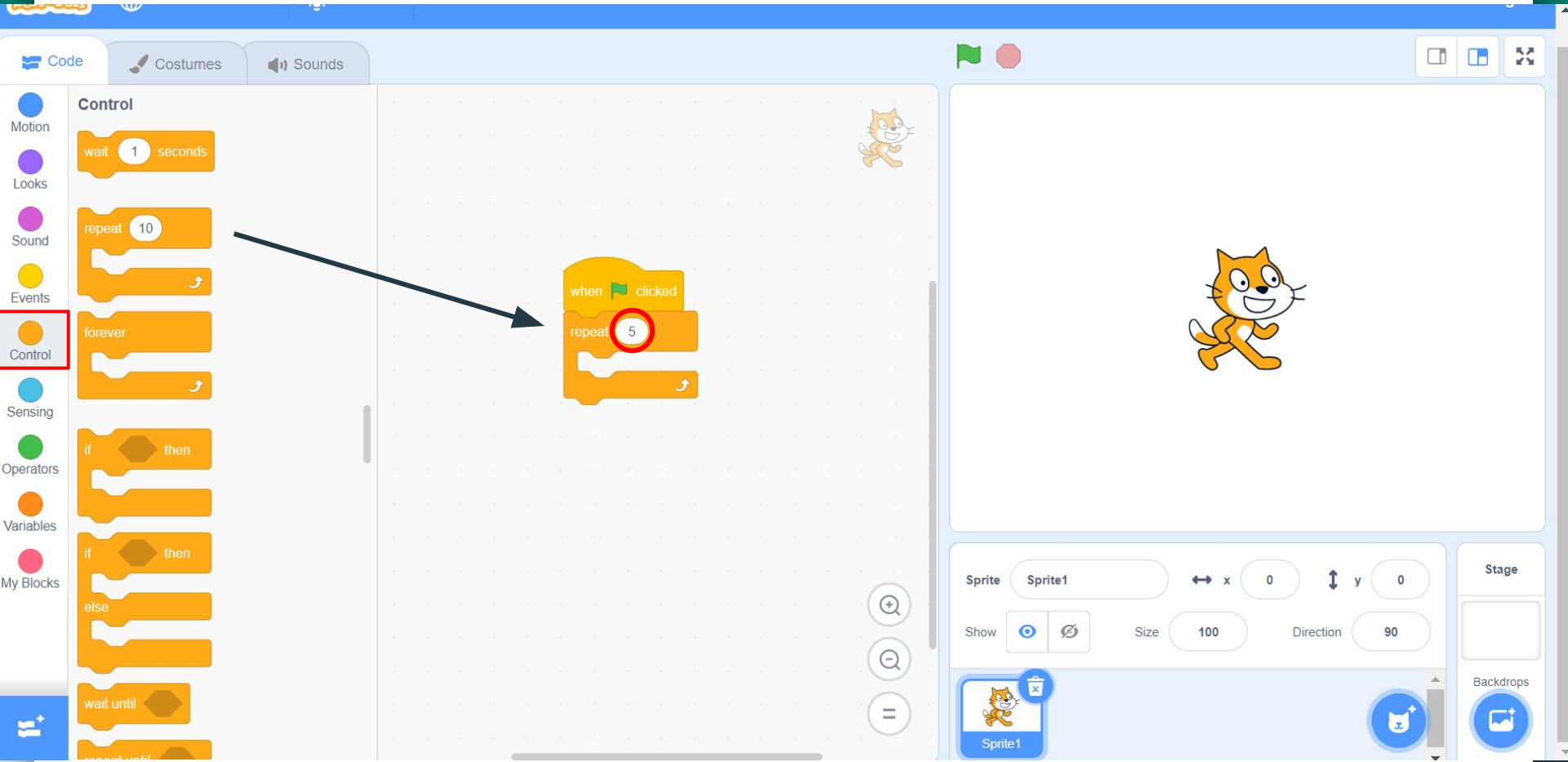

File
Edit
Tutorials
Join Scratch
Sign In
Code
Costumes
Sounds
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
Motion
move 10 steps
turn 15 degrees
turn 15 degrees
go to random position
go to x: 0 y: 0
glide 1 secs to random position
glide 1 secs to x: 0 y: 0
point in direction 90
point towards mouse-pointer
change x by 10
Sprite
Sprite1
x 0
y 0
Show
Size 100
Direction 90
Stage
Backdrops
Sprite1

