Programming in Scratch

Computer Science 111
Boston University

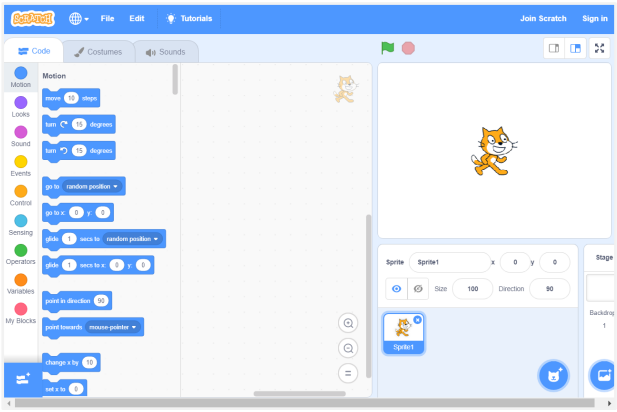
Vahid Azadeh-Ranjbar, Ph.D.

Get ready for voting! (B1 lecture)

- Using your phone **(preferred)**:
  1. Open the Top Hat app.
  2. Login as needed using your Top Hat account info.
  3. Select our course.
     - if you don't see it, click *Add Course* and use **join code 812992**
  4. Click on *Lecture*, and wait for the first question!
- Using your browser:
  1. Enter this URL: **https://app.tophat.com/e/812992**
  2. Login as needed using your Top Hat account info.
  3. If asked about enrolling, click *Enroll*.
  4. Click on *Lecture*, and wait for the first question!
- **Get into groups of 3** with whomever is near you.
- **Take out your paper notebook.**

Scratch

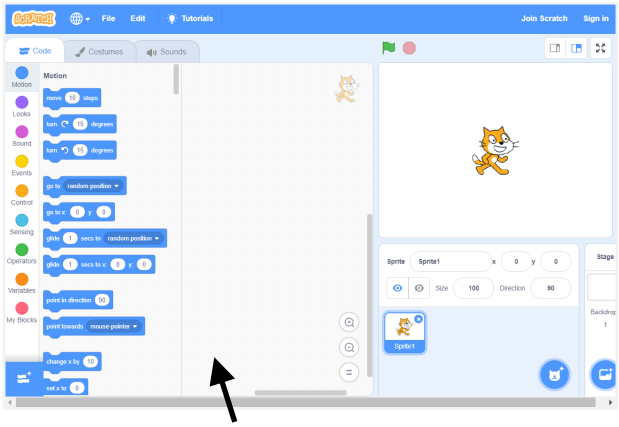
- A simple but powerful graphical programming language
  - developed at the MIT Media Lab
  - makes it easy to create animations, games, etc.



https://scratch.mit.edu/projects/editor

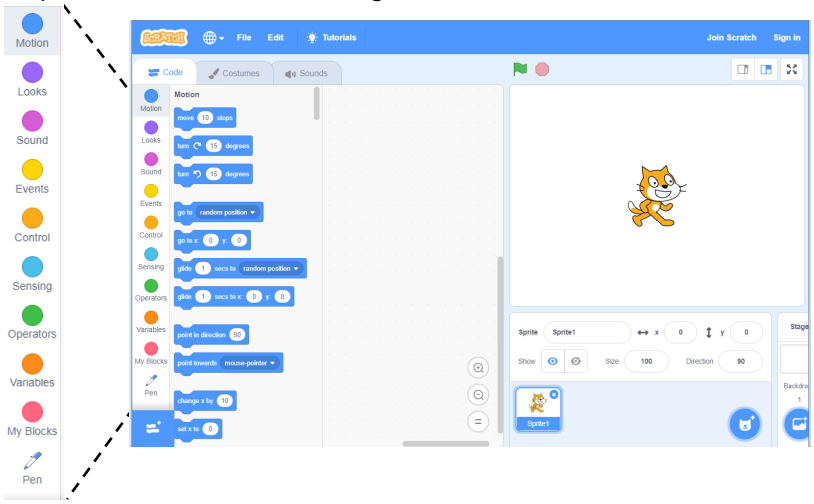
Scratch Basics

- Scratch programs (*scripts*) control characters called *sprites.*
- Sprites perform actions and interact with each other on the *stage.*



Program Building Blocks

- Grouped into color-coded categories:



- The shape of a building block indicates where it can go.

