

# STORY TIME!

Story creation with Scratch provides opportunities to explore a variety of computational concepts and skills. Here are some blocks that are frequently useful in stories.

## WAIT

Insert a pause



## SAY/THINK

Have a speech or thought bubble appear over a sprite



## SOUNDS

Play recorded audio



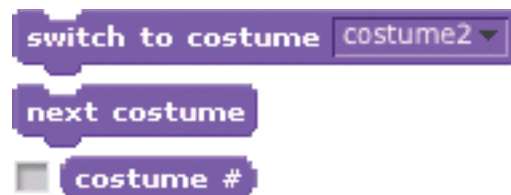
## VISIBILITY

Make a sprite appear or disappear



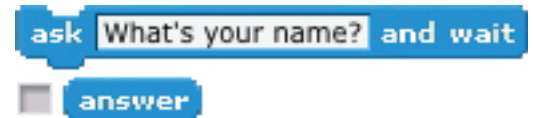
## COSTUMES

Change the appearance of your sprite



## ASK

Get input to use in a project



## STRINGS

Test, access, and change words and sentences



## COORDINATE

Synchronize actions between and within sprites



# SLIDESHOW



Create your own slideshow – a collection of background images accompanied by audio narration.

1 script  
6 background images  
6 recorded sounds

record your narration right in Scratch with the sound recorder

Background Name	Dimensions	Size
ScratchDay	480x168	92 KB
ScratchDay-Hong	480x321	595 KB
ScratchDay-Guat	480x360	665 KB
ScratchDay-Swit	480x360	650 KB
ScratchDay-MIT	480x360	649 KB
ScratchDay-Cat	480x360	636 KB

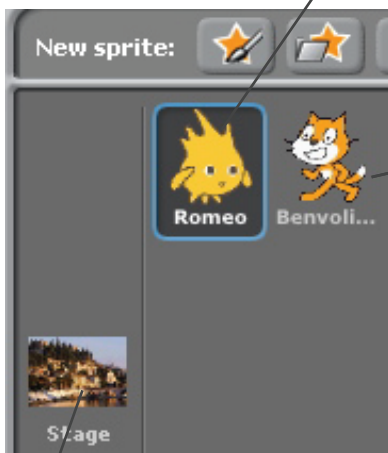
Sound Part	Duration	Size
part1	0:00:11	491 KB
part2	0:00:06	241 KB
part3	0:00:06	251 KB
part4	0:00:08	338 KB
part5	0:00:10	436 KB
part6	0:00:11	491 KB

```
when green flag clicked
  switch to background ScratchDay
  play sound part1 until done
  next background
  play sound part2 until done
  next background
  play sound part3 until done
  next background
  play sound part4 until done
  next background
  play sound part5 until done
  next background
  play sound part6 until done
  next background
```

# CONVERSATION



Get two characters talking to each other. Use the **say** and **wait** blocks to coordinate the conversation.



1 script

1 script

no scripts  
add a background image

```
when clicked
say Ay me! Sad hours seem long. for 2 secs
wait 2 secs
say Not having that, which, having, makes them short. for 2 secs
wait 2 secs
say Out-- for 2 secs
wait 2 secs
say Out of her favour, where I am in love. for 2 secs

when clicked
wait 2 secs
say What sadness lengthens Romeo's hours? for 2 secs
wait 2 secs
say In love? for 2 secs
wait 2 secs
say Of love? for 2 secs
wait 2 secs
say Alas, that love, so gentle in his view, for 2 secs
say Should be so tyrannous and rough in proof! for 2 secs
```

# CATLIBS



Dynamically create a surprising story. Use the **ask**, **say**, **join** blocks, and **variables** to compose a short story based on suggested words.

```
when clicked
say Welcome to CatLibs! for 2 secs
say Let's create a story together. for 2 secs
ask Please give me the name of a friend: and wait
set friend's name to answer
ask Please give me the name of an animal: and wait
set animal to answer
ask Please give me the name of a place: and wait
set place to answer
ask Please give me the name of a thing: and wait
set thing to answer
say One day, for 2 secs
say join friend's name join and a animal for 2 secs
say join went to join place join to see a thing for 2 secs
```

no scripts



1 script  
4 variables



# SCENES



Use the **broadcast** and **when I receive** blocks to create a multi-scene story.