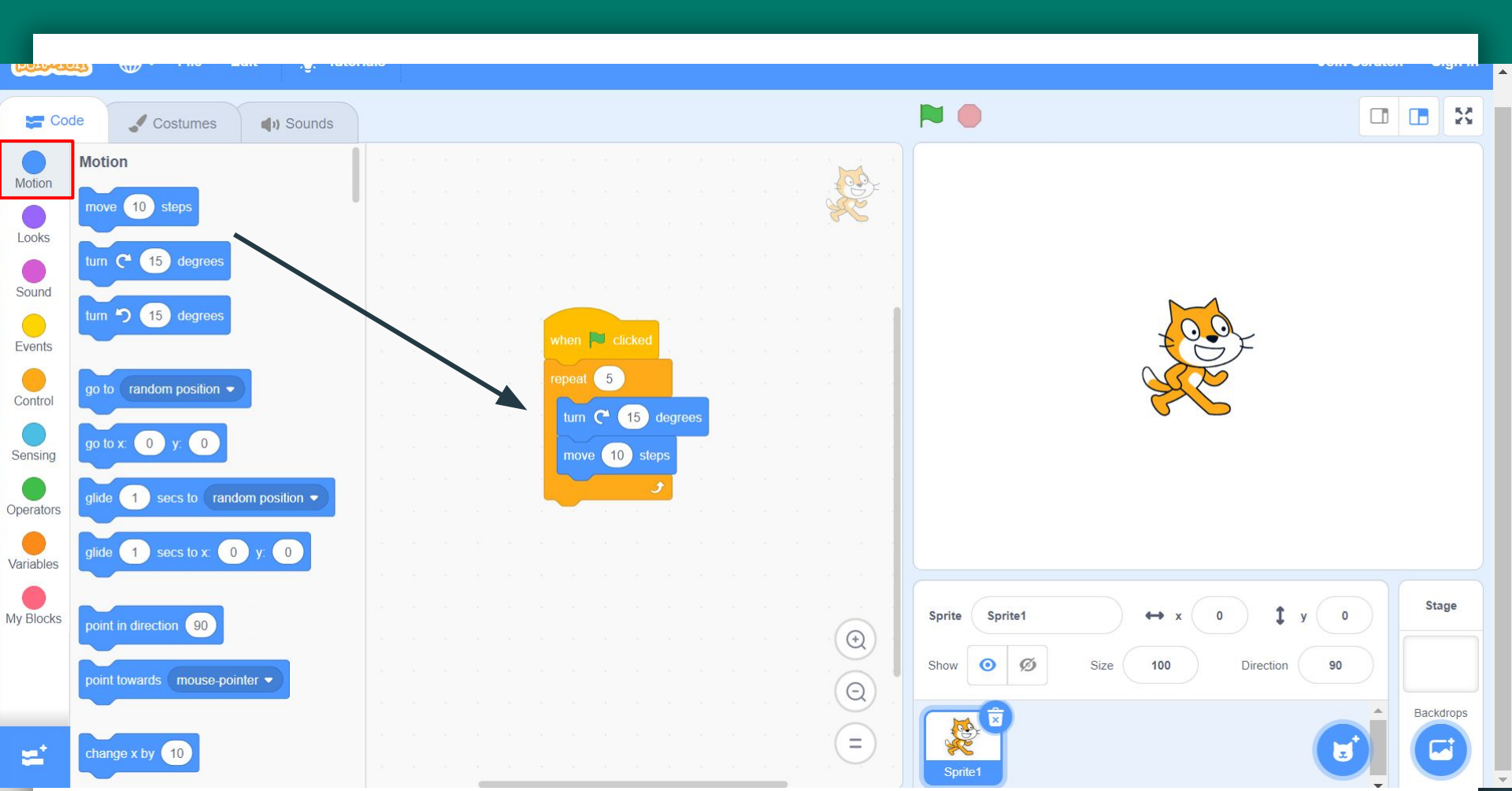

File
Edit
Tutorials
Join Scratch
Sign In
Code
Costumes
Sounds
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
Motion
move 10 steps
turn 15 degrees
turn 15 degrees
go to random position
go to x: 0 y: 0
glide 1 secs to random position
glide 1 secs to x: 0 y: 0
point in direction 90
point towards mouse-pointer
change x by 10
Sprite
Sprite1
x 0
y 0
Show
Size 100
Direction 90
Stage
Backdrops
Sprite1

