

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!

 @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!

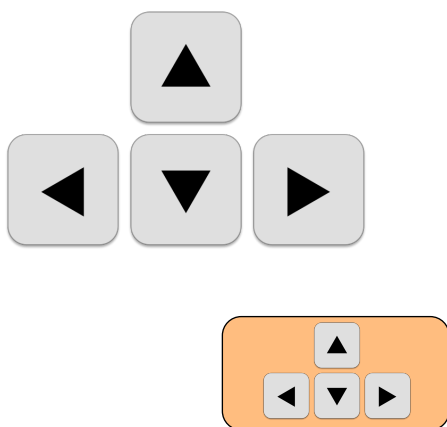
WASABI

“nuclear horseradish” (urbandictionary.com)

“really hot stuff” (ginglexia!)



USING KEYS



```

when up arrow key pressed
change y by 10

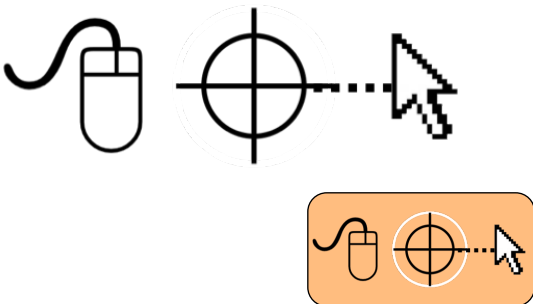
when down arrow key pressed
change y by -10

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

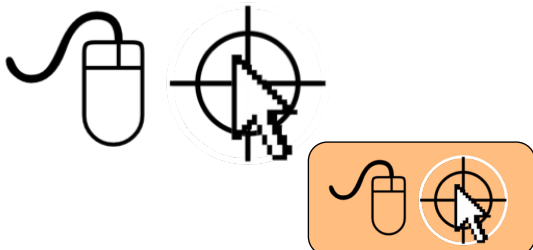


```

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

002

ON MOUSE

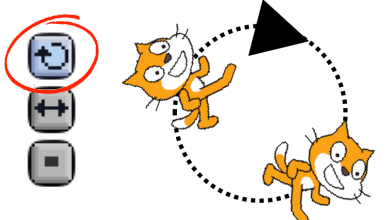


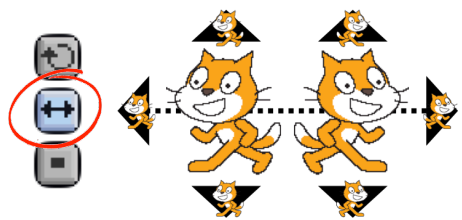
```

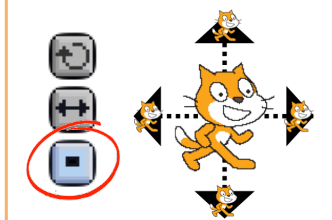
when green flag clicked
forever
go to x: mouse x y: mouse y
            
```

003

HINTS ...







RANDOM TURNS

004

RANDOM GLIDE

005


CHASING SPRITE

006

HINTS ...

SHOOTING

SHOOTER



```

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

when space key pressed
  broadcast fire and wait
  
```

MISSILE



```

when green flag clicked
  go to front
  go back 1 layers
  hide

when I receive fire
  go to wizard
  change x by 50
  change y by 15
  show
  repeat until x position > 220
    change x by 20
  hide

when I receive hit
  hide
  
```

TARGET




```

when green flag clicked
  switch to costume ghou1-a
  show
  forever
    if touching ball?
      switch to costume ghou1-b
      broadcast hit
      say Hit! for 1 secs
      hide
      wait 1 secs
      switch to costume ghou1-a
      show

when green flag clicked
  forever
    glide 1 secs to x: 160 y: pick random -180 to 180
  
```



CLICKING

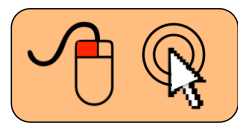
```

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



```

when clicked
  go to front
  forever
    go to x: mouse x y: mouse y
    if mouse down? and touching ghost?
      say Hit! for 0.5 secs
    if mouse down? and not touching ghost?
      say Miss! for 0.5 secs
  
```

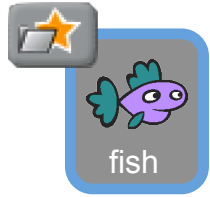


008

CHASING