add 3 background images



4 scripts

4 scripts

2 scripts

```

when green flag clicked
  broadcast scene-boston and wait
  broadcast scene-paris and wait
  broadcast scene-calro and wait
  
```

```

when I receive scene-boston
  switch to background boston
  
```

```

when I receive scene-paris
  switch to background paris
  
```

```

when I receive scene-calro
  switch to background calro
  
```

```

when green flag clicked
  go to x: -200 y: -100
  when I receive scene-boston
    say I love Boston. for 2 secs
    glide 1 secs to x: -80 y: -100
  
```

```

when I receive scene-paris
  say But Paris is great. for 2 secs
  glide 1 secs to x: 40 y: -100
  
```

```

when I receive scene-calro
  say Whoa! for 2 secs
  
```

```

when green flag clicked
  hide
  go to x: 200 y: -100
  when I receive scene-calro
    show
    wait 2 secs
    say Welcome to Cairo! for 2 secs
  
```

# LET'S PLAY!

Like the other genres we've already explored, games provide numerous opportunities to explore a variety of computational concepts and skills. Here are some blocks that are frequently useful in games.

## TOUCHING

See if two sprites are touching or if a sprite is touching a color



## VISIBILITY

Make a sprite appear or disappear



## RANDOM

Get a computer-generated number from within a specified range



## TIMING

Have the computer keep track of time for you



## STRINGS

Test, access, and change words and sentences



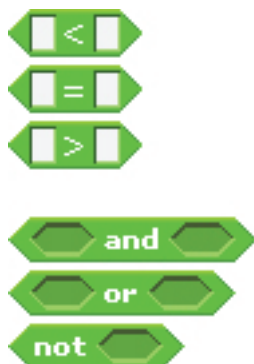
## VARIABLES

Store a number or string in a container to access later



## COMPARE

Compare values to help make decisions within your game

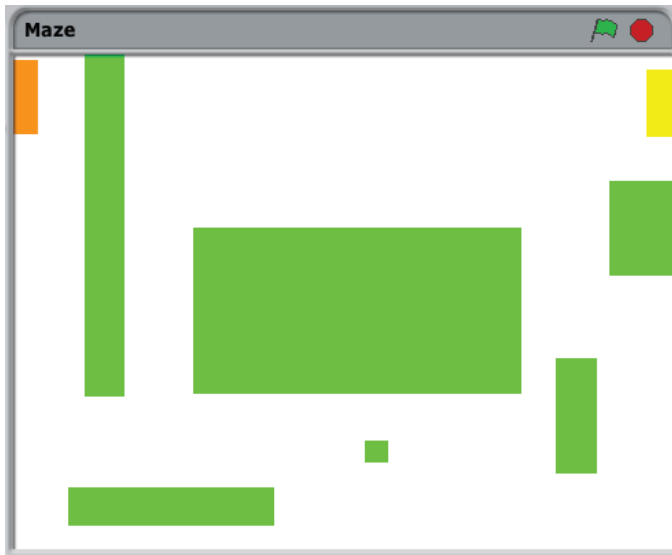


## KEY PRESS

Make a sprite respond when different keys are pressed



# MAZE



## GOAL

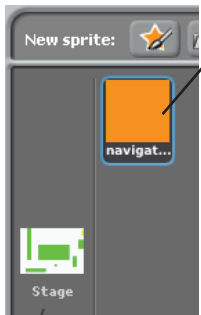
Get from the start of the maze to the end

## RULES

Don't touch the green walls

## OUTCOME

Win when the yellow marker is reached



7 scripts

move the sprite around

```

when down arrow key pressed
  point in direction 180
  move 10 steps
  
```

```

when up arrow key pressed
  point in direction 0
  move 10 steps
  
```

```

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

```

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

have the sprite bounce off the green walls

```

when green flag clicked
  go to x: -249 y: 149
  
```

```

when green flag clicked
  forever if touching color green
    turn 180 degrees
    move 10 steps
  