Program Building Blocks: Statements

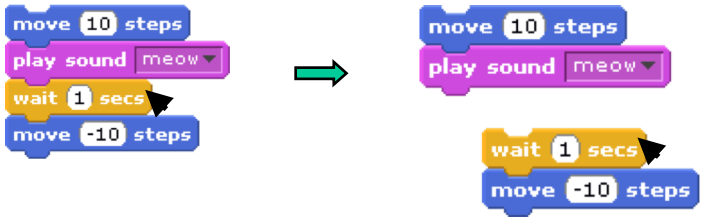
- Statement = a command or action

- Statements have bumps and/or notches that allow you to stack them.
  - each stack is a single script

- A statement may have:
  - an *input area* that takes a value (10, 1, etc.)
  - a pull-down menu with choices (meow)

Program Building Blocks: Statements (cont.)

- Clicking on any statement in a script executes the script.
- When rearranging blocks, dragging a statement drags it *and* any other statements below it in the stack.
  - example: dragging the *wait* command below



Flow of Control

- Flow of control = the order in which statements are executed
- By default, statements in a script are executed sequentially from top to bottom when the script is clicked.

- *Control blocks* (gold in color) allow you to affect the flow of control.
  - simple example: the *wait* statement above pauses the flow of control

Flow of Control: Repetition

- Many control statements are C-shaped, which allows them to control other statements.
- Example: statements that repeat other statements.

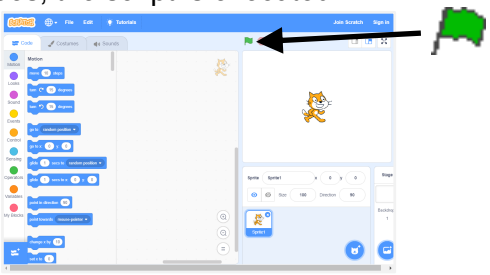
- Drag statements inside the opening to create a repeating stack.

- In programming, a group of statements that repeats is known as a *loop*.

Flow of Control: Responding to an Event

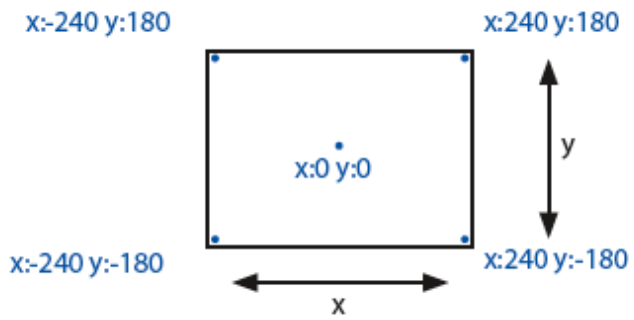
- *Hat blocks* (ones with rounded tops) can be put on top of a script.

- They wait for an event to happen.
  - when it does, the script is executed



Stage Coordinates

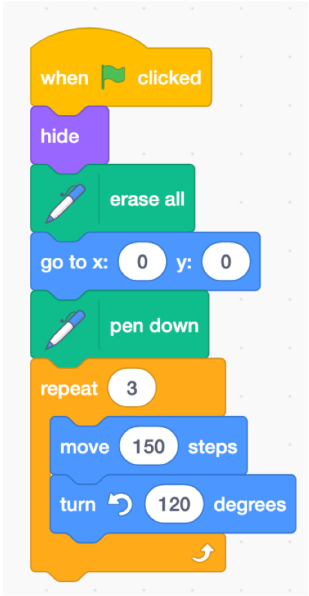
- Dimensions: 480 units wide by 360 units tall
- Center has coordinates of 0, 0



Peer Instruction

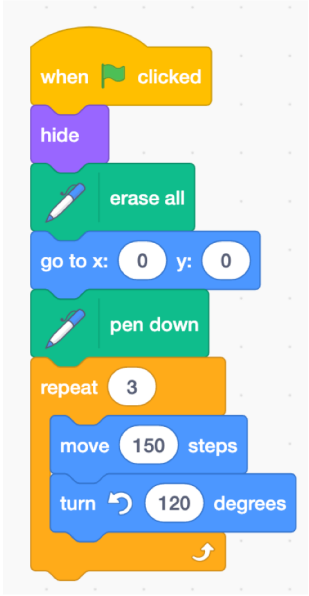
- **Get into groups of 3** with whomever is near you.
- Basic process:
  1. Question posed
  2. Solo vote (no discussion yet)
  3. Small-group discussions
     - explain your thinking to each other
     - come to a consensus
  4. Group vote - each person in a group should vote the same
  5. Class-wide discussion

What does this program draw?



