

# Animated Activities

Philadelphia Area Math Teachers Circle  
Steve Garland  
May 17, 2016

5:15 - 5:20 Slides 1-2

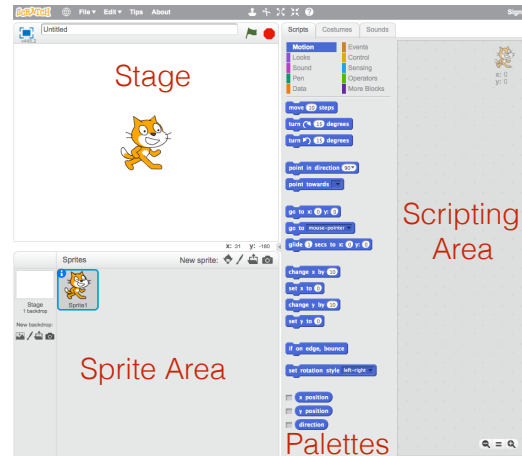
# The Problem

- Teach the cat how to walk to the girl without bumping into the fire-eating dragon
- Using the Scratch visual programming system



# Setting the Scene

- [scratch.mit.edu](http://scratch.mit.edu)



5:20 - 5:30 Slide 3 plus Scratch demonstration

Scratch: set desert backdrop, move cat below cactus, add Alex facing the cat

Mathematical note: two ways to have Alex face the cat, rotation and reflection

Coding: snap *point towards*, *set rotation style* blocks together

*Sequence* is the first of three fundamental ways to specify the flow of control

# Warm-Up Exercise

- Teach the cat how to move to the girl

5:30 - 5:45 Slide 4: Warm-up exercise

Show slides 5-7 as participants work in pairs on exercise

## Warm-Up Exercise ...

- Teach the cat how to move to the girl
- Make the cat appear to walk

## Warm-Up Exercise ...

- Teach the cat how to move to the girl
- Make the cat appear to walk
- Make the cat start walking when you click it

## Warm-Up Exercise ...

- Teach the cat how to move to the girl
- Make the cat appear to walk
- Make the cat start walking when you click it
- Don't depend on where the cat starts

# Warm-Up Exercise

## One Solution



5:45 - 6:00 Debriefing

10 minutes: Project participants' solutions on the smart board

5 minutes: Slides 8-10



# Basic Structures for Code

## *Flow*

- Sequence (straight-line code)
- Choice (branching or conditional code)
- Repetition (loop)

Solution uses sequence and repetition

Choice is coming soon!

# Basic Structures for Code

## *Flow*

- Sequence (straight-line code)
- Choice (branching or conditional code)
- Repetition (loop)

## *Organization*

- Procedure (method, function, subroutine)

More later about another important structure

# Enter, the dragon!



```
when green flag clicked
  go to x: 0 y: -100
  set rotation style left-right
  point in direction -90
  forever
    if touching Cat? then
      switch costume to dragon1-b
    else
      switch costume to dragon1-a
```

6:00 - 6:10 Slides 11-12, followed by brainstorming and Slide 13  
Notice use of choice

# The Problem

- Teach the cat how to walk to the girl without bumping into the fire-eating dragon
- But how?



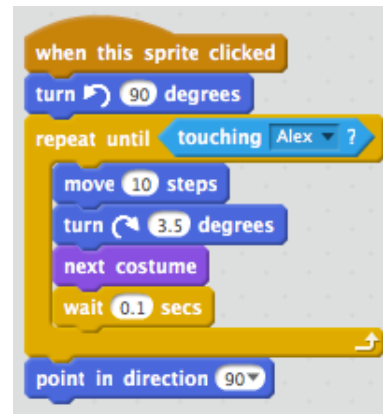
# Some Things to Try

- Walk in a semicircle.
- Jump over the dragon.
- Bounce off the top of the screen.
- Hug the edges of the screen.
- ... and lots more!

# Semicircular Path

Why 3.5 degrees?

Experimentation  
or  
some simple math



6:30 - 6:45 Debriefing (Slides 14-18 held in reserve)

First project participants' solutions on the smart board

Use Slides 14-18 to discuss connections to mathematics

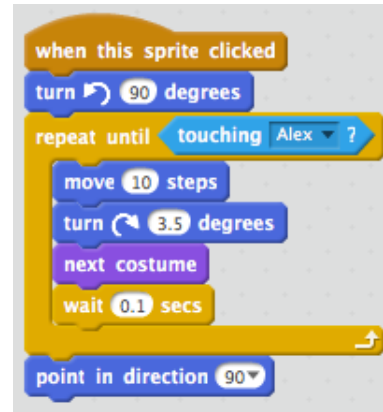
## Semicircular Path

Why 3.5 degrees?

Path diameter: 326

Path length: 512

Turn:  $10 \cdot 180 / 512$



6:30 - 6:45 Debriefing (Slides 14-16 held in reserve)

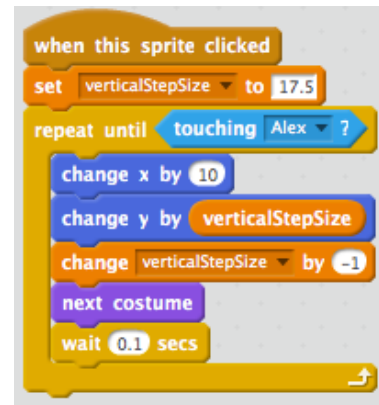
First project participants' solutions on the smart board

Use Slides 14-16 to discuss connections to mathematics

# Jump Over Dragon

Why 17.5?