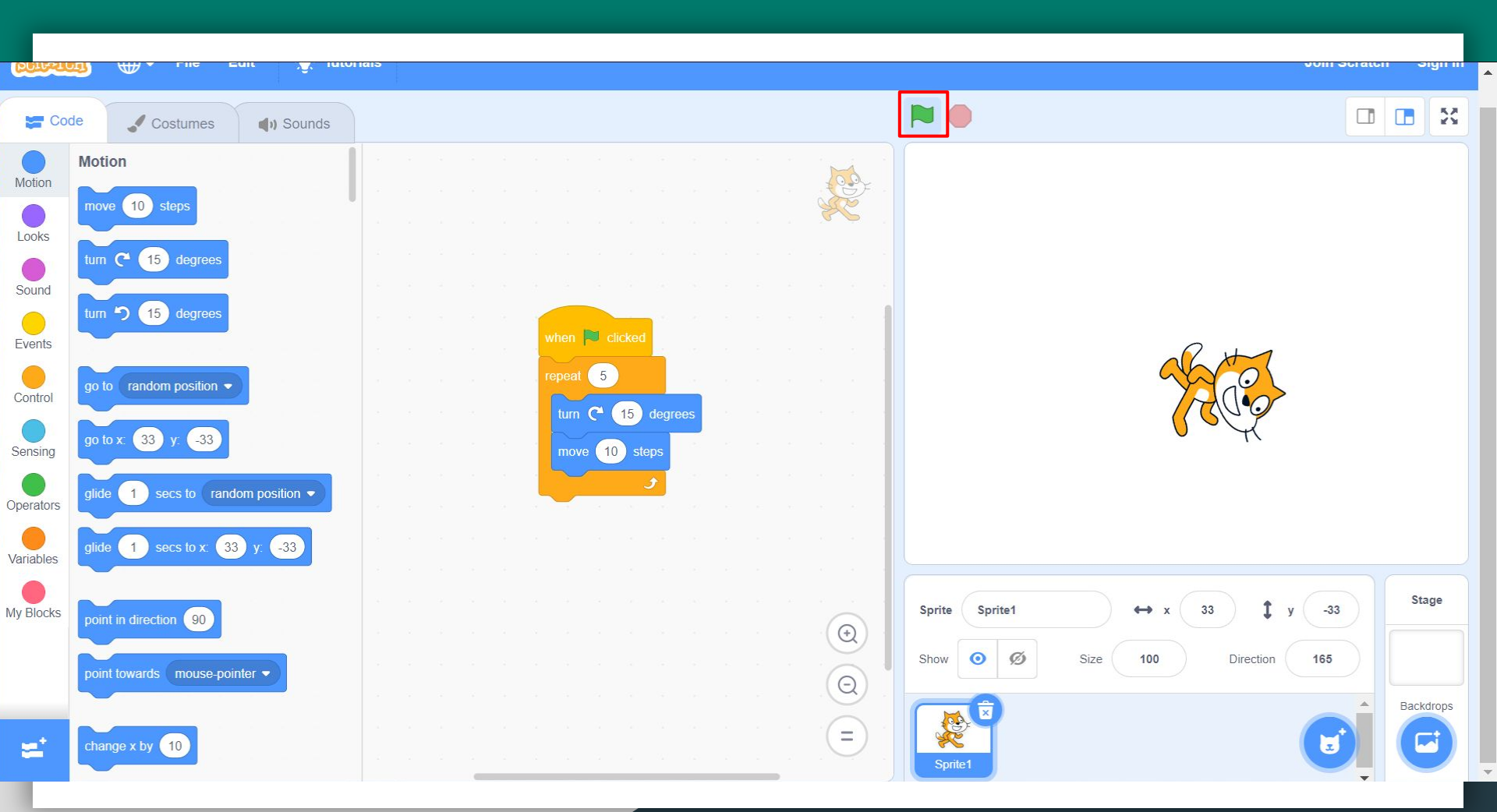

File
Edit
Tutorials
Join Scratch
Sign In
Code
Costumes
Sounds
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
Motion
move 10 steps
turn 15 degrees
turn 15 degrees
go to random position
go to x: 0 y: 0
glide 1 secs to random position
glide 1 secs to x: 0 y: 0
point in direction 90
point towards mouse-pointer
change x by 10
Sprite
Sprite1
x 0
y 0
Show
Size 100
Direction 90
Stage
Backdrops
Sprite1


File
Edit
Tutorials
Join Scratch
Sign In
Code
Costumes
Sounds
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
Events
when clicked
when space key pressed
when this sprite clicked
when backdrop switches to backdrop1
when loudness > 10
when I receive message1
broadcast message1
broadcast message1 and wait
Control
Sprite
Sprite1
x
0
y
0
Show
Size
100
Direction
90
Stage
Backdrops
Sprite1


Code
Costumes
Sounds
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
Events
when clicked
when space key pressed
when this sprite clicked
when backdrop switches to backdrop1
when loudness > 10
when I receive message1
broadcast message1
broadcast message1 and wait
Control
when clicked
Sprite
Sprite1
x 0
y 0
Show
Size 100
Direction 90
Stage
Backdrops
Sprite1


Code
Costumes
Sounds
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
Control
wait 1 seconds
repeat 10
forever
if then
if then
else
wait until
when clicked
repeat 5
Sprite
Sprite1
x 0
y 0
Show
Size 100
Direction 90
Stage
Backdrops
Sprite1


Code
Costumes
Sounds
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
Motion
move 10 steps
turn 15 degrees
turn 15 degrees
go to random position
go to x: 0 y: 0
glide 1 secs to random position
glide 1 secs to x: 0 y: 0
point in direction 90
point towards mouse-pointer
change x by 10
when clicked
repeat 5
turn 15 degrees
move 10 steps
Sprite
Sprite1
x 0
y 0
Show
Size 100
Direction 90
Stage
Backdrops
Sprite1


File
Edit
Tutorials
Join Scratch
Sign In
Code
Costumes
Sounds
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
Motion
move 10 steps
turn 15 degrees
turn 15 degrees
go to random position
go to x: 33 y: -33
glide 1 secs to random position
glide 1 secs to x: 33 y: -33
point in direction 90
point towards mouse-pointer
change x by 10
when clicked
repeat 5
turn 15 degrees
move 10 steps
Sprite
Sprite1
x
33
y
-33
Show
Size
100
Direction
165
Stage
Backdrops
Sprite1

ScratchJr

ScratchJr

ScratchJr

ScratchJr

ScratchJr

Computational Thinking

Computational Thinking

Considered the skill of solving problems, designing systems, and understanding human behavior based on computer science concepts.