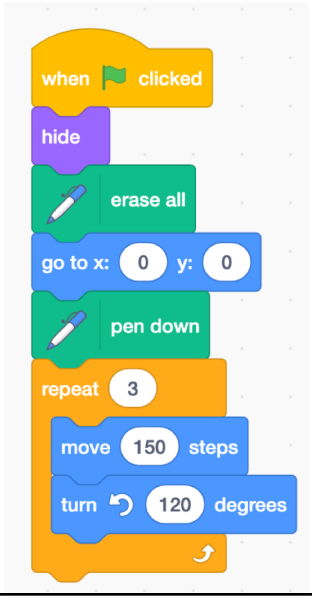
A. a triangle
B. a square
C. an unconnected sequence of line segments
D. none of the above

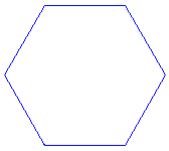
What does this program draw?



A. **a triangle**
B. a square
C. an unconnected sequence of line segments
D. none of the above

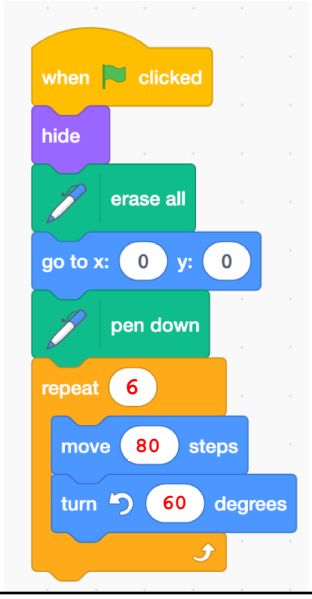
How many changes would be needed to draw this figure instead? (What are they?)

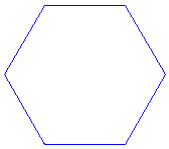




A. one
B. two
C. three
D. more than three

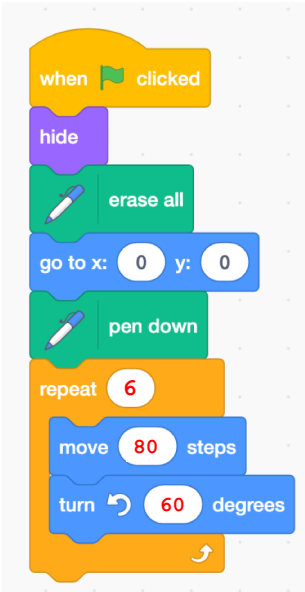
How many changes would be needed to draw this figure instead? (What are they?)

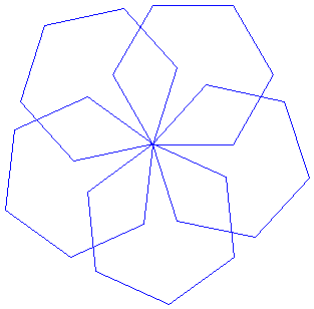




A. one
B. **two**
C. **three**
D. more than three

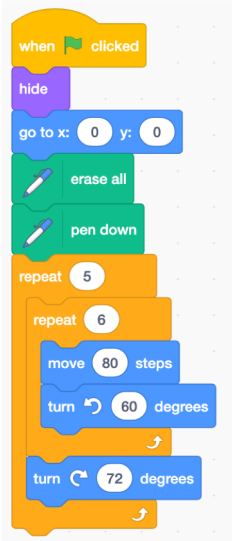
How could we draw this figure?





Flow of Control: Repeating a Repetition!

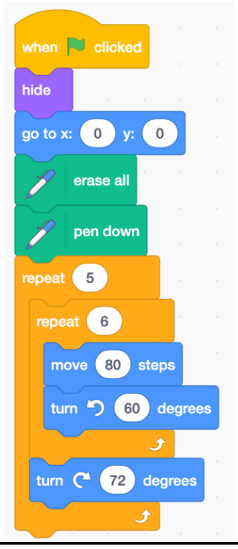
- One loop inside another loop!
  - known as a *nested loop*