Experimentation  
or  
some simple math





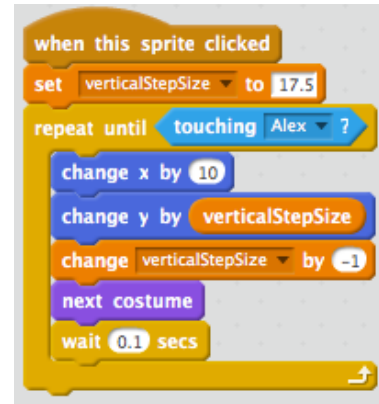
# Jump Over Dragon

Why 17.5?

32 times through loop.  
Height after 16 times should  
be about 160.

$$\begin{aligned}160 &= v + (v-1) + \dots + (v-15) \\ &= 16v - 15 \cdot 16/2 \\ &= 16v - 120\end{aligned}$$

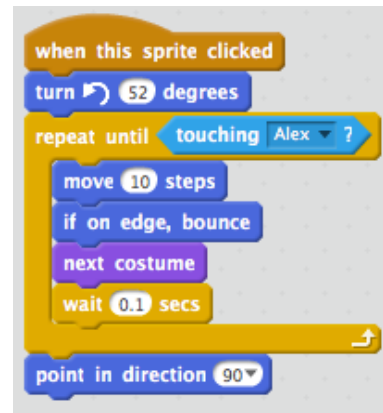
$$v = 17.5$$



# Bounce Off Top Edge

Why 52?

Need some trigonometry!

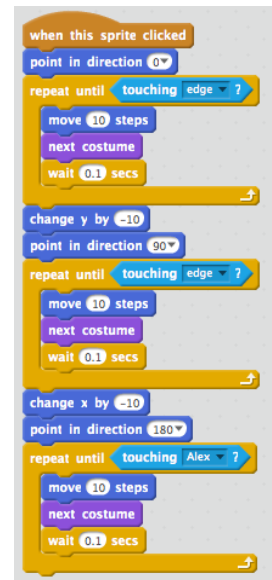


```
when this sprite clicked
  turn 52 degrees
  repeat until touching Alex ?
    move 10 steps
    if on edge, bounce
    next costume
    wait 0.1 secs
  point in direction 90
```

The image shows a Scratch script designed to make a sprite bounce off its top edge. The script starts with a 'when this sprite clicked' event block. It then turns the sprite 52 degrees. A 'repeat until' loop follows, which continues as long as the sprite is not touching a sprite named 'Alex'. Inside the loop, the sprite moves 10 steps, checks for an edge and bounces if one is reached, changes to the next costume, and waits for 0.1 seconds. After the loop ends, the sprite is pointed in the 90-degree direction.

## Hug Edges of the Screen

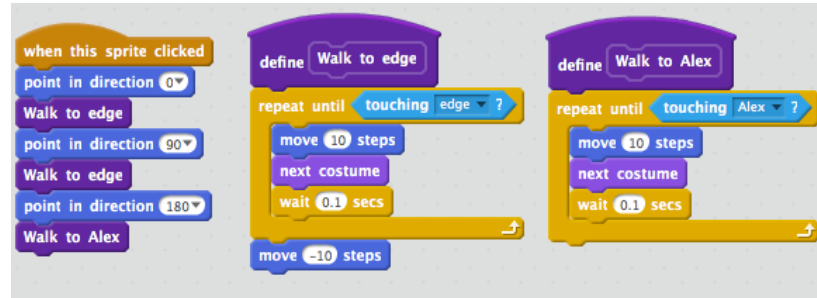
Repetitive code is hard to understand!



6:45 - 6:50 Debriefing, continued

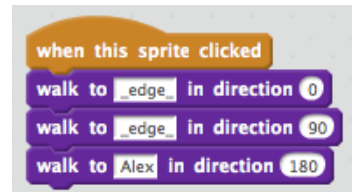
Slides 19-21

# Hug Edges of the Screen

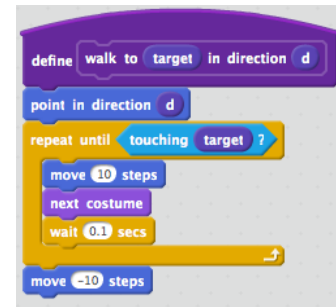


- Defining new blocks makes code easier to understand

# Hug Edges of Screen



```
when this sprite clicked
  walk to _edge in direction 0
  walk to _edge in direction 90
  walk to Alex in direction 180
```



```
define walk to target in direction d
  point in direction d
  repeat until touching target ?
    move 10 steps
    next costume
    wait 0.1 secs
  move -10 steps
```

- Defining a more general block makes the code even better

## Connections to Common Core Math Standards

- Graph points on the coordinate plane to solve real-world and mathematical problems (5.G)
- Reason about and solve one-variable equations and inequalities (6.EE)
- Know the formulas for the area and circumference of a circle (7.G)
- Understand congruence and similarity using physical models, ..., or geometry software; understand and apply the Pythagorean Theorem (8.G)

6:50 Wrap-up, Slides 22-24, leaving time for door prizes, etc.

# Math Practices

MP1. Make sense of problems and persevere in solving them

MP2. Reason abstractly and quantitatively

MP3. Construct viable arguments and critique the reasoning of others

MP4. Model with mathematics

MP5. Use appropriate tools strategically

MP6. Attend to precision

MP7. Look for and make use of structure

# CSTA Practices

1. Recognizing and representing computational problems
2. Developing and using abstractions
3. Creating computational artifacts
4. Testing and iteratively refining
5. Fostering an inclusive computing culture
6. Communicating about computing