

SCRATCH WASABI

Edition

SUSHI-WHOSHI? Using These Cards (1)

COUNTDOWN (30-0)

Variables: Time

```

when green flag clicked
set Time to 30
repeat until Time = 0
    wait 1 secs
    change Time by -1
end
broadcast end_game

```

COUNTUP (0-30)

Variables: Time

```

when green flag clicked
set Time to 0
repeat until Time = 30
    wait 1 secs
    change Time by 1
end
broadcast end_game

```

WAITING

Stage script:

```

when green flag clicked
go to x: 0 y: 0
hide
repeat (80)
    broadcast waiting
    broadcast ready to play
end

```

Waiting costume:

Costumes:

Spinner costume:

Scripts for the stage:

Scripts for the sprite:

Variables you will need to create:

MOVE ME!... Sprites (2)

MOVE ME!... Controlling Sprites (1)

3, 2, 1!... Countdowns & Timers (1)

3, 2, 1!... & Timers (2)

Variables: digit, digit2, digit3, digit4

Costumes for the sprite:

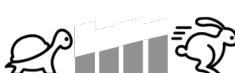
simple example project available in our on-line studio

available on individual wasabi cards

CoderDojo Sushi Edition
developed by @ginglexia

download @ kata.coderdojo.com/wiki/EU_Scratch_Sushi/
more info @ www.coderdojo.com
projects @ scratch.mit.edu/studios/1170890

CC BY NC SA



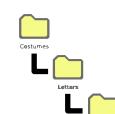
higher = faster
lower = slower



higher = slower
lower = faster



higher = smaller
lower = bigger



finding costumes



WHAT IS IT?

“Scratch Wasabi” are a series of “ingredients” cards which display the basics of Scratch in an easy to understand format. They are designed to help young coders get started with creating their own Scratch projects – especially games!

We’ve chosen some of the most common components we use and put them together in a way that we hope lets coders “pick and mix” what they want in their own project. For example you might want to think about how you want to control a sprite ... Using the keyboard?? Using the mouse?? Moving randomly with a Script?? How do you want to control the game ... against the clock?? keeping time?? pausing?? ... and so on ...

These cards were inspired by the excellent HTML Sushi Cards originally developed by CoderDojo Bray (and available on kata.coderdojo.com). Like the HTML Sushi we’d suggest these A4 sheets are printed out double-sided (or back-to-back) and then laminated for easy re-use! There are also a series of A5 cards which contain the individual components on these sheets.

For each component we’ve also created an example Scratch project file in our studio which you can find on the scratch.mit.edu website – here you can see the individual elements in action!

In future versions we’re going to add more elements and also provide some example “recipe” projects which bring these elements together to create full blown Scratch games! We really hope you find these useful and enjoy creating your own projects!

Ninja TC & Ginger Ninja ... aka Ginglexia!



WHY SCRATCH v1.4?

We’ve deliberately chosen to present Wasabi using Scratch v1.4!

- **It’s still the most commonly used version in schools!**
- It comes with most Raspberry Pi Operating Systems!
- (We’re currently working on a Wasabi sheet to show the differences with v2 ... watch this space...)

WHY JUST PICTURES?

We’ve tried to use a few words as possible!

- To make it more accessible to younger coders!
- To make translation easier!
- To make it more accessible to children with Dyslexia and other learning challenges!
- To help young coders teach others with some simple content!