```

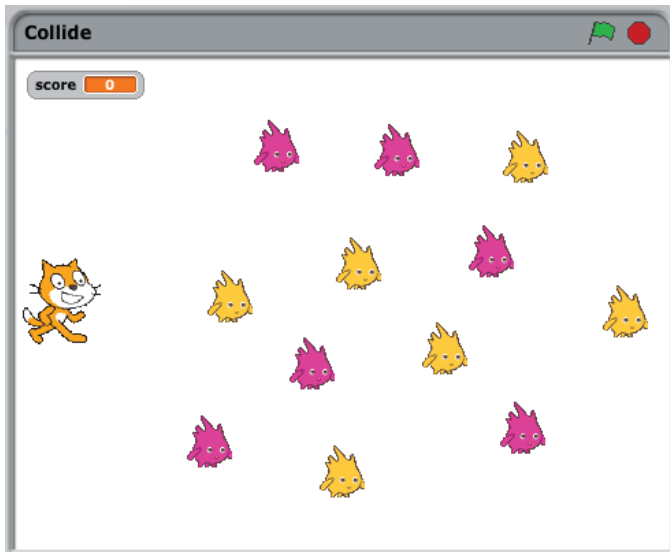
```

when green flag clicked
  wait until touching color yellow
  say You Win! for 2 secs
  
```

players wins when sprite reaches the yellow end marker

no scripts, draw a maze-like background with colored walls and a differently colored end marker

# COLLIDE



## GOAL

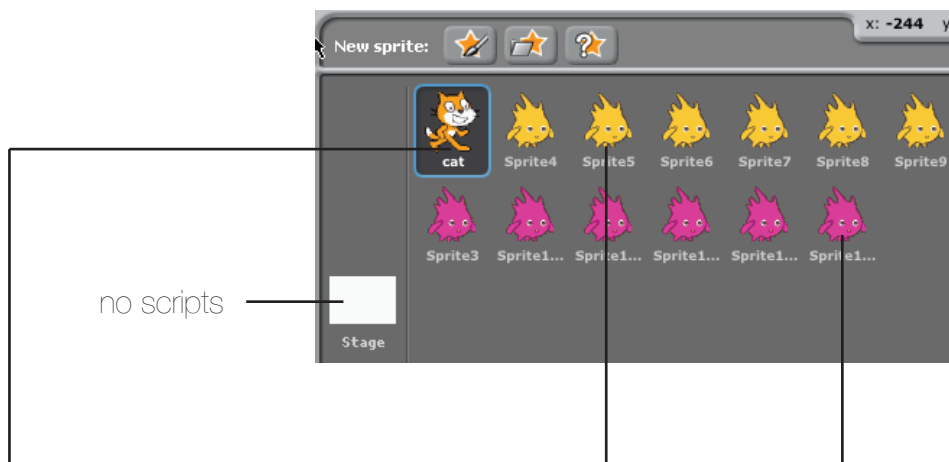
Help the cat navigate a gobo minefield

## RULES

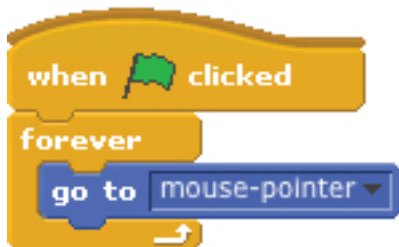
Collect yellow gobos to earn points, avoid pink gobos to avoid losing points

## OUTCOME

Maximize your score

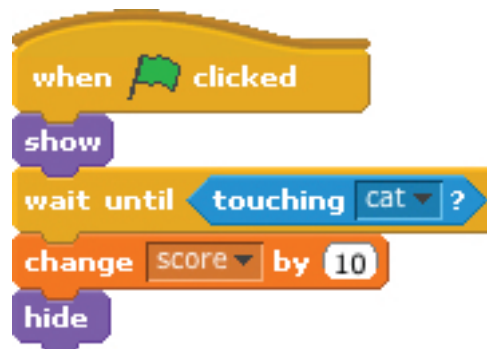


reset the cat's position and the score

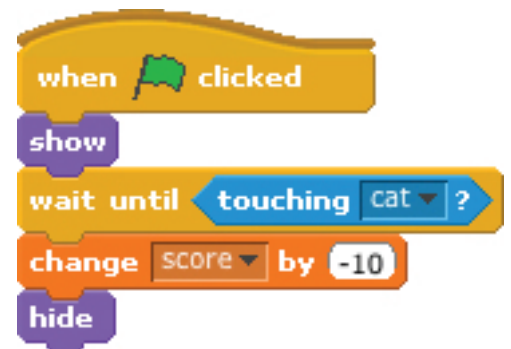


have the cat follow the mouse cursor

when the cat collides with a yellow gobo, the gobo disappears and the score increases by 10