Positive evidence answers first problem of comprehension.

Two methods

Methods using Scratch and ScratchJr

Scratch and Metaphors

MECOPROG

Example: Loops with hand mixer, conditionals with intelligent fridge

132 elementary students

ROMT, CONT, and PCNT tests

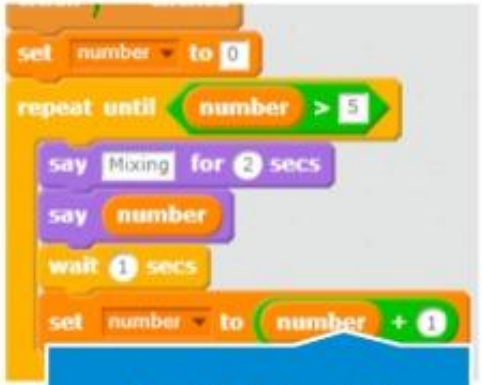
Block 1 Introduction (program,...,I/O)

Block 2 Intelligent fridge (conditionals)

Block 3 Hand mixer (loops)

Block 1 Scratch

Block 2 Scratch



Block 3 Scratch

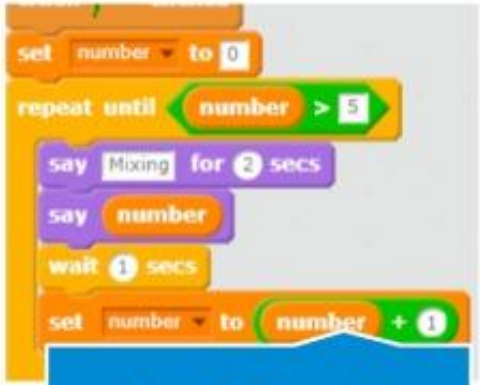
Block 1 Introduction (program,...,I/O)

Block 2 Intelligent fridge (conditionals)

Block 3 Hand mixer (loops)

Block 1 Scratch

Block 2 Scratch



Block 3 Scratch

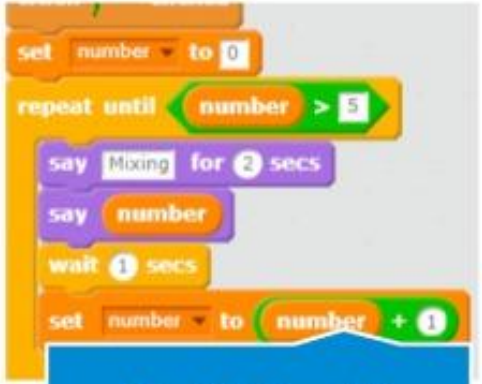
Block 1 Introduction (program,...,I/O)

Block 2 Intelligent fridge (conditionals)

Block 3 Hand mixer (loops)

Block 1 Scratch

Block 2 Scratch



Block 3 Scratch

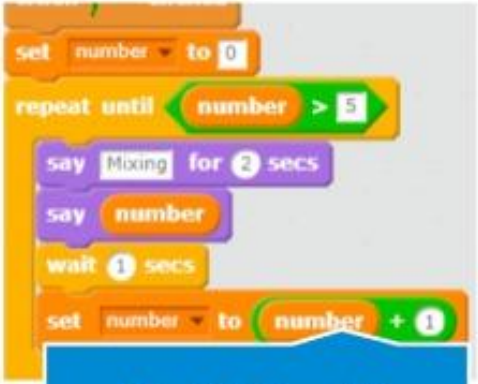
Block 1 Introduction (program,...,I/O)

Block 2 Intelligent fridge (conditionals)

Block 3 Hand mixer (loops)

Block 1 Scratch

Block 2 Scratch



Block 3 Scratch

Study 1 Results

	PCN			CON			ROM		
	Mdn	M	SD	Mdn	M	SD	Mdn	M	SD
Pre	8.57	8.37	1.25	2.69	2.77	1.32	4.28	4.23	1.36
Post	9.28	8.99	1.05	5	5.08	1.59	4.64	4.77	1.56

Study 1 Results

	PCN			CON			ROM		
	Mdn	M	SD	Mdn	M	SD	Mdn	M	SD
Pre	8.57	8.37	1.25	2.69	2.77	1.32	4.28	4.23	1.36
Post	9.28	8.99	1.05	5	5.08	1.59	4.64	4.77	1.56

ScratchJr and Solve Its

57 kindergarten through 2nd grade students

Lessons introduced computer science ideas

Three modules: Interactive collage, animated story, and interactive game

Solve Its

Measured: observation, memory, and reasoning

Recorded errors made

Study 2 results

Errors:

- Kindergarteners: 3 per question
- First Graders: 1.8 per question
- Second Graders: 1.4 per question

Study 2 results cont.

