CSC 221: Introduction to Programming

Fall 2011

Introduction to programming in Scratch

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

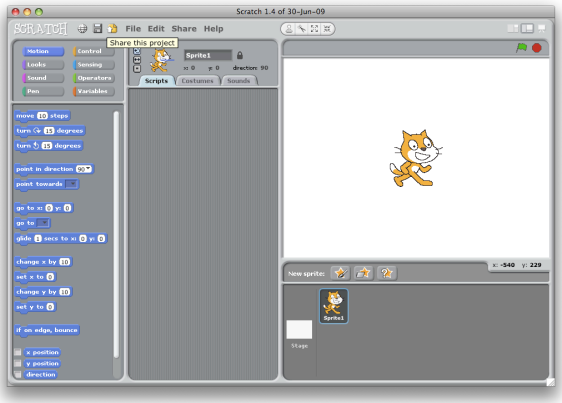
Scratch

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

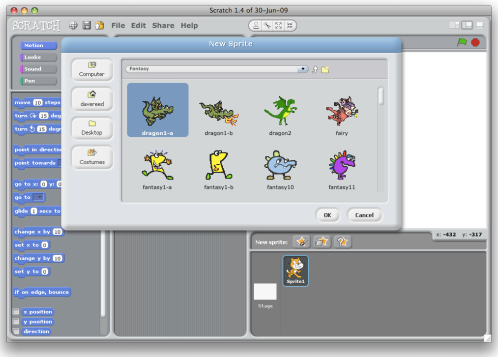
  by default, the Scratch cat sprite loads
- can add properties and behaviors by dragging blocks from the left pane to the middle pane



Scratch is free at scratch.mit.edu

- Web site contains numerous guides & tutorials
- almost 2 million projects uploaded & viewable

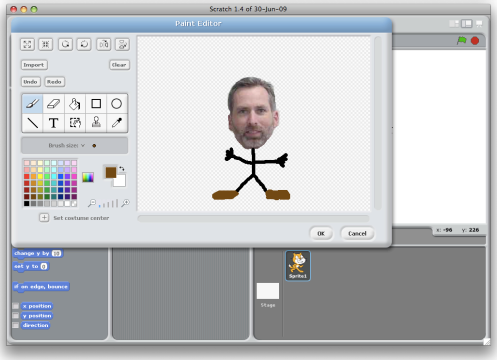
Adding sprites



can select from a provided collection of sprites

- click on the button with star & folder
- collections include animals, fantasy characters, people, things, …

can create a new sprite using the Paint Editor



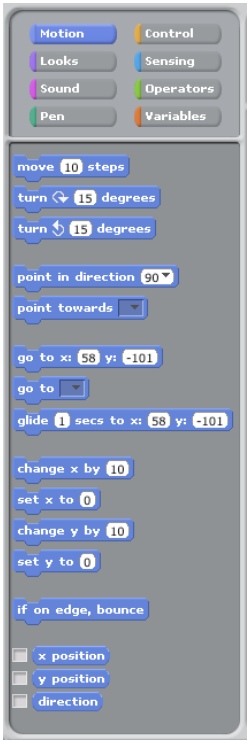
- click on the button with star & paint brush
- can draw, add shapes, fill, erase, rotate, resize, add text, …
- can even import an image, then edit as desired (e.g., erase unwanted parts)

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:

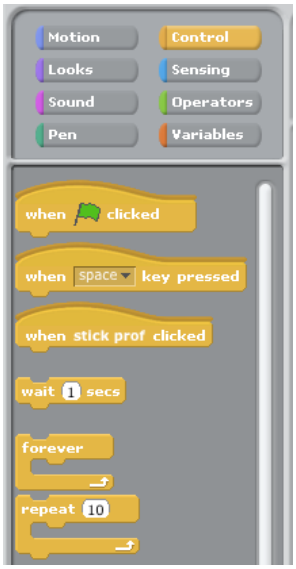
- coordinates: center = (0,0)
  bottom-left = (-240, -180)
  top-right = (240, 180)
- directions: 0 = up
  90 = right
  180 = down
  270 = left
- can control whether the sprite turns as it moves by selecting one of:

