

CSC 221: Introduction to Programming

Fall 2018

Introduction to programming in Scratch

- animation sprites
- motion, control & sensing
- costume changes
- variables & state
- interacting sprites & broadcasts
- animation & game programming

1

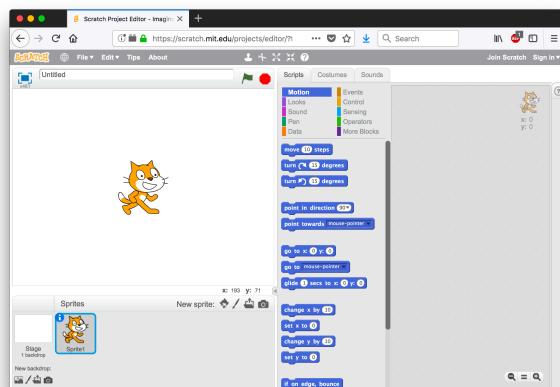
Scratch

Scratch is a simple environment for creating animations & games

- developed at MIT to introduce programming & creative design
- great for making programming and object-oriented concepts concrete
 - sprite:* an animation object with properties and behaviors

by default, the Scratch cat sprite loads

- can add properties and behaviors by dragging blocks from the middle pane to the right pane



Scratch is free & online at scratch.mit.edu

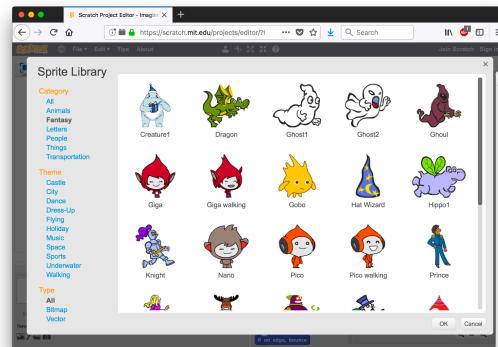
- Web site contains numerous guides & tutorials
- more than 38 million projects online
- you can download to/upload from your computer, or create a free account and store projects online

2

Adding sprites

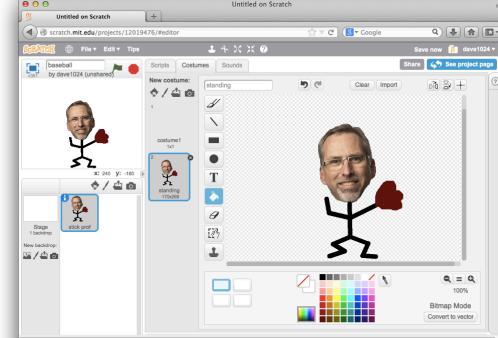
can select from a provided collection of sprites

- click on the cartoon character next to New Sprite:
- collections include animals, fantasy characters, people, things, ...



can create a new sprite using the Paint Editor

- click on the paint brush
- can draw, add shapes, fill, erase, rotate, resize, add text, ...
- can even import an image, then edit as desired (e.g., erase unwanted parts)



can move sprites around the stage using the mouse, shrink or grow by right-clicking

Sprite motion

each sprite can move around the stage

- select the desired (blue) Motion block
- drag that block to the Script panel for the sprite
- click on that block to execute it

note:

- coordinates: center = (0,0)
bottom-left = (-240, -180)
top-right = (240, 180)
- directions: 0 = up
90 = right
180 = down
270 = left
- can control whether the sprite turns using the "set rotation style" block