- How many times is the *move* statement executed above? **30**

Making Our Program Easier to Change

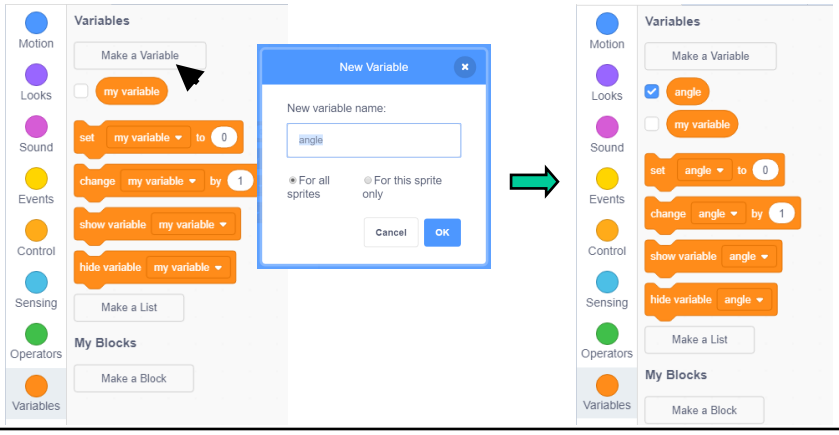
- It would be nice to avoid having to manually change *all* of the numbers.
- Take advantage of relationships between the numbers.



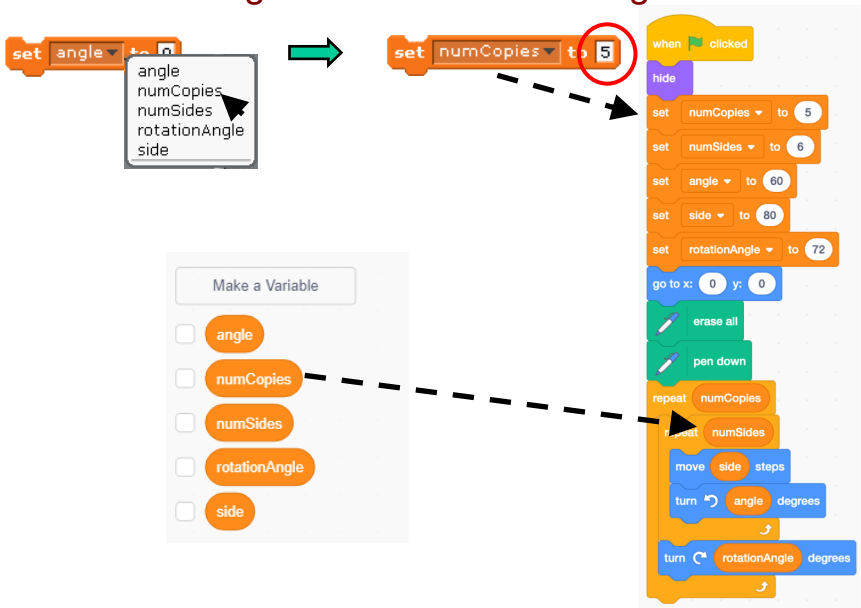
Program Building Blocks: Variables

- A *variable* is a named location in the computer's memory that is used to store a value.
- Can picture it as a named box: 

- To create a variable:



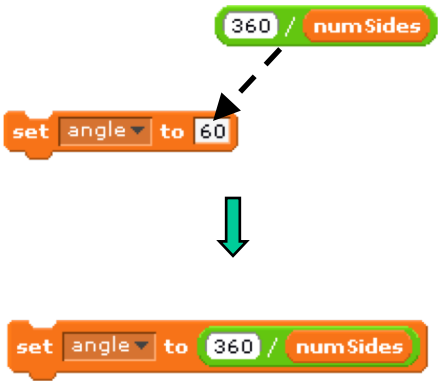
Using Variables in Your Program



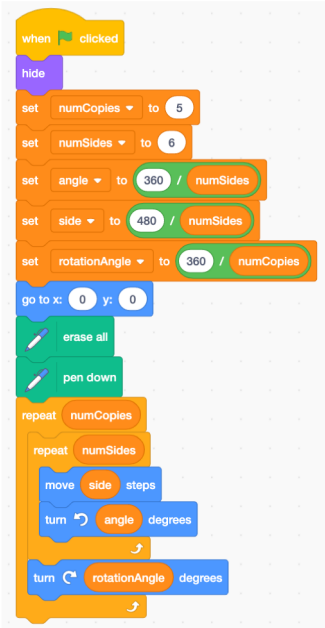
note: you must drag a variable into place, <u>not</u> type its name

Program Building Blocks: Operators

- Operators create a new value from existing values/variables.