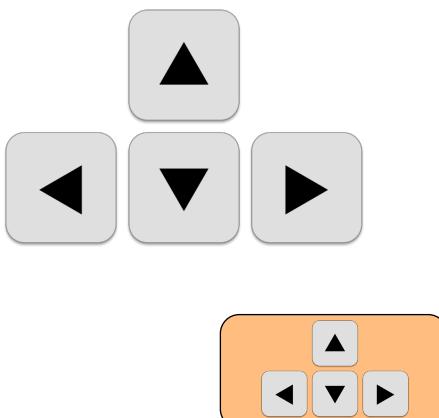
WASABI

“nuclear horseradish” (urbandictionary.com)

“really hot stuff” ([ginglexia!](http://ginglexia.com))



USING KEYS



```

when up arrow key pressed
change y by 10

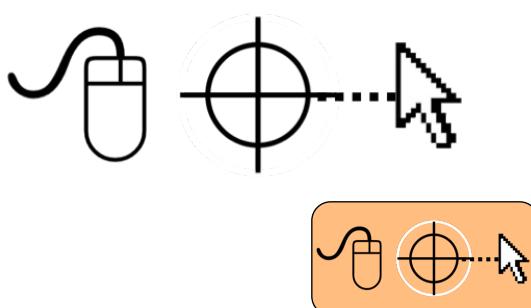
when down arrow key pressed
change y by -10

when right arrow key pressed
point in direction 90
move 10 steps

when left arrow key pressed
point in direction -90
move 10 steps
  
```

001

CHASING MOUSE

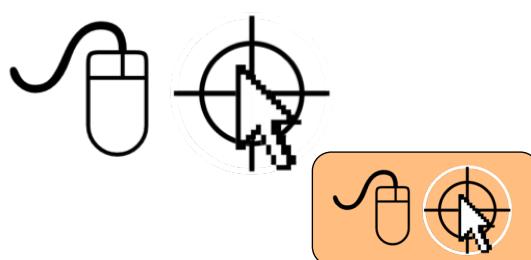


```

when green flag clicked
forever
  point towards mouse-pointer
  move 5 steps
  
```

002

ON MOUSE

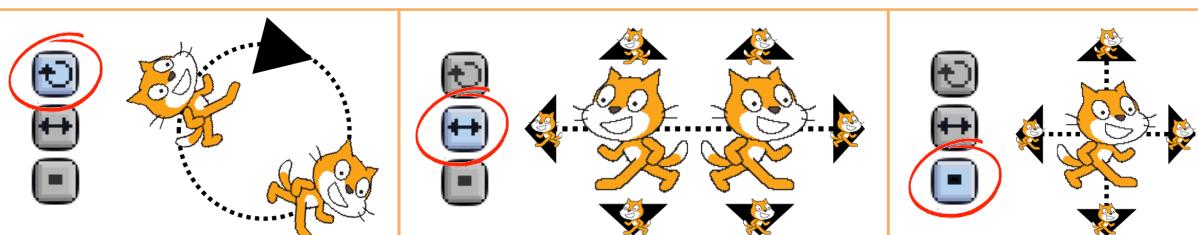


```

when green flag clicked
forever
  go to x: mouse x y: mouse y
  
```

003

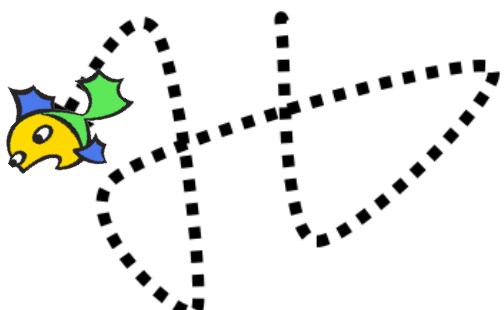
HINTS ...



#codeEU SCRATCH WASABI Edition

MOVE ME!... Controlling Sprites (2)

RANDOM TURNS

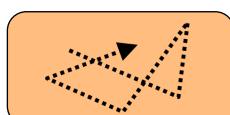


```

when green flag clicked
forever
  move (20) steps
  turn (pick random 1 to 10) degrees
  wait (0.1) secs
  if on edge, bounce
  turn (pick random 1 to 10) degrees
  