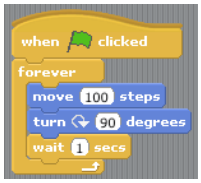
4

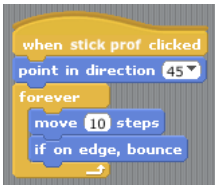
Sprite control

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



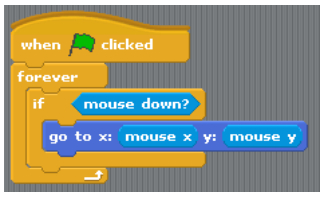


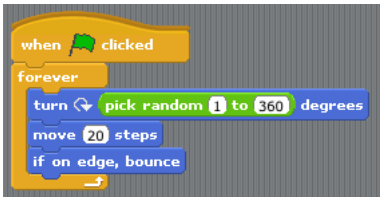


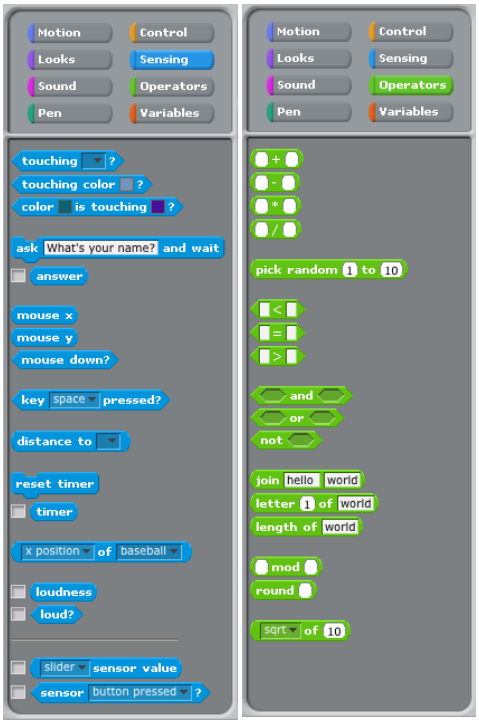
5

Sprite sensing

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







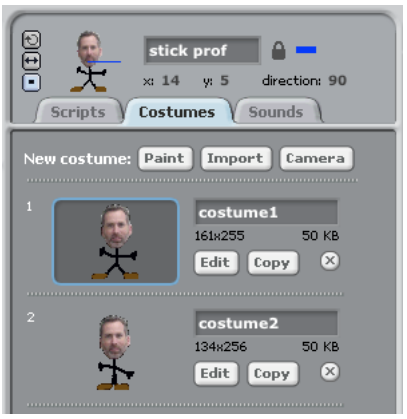
6

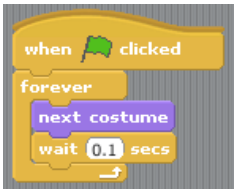
Changing costumes

can create multiple costumes for a sprite

- click on the Costumes tab
- either:
  1. click on Paint to draw a new costume
  2. click on Import to load an image
  3. click on Copy to make a copy of an existing costume, then Edit to modify it

switch to costume and *next costume* from the (purple) Looks blocks change the sprite's costume



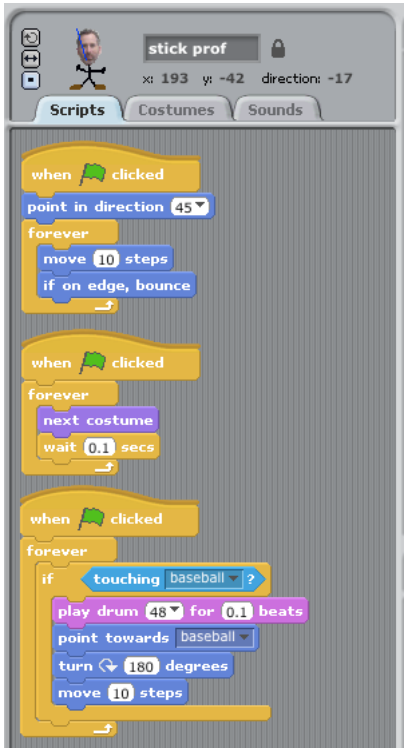


7

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

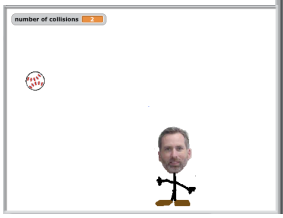
Variables & state

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

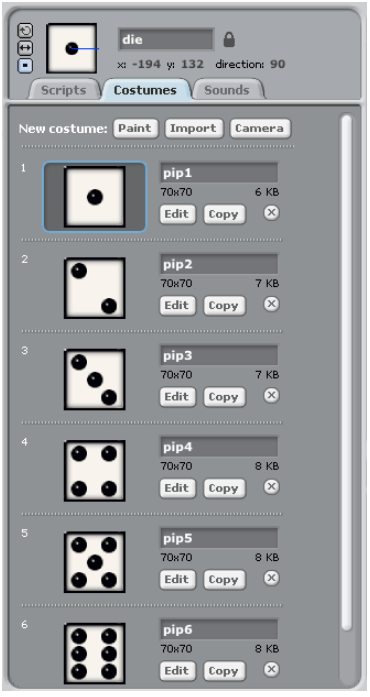


Die example

suppose we want to develop an animation that involves dice (e.g., online casino)

- could create a sprite that represents a die
- has 6 costumes, corresponding to die faces
- when clicked, flips through a random number of costumes

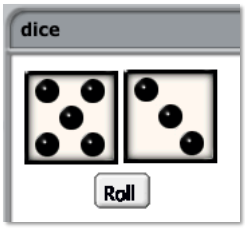
why between 14 and 19?



Coordinating sprites

if we want two dice, how do we coordinate them?

- could pick a key, have both dice roll when that key is pressed
- better solution: put a button on the stage & have both dice roll when the button is clicked
- requires the button to broadcast a message when clicked & dice to react to that message



11

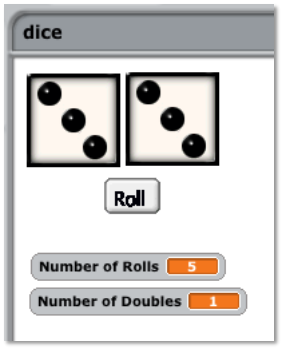
Counters

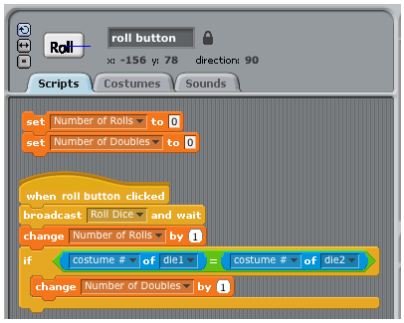
if we want to keep track of the rolls, again need variables to maintain the state

- create a variable for the number of dice rolls and the number of doubles obtained
- every time the button is clicked, the number of rolls goes up by one
- if doubles were rolled (i.e., the dice costumes are identical), the number of doubles goes up by one

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





12

Animations & games

numerous projects are included with the Scratch download

- almost 2 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 implements those behaviors
- you test and debug the projects to make them work as desired

13