Our Program with Variables and Operators

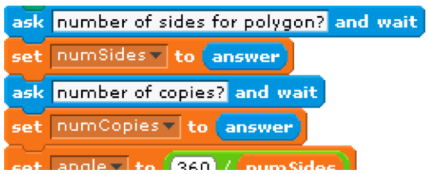


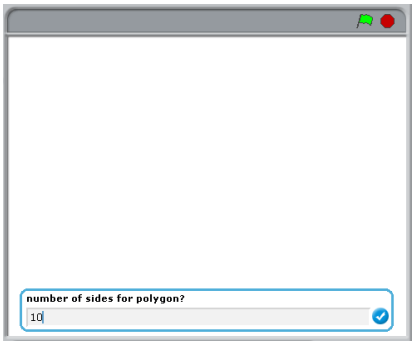
Getting User Input

- Use the *ask* command from the *sensing* category.

- The value entered by the user is stored in the special variable *answer*, which is also located in the sensing category.

- Allowing the user to enter *numSides* and *numCopies*:





Our Program With User Inputs

Program Building Blocks: Boolean Expressions

- Blocks with pointed edges produce *boolean* values:
  - *true* or *false*
- Boolean operators:

Reports true if first value is less than second.

Reports true if two values are equal.

Reports true if first value is greater than second.

Reports true if both conditions are true.

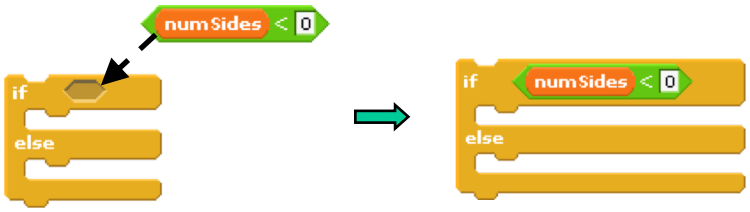
Reports true if either condition is true.

Reports true if condition is false; reports false if condition is true.

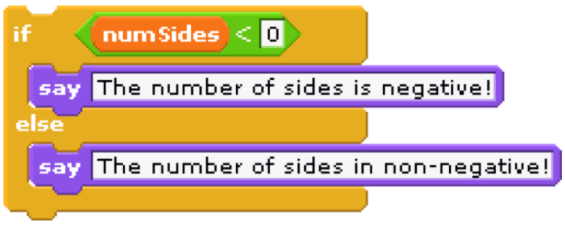
Flow of Control: Conditional Execution

- conditional execution = deciding whether to execute one or more statements on the basis of some condition
- There are C-shaped control blocks for this:

- They have an input area with pointed edges for the condition.

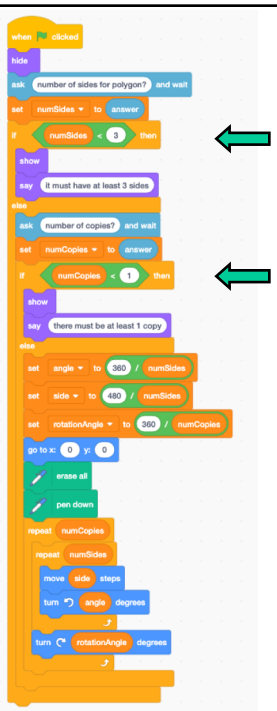


Flow of Control: Conditional Execution (cont.)



- If the condition is true:
  - the statements under the *if* are executed
  - the statements under the *else* are <u>not</u> executed
- If the condition is false:
  - the statements under the *if* are <u>not</u> executed
  - the statements under the *else* are executed

How can we deal with invalid user inputs?



Final version

- We use two `if-else` statements to check for invalid inputs:
  - one checks for `numSides < 3`
  - one checks for `numCopies < 1`
- If an invalid input is found, we:
  - show the sprite
  - have the sprite say an error message
  - end the program
- Otherwise, we continue with the rest of the program.

More Info on Scratch

- We're using the latest version:
  https://scratch.mit.edu/projects/editor
- Creating a Scratch account is <u>not</u> required for this course.

Other Announcements

- Complete Lab 0 ASAP.
  - see the Labs section of the course website
- Check Blackboard later today for Monday's pre-lecture tasks.
  - due by 10 a.m. on Monday