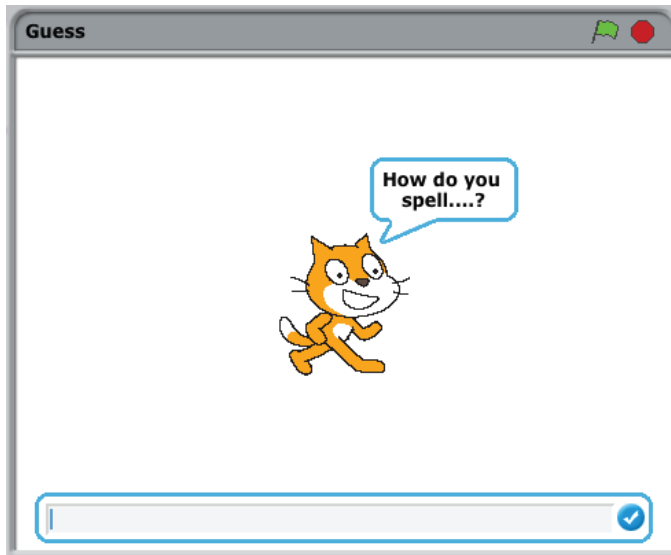


when the cat collides with a pink gobo, the gobo disappears and the score decreases by 10





# GUESS



## GOAL

Test your spelling abilities

## RULES

Type the words spoken by the cat

## OUTCOME

Learn whether you spelled each word correctly

create a list of words and audio-record their pronunciations



Make a list

Delete a list

words

add thing to words



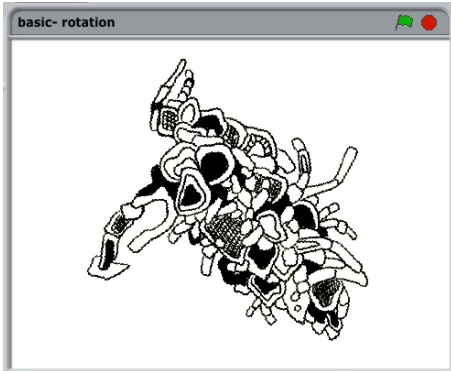
1 script

no scripts



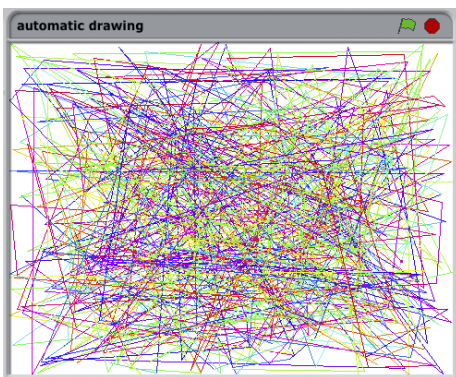
# GOT ART?

Creating art projects offers a slightly different perspective when using Scratch and a chance to think of new and exciting ways to experiment with computational concepts and skills. The following five projects will enable you to explore Scratch by creating abstract works.

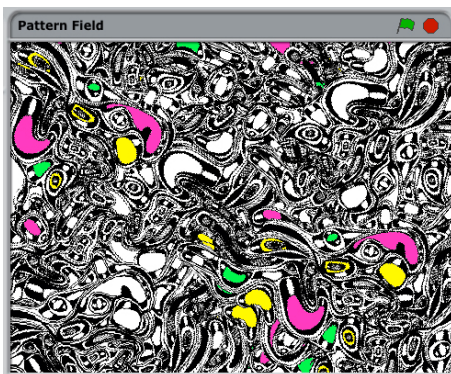


1) **Back to the Basics:** learn some basic block combos that are important to this project series

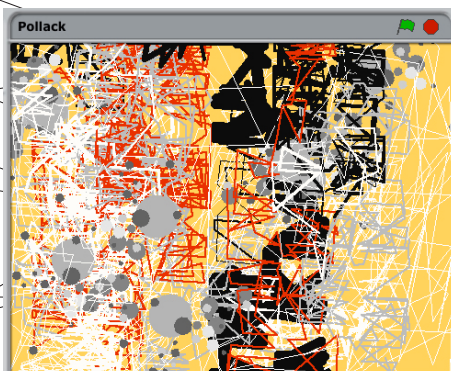
2) **Randomization:** learn some ways to switch up the scripts from “Back to the Basics” using the “Pick Random” block