```

004

RANDOM GLIDE

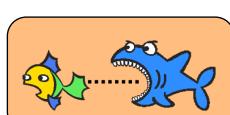
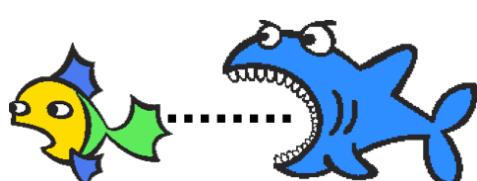


```

pick random (-180) to (180)
when green flag clicked
forever
  glide (1) secs to x: (pick random -240 to 240) y: (pick random -240 to 240)
  
```

005

CHASING SPRITE

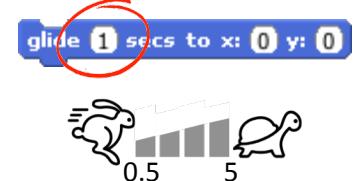
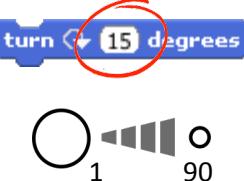
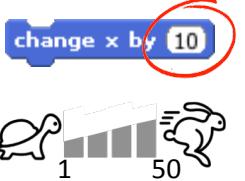
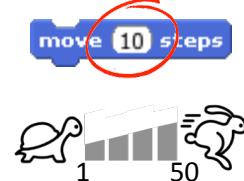


```

when green flag clicked
forever
  point towards fish
  move (5) steps
  
```

006

HINTS ...



SHOOTER

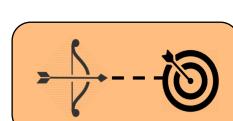
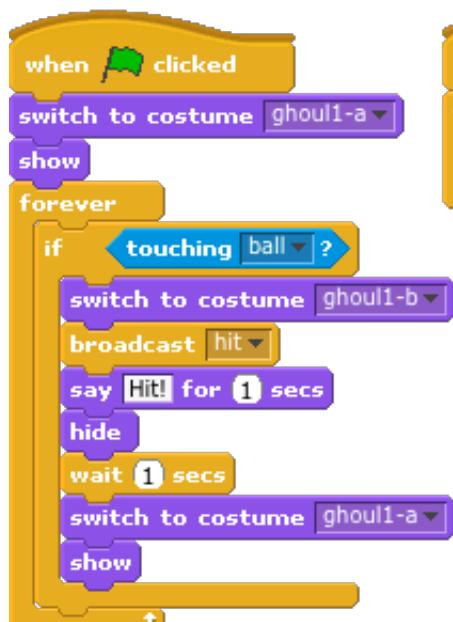


MISSILE



SHOOTING

TARGET



007



CLICKING

```

when green flag clicked
forever
  glide (1 secs to x: pick random -240 to 240) y: pick random -180 to 180

```



```

when green flag clicked
go to front
forever
  go to x: (mouse x) y: (mouse y)
  if (mouse down? and touching [ghost v]) then
    say [Hit!] for (0.5) secs
  else
    if (mouse down? and not touching [ghost v]) then
      say [Miss!] for (0.5) secs

```



008

CHASING

```

when green flag clicked
show
forever
  move (20) steps
  wait (0.5) secs
  turn (pick random 1 to 10) degrees
  if on edge, bounce
  turn (pick random 1 to 10) degrees
  if (touching [Shark v])
    broadcast [got_me v]
    hide
    wait (1) secs
    go to x: (pick random -240 to 240) y: (pick random -180 to 180)
    show

```



```

when green flag clicked
show
switch to costume [shark1-a v]
forever if (distance to mouse-pointer) > (10)
  point towards mouse-pointer
  move (5) steps

```



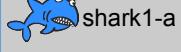
```

when I receive [got_me v]
repeat (2)
  switch to costume [shark1-b v]
  wait (0.1) secs
  switch to costume [shark1-a v]