```

when 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?
      broadcast got_me
      hide
      wait 1 secs
      go to x: pick random -240 to 240 y: pick random -180 to 180
      show
  
```



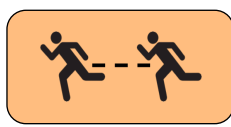
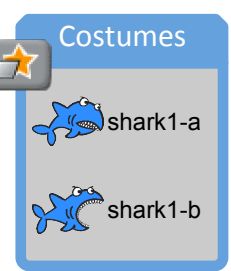
```

when clicked
  show
  switch to costume shark1-a
  forever if distance to mouse-pointer > 10
    point towards mouse-pointer
    move 5 steps
  
```



```

when I receive got_me
  repeat 2
    switch to costume shark1-b
    wait 0.1 secs
    switch to costume shark1-a
  
```



009

COUNTDOWN (30-0)

```

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

010

COUNTUP (0-30)

```

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

011

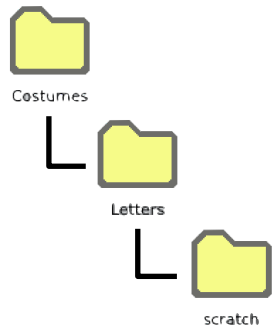
WAITING

```

when clicked
  go to x: 0 y: 0
  hide
  broadcast waiting
  repeat 80
    broadcast ready to play
  
```

012

0:00:00



Variables

- digit1
- digit2
- digit3
- digit4

STOPWATCH

0 minutes : divider2 0 tens : divider1 0 tenths

Costumes

- 0 zero
- 1 one
- 2 two
- 3 three
- ⋮
- 9 nine

Costumes

- 0 zero
- 1 one
- 2 two
- 3 three
- ⋮
- 5 five

Costumes

- 0 zero
- 1 one
- 2 two
- 3 three
- ⋮
- 9 nine

Costumes

- 0 zero
- 1 one
- 2 two
- 3 three
- ⋮
- 9 nine

```

when clicked
  set digit4 to 0
  switch to costume zero
  
```

```

when clicked
  set digit3 to 0
  switch to costume zero
  
```

```

when clicked
  set digit2 to 0
  switch to costume zero
  
```

```

when clicked
  forever
    set digit1 to 0
    switch to costume zero
    wait 0.1 secs
    repeat 9
      change digit1 by 1
      next costume
      wait 0.1 secs
    broadcast digit2
  
```

```

when I receive digit4
  next costume
  change digit4 by 1
  if digit4 = 10
    broadcast digit5
    set digit4 to 0
  
```

```

when I receive digit3
  next costume
  change digit3 by 1
  if digit3 = 6
    broadcast digit4
    set digit3 to 0
  
```

```

when I receive digit2
  next costume
  change digit2 by 1
  if digit2 = 10
    broadcast digit3
    set digit2 to 0
  
```

013

WITHOUT BROADCAST

```

when 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 to no
  
```

```

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

Variables

- animal
- clicked



x 5

014

go to x: 0 y: 0	go to x: 150 y: 70	go to x: 150 y: -70	go to x: -150 y: 70	go to x: -150 y: -70
set animal to cat	set animal to parrot	set animal to bee	set animal to crab	set animal to dog

WITH BROADCAST

```

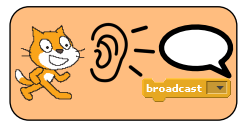
when I receive reset
  go to x: 0 y: 0

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

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

Variables

- animal



x 5

015

POINT AND SHOOT

SHOOTER