Solve It Task 2: Two characters take turns doing actions

Kindergarteners couldn’t solve

13% of first graders found solution

40% of second graders found solution

Higher grade = better problem grasping and solving

Advantages

Advantages

Early introduction helps:

- Cognitive skills
- Visual memory
- Language skills
- Manage uncertainty
- Assess problem difficulty
- Use of modularization

- Easier time with advanced courses later on
- More knowledgeable in careers

Study 3

2,871 introductory college computer science students

Survey students about content and pedagogy of computer science courses before college

College professors shared final grades with researchers

	Model 1: Controls		Model 2: Main Effects		Model 3: Interactions	
	b	(se)	b	(se)	b	(se)
Intercept	77.55***	(1.53)	78.48***	(1.56)	80.57***	(1.71)
Student-level variables						
Avg. parent education	0.39	(0.22)	0.34	(0.22)	0.33	(0.22)
Help at home	0.38	(0.65)	0.33	(0.65)	0.46	(0.65)
Vocabulary	−0.04	(0.21)	−0.11	(0.21)	0.23	(0.25)
Male	−1.06*	(0.50)	−1.19*	(0.50)	−1.22*	(0.49)
Hispanic	−1.08	(0.73)	−1.11	(0.73)	−1.02	(0.73)
Black	−5.01***	(0.87)	−4.95***	(0.87)	−4.82***	(0.87)
Asian	−0.67	(0.57)	−0.60	(0.57)	−0.68	(0.56)
Other race	−1.59*	(0.75)	−1.47*	(0.74)	−1.61*	(0.74)
SAT Math/100	1.16***	(0.24)	1.07***	(0.24)	0.74**	(0.27)
Freshman	1.41**	(0.47)	1.13*	(0.47)	1.09*	(0.47)
Coding			0.88***	(0.24)	1.05***	(0.25)
Non-coding computer use			−0.67**	(0.22)	−0.65**	(0.22)
Class-level variable						
Innovative college CS	1.75	(0.91)	1.81*	(0.91)	−7.28*	(3.04)
Interactions						
Help at home X coding					−1.85**	(0.67)
Innovative X SAT math/100					1.46**	(0.47)
Innovative X vocabulary					−1.22**	(0.46)
N (classes)		177		177		177
N (students)		2871		2871		2871
Pseudo-R^2		9.0%		10.0%		11.0%

Note: Significant *p*-values indicated by * ($p \leq .05$), ** ($p \leq .01$), and *** ($p \leq .001$). Parentheses indicate standard errors.

	Model 1: Controls		Model 2: Main Effects		Model 3: Interactions	
	b	(se)	b	(se)	b	(se)
Intercept	77.55***	(1.53)	78.48***	(1.56)	80.57***	(1.71)
Student-level variables						
Avg. parent education	0.39	(0.22)	0.34	(0.22)	0.33	(0.22)
Help at home	0.38	(0.65)	0.33	(0.65)	0.46	(0.65)
Vocabulary	−0.04	(0.21)	−0.11	(0.21)	0.23	(0.25)
Male	−1.06*	(0.50)	−1.19*	(0.50)	−1.22*	(0.49)
Hispanic	−1.08	(0.73)	−1.11	(0.73)	−1.02	(0.73)
Black	−5.01***	(0.87)	−4.95***	(0.87)	−4.82***	(0.87)
Asian	−0.67	(0.57)	−0.60	(0.57)	−0.68	(0.56)
Other race	−1.59*	(0.75)	−1.47*	(0.74)	−1.61*	(0.74)
SAT Math/100	1.16***	(0.24)	1.07***	(0.24)	0.74**	(0.27)
Freshman	1.41**	(0.47)	1.13*	(0.47)	1.09*	(0.47)
Coding			0.88***	(0.24)	1.05***	(0.25)
Non-coding computer use			−0.67**	(0.22)	−0.65**	(0.22)
Class-level variable						
Innovative college CS	1.75	(0.91)	1.81*	(0.91)	−7.28*	(3.04)
Interactions						
Help at home X coding					−1.85**	(0.67)
Innovative X SAT math/100					1.46**	(0.47)
Innovative X vocabulary					−1.22**	(0.46)
N (classes)		177		177		177
N (students)		2871		2871		2871
Pseudo-R^2		9.0%		10.0%		11.0%

Note: Significant *p*-values indicated by * ($p \leq .05$), ** ($p \leq .01$), and *** ($p \leq .001$). Parentheses indicate standard errors.