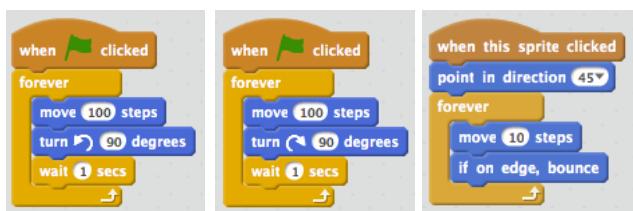


4

Sprite control

the (yellow) Control blocks allow for repeated or conditional behavior

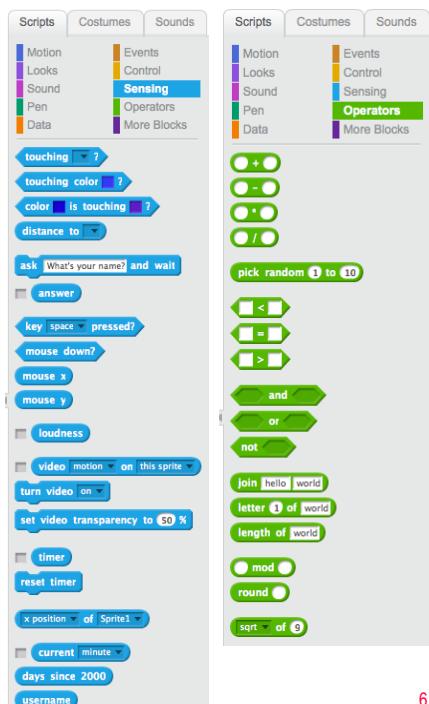
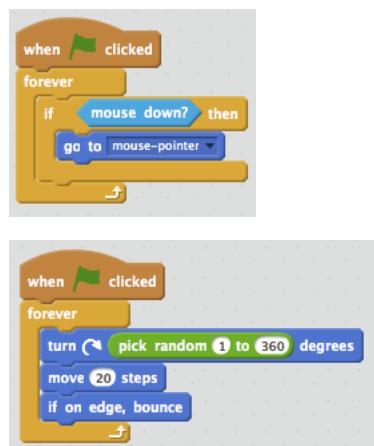
- *when* blocks allow for behaviors to be triggered by events (click on green flag, press a key, click on sprite)
- *wait* block causes execution to delay a set amount of time
- *forever* block encloses other block(s) that are to be executed over and over, indefinitely
- *repeat* block encloses other block(s) that are to be executed a set number of times



5

Sprite sensing

other blocks allow the sprite to sense the environment and make comparisons

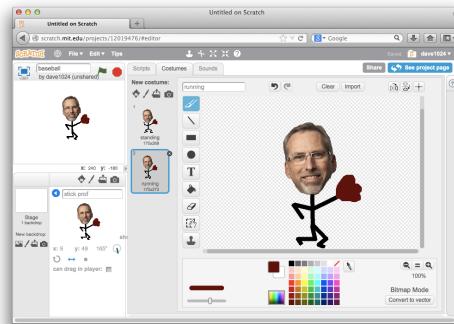


6

Changing costumes

can create multiple costumes for a sprite

- click on the Costumes tab
- either:
 1. click on brush to draw a new costume
 2. click on file to load an image
 3. right click on an existing costume to make a duplicate, then edit



switch to costume and next costume

from the (purple) Looks blocks
change the sprite's costume



7

Interacting sprites

can have multiple sprites active at once

- can sense and react to each other
- note use of sound when sprites collide -- can also record sounds (or even play music), but much slower!



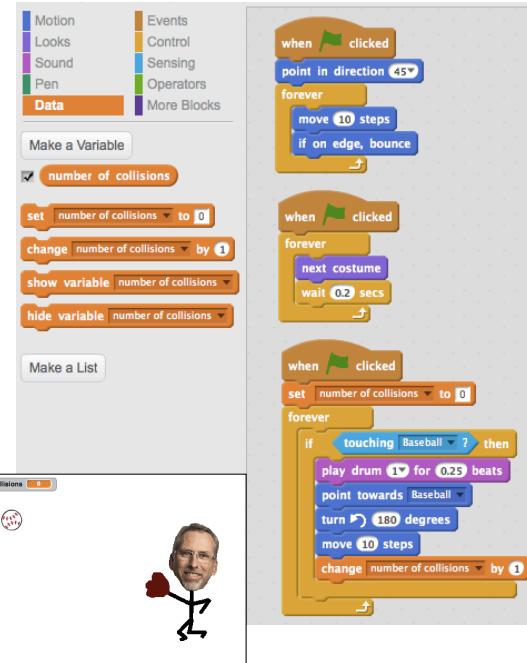
8

Variables & state

the state of a sprite is its current set of properties (e.g., location, color, heading)

can add new properties to a sprite's state with variables

- a *variable* is a name that refers to some value/property
- create a variable by clicking on Make a Variable
- by default, the variable is displayed on the stage
- a sprite can access and update a variable using the *set* and *change* (orange) blocks