```



Costumes



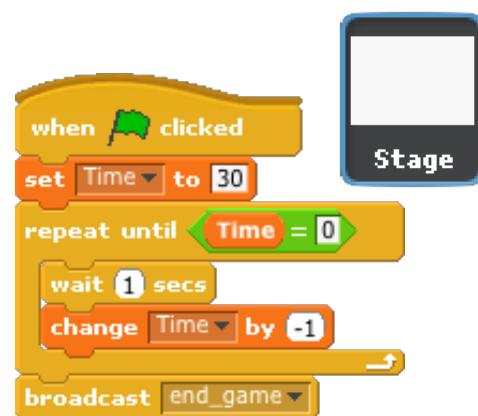
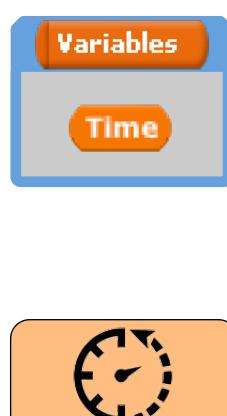
009



#codeEU SCRATCH WASABI Edition

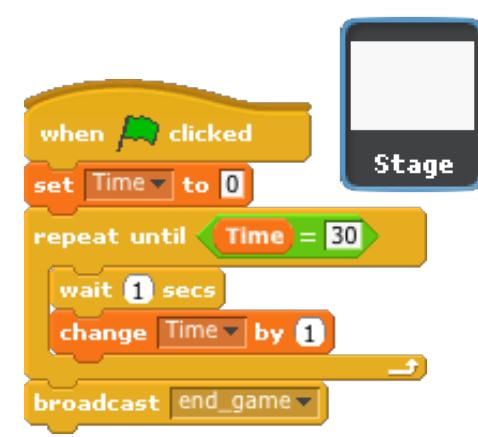
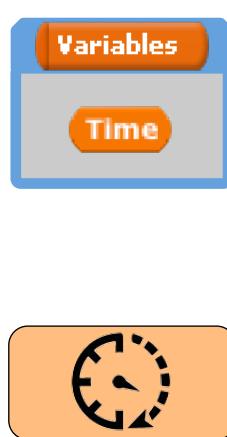
3, 2, 1!...
Countdowns & Timers (1)

COUNTDOWN (30-0)



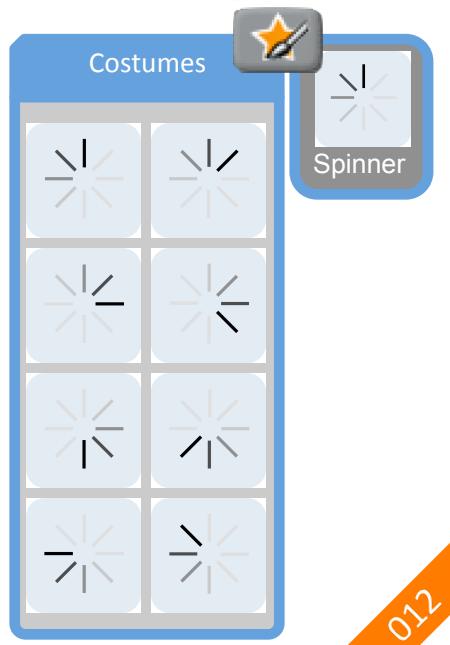
010

COUNTUP (0-30)