	Model 1: Controls		Model 2: Main Effects		Model 3: Interactions	
	b	(se)	b	(se)	b	(se)
Intercept	77.55***	(1.53)	78.48***	(1.56)	80.57***	(1.71)
Student-level variables						
Avg. parent education	0.39	(0.22)	0.34	(0.22)	0.33	(0.22)
Help at home	0.38	(0.65)	0.33	(0.65)	0.46	(0.65)
Vocabulary	−0.04	(0.21)	−0.11	(0.21)	0.23	(0.25)
Male	−1.06*	(0.50)	−1.19*	(0.50)	−1.22*	(0.49)
Hispanic	−1.08	(0.73)	−1.11	(0.73)	−1.02	(0.73)
Black	−5.01***	(0.87)	−4.95***	(0.87)	−4.82***	(0.87)
Asian	−0.67	(0.57)	−0.60	(0.57)	−0.68	(0.56)
Other race	−1.59*	(0.75)	−1.47*	(0.74)	−1.61*	(0.74)
SAT Math/100	1.16***	(0.24)	1.07***	(0.24)	0.74**	(0.27)
Freshman	1.41**	(0.47)	1.13*	(0.47)	1.09*	(0.47)
Coding			0.88***	(0.24)	1.05***	(0.25)
Non-coding computer use			−0.67**	(0.22)	−0.65**	(0.22)
Class-level variable						
Innovative college CS	1.75	(0.91)	1.81*	(0.91)	−7.28*	(3.04)
Interactions						
Help at home X coding					−1.85**	(0.67)
Innovative X SAT math/100					1.46**	(0.47)
Innovative X vocabulary					−1.22**	(0.46)
N (classes)		177		177		177
N (students)		2871		2871		2871
Pseudo-R^2		9.0%		10.0%		11.0%

Note: Significant *p*-values indicated by * ($p \leq .05$), ** ($p \leq .01$), and *** ($p \leq .001$). Parentheses indicate standard errors.

	Model 1: Controls		Model 2: Main Effects		Model 3: Interactions	
	b	(se)	b	(se)	b	(se)
Intercept	77.55***	(1.53)	78.48***	(1.56)	80.57***	(1.71)
Student-level variables						
Avg. parent education	0.39	(0.22)	0.34	(0.22)	0.33	(0.22)
Help at home	0.38	(0.65)	0.33	(0.65)	0.46	(0.65)
Vocabulary	−0.04	(0.21)	−0.11	(0.21)	0.23	(0.25)
Male	−1.06*	(0.50)	−1.19*	(0.50)	−1.22*	(0.49)
Hispanic	−1.08	(0.73)	−1.11	(0.73)	−1.02	(0.73)
Black	−5.01***	(0.87)	−4.95***	(0.87)	−4.82***	(0.87)
Asian	−0.67	(0.57)	−0.60	(0.57)	−0.68	(0.56)
Other race	−1.59*	(0.75)	−1.47*	(0.74)	−1.61*	(0.74)
SAT Math/100	1.16***	(0.24)	1.07***	(0.24)	0.74**	(0.27)
Freshman	1.41**	(0.47)	1.13*	(0.47)	1.09*	(0.47)
Coding			0.88***	(0.24)	1.05***	(0.25)
Non-coding computer use			−0.67**	(0.22)	−0.65**	(0.22)
Class-level variable						
Innovative college CS	1.75	(0.91)	1.81*	(0.91)	−7.28*	(3.04)
Interactions						
Help at home X coding					−1.85**	(0.67)
Innovative X SAT math/100					1.46**	(0.47)
Innovative X vocabulary					−1.22**	(0.46)
N (classes)		177		177		177
N (students)		2871		2871		2871
Pseudo-R^2		9.0%		10.0%		11.0%

Note: Significant *p*-values indicated by * ($p \leq .05$), ** ($p \leq .01$), and *** ($p \leq .001$). Parentheses indicate standard errors.