3) **Automatic Drawing:** create a self-creating drawing using the skills learned from Project 1 and 2



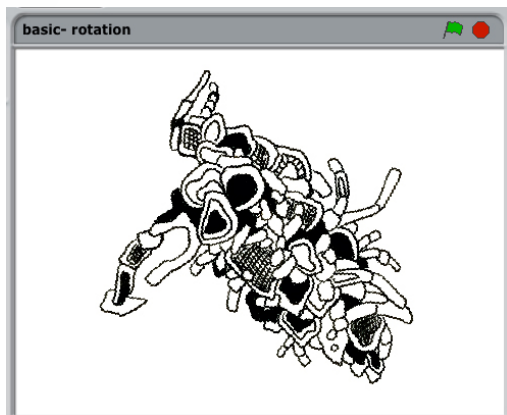
4) **Pattern Field:** create a partly interactive pattern field based on repeating units



5) **Make a Pollock (CHALLENGE):** create your own Pollock-esque original using blocks combinations learned from previous projects and experimentation

# Back to the Basics

**objective:** learn some basic block combos that are important to this project series



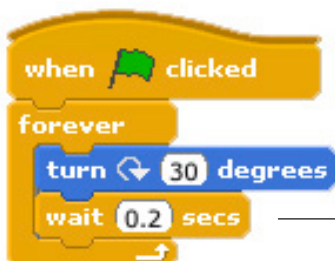
## Behind the Scenes:



**stage:** no scripts, no background

**sprite1:** image without point symmetry

## Rotation



allows the rotation to occur at a time interval so you can see it move



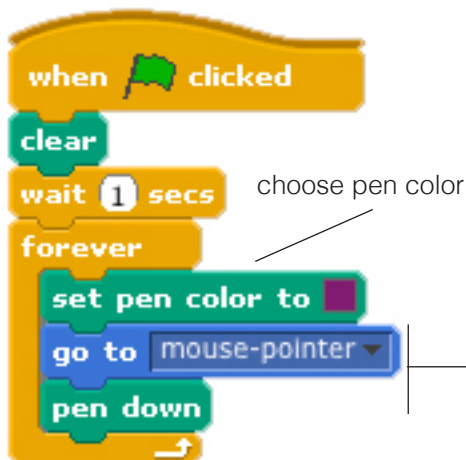
to stop the script

sprite1 follows mouse pointer repeatedly

## Movement



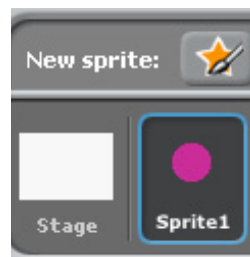
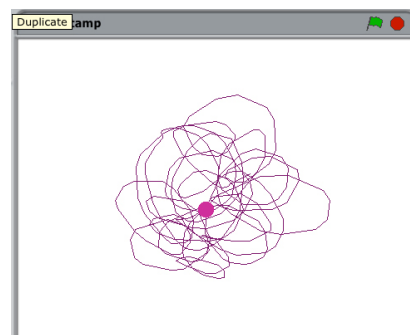
## Stamping (Drawing)