011

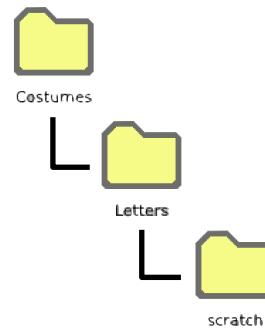
WAITING



012



0:00:0

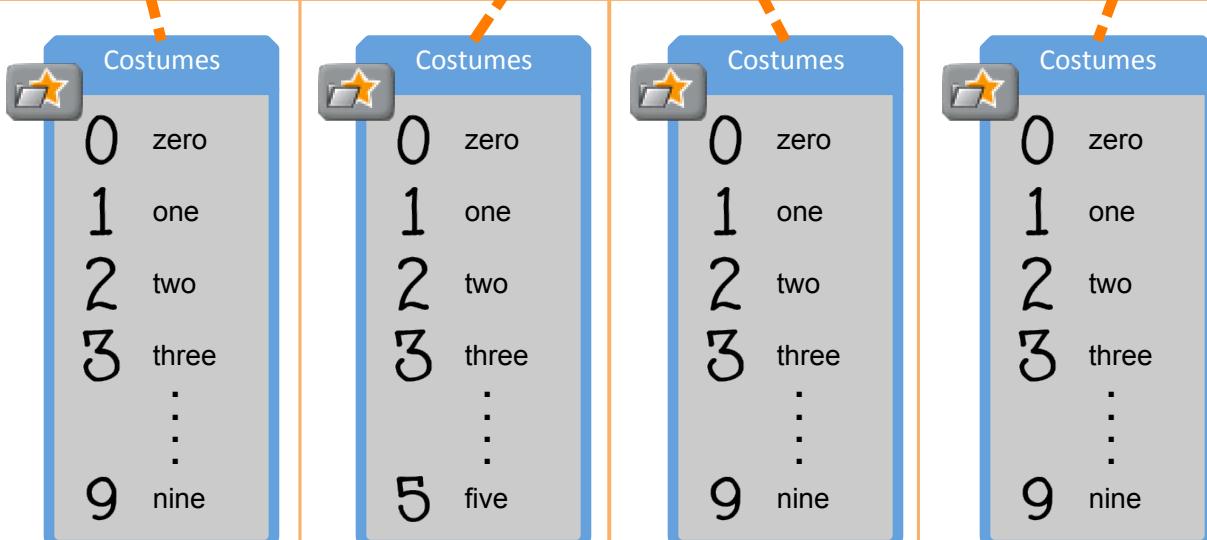


Variables

- digit1
- digit2
- digit3
- digit4



STOPWATCH



013



WITHOUT BROADCAST

```

when green flag clicked
forever
  go to x: 0 y: 0
  if clicked = yes
    glide 1 secs to x: [x position] of animal y: [y position] of animal
    wait 1 secs
    set [clicked v] to [no]
  end
end

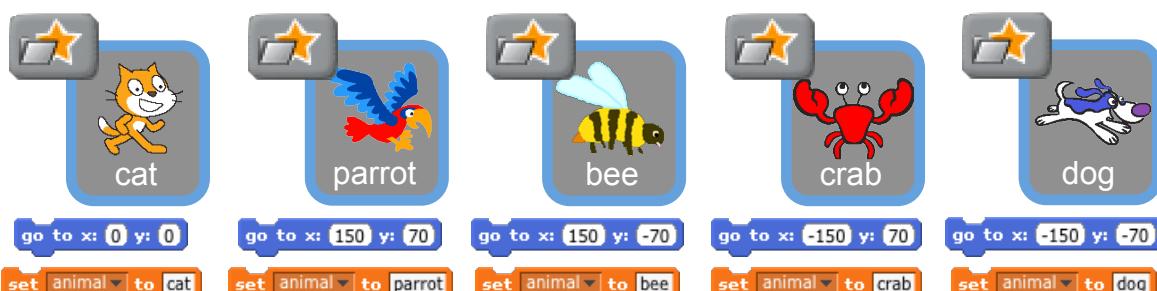
when [sprite name] clicked
set [animal v] to [sprite name]
set [clicked v] to [yes]
  
```



x 5



014



WITH BROADCAST

```

when I receive [reset v]
go to x: 0 y: 0

when [sprite name] clicked
set [animal v] to [sprite name]
broadcast [go to animal v] and wait
wait 1 secs
broadcast [reset v]

when I receive [go to animal v]
glide 1 secs to x: [x position] of animal y: [y position] of animal
  
```



x 5



015



POINT AND SHOOT

SHOOTER

```

when green flag clicked
  go to x: -150 y: 0
  broadcast play

when space key pressed
  broadcast fire and wait

when left arrow key pressed
  turn 15 degrees

when right arrow key pressed
  turn -15 degrees