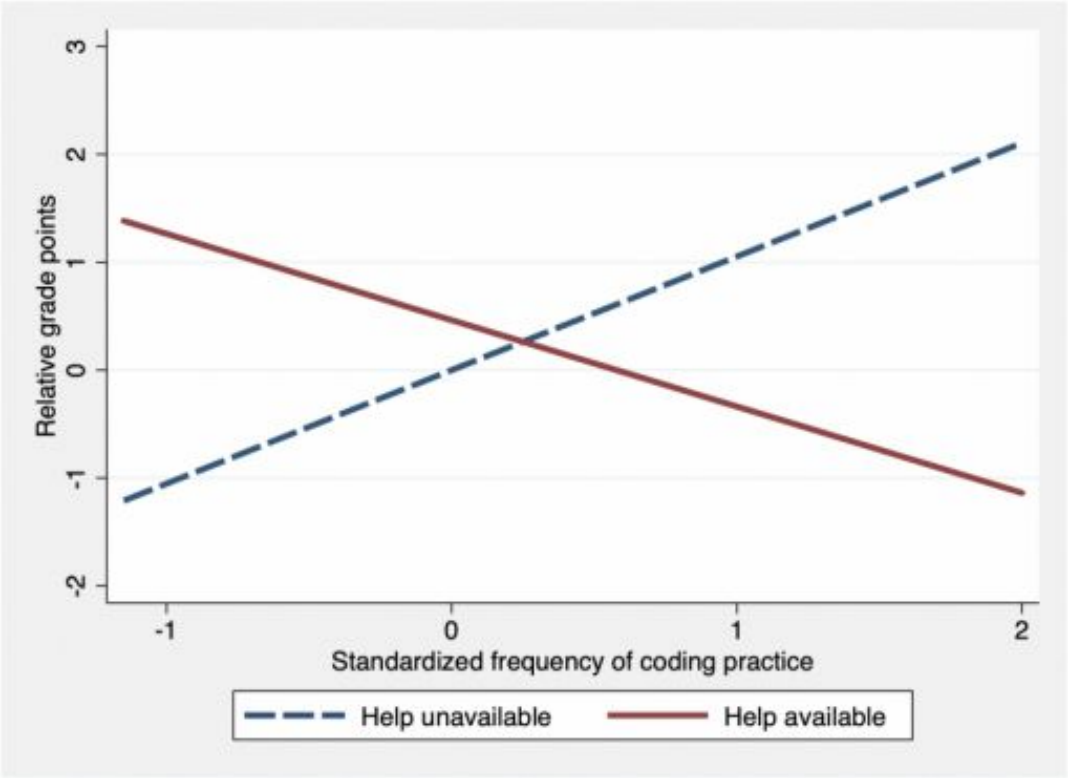
	Model 1: Controls		Model 2: Main Effects		Model 3: Interactions	
	b	(se)	b	(se)	b	(se)
Intercept	77.55***	(1.53)	78.48***	(1.56)	80.57***	(1.71)
Student-level variables						
Avg. parent education	0.39	(0.22)	0.34	(0.22)	0.33	(0.22)
Help at home	0.38	(0.65)	0.33	(0.65)	0.46	(0.65)
Vocabulary	−0.04	(0.21)	−0.11	(0.21)	0.23	(0.25)
Male	−1.06*	(0.50)	−1.19*	(0.50)	−1.22*	(0.49)
Hispanic	−1.08	(0.73)	−1.11	(0.73)	−1.02	(0.73)
Black	−5.01***	(0.87)	−4.95***	(0.87)	−4.82***	(0.87)
Asian	−0.67	(0.57)	−0.60	(0.57)	−0.68	(0.56)
Other race	−1.59*	(0.75)	−1.47*	(0.74)	−1.61*	(0.74)
SAT Math/100	1.16***	(0.24)	1.07***	(0.24)	0.74**	(0.27)
Freshman	1.41**	(0.47)	1.13*	(0.47)	1.09*	(0.47)
Coding			0.88***	(0.24)	1.05***	(0.25)
Non-coding computer use			−0.67**	(0.22)	−0.65**	(0.22)
Class-level variable						
Innovative college CS	1.75	(0.91)	1.81*	(0.91)	−7.28*	(3.04)
Interactions						
Help at home X coding					−1.85**	(0.67)
Innovative X SAT math/100					1.46**	(0.47)
Innovative X vocabulary					−1.22**	(0.46)
N (classes)		177		177		177
N (students)		2871		2871		2871
Pseudo-R^2		9.0%		10.0%		11.0%

Note: Significant *p*-values indicated by * ($p \leq .05$), ** ($p \leq .01$), and *** ($p \leq .001$). Parentheses indicate standard errors.

	Model 1: Controls		Model 2: Main Effects		Model 3: Interactions	
	b	(se)	b	(se)	b	(se)
Intercept	77.55***	(1.53)	78.48***	(1.56)	80.57***	(1.71)
Student-level variables						
Avg. parent education	0.39	(0.22)	0.34	(0.22)	0.33	(0.22)
Help at home	0.38	(0.65)	0.33	(0.65)	0.46	(0.65)
Vocabulary	−0.04	(0.21)	−0.11	(0.21)	0.23	(0.25)
Male	−1.06*	(0.50)	−1.19*	(0.50)	−1.22*	(0.49)
Hispanic	−1.08	(0.73)	−1.11	(0.73)	−1.02	(0.73)
Black	−5.01***	(0.87)	−4.95***	(0.87)	−4.82***	(0.87)
Asian	−0.67	(0.57)	−0.60	(0.57)	−0.68	(0.56)
Other race	−1.59*	(0.75)	−1.47*	(0.74)	−1.61*	(0.74)
SAT Math/100	1.16***	(0.24)	1.07***	(0.24)	0.74**	(0.27)
Freshman	1.41**	(0.47)	1.13*	(0.47)	1.09*	(0.47)
Coding			0.88***	(0.24)	1.05***	(0.25)
Non-coding computer use			−0.67**	(0.22)	−0.65**	(0.22)
Class-level variable						
Innovative college CS	1.75	(0.91)	1.81*	(0.91)	−7.28*	(3.04)
Interactions						
Help at home X coding					−1.85**	(0.67)
Innovative X SAT math/100					1.46**	(0.47)
Innovative X vocabulary					−1.22**	(0.46)
N (classes)		177		177		177
N (students)		2871		2871		2871
Pseudo-R^2		9.0%		10.0%		11.0%

Note: Significant *p*-values indicated by * ($p \leq .05$), ** ($p \leq .01$), and *** ($p \leq .001$). Parentheses indicate standard errors.