choose pen color



pen draws, following mouse-pointer

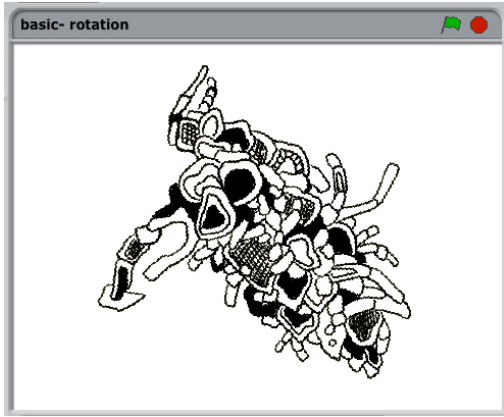


**stage:** no scripts, no background

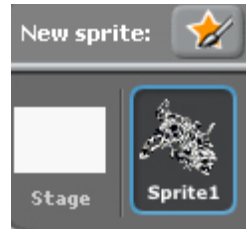
**sprite1:** simple, a dot works

# Randomization

**objective:** learn some ways to switch up the scripts from “Back to the Basics” using the “Pick Random” block



## Behind the Scenes:



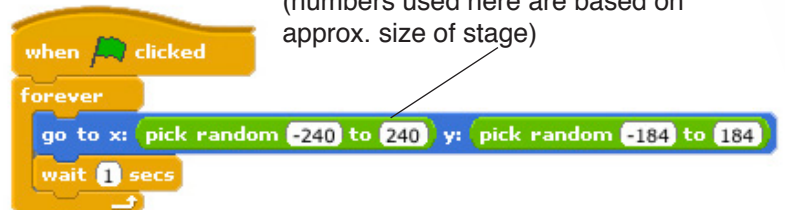
**stage:** no scripts, no background

**sprite1:** image without point symmetry

## Random Rotation



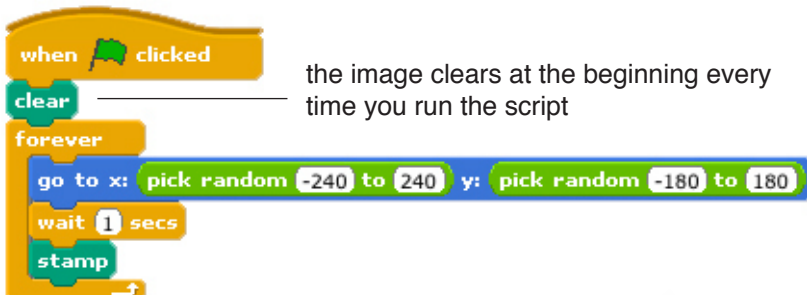
## Random Movement



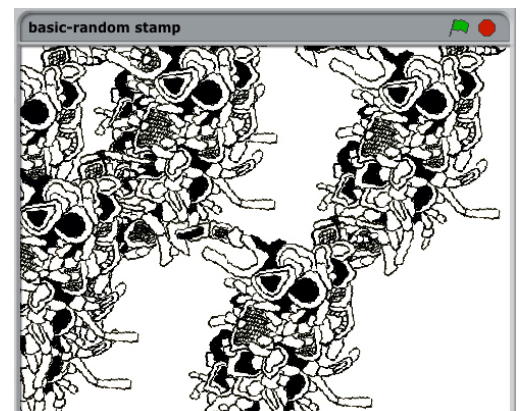
sprite moves to a random position (numbers used here are based on approx. size of stage)



## Random Stamping

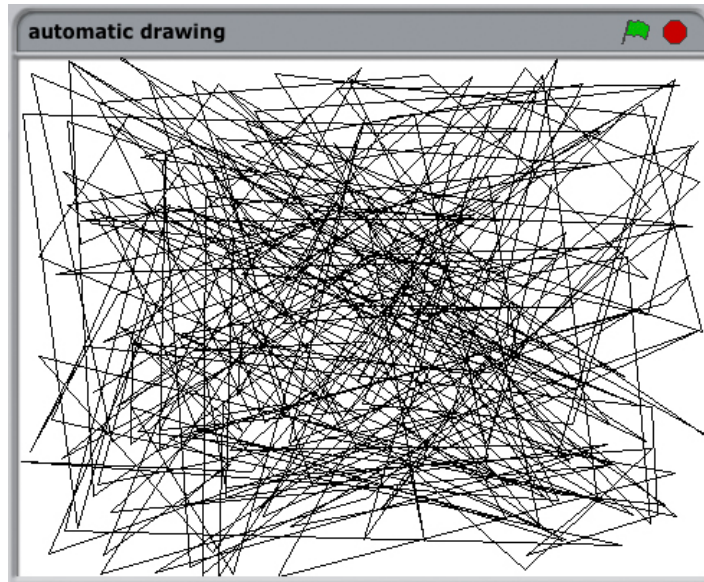


the image clears at the beginning every time you run the script

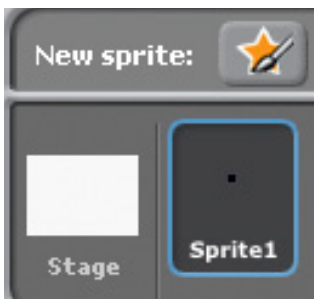


# Automatic Drawing

**objective:** create a self-creating drawing



## Behind the Scenes:



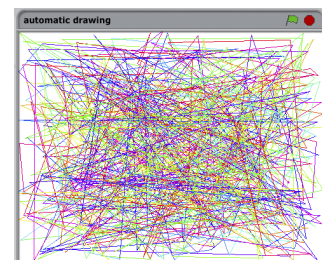
**stage:** no scripts, no background

**sprite1:** simple, a dot works

(optional)