```

when 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
  hide

when I receive hit
  hide
  
```

TARGET



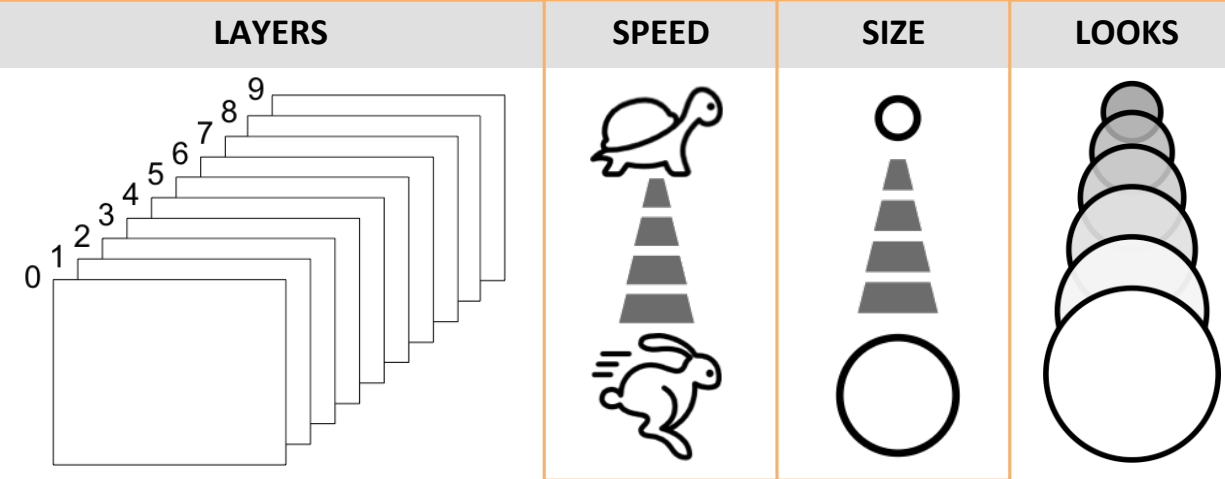
```

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

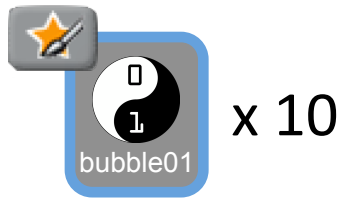
when I receive play
  forever
    if touching bananas ?
      broadcast hit
      say Yum! for 1 secs
  
```



HINTS: DESIGN IN 3D

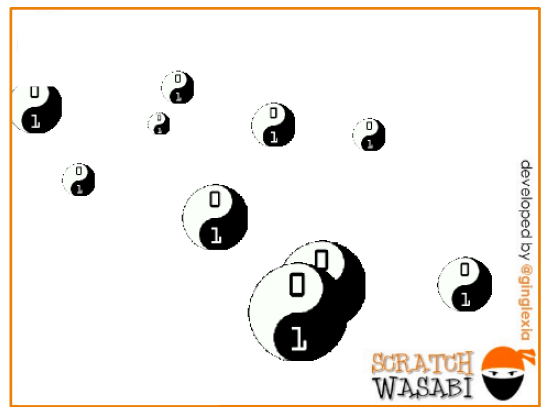


BUBBLING UP



Variables

- layer
- x_position