```

MISSILE

```

when I receive play
  go to front
  go back 1 layers
  hide

when I receive fire
  go to boy
  point in direction direction of boy
  show
  repeat until touching edge?
    move 10 steps
  end
  hide

when I receive hit
  hide

```

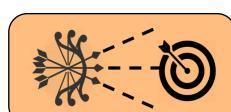
TARGET

```

when I receive play
  forever
    glide 1 secs to x: 160 y: pick random -180 to 180

when I receive play
  forever
    if touching bananas?
      broadcast hit
      say Yum! for 1 secs

```



016



HINTS: DESIGN IN 3D

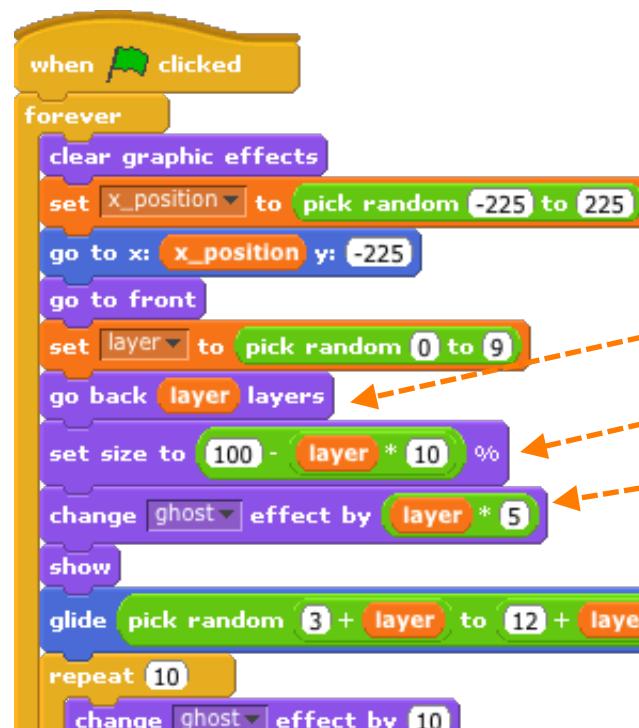
LAYERS	SPEED	SIZE	LOOKS

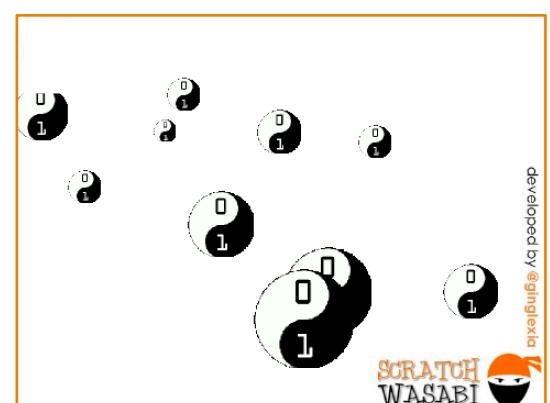
BUBBLING UP


x 10

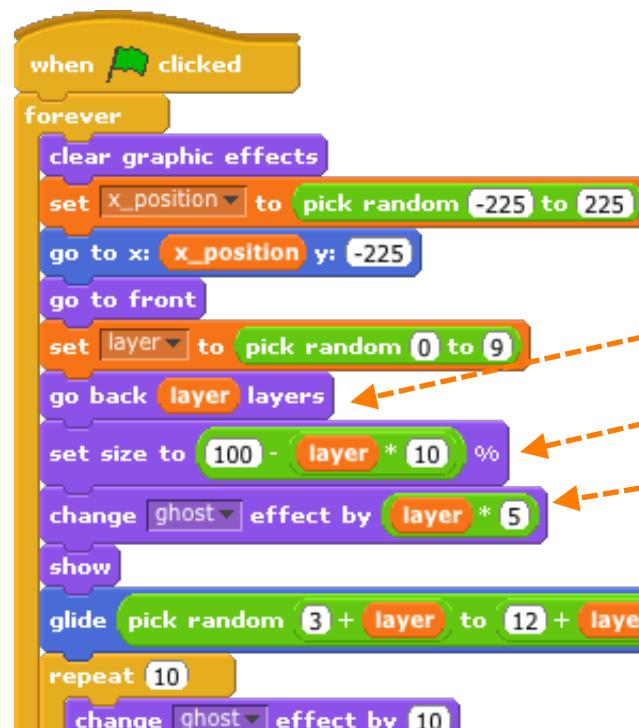
Variables

- layer
- x_position





developed by @gingexia



LAYERS

SIZE

LOOKS

SPEED



017



HONTS: DESIGN IN 3D

LAYERS	SPEED	SIZE	LOOKS

FLYING ACROSS

```

when green flag clicked
forever
  clear graphic effects
  go to front
  set [layer v] to [pick random (0) to (9)]
  set [y-position v] to [-80 + (layer * 20)]
  go to x: -255 y: [y-position v]
  go back [layer] layers
  set size to [100 - (layer * 10)] %
  change [brightness v] effect by [(layer * -5) %]
  show
  glide [3 + (layer) secs] to x: 255 y: [y-position v]
  hide

```

developed by @ginglexia

SCRATCH WASABI

LAYERS

SIZE

LOOKS

SPEED

018



PRESS 'P' TO PAUSE

```

when green flag clicked
go to x: 0 y: 0
forever if [playing = yes]
  move (50) steps
  turn (pick random (1) to (10)) degrees
  wait (0.1) secs
  if on edge, bounce
  turn (pick random (1) to (10)) degrees
end

```



```

when green flag clicked
set [playing v] to [yes]
when [p] key pressed
  if [playing = yes]