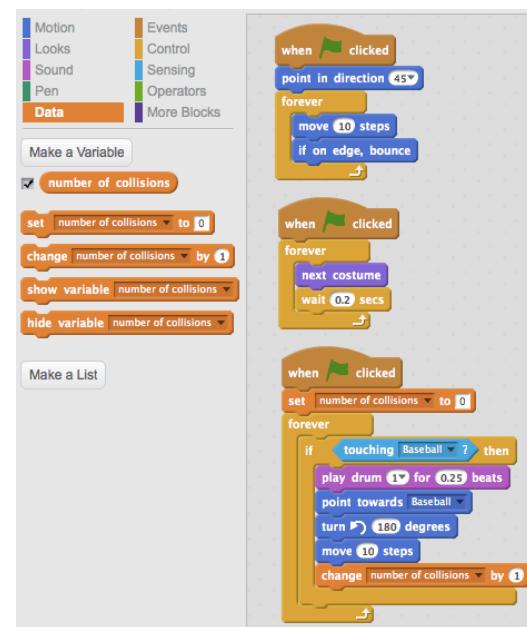


9

Counters

in general, a *counter* is a variable that keeps track of some event

- initially, the counter value is 0 (nothing has happened yet)
- each time the event occurs, the variable value goes up by one
- it is VERY important that the initialization happens OUTSIDE the loop -- WHY?



10

New game

we want to repurpose the stick prof and baseball into a game

- the stick prof is controlled by the mouse
 - by moving the mouse, the user can move the stick prof left and right
- the baseball drops out of the sky from a random location
 - the user must move the stick prof so that the ball lands in the glove
 - if caught, the player's score goes up and a new ball drops
 - if missed, then the game ends
- would like to make it easy to start and restart the game
 - when the green flag is clicked, the stick prof is positioned & instructions displayed
 - when the user hits space bar, then balls start dropping
 - when the game ends, a message is displayed and the space bar restarts

11

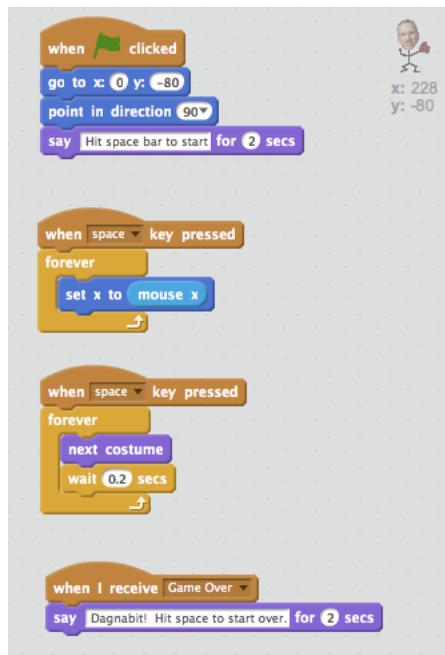
Ball drop game

it makes sense for the stick prof to handle the start and end of the game

- but the baseball knows when it hits the ground
- how do they communicate and coordinate?

one sprite can broadcast a message when an event occurs (e.g., ball hits the ground)

- another sprite can receive the broadcast and react



12

Ball drop game

note: there is only one ball

- when it is caught, it reappears at the top at a new position
- when it hits the ground, it broadcasts the "Game Over" message and stops the script

what if we changed the number?

change y by -20



13

Animations & games

numerous projects are included with the Scratch download

- more than 38 million projects uploaded to the scratch.mit.edu site
- lots of animations
 - stories with characters interacting (especially cats & anime)
 - tutorials, dance & music, drawing, ...
- lots of games
 - interactive video games (e.g., tetris, pacman, breakout)
 - casino games, word games, stick figure movement, ...

recall: *programming* is the process of designing, writing, testing and debugging algorithms that can be carried out by a computer

➔ Scratch is programming!

- you design the sprites (objects), their properties & behaviors, their interactions
- you create the scripts (algorithms) that implement those behaviors
- you test and debug the projects to make them work as desired

14