change pen color by pick random -30 to 30

adding this block after "if on edge, bounce" block will create a crazy colorful image!



```
when green flag clicked
  clear
  pen up
  set pen color to black
  go to x: pick random -240 to 240 y: pick random -180 to 180
  repeat until key space pressed?
    pen down
    go to x: pick random -240 to 240 y: pick random -180 to 180
    if on edge, bounce
    change pen color by pick random -30 to 30
```

a new drawing starts every time

choose pen color, "pen up" so it does not start drawing

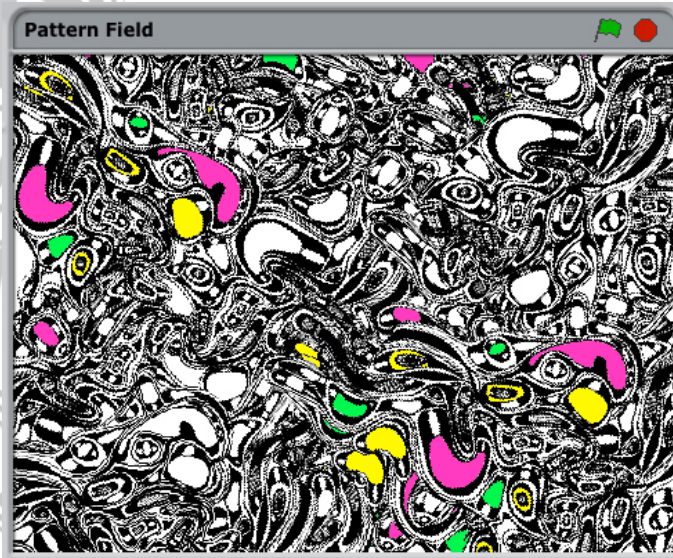
pen moves to a random position

to stop the drawing

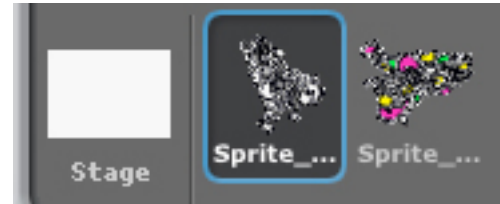
pen draws as it moves continuously to a random position on the stage (numbers used here are based on approx. size of stage)

# Pattern Field

**objective:** create a partly interactive pattern field based on repeating units



## Behind the Scenes:



**stage:** no scripts, no background

**sprite 1:** a found or hand drawn image with a striking pattern, preferably with interesting large spaces

**sprite 2:** an edited version of sprite 1 with some of the large spaces filled with bright color

*the image will be half interactive and half automatic, therefore, two sprites will be made*

## Sprite 1

```
when green flag clicked
hide
wait 1 secs
forever
  show
  go to mouse-pointer
  turn pick random 1 to 360 degrees
  wait 1.5 secs
  stamp
```

ensures the stage will be empty when the project is started

image will follow your pointer

sprite will rotate before it's stamped, for variation

so that the patterns "print" on the stage ever 1.5 seconds



example Sprite 1

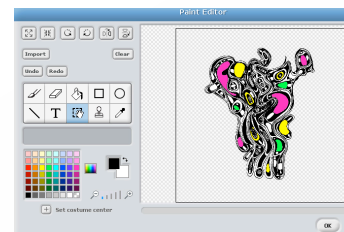
```
when space key pressed
hide
clear
```

the pattern field is cleared when the space key is pressed

## Sprite 2

```
when green flag clicked
hide
wait 1 secs
forever
  go to x: pick random -250 to 250 y: pick random -180 to 180
  turn pick random 1 to 360 degrees
  show
  wait 3 secs
  stamp
```

sprite moves to a random position (numbers used here are based on approx. size of stage)



Sprite 2 made by editing Sprite 1 in Scratch

```
when space key pressed
hide
clear
```