    set [playing v] to [no]
  else
    set [playing v] to [yes]
  end
end

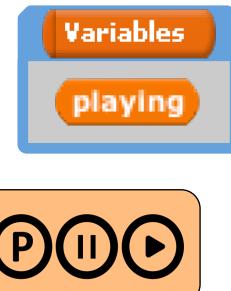
```



```

when green flag clicked
forever if [playing = yes]
  point towards [fish v]
  move (5) steps
end

```



playing



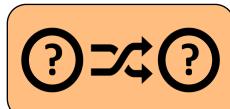
019

RANDOM TRIGGERS

```

when green flag clicked
forever
  hide
  go to x: 0 y: 0
  set [random_1 v] to [pick random (1) to (10)]
  set [random_2 v] to [pick random (1) to (10)]
  wait (0.5) secs
  if [random_1 = random_2]
    show
    think [2 random numbers the same!] for (2) secs
  end
end

```



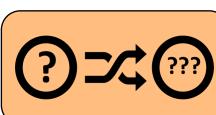
```

when green flag clicked
forever
  hide
  go to x: (pick random (-240) to 240) y: (pick random (-180) to 180)
  wait (pick random (0.1) to (1)) secs
  show
  wait (pick random (0.5) to (2)) secs
end

```



020



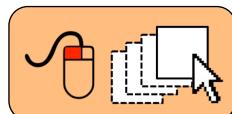
021



DRAG 'N' DROP



```
when green flag clicked
go to x: -100 y: -50
```



```
when green flag clicked
go to x: 0 y: 0
forever
if touching cat? and not mouse down?
  go to x: [x position of cat] y: [y position of cat] + [55]
if not touching cat? and not mouse down?
  go to x: 0 y: 0
```

022

SMOOTH MOVES

```
when green flag clicked
set control to keys
```



```
when C key pressed
if control = keys
  set control to mouse
else
  set control to keys
```



```
when green flag clicked
forever
if control = keys
  if key right arrow pressed?
    change x by 10
  if key left arrow pressed?
    change x by -10
  if key up arrow pressed?
    change y by 10
  if key down arrow pressed?
    change y by -10
if control = mouse
  set x to mouse x
  set y to mouse y
```



023