Graph of Interactions



	Model 1: Controls		Model 2: Main Effects		Model 3: Interactions	
	b	(se)	b	(se)	b	(se)
Intercept	77.55***	(1.53)	78.48***	(1.56)	80.57***	(1.71)
Student-level variables						
Avg. parent education	0.39	(0.22)	0.34	(0.22)	0.33	(0.22)
Help at home	0.38	(0.65)	0.33	(0.65)	0.46	(0.65)
Vocabulary	−0.04	(0.21)	−0.11	(0.21)	0.23	(0.25)
Male	−1.06*	(0.50)	−1.19*	(0.50)	−1.22*	(0.49)
Hispanic	−1.08	(0.73)	−1.11	(0.73)	−1.02	(0.73)
Black	−5.01***	(0.87)	−4.95***	(0.87)	−4.82***	(0.87)
Asian	−0.67	(0.57)	−0.60	(0.57)	−0.68	(0.56)
Other race	−1.59*	(0.75)	−1.47*	(0.74)	−1.61*	(0.74)
SAT Math/100	1.16***	(0.24)	1.07***	(0.24)	0.74**	(0.27)
Freshman	1.41**	(0.47)	1.13*	(0.47)	1.09*	(0.47)
Coding			0.88***	(0.24)	1.05***	(0.25)
Non-coding computer use			−0.67**	(0.22)	−0.65**	(0.22)
Class-level variable						
Innovative college CS	1.75	(0.91)	1.81*	(0.91)	−7.28*	(3.04)
Interactions						
Help at home X coding					−1.85**	(0.67)
Innovative X SAT math/100					1.46**	(0.47)
Innovative X vocabulary					−1.22**	(0.46)
N (classes)		177		177		177
N (students)		2871		2871		2871
Pseudo-R^2		9.0%		10.0%		11.0%

Note: Significant *p*-values indicated by * ($p \leq .05$), ** ($p \leq .01$), and *** ($p \leq .001$). Parentheses indicate standard errors.

Limitations

Limitations

Limitations

- Availability of confident computer science teachers
- Proper resources for teachers

Tackling Limitations

Two studies: Using workshops and interviewing current computer science teachers

Workshop study: 25 teachers, week long workshop

Curriculum: Recursion, analysis, machine language, and theoretical computer science topics

Teachers came out more confident and ready

Tackling Limitations

Push for computer science education around the globe

Need a large amount of computer science teachers

United States computer science teacher field flawed

Certification programs

57% of computer science teachers teach other content areas

Tackling Limitations

Interviewing study: 24 high school computer science teachers

Interview gathered information on challenges faced by teachers in classroom

Tackling Limitations

Interviewing study: 24 high school computer science teachers

Interview gathered information on challenges faced by teachers in classroom

Challenges found:

- Teaching in the classroom
  - Content
  - Pedagogy
  - Assessment

Tackling Limitations

Interviewing study: 24 high school computer science teachers

Interview gathered information on challenges faced by teachers in classroom

Challenges found:

- Teaching in the classroom
  - Content
  - Pedagogy
  - Assessment
- Compounding factors
  - Lack of teacher prep.
  - Isolation
  - Information technology

Tackling Limitations

Interviewing study: 24 high school computer science teachers

Interview gathered information on challenges faced by teachers in classroom

Challenges found:

- Teaching in the classroom
  - Content
  - Pedagogy
  - Assessment
- Compounding factors
  - Lack of teacher prep.
  - Isolation
  - Information technology
- Support
  - Organized repository
  - Community

Tackling Limitations

Possible Solutions:

- Professional workshops
- More proper certification programs
- Support community
- Online tool repository

Conclusion

Conclusion

New, developing field

Conclusion

New, developing field

Young children can grow skill of computational thinking through visual programming languages with or without metaphors

Early education of computer science holds many advantages

Conclusion

New, developing field

Young children can grow skill of computational thinking through visual programming languages with or without metaphors

Early education of computer science holds many advantages

Overhaul of computer science certification programs and more available resources

Questions?

References

Images:

- Myself
- https://www.scratchjr.org/learn/interface
- https://www.teachhub.com/teaching-strategies/2019/09/5-teaching-strategies-to-keep-class-interesting/
- Sources

References

Sources:

- D. D. Bowman. Declining talent in computer relatedcareers.Journal of Academic Administration in HigherEducation, 14(1):1–4, 2018
- H. Burgiel, P. M. Sadler, and G. Sonnert. Theassociation of high school computer science content andpedagogy with students' success in college computer science.ACM Transactions on Computing Education(TOCE), 20(2):1–21, 2020
- D. Leyzberg and C. Moretti. Teaching cs to cs teachers:Addressing the need for advanced content in k-12professional development. InProceedings of the 2017ACM SIGCSE technical symposium on ComputerScience Education, pages 369–374, 2017
- D. P´erez-Mar´ın, R. Hij´on-Neira, A. Bacelo, andC. Pizarro. Can computational thinking be improvedby using a methodology based on metaphors andscratch to teach computer programming to children?Computers in Human Behavior, 105:105849, 2020
- A. Strawhacker and M. U. Bers. What they learn whenthey learn coding: investigating cognitive domains andcomputer programming knowledge in young children.Educational Technology Research and Development,67(3):541–575, 2019.
- A. Yadav, S. Gretter, S. Hambrusch, and P. Sands.Expanding computer science education in schools:understanding teacher experiences and challenges.Computer Science Education, 26(4):235–254, 2016