```

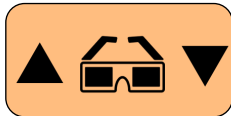
when clicked
  forever
    clear graphic effects
    set x_position to pick random -225 to 225
    go to x: x_position y: -225
    go to front
    set layer to pick random 0 to 9
    go back layer layers
    set size to 100 - layer * 10 %
    change ghost effect by layer * 5
    show
    glide pick random 3 + layer to 12 + layer secs to x: x_position y: 180
    repeat 10
      change ghost effect by 10
    hide
  
```

LAYERS

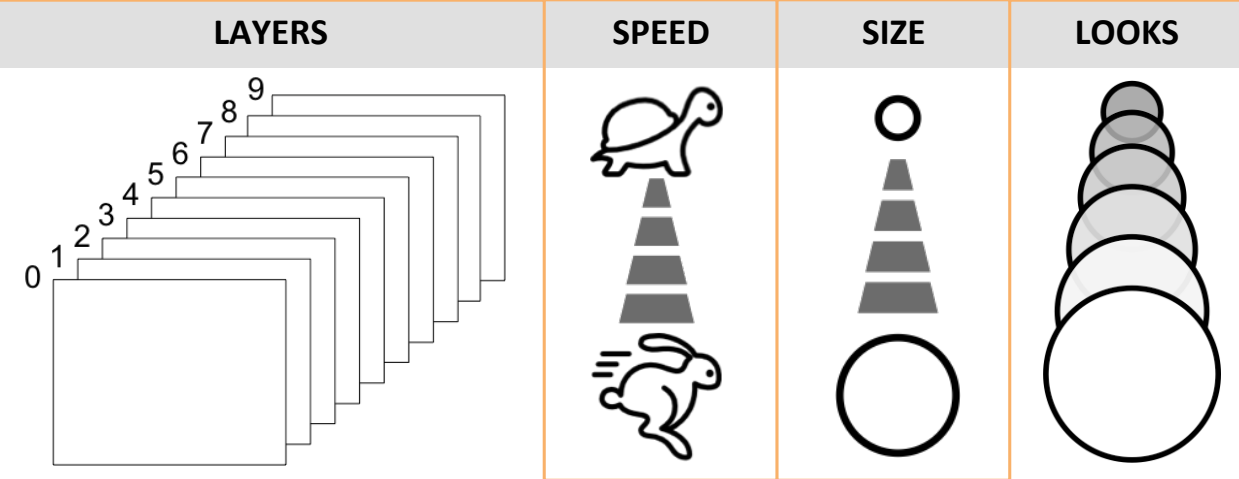
SIZE

LOOKS

SPEED



HONTS: DESGIN IN 3D



FLYING ACROSS

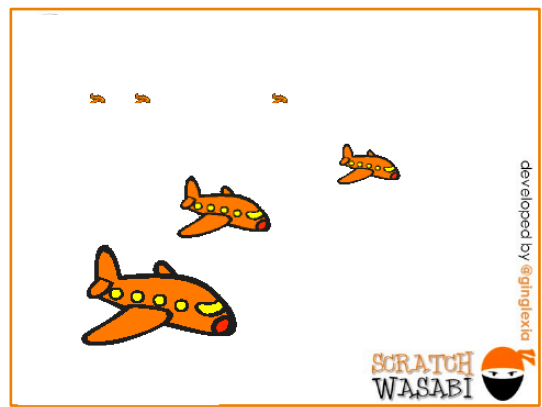
x 6

 plane01

Variables

 layer

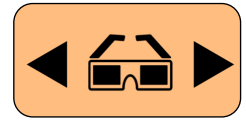
 y_positlon



```

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

- LAYERS
- SIZE
- LOOKS
- SPEED



PRESS 'P' TO PAUSE

```

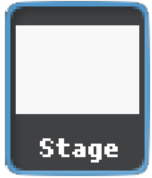
when green flag clicked
  go to x: 0 y: 0
  forever if playing = yes
    move 50 steps
    turn pick random 1 to 10 degrees
    wait 0.1 secs
    if on edge, bounce
    turn pick random 1 to 10 degrees
  
```



```

when green flag clicked
  set playing to yes

when p key pressed
  if playing = yes
    set playing to no
  else
    set playing to yes
  
```



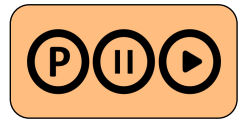
```

when green flag clicked
  forever if playing = yes
    point towards fish
    move 5 steps
  
```



Variables

playing

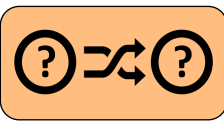


019

RANDOM TRIGGERS

```

when green flag clicked
  forever
    hide
    go to x: 0 y: 0
    set random_1 to pick random 1 to 10
    set random_2 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
  
```



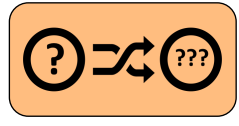
Variables

random_1

random_2

```

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
  
```



020

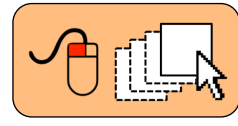
021

DRAG 'N' DROP



```

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



```

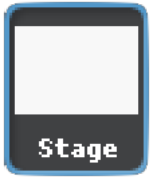
when clicked
  go to x: 0 y: 0
  forever
    if touching cat? and not mouse down?
      go to x: x position of cat y: y position of cat + 55
    if not touching cat? and not mouse down?
      go to x: 0 y: 0
  
```

022

SMOOTH MOVES

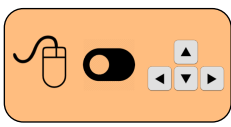
```

when clicked
  set control to keys
  
```



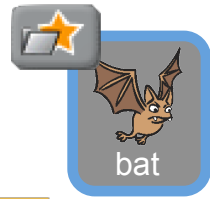
```

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



